1-1-2017

Product Development Resilience Through Set-Based Design

Stephen H. Rapp
Wayne State University,

PRODUCT DEVELOPMENT RESILIENCE THROUGH SET-BASED DESIGN

by

STEPHEN HORTON RAPP

DISSERTATION

Submitted to the Graduate School

of Wayne State University,

Detroit, Michigan

in partial fulfillment of the requirements

for the degree of

DOCTOR OF PHILOSOPHY

2017

MAJOR: INDUSTRIAL ENGINEERING

Approved By:

____

Advisor Date

____

Advisor

____

____

____

DEDICATION

To my dear wife: Polly Lynn Rapp

"Tu sei la mia vita, mio amore e la mia anima gemella."

To my children: Lydia, Joshua & Lauren, Caleb and Christina & Stephen

To my siblings: David & Kim and Susan & Steve

To my mom and her eight siblings: Betty Browand Horton

To my dad and step-mom: Dr. Robert Samuel Rapp, ThD and Clara Rapp

To the men who shaped me: my uncle, Stan Rapp and father-in-law, Norm Smith

To my paternal grandparents: Stanley and Marian Rapp who taught me -

"Machst du es richtig, mein junge Sohn."

To my maternal grandparents: Dr. Glover Horton, MD and Delilah Horton

To my inspiring aunt and uncle: Ruth Rapp Donaldson and Dr. Alan Donaldson, PhD

To all my Harnish aunts who just loved me non-stop when I was a lad

To my loving aunt who taught me to swim: Rosalyn Pearson Rapp

To my uncle and aunt who never gave up finding me: Samuel & Susan Browand

To my amazing, loving and wonderful mom-in-law: Marcene Conrad Smith

To my uncle and aunt who fed me and taught me to fish: Walter & Joyce Bergey

To my Michigan family: "Der Geschicte von Proschek", who are more Pennsylvania Dutch than me.

ACKNOWLEDGEMENTS

This dissertation represents the end of a long, almost 10-year trail that I undertook to refresh my education and life skill set while still working. Between my life classmates and extraordinary faculty like Dr. Kenneth Chelst, Dr. Julia Gluessing and so many others, I experienced a true renaissance and expansion of my life skills. Although the road was long, the Global Executive Track program made it not just a reward at the end, but a rewarding and fulfilling total experience. Thank you all very much.

I specifically want to recognize each of my dissertation committee members, not only as amazing faculty, but also as mentors. Dr. Alper Murat helped me go beyond my optimization background as a friend, guide and teacher. He helped me synergize all the analytics. Dr. Norbert Doerry gave me the initial break through thought that direct, single point optimization for product development was fraught with issues. He has been a steady realist and guide for the research. His combination skill set of real development and analytics was a huge help. Dr. Leslie Monplaisir has been a steady hand of support and reason from Day One. His door was always open. Dr. Gary Witus, a co-advisor, helped me flesh out the initial concept of hybridizing optimization into a different valuation structure. He gave me so much of his time and life to bring this project to fruition. Dr. Ratna Babu Chinnam, my other advisor, was simply a blessing. His expertise in Markov Decision Processes was crucial, but more importantly, he was there for me the entire time. Between his support and tutelage early in my classes, through the case studies and finally completing the dissertation, he was both a partner and leader to me. I cannot say thank you enough, to you all.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: RESEARCH DESCRIPTION

PROBLEM BACKGROUND

The Problem Itself

Product Development (PD) remains an uncertain process fraught both with reward and risk wrapped up by the uncertainties of the engineering development itself and outside changing factors over the timeline of each program. Particularly, changing requirements during product development, play havoc with programs. The multiple stakeholders with their different concerns, constraints, changeable priorities and the uncertainty in the cost and engineering characteristics of subsystem technologies all impact the total system choice through the program life. Set-Based Design (SBD) methodology as a candidate engineering development approach promises to add resiliency into the PD process.

While there are great qualitative examples from Toyota (Liker 1999), Schlumberger (Madhavan 2008) and others, SBD lacks a rigorous mathematical, quantitative formalization. Emergence of computational, combinatorial design generation and evaluation tools and methods continue, but there are limitations in their use and application as they are all focused on generating “point solutions”. In general, they create efficient “bags” of Pareto optimal or near optimal solution points oriented to a single system design. This is a great weakness because the supposedly optimal point solution early in the design will undergo unknown changes that directly affect the design space and ultimate program success. Thus, no insight into system design decisions, that are robust and resilient to time and cost changes, occurs. The additional ongoing requirements changes in thresholds and priorities can shatter the vulnerable or brittle point solutions when subsystem technology characteristics turn out to be different than expected.

A classic example of this was the Future Combat Systems (FCS) program that was to develop a family of light, but still well protected and network integrated armored fighting vehicles. The Army outsourced the System Integration to Boeing which created organizational cultural problems. However, the key issue documented for the program failure was the risk and uncertainty associated with the requirements as they shifted and morphed over time (Bradford 2011). Technically from the author's experience, the key underlying issue was that the needed light-weight armor subsystem did not develop and mature to expectation. This impacted the system weight margin so severely that at one point there were no feasible solutions out of the millions of combinations as the design effort progressed. The program had to raise the system weight threshold to find feasibility. The associated increased costs with the higher weight and other unfulfilled requirements eventually caused the Army to cancel the program. Basically, the program's PD process was too brittle to survive the point based design uncertainties as they came to fruition and created large cost breeches.

Furthermore, from the author's experience, the Earned Value Management System utilized to ensure the system integrators delivered on-time and on-budget, posted near to perfect and the Army Program Office successfully passed GAO auditing even though the program was cancelled. This implies that the incremental contractor deliveries met Army expectations while the Army successfully managed the team, but the program itself failed. The only rational explanation then, is that the unknown program risks, juxtapose to the requirements, manifested themselves to such a degree, that the program reserve was unable to contain the program cost breech.

The terms design, trade and state space generally mean the same thing, but are used to create different emphases in this proposal. State space is used when considering more mathematical aspects. Trade space is used specifically in the context of structured decisions. Design space is used and focused on design decisions occurring between the initiation of requirements and the completion of design and testing prior to full production.

Set-Based Design Background

Set-Based Design (SBD) is based on the principle of satisficing given multiple constraints (requirements). Different stakeholders may be the source of different constraints. The requirement thresholds change over time. The relative priorities, i.e., the willingness of stakeholders to relax one requirement to be able to restrict another requirement and still have feasible solutions, also change over time. SBD is an iterative process in which design and requirements evolve in parallel, in which stakeholders restrict and relax requirements with feedback regarding feasible solutions in design space given the requirements (Singer 2009). It is a concurrent engineering process that helps stakeholders understand the interdependencies among the requirements and impact on design as they work to develop the performance specification and preliminary design.

The guiding principle of SBD is to consider the set of feasible designs. As requirements are tightened and re-balanced, the feasible design space is restricted until one, or a few distinct, alternative solutions remain. The design space can also be restricted based on additional feasibility assessments and test results. SBD does not presume there is an objective function for the system, or even a collection of partial objective function for Pareto-optimality. A design, i.e., a point solution, is feasible if it meets all the constraints.

STATEMENT OF THE PROBLEM

With increased execution speed requirements and cost scrutiny of modern large-scale programs, current execution strategies need resiliency to deal effectively with unknowns, risk and program changes. Set-Based Design itself is a promising capability supporting resiliency and change during program execution. More research is needed to create a functioning SBD trade space framework and eventual toolset. Additionally, current SBD uses are primarily qualitative in nature. Our research will also advance SBD quantitative analysis to engender and develop program Product Development (PD) resiliency.

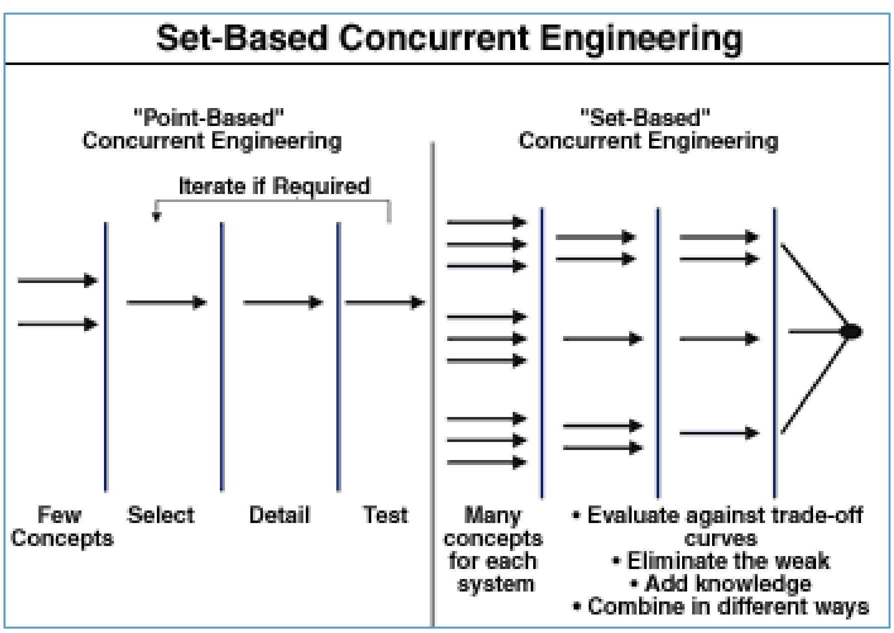

FIGURE 1 - POINT VS. SET BASED DESIGN CONCEPTS

Figure 1 shows the general conceptual difference between point-based design and set-based design (Miller 1993).

Great capability exists with current system design optimization models, but actual PD resiliency, that is set-based, is lacking since analytical mathematical tools, even if combinatorial based, only provide point solutions. The narrow point solutions eventually create design issues and unplanned changes as the design uncertainty lifts over time. This proposal addresses framework creation to enable a resilient set-based design process for complex programs. The GAO itself has recognized the need for a greater diversity of design concepts earlier to lower risk and design problems later (Martin 2012). SBD holds great promise when coupled with high speed accurate optimization to keep a richer, less failure prone design set to accomplish PD needs both in budget and on time. The GAO specifically identified and correlated program success with richer alternatives or solution sets and program failure to those that had less. However, that correlation was only related to the Analysis of Alternatives (AoA) which is wholly associated to very high-level requirements at the program start.

Particularly, this framework will focus on the PD time segment from initial high-level requirements, to design completion prior to limited production. This is specifically, from the completion of the AoA which occurs just before the Alternatives System Review (ASR) through the Critical Design Review (CDR) for defense programs. Currently, defense programs tend to neck down and lock conceptual elements very early in design, limiting the program office's later decision making as unknowns become known. Commercial manufacturing generally follows this process, but lacks the formalism of government funded PD. The framework will be relevant for both commercial and government sectors. By combining set-based design in a framework structure that utilizes modern high speed mathematical analytics, a more open and rich design trade space can be maintained deeper into programs, allowing for a less risky and more flexible program office decision capability.

Figure 2 shows the PD timeline for formalized defense sector programs. A, B and C are formal program milestones. Accompanied with the seven technical reviews a typical defense program consists of ten or even more epochs or decision points. The reviews have informal decisions that directly impact the formal milestone decisions. Business programs typically have the same work but are not formalized. Our use of epochs in the framework represent a more generic approach but are decision points nonetheless. Both defense and business should view design reviews and formal milestones as epochal decision points.

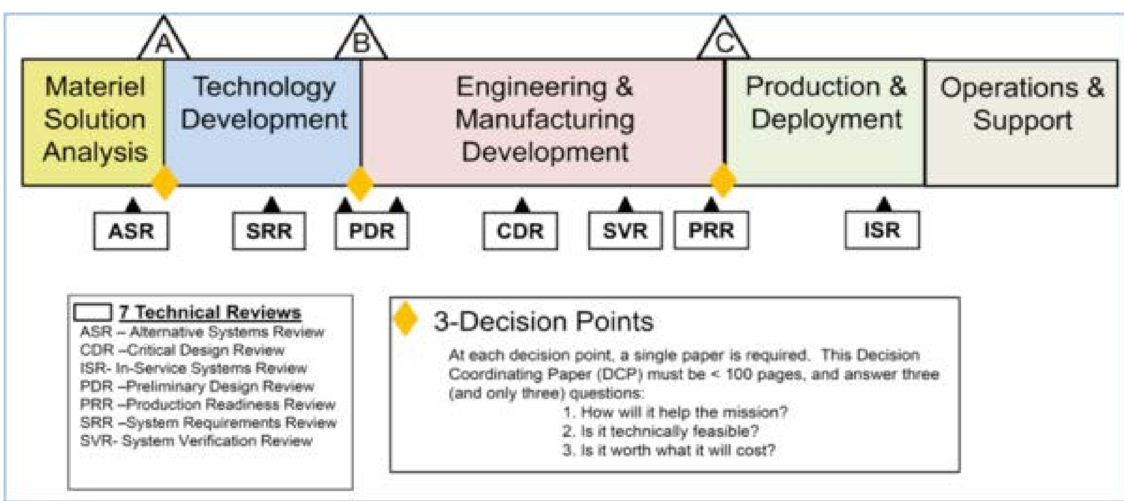

FIGURE 2 - DEFENSE SECTOR PD TIMELINE

PURPOSE OF THE STUDY

GENERAL GOAL

As a general goal, we will extend SBD to improve resilience of the development process with respect to changes in requirements and to uncertainties in the availability of subsystem/component alternatives during system development. The research will develop a conceptual framework that will describe an implementation of the resilient research itself that integrates mathematical analysis with the richness of SBD. The research will combine the strengths of SBD and optimization into a new mathematical hybrid approach supporting the resilient design process itself and by association, derivative resilient designs that are neither cost prohibitive, nor brittle to requirements and uncertainty. Finally, the research will develop a current case or example as a Proof of Concept (POC) to validate the framework and provide insight for eventual instantiation of a technical algorithm and solution.

RESILIENT PD PROCESS CHARACTERIZATION AND GOAL ASSUMPTIONS

A Resilient PD Process needs to be characterized in two macro manners. The first is the ability to carry multiple efficient designs forward and deeper into the PD process than what is current in the singular design process. The second is the ability to change or modify the design or designs on the fly, when uncertainties resolve and technical, cost or schedule risks come to fruition. This may be before or after production. This is the essence of why SBD was developed into Concurrent Engineering (CE) at Toyota in the 1990's.

The development of SBD has been heavily focused toward developing the macro processes inside business itself and has only recently moved to actual formulation structures. Those structures have recognized the role of bringing in mathematical analysis to work and analyze the set solutions through PD (Singer 2009). However, the research is extremely seminal. The fundamentals of what and how mathematical analysis itself is conducted and maintained, has not been addressed.

The very first mathematical issue that must be resolved is the trade space representation or region definition of a set solution. This is because set solutions typically include multiple point solutions that may be optimal, suboptimal or potentially infeasible. Although, all solutions reside in the trade space of the system choices, those choices themselves are not static and will morph over time. An entire new topology must be constructed to properly deal with set solutions. Supporting data structures need to be described and properly sized to support the potentially geometric increase in the solution dimensions themselves, since a set solution must carry greater data than a singular solution.

This is a significant challenge requiring a large effort. This dissertation's macro scope will be developing the framework with a focus on the topology of the design space, specifically the design space region characterization for set solutions. The research will directly consider the time effects associated with evolving requirements, cost basis changes, uncertainty and risk as key metrics for the future SBD hybridized analytical framework.

RESEARCH OBJECTIVES

Objective 1: Develop a rigorous framework extending SBD with formal definition and an analytical treatment of the design space created by system requirements, into feasible regions or "islands". The framework will provide a meaningful organization of the collection of feasible point solutions into regions such that:

1. The regions are distinct from each other in a meaningful sense relative to the design process and design changes
2. Within each region, the point solutions are like each other in a meaningful sense relative to the design process and design changes

Objective 2: Develop analytic methods to address the sensitivity of the topology of design space (i.e., the organization into regions) to the constraint levels, to understand critical levels of the constraints, and the effect of constraint levels on feasible regions of design space.

Objective 3: Extend the SBD framework to incorporate uncertainty in subsystem/component availability.

Objective 4: Develop an approach to collapse a feasible region of design space into one or more characteristic point designs that are most resilient with respect to changes in the constraints, i.e., such that changes in constraints tend not to change the characteristic point(s) for the region.

CHAPTER 2: REVIEW OF THE LITERATURE

The Literature Review defines the current state of the Body of Knowledge (BoK) into six categories and definitions:

1. Set-Based Design – General (important background)
2. Set-Based Design – Formulations (related algorithm/framework development)
3. Trade Space and Design Region Exploration
4. Resilient Processes for Design and Development (other advancements related to but not SBD)
5. Mathematical Tools and Methods
6. Risk and Uncertainty Effecting Development.

SET-BASED DESIGN – EXPERIENCE AND HISTORICAL EVOLUTION

SBD started at Toyota and is an extension of the Toyota Production System (TPS). More specifically, it is the analytical extension of their Concurrent Engineering process (Liker 1999). It also naturally flows from their evolutionary improvement process. SBD is primarily concerned with not down selecting to a single design too early and with maintaining design data for continuous reuse, updating and improvement.

Toyota learned early on to maintain multiple options throughout the design process. Process uncertainty directly impacts planned system integration. Multiple companies have tried to copy TPS. While some have been successful, others haven't. Typically, non-success is associated with not accepting the entire process. This extends to the use of SBD.

The U.S. Navy has decided to utilize SBD for their design process. Literature examples of military SBD programs are still seminal. The only current example is the Ship-to-Shore Connector Program. This is an ongoing program to replace the LCAC. The driving factors for choosing SBD for the Ship to Shore Program included a short design timeframe, starting from a blank sheet of paper and tight cost constraints. (Mebane 2011) Of note, this is the only defense program in execution that is using SBD, although there are several programs much earlier in cycle that want to utilize SBD.

The U. Navy is hoping to use SBD to affect rapid concurrent developments and to defer detailed specifications until trade-offs are more fully understood and they are uncertainty has been removed (Singer 2009). Their version of SBD differs from Toyota's process in that it must be compatible with the DOD acquisition process. This means that the requirements process is more deliberate than Toyota's. Additionally, the Navy's developmental process is more revolutionary than evolutionary compared to Toyota's. The U.S. Navy is already seeing improvements in their design practice with the use of SBD. They have even built the case for the use of SBD and design resilience in their Ship Design Manager Manual. (Unknown 2012)

The Navy as well as the Army are looking at using SBD to reverse a lengthy series of costly and failed large programs. The literature documents the Navy's complete reversal of changing its primary destroyer ship type from the newer Zumwalt class to the Burke class. This was done to enable a new major AAW upgrade. The Zumwalt class's design would have required a costly upgrade. The Navy contractors for the Zumwalt class could pick from a fixed set of combat requirements (an acquisition reform change) that allowed the Zumwalt class to be less compatible than the Burke class. Thus, the Zumwalt class's initial AAW structure was decided from a tighter design space. The GAO audit of this fiasco referenced the need and recommended that more alternatives early in the PD cycle would help reduce later design failures. (Martin 2012)

The key take-away from this section of the literature review is that SBD holds great promise and it has proven itself with several very good qualitative decisions. However, there is no analytical and quantitative vision or framework that has been developed to create and support a resilient PD process.

(Liker 1999)	(Singer 2009)	(Mebane 2011)	(Martin 2012)	(Unknown 2012)
Title: Toyota's Principles of Set-Based Concurrent Engineering	Title: What is Set-Based Design?	Title: Set-Based Design and the Ship to Shore Connector (SSC)	Title: Additional Analysis and Oversight Required to Support Navy Combatant Plans	Title: Ship Design Manager and Systems Integration Manager Manual
This is the definitive article that describes how SBD or Set Based CE (SBCE) is a key subset to Toyota's PD process. Key emphasis is made to holding back design "neck-downs" until necessary. Process uncertainty is the key driver in	This paper documents the Navy's need for evolving models and analysis tools to be compatible with, among other things, set-based design (SBD). SBD allows more of the design effort to proceed concurrently and defers detailed	This paper discusses the changes that a government-led design presents to design approach, including schedule, organization structure, and methodology using the SSC program as an example. The necessity for	This GAO audit documents the Navy's complete reversal of a primary ship design from the Zumwalt to the Burke class. A key recommendation to creating and maintaining more alternatives early in the PD process gives credence to SBD methods. Documentation of	This manual serves as a guide to managers involved in Ship Design. SBD and design resiliency are both referenced and desired. No capability to execute and enable a resilient

effectively conducting engineering with manufacturing.	specifications until trade-offs are more fully understood. This paper describes SBD principles, citing improvements in design practice that have set the stage for SBD, and relating these principles to current Navy ship designs.	implementing SBD was the desire to design SSC from a blank sheet of paper in a short time frame - 12 months. This paper describes SBD's application to SSC, the program challenges and methodology adoption.	sunk costs associated with "necking-down" design solutions too early are a key finding in the audit study.	design process is given. The need to having an SBD-based design process is implied.

TABLE 1 - SET BASED DESIGN - GENERAL

SET-BASED DESIGN – FORMULATIONS

The literature documents several new potential SBD formulations. The literature asserts that SBD is superior over traditional point design since SBD maintains a set of solutions later into the design allowing flexible choices and design changes because the solution set is not "necked down" to a single solution that may fail. This flexibility allows for the eventual solution to overcome design failure associated with uncertainty. (Gray 2011)

The literature proposes the use of fuzzy logic systems as a mechanism to inject probabilistic logic into the design process itself. Additionally, the use of fuzzy logic allows designers and engineers to better communicate uncertainty in the system design itself. The use of fuzzy design, was documented in the research to better model uncertainty and the negotiation of design variables. This was especially true when dealing with highly constrained designs. Fuzzy logic is a potential mechanism that could be applied in the framework as a black box process to represent uncertainty and risk. It is yet to be determined if there is another analytical mechanism available that could perform that role. (Gray 2011)

(Madhavan 2008)	(Gray 2011)	(Gray 2011)
Title: An Industrial Trial of an SBD Example	Title: An Applied SBD Communications System	Title: Enhancement of SBD Via Introduction of Uncertainty
A set-based multi-scale and multi-disciplinary design method has been proposed in which distributed designers manage interdependencies by exchanging targets and Pareto sets of solutions. The research showed that SBD has the potential to reduce costly iterations between design teams, relative to centralized optimization approaches, while expanding the variety of high quality, system-wide solutions. An industrial trial, with Schlumberger converted the existing Schlumberger design process into a SBD process. Results indicate that SBD delivers the benefits predicted in the laboratory, along with a host of advantageous side effects, such as a library of back-up design options for future design projects.	This presentation covers how Set-based design (SBD) is superior over traditional Point design due to the total uncertainty in design through the PD process. Use of Fuzzy Logic Systems (FLSs) to share design communications of system structure and components example are covered. The purpose is to affect a slower necking down of the set design space.	This dissertation looked at the effects of introducing uncertainty representation into a set-based design process. The hypothesis was that the introduction of design uncertainty would enhance the facilitation of set-based design practices. The results of this experimental research have shown that the inclusion of uncertainty modeling in the set-based design process for the negotiation of design variables enhances the overall set-based design progression, especially when working with highly constrained designs. This research has led to the enhancement of the set-based design process by providing capabilities to now represent uncertainty in the set-based design space though the use of fuzzy logic systems.

TABLE 2 - SET BASED DESIGN - FORMULATIONS

A practical example considered an industrial trial of a collaborative design that had Pareto combinatorial solutions. Their research showed that SBD could reduce costly iterations between design teams. Schlumberger, the manufacturer, converted their design process into an SBD process. They documented the variety of solutions as being richer and of higher quality relative to earlier centralized single-point optimization approaches. Future needs were described to develop a library of backup design options and development of SBD tools into their collaborative design process. (Madhavan 2008)

The literature concerning formulations showed that SBD is a better concept for maintaining flexibility in the design process, but that needs to be integrated with mathematical and analytical tools and processes. Both theorization and practical examples of analytical frameworks that directly address the uniqueness of the set-based design space are lacking or at best highly seminal.

TRADE SPACE AND DESIGN REGION EXPLORATION

The BoK does contain research into design space and regionalization for solutions. However, it is almost, if not completely focused on point solutions. However, we still gain some insights. Early design space methods, even though focused on point solutions, recognize that the actual state or trade space has specific mathematical characteristics that need to be exploited and analyzed to fully understand the design space. Issues around orthogonality and other mathematical topologies need exploitation to understand and utilize the design space properly. (Gries 2004)

Further insight extends that proper design space exploration directly impacts both design convergence and program schedule during design. Pareto analysis and a movement away from hard point solutions is discussed but the conceptualization of formalizing solutions away from point to set is lacking. (Thangaraj 2010)

Finally, functional design in the SBD context is reviewed. The U. S. Navy has recently developed a design space exploration and analysis tool which facilitates the sharing of design data and its preference structure. SBD was shown to do a good job of sharing variable, feasibility, and robust information for the design. The paper also proposed future work to develop and integrate more functional design information finally the paper related need to further develop design trade space exploration into the SBD process itself. However, the need for a mathematical topology for the set solutions themselves and the need to understand and manipulate the set solution design trade space is not covered. (McKenney 2012)

(Gries 2004)	(Thangaraj 2010)	(McKenney 2012)
Title: Methods for Evaluating Design Space Early in Design	Title: Rapid Design Space Exploration	Title: Influence of Functional Design in SBD
This paper gives an overview of methods used for design space exploration (DSE) of micro-architectures and systems. The DSE problem generally considers two orthogonal issues: (I) Single design point evaluation, (II) design space coverage during exploration. The latter question arises since an exhaustive exploration of the design space is usually prohibitive	Rapid and effective design space exploration at all stages of a design process enables faster design convergence. This is particularly important during the early stage of a design where design decisions impact design convergence. This paper describes a methodology for design space exploration using design target prediction models.	This article documents the first SBD application to a U.S. Navy design – the Ship to Shore Connector (SSC). While the program was deemed successful, SBD process improvements for ship design were described.

due to the sheer size of the design space. We explain trade-offs linked to the choice of appropriate evaluation and coverage methods. The designer must balance the following issues: the accuracy of the evaluation, the time it takes to evaluate one design point (including the implementation of the evaluation model), the precision/granularity of the design space coverage, and, finally, and the possibilities for automating the exploration process. We also summarize common representations of the design space and compare current system and microarchitecture level design frameworks. This review eases the choice of a decent exploration policy by providing a comprehensive survey and classification of recent related work. It is focused on system-on-a-chip designs, particularly those used for network processors. These systems are heterogeneous in nature using multiple computation, communication, memory, and peripheral resources.	These models are driven by legacy design data, technology scaling trends and, an in-situ model-fitting process. Experiments on ISCAS benchmark circuits validate the feasibility of the proposed approach and yielded power centric designs that improved power by 7–32% for a corresponding 0–9% performance impact; or performance centric designs with improved performance of 10.31–17% for a corresponding 2–3.85% power penalty. Evolutionary algorithm based Pareto-analysis on an industrial 65nm design uncovered design tradeoffs which are not obvious to designers and optimize both power and performance. The high-performance design option of the industrial design improved the straight-ported design's performance by 29% with a 2.5% power penalty, whereas the low power design option reduced the straight-ported design's power consumption by 40% for a 9% performance penalty.	This paper introduces a design space exploration and analysis tool, which was developed to facilitate sharing preference-based design data. Compounding effects of variables, feasibility information, and robustness analyses are shared in the design groups using set-based communications. Finally, this paper outlines future work related to the integration of functional design information, as provided by the design space exploration tool, into the SBD process.

Table 3 - Trade Space and Design Region Exploration

RESILIENT PROCESSES FOR DESIGN AND DEVELOPMENT

The literature is full of ideas to make the design and development process more resilient in a qualitative sense, but is generally lacking in an integrated, quantitative sense. These papers identified obstacles and gaps that will be addressed in our research. Multiple reviews of maturity assessment, data quality assessment and general design decision support were found. The literature also showed general improvement ideas around better collaboration and injection of sustainability into the PD process to generate more resilience.

Maturity assessment is very important in reducing uncertainty between technological capability and cost. The less mature an assessment, the greater the risk and uncertainty is to successfully complete the design and move to production. Multiple maturity assessment techniques are available and vary in performance and capability. However, maturity modeling and the use of maturity metrics in actual system decision models is not well integrated. (Azizian 2009). This is a major concern area for defense programs that are traditionally much more ***kaikaku*** i.e. breakthrough than ***kaizen*** i.e. gradual improvement focused such as at Toyota. Both sectors want a balanced design approach, but defense will always look for more breakthrough technologies.

Data quality assessment is also referenced in the literature. Clearly, data quality directly impacts all designs. Multiple methodologies are available to select, customize, apply and improve data. However, the application to design data is not well documented. Data quality assessment is found more in the use and development of software specifically and less in design generally. Similarly, to technical maturity assessment, data quality assessment needs to be properly measured and used in a coherent manner for design processes and design data. (Batini 2009)

PD as a process, requires the design team to share a similar culture for complete sharing of design information and data. Design logic can be viewed quite differently and color decisions based on geographical and cultural differences. Given that many companies and organizations operate globally, this simple problem perturbs the design in both technical value and data quality. Additionally, new concerns such as sustainable development and ecological efficiency add to designs greater requirements. This bolsters the need for a SBD approach that will allow balancing sustainability needs with both technical performance and system cost. (Fiksel 2003) Very recent examples of this trend are available. The most striking is perhaps Elon Musk's ventures: Solar City and Tesla. Mathematically, sustainability adds another dimension of valuation that is time dependent into the SBD state space.

Trade space exploration is another critical need documented in the literature, for effective design. Papers range from discussing the complexities and cognition effects that people have in a qualitative sense (Daskilewicz 2012), to an actual quantitative review of trade space exploration tools (Spero 2014). These framework papers supported the need for structure to trade space exploration but fell short of exploring the actual unique mathematical topology required for set solutions in their state space.

Critical program decisions are made based on the outcome of trade studies that employ both qualitative and quantitative analyses. Particularly, trade space exploration for engineered resilient systems is envisioned to coalesce pertinent information tuned to specific decision-makers and their needs. The literature documented the need for trade space exploration to ensure that attributes and data are accurate. The study presented best common practices for trade space exploration and that new tools were needed to track and document a richer trade space in order to create not just resilient solutions but a resilient process for design. (Spero 2014)

(Fiksel 2003)	(Azizian 2009)	(Batini 2009)	(Daskilewicz 2012)
Title: Designing Resilient, Sustainable Systems	Title: A Comprehensive Review and Analysis of Maturity Assessment Approaches	Title: Methodologies for Data Quality Assessment and Improvement	Title: Rave: A Framework to Facilitate Research in Design Decision Support
Sustainable development requires a systems approach. The development of sustainable systems is challenging because economic, environmental and social factors need balanced through the system life cycle. Traditional systems engineering is vulnerable to unforeseen factors. Resiliency leverages design diversity, efficiency, adaptability, and cohesion. Ecological efficiency improvements help, create value but broader systems thinking is needed. A design protocol is presented. The approach encourages explicit consideration of resilience in the total system design process.	This paper is a literature review of leading research and industrial practices for technology maturity assessment techniques. The focus is on balancing cost and technical maturity. The paper provides a review and analysis of maturity assessment techniques to provide a more resilient selection criteria for decision makers.	The paper reviews methodologies that help the selection, customization and application of data quality assessment and improvement techniques. It provides a description of such methodologies and compares them along several dimensions.	Product performance and value are limited by physics, economics and human cognition. Product Design improvement is a direct result of technology advancement and its advantageous application. Engineer training and experience limit design, but so do the tools that support decision making. These limitations imply substantive cognitive challenges from alternative conception through trade studies.

Table 4- Resilient Processes for Design and Development - 1

(Dremont 2013)	(Holland 2013)	(Spero 2014)
Title: Maturity Integrated in a Meta Model of Knowledge to Help Decision Making in Design	Title: Engineered Resilient Systems: A DoD Science and Technology Priority Area	Title: Tradespace Exploration for the Engineering of Resilient Systems
Different people and interaction in design decisions form part of PD. While collaborative, geographic and cultural differences color decisions. The PD process can be represented by the main sub processes from conception through design to production. Preliminary design represents the early stages of the design cycle. Of interest to us, particularly at this level, is the fact that the product being defined is the fulcrum of the PD process due to high uncertainty from a lack of important knowledge at this stage. Flexible decision support is needed early in PD.	This document defines the USG goals to add resilience to their PD process. The USG is concerned with expanding upon “fast, informed, adaptable” goals toward producing more complete and robust requirements, and generating requirements that consider many more alternative scenarios or designs. Their position supports SBD to create a more resilient PD process. They are also wanting to manage the perturbations or uncertainty associated with the original requirements as tactical usage evolves with adaptive enemies.	Tradespace exploration supports the Systems Engineering Technical Management Process of Decision Analysis by identifying compromises, revealing opportunities, and communicating the impacts of decisions across a system’s development lifecycle. Critical program decisions are made based on the outcomes of trades and the data coming out of tools and methods employing qualitative and quantitative analyses. Tradespace exploration for Engineered Resilient Systems (ERS) is envisioned to coalesce pertinent, timely information tuned to specific decision makers providing a holistic view of decision impacts on required system capabilities. This study reveals a fundamental insufficiency of ERS trade space exploration. What is needed is a deeper understanding of how these tools are used when performing tradespace exploration. This will enable users to better assess tradespace exploration and their tools. A review of 81 candidate tradespace exploration tools is provided. This study addresses the need to assemble a “best common practice” process for requirements, attribute definition and tool selection. A paradigm shift towards common tradespace methods, tools, cost models, and steps is emphasized.

Table 5- Resilient Processes for Design and Development – 2

The defense sector recognizes the need for greater adaptability and resiliency in programs. These are current, stated high priority goals to achieve. (Holland 2013) SBD is recognized as a process that will not only open the trade space, but also consider uncertainty. The usage of mathematical analytics and combinatorial tools are both implied and assumed. This logic is also carried into discussions on the need to understand and leverage maturity in PD (Dremont 2013). The author's dissertation directly ties on-going design maturity to reduce the high uncertainty early in PD. He proposes that maturity meta-modeling must be integrated into PD as a framework.

MATHEMATICAL TOOLS AND METHODS

The literature in this area is rather sparse. Generally, the BoK recognizes that using probabilistic methods is the main path forward in softening point-based solutions and recognizes that point-based solutions can create uninformed and even incorrect design directions. Then as uncertainty is lifted, recognition of the incorrect path is revealed. The literature here tends to view robustness more from the POV that the point solutions need to be looked at near where they're at, as opposed to carrying a set of solutions that directly covers design uncertainty and risk. Thus, regional description is for the area around the point solutions themselves. Recognition is given toward understanding that what may appear to be suboptimal solutions are the source of the final best design. This dovetails with the Toyota experience and their general focus to employ smaller, less uncertain changes rather than revolutionary break through designs. The key take-away here is that the solution will need to at least soften the point solutions probabilistically to more accurately reflect the solution's uncertainty.

Other research touching the very nature of solution clusters gives insight into how regions can be described since clusters share may joint edges relative to non-cluster members. Significant clustering research has been on-going for twenty to thirty years and forms the basis of data mining and the usage of genetic algorithms. Unfortunately, the clustering research has been focused to look at state space entities in current time or as a group stepping forward through time. There has been no application toward using clustering techniques through time and considering uncertainty. However, the uniqueness of the actual graph structures themselves does give insight toward framework development (Fortunato 2010).

Further research in this matter looks at the actual response surfaces of combinatorial solutions. The key point looking forward into research is that multiple convex hulls are required to conduct response surface approximation. A convex hull in this instance is a surface over a convex region described by multiple hyper-planes and their intersections. This is a key point, but further questions as to how to categorize multiple convex hulls and to carry them in a set solution will need further investigation. It is acknowledged that regions of solutions are needed to describe the solution space in the face of uncertainty (Goel 2006). In this vein, another article postulated adding probabilistic bounds to soften the point solutions. The author also tied the softening of the bounds directly to the cost of a robust design itself (Bertsimas 2004). This concept clearly leads toward, but does not reach our conceptualization of utilizing set solutions to both make individual designs more robust and to make the PD process more resilient. These analytical papers project from the point solution paradigm and fall short of touching and developing mathematics for set solutions.

(Bertsimas 2004)	(Goel 2006)	(Fortunato 2010)	(Azadian 2011)
Title: The Price of Robustness	Title: Response Surface Approximation of Pareto Optimal Front in Multi-Objective Optimization	Title: Community Detections in Graphs	Title: Dynamic Routing of Time-Sensitive Air Cargo Using Real-Time Information
Allowance for suboptimal solutions in linear programming problems is explored in this paper. Nominal values of the data are used to ensure solution feasibility and near optimality when the data changes. This paper investigates ways to decrease the price of robustness by adding probabilistic bounds.	A systematic approach is presented to approximate the Pareto optimal front (POF) by a response surface approximation. The approximated POF can help visualize and quantify trade-offs among objectives to select compromise designs. The bounds of this approximate POF are obtained using multiple convex-hulls. The POF is approximated using a quintic polynomial. The compromise region quantifies trade-offs among objectives.	Graph representation of systems' community structure, or clustering leverages the different organization of vertices in clusters where many edges join vertices of the same cluster while comparatively few edges join vertices out of the clusters. Such clusters, are independent compartments of a graph or regions of state space. This problem is hard and not solved, despite huge effort by a large interdisciplinary community of scientists. Significance of clustering and how methods should be tested and compared.	The route planning of time sensitive air cargo and delays is the problem reviewed. This paper is unique in that it directly models uncertainty through a novel Markov decision model for a true optimization problem.

TABLE 6 - MATHEMATICAL TOOLS AND METHODS

The final article reviewed (Azadian 2011), gives the clearest indication as to our framework development. The paper recognized the need to maintain a set of solutions for the routing of air cargo. The problem uncertainty is associated with routing and scheduling of cargo movement where the cargo system, weather and the maintenance of mechanical systems all interact. Since routes can modify both through locations and time, albeit in a nearer, shorter time window than the system design time window, it provides the greatest insight into research going forward. Considering all the other research, its proposal of using Markov decision modeling is insightful. The key take-away from this group of literature is that set solutions may need to have some form of Pareto optimal front approximation in some type of Markov chain modeling. The size of the support structure for set problems could very well overwhelm data storage and manipulation capabilities. Research is needed in both how to and how much of this logic to employ and to create both a sufficient and efficient SBD solution in the state space, with proper regionalization that will provide PD process resilience.

RISK AND UNCERTAINTY EFFECTING DEVELOPMENT

Our scope for the inclusion of risk and uncertainty in the framework will be a black box approach. The actual mathematical description and analytical development of risk and uncertainty for SBD will be done in parallel with this dissertation. However, risk and uncertainty still need to be addressed and modeled in the framework.

Inclusion of Uncertainty in PD runs the gamut. The lead paper discusses the need to conduct PD, integrated with Risk Management utilizing a Risk-driven Design (RdD) framework. The paper although conceptual highlights the need to have risk embedded in a PD framework to improve PD process resilience. This article builds part of the case for the research we plan to do. Quantitative analytics were discussed but referred to as a future expansion need into the BoK (Bertsimas 2004).

(Bassler 2011)	(Oehman 2012)	(Ratiu 2004)
Title: Risk-Driven Design - Integrating Risk Management with Product Development	Title: Lean Enablers for Managing Engineering Programs	Title: Using History Information to Improve Design Flaws Detection
This Product Development (PD) thesis investigates the integration of risk management and uncertainty as an emergent property of the PD approach itself. The Risk-driven Design (RdD) framework is used as a guiding concept for this research because RdD is the only risk framework that consists of solution-neutral principles. It represents objectives or outcomes of successful RM instead of prescribing specific and external processes on how to manage risk. Characteristics of resilience are introduced. Common PD frameworks were also analyzed. The analysis shows that existing PD frameworks only partially address resiliency in the design process. Design for Six Sigma methods yield the most comprehensive risk management oriented PD approach. Results from a comprehensive survey among North American companies with 185 respondents showed that companies currently focus their RM effort too much on technological risks and too little on customer related risks. A modified PD framework that utilizes RdD is proposed in the thesis.	This paper presents lean enablers for managing engineering programs. Programs fail or succeed primarily based on people, not processes or tools. Unstable requirements increase the cost of R&D. Trade space exploration, should include sets of architectures. This paper supports the need for a more resilient design process that considers the impact of requirement uncertainty and lack of trade space on program costs.	This paper addresses how uncertain design flaws directly impact system maintainers. Accurate and automatic identification of the design problems are needed. Essential information as design problems appears later and evolve over time. The approach to use historical information of the suspected flawed structure may increase the accuracy of the automatic problem detection. A large-scale case study is shown to improve the accuracy of the detection of God Classes and Data Classes, and additionally how it adds valuable semantic information about the evolution of flawed design structures. This is an example of how design uncertainty may be reduced.

Table 7 - Risk and Uncertainty Effecting Development – 1

A key takeaway was that Lean Six Sigma practices appeared to improve PD process resilience. This correlates with other articles that tie the potential use of probabilistic math to PD process resilience. This thought is echoed and modified in another article that presents using lean enablers to deal with requirement uncertainty and to open up the design trade space (Oehman 2012). Both papers point to Six Sigma to improve PD process resilience for programs.

(Wang 1995)	(Zhou 2011)
Title: A Framework for Analysis of Data Quality Research	Title: Application of Data Maturity in Product Development Process Control
Organizational databases are pervaded with data of poor quality. However, there has not been an analysis of the data quality literature that provides an overall understanding of the state-of-art research in this area. Using an analogy between product manufacturing and data manufacturing, this paper develops a framework for analyzing data quality research, and uses it as the basis for organizing the data quality literature. This framework consists of seven elements: management responsibilities, operation and assurance costs, research and development, production, distribution, personnel management, and legal function. The analysis reveals that most research efforts focus on operation and assurance costs, research and development, and production of data products. Unexplored research topics and unresolved issues are identified and directions for future research provided.	In the concurrent and collaborative product development process, the realization of quantitative analysis and overall control of the development process management is the key to improving PD. The paper throughout describes the concept and role of data maturity, new methods to conduct quantitative analysis and overall control of collaborative product development process based on the idea of data maturity was proposed in this paper. Based on the explanation of the theory of data maturity to control the process, the differences and relationship between data maturity and milestone was analyzed. The division method of data maturity level was discussed, and the corresponding relationship between each data maturity level and completion degree of the digital product model was illustrated also. In addition, the overall form of the product collaborative development process based on data maturity was provided. Finally, the detailed process between product structure design, process design and tooling design driven by each data maturity were explored, which realized the quantitative analysis and overall control of the product development process.

TABLE 8- RISK AND UNCERTAINTY EFFECTING DEVELOPMENT - 2

The next grouping looks at uncertainty and risk elements that directly impact the design and not the risk process itself. They include: usage of history information (Ratiu 2004), data quality (Wang 1995) and data maturity (Zhou 2011). Ratiu postulates that history information of designs is not only valuable to baseline new design elements, but that it also provides insight into reducing design flaws. This echoes defense sector concerns that maintainer data has been ignored at great cost to new programs and that there is great value in using history information to reduce design uncertainty itself. Wang's article, although dated, is still valid today. Data quality, if not dealt with directly impacts design. He provides a framework of data quality research that is useful. Unfortunately, there is not a newer similar framework paper to reference.

Zhou's article ties in with other papers because it directly correlates the need to actively mature product data to reduce uncertainty and risk in PD. His example which directly looked at not only design but tooling design added further conviction that uncertainty must be addressed and then as the data matures it should directly evolve the design and neck the set solutions down.

LITERATURE REVIEW SYNTHESIS

The BoK supports the theoretical foundation that uncertainty and risk severely impact designs as they morph and change in the PD cycle. Multiple foundational examples of failed programs, particularly complex and high cost programs are directly linked to a lack of rich alternatives in a robust design space. Furthermore, failures are tied to design process brittleness that is unable to deal with subsystem, assembly or component failures associated with uncertainties being realized through time in the PD process. The ability of SBD to both wax and wane set solutions in the face of risk and uncertainty not only provides great promise but there are multiple qualitative examples that have resulted in improved designs and the derived resulting industrial profits.

SBD is not only promising with designs, but there are companies and programs that are effectively using it to remove design process brittleness to improve their PD processes. All the literature points to the need of coupling quantitative mathematical analysis into the PD process to create not just more process resiliency, but cost-effective and technically improved solutions. The need for a framework that couples SBD with mathematical analytics is both real and overdue. Novel quantitative processes in point solution methodology whether with genetic algorithms or cutting edge combinatorial, stochastic based optimization are continuing to be worked on and documented in the BoK. However, extensions into or in the usage of an analytical process coupling SBD logic with fast analytical mathematical tools in a framework or in practice are missing or so seminal as to be unavailable now.

Without adding a quantitative basis to SBD, it will remain merely a very good, but qualitative effort associated with organizational conference room decision making. This is not disparaging, but rather is encouraging in that SBD has already proven itself in decision making. Therefore, adding a quantitative nature to SBD that keeps the richness of SBD facing and agilely dealing with risks in PD, while adding a soft heuristical or even a highly flexible optimization process, is both unique and novel.

CHAPTER 3: METHODOLOGY

THEORETICAL FRAMEWORK

The theoretical mathematical framework (in purple) below shows the macro data processes at high level, that the framework will need to work and interact with, as it progresses through epochs of the Product Development cycle. We will develop framework formalisms and then showcase them in the Proof of Concept example problem. Additionally, Figure 3 also shows the real-world interactions and connections that a framework must maintain to its evolving data sources. Although outside of scope in this research (in blue), it should be noted that no framework can be resilient without easy and quick access to good data. Set-Based Design considers the uncertainty of data, but it will function better with good data.

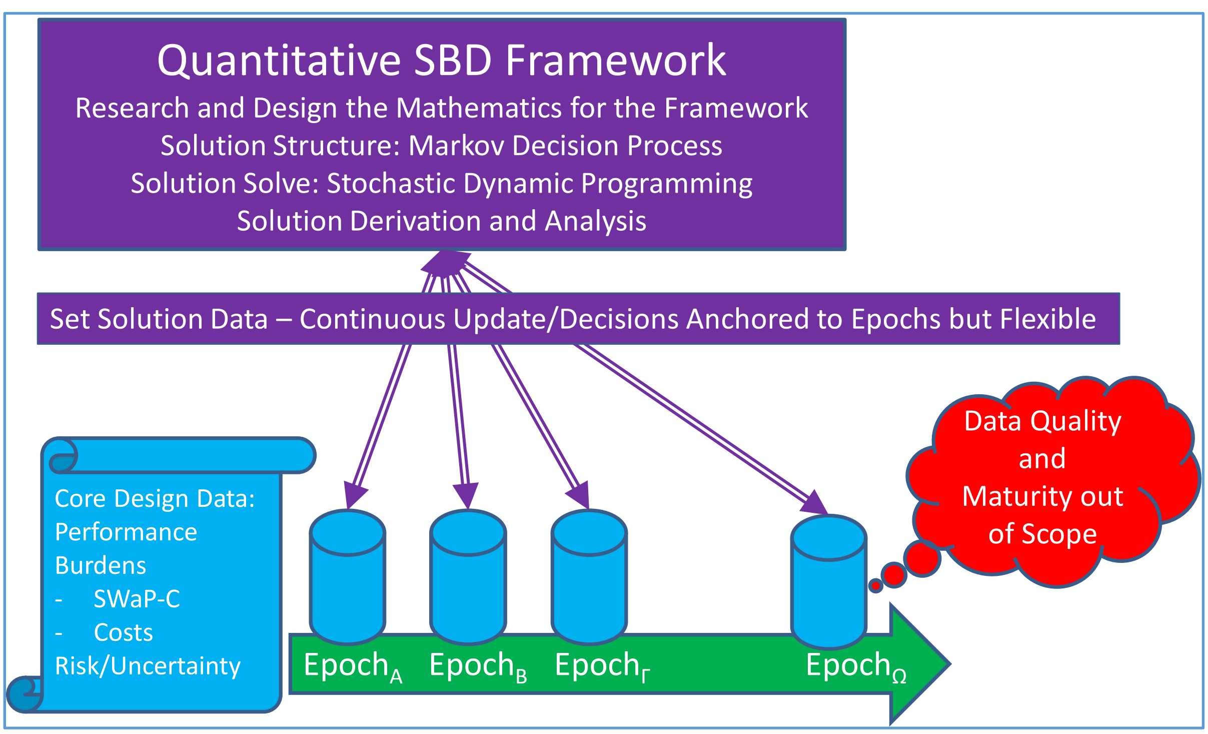

FIGURE 3 – HIGH LEVEL SBD BASED PD FRAMEWORK

Fundamentally, our framework will focus development into creating and employing a continuous decision process where SBD solutions are carried forward through design milestones or epochs. Current milestone thinking is that there is fundamentally a "necking down" of potential solutions at every milestone. Our framework will keep set solutions longer into the process to avoid early mistakes as the uncertainty veil lifts and to also provide a rational, technology set of alternatives if an earlier "neck down" in the set solution proves to be deleterious.

The future framework will allow for continuous design updating by developing and maintaining regional characterizations that provide insight into technology interplay and toward dependencies that move prove to be deleterious to the system solution as uncertainty lifts.

Significant effort, as raised in the earlier questions is required to understand how to apply and correlate solution information for set solutions is going to be done. This is truly a new frontier. Great effort has been made in the past to create mathematically rigorous and correct algorithms and solutions for point based systems. Further effort has been made to describe SBD to avoid the uncertainty trap that proven time and time again to fail the very best and most expensive programs. However, little to no formulation has been accomplished directly with SBD outside of singular programs. More qualitative process improvement has occurred, but the point remains that point solution mathematics are far ahead of SBD.

This framework will lead in hybridizing SBD with current mathematical techniques. Research and insight into how set solutions and their regions will be conducted to provide a rational path beyond the current heuristic thinking. Significant effort will have to be made in understanding the data storage, data mining and analytical manipulation of set solutions.

Finally, both cost effective logic and uncertainty logic will be employed to more accurately reflect the true unknowns as preliminary design concepts morph through time and modify in the face of evolving requirements, programmatic impacts and technology movement. Continuous system integration processes, as designs morph need to become resilient. It appears that the next major research area will be in the coherent application of SBD logic with mathematical analysis inside a data-driven framework that can provide timely guidance through time that is sensitive and inclusive of uncertainty.

RESEARCH QUESTIONS

Research questions include how to represent regions of design-capability space in a manner suitable to perform both the constraint restriction and relaxation operations of set-based design and the partition design-capability space into connected regions. This is two sides of the same coin which focuses on how to explain the distinguishing characteristics of different regions. This leads to following considerations:

What properties should formulation of "regions" have?

What sorts of operations should be performed for the set based design/resilient process itself and separately the resulting resilient system designs?

Conditions of uncertainty will include: the time and cost of technology maturation, the funding of that maturation, and specifically the individual performance and burden elements of the subsystem options and the system itself. Burdens will include: Size, Weight and Power – Coolant needed (SWaP-C), costs, risks and required Reliability, Availability, Maintainability – Durability (RAM-D). This leads to following considerations:

How do the design choices, both point and set based, effect technology maturation uncertainties?

Can these uncertainties be reduced or better managed with SBD?

The interplay of design choices and dynamic funding relative to system and sub-system characteristics is an area of exploration. What level of cost improvement can be expected with SBD?

Funding for different technologies can be correlated for functional reasons that are not related to the underlying scientific, engineering challenges. Technologies may compete to perform the same function, and by extension compete for funding, which when given resource scarcity usually results in a cancellation of at least one. Technologies may be synergistic where they perform different functions with a positive interaction effect (the whole is more than the sum of the parts), in which case funding and/or development progress for one increases the likelihood of funding for the other. What are the interplays between the main program forwarding its key technologies and smaller off-shoot engineering change efforts? Commercially, this same question relates between ***kaizen*** (evolutionary, continuous change) and ***kaikaku*** (revolutionary, radical change) and its impact conducting concurrent engineering. (Liker 1999)

Conditions of uncertainty related to operational needs and conditions include the "threshold" and "objective" capability requirements. This is where system function performance and life cycle parameter levels are measured relatively as to the importance of the different capabilities and their employment. The level of capabilities provided by a system can affect the operational needs and conditions: adaptive adversaries will choose theaters, tactics, and materiel that avoid the strengths and exploit the limitations of our systems. Can this uncertainty impact be modeled in the framework?

The region representation should be symmetric with respect to time in acquisition, i.e., the same representation used for necking-down the option set, should also represent the potential upgrade options. How will the regions characterizations modify or change through program time?

Initial clustering and technology off-ramp solutions need identified. The following technical questions apply:

1. How does the clustering define and characterize the regions?
2. How will the framework define the on-going clustering over time?
3. Optimization runs for each "set element" are focused on individual solutions, but by definition, each run may include a vast number of pareto-optimal solutions. How will this modify the definitions of the set itself?
4. Are database technologies sufficient to maintain a vast super set of mathematical solutions where each solution set represents a single discrete "set element"?

IMPORTANCE OF THE STUDY

The dissertation will extend the concepts and methods of set-based design with a rigorous, practical theoretical framework to improve resilience in the PD process itself, under conditions of uncertainty that represent risk and ultimately program opportunity. This is under developed and there is little to no information in the BoK beyond basic qualitative examples.

A resilient system process is by definition, adaptable and extensible. Resilient systems that are instantiated in a set-based framework are economical for modification and allow additional or enhanced capabilities to be applied during design. This will allow programs to meet changing operational needs and conditions as well as exploitation of newly mature technologies. Currently, only qualitative examples of SBD-enabled resilient PD processes are known. This effort will extend SBD research into a mathematically rigorous, quantitative venue.

A resilient system process is also robust operating effectively in a wide range of external conditions to include incremental system degradation from damage to or degraded capabilities of internal components. Resilient systems are trustworthy having predictable performance in a wide range of natural conditions and during damaged or degraded operational modes. This effort will enhance PD process resiliency by allowing and defining quantifiable SBD analytics into a verifiable mathematical framework.

A resilient system process is reliable having lower rates of critical mission failure and provide backup reduced modes to allow for easier maintenance recovery. They also have low accumulation rates of maintenance-action failures and lower system failure rates due to fewer total component failures.

The primary focus of this research is on the “adaptable and extensible” aspect of resilience. Once the concepts and methods are developed, examination for application and adaption will be addressed for other aspects of resilience. This research area is ground breaking.

Resilient design can be viewed from multiple perspectives. System development can be resilient regarding the design specifications and architecture as the design morphs during concept and engineering development in the PD cycle. A completed system design itself can be resilient where the design and manufacturing carries variations to deal with the unknowns that become real during the PD cycle. Additionally, components and assemblies can be individually resilient so that equipment can be upgraded or repurposed during RESET and RECAP phases. Concurrent, program integration of this type and magnitude is unknown in the defense sector and can only be seen in a qualitative sense from the Toyota and other commercial examples.

Conceptually, set based design, keeps a broad a range of options throughout PD and specifically the design phase. (Singer 2009) The design process itself allows iteratively restricting and relaxing constraints to define the “open” options, until a single design remains. Constraints come from different stakeholders. This is a resilient development perspective.

Set based design can be extended to the resilient system design perspective. Here, a specific design is viewed as the set of all potential future options and variants which in the systems perspective is a combinatorial set of subsystems. The system is a platform that can support a system family and continuously is open for upgrading and modification. In this perspective, set based design keeps the greatest range of potential capabilities and future options open, subject to uncertain funding and operational conditions. The focus of this research is on designing resilient systems or platforms and by later extension to development.

The key mathematical focus extends from the need to have the PD process itself resilient. Point solutions, if optimal, are in a convex set of feasible solutions where the optimal solution can be found. That set of point solutions occupies a state space or a trade space that is convex. Other heuristic based mathematical solutions may be employed if the region of feasible solutions, the trade space, is either too big or not completely convex. In that situation, the best solution is understood as requiring additional qualitative analysis, but it remains the best point solution or is good enough that it can represent the best point solution. Combinatorial optimization yields multiple pareto-optimal solutions in a convex trade space, but it differs in that it is sophisticated with its weighting of the metrics. Thus, one may have a most reliable solution, a cheapest solution, a best-balanced solution in the combinatorial scenario. However, again these are point solutions for a given discrete point in time.

Set solutions have a region associated with them where they are characterized by a given technology or other set decision factor such as: cost or even programmatic issues outside of PD. Additionally, it is understood to carry uncertainty with it. These regions then quite simply contain and describe a group of related point solutions that are an aggregation which is the set solution. A combinatorial or other mathematical tool may be able to identify that set's region for a given snapshot in time. However, going over time, the very changes of the technologies and reduction in uncertainty, as risks dissipate or evolve, adds a degree of fluidity required to carry and map that set forward through time as the set itself evolves. Regionalization of sets, their descriptions and needed development to extend mathematical analytic tools is central to this research. Most importantly this is a way point in SBD research, as very little effort if any effort, has been made to add system level quantitative analysis for SBD solutions. Understanding and knowing how to use set solution regions is the next step to adding resiliency to the PD process.

RESEARCH SCOPE

1. The framework will focus on the formalisms needed to adequately describe state space regions for the SBD problem
    a. Region topology
    b. Inclusion of cost and uncertainty weighted versus technical performance
    c. Other burden parameters such as size, weight, etcetera will be covered
    d. State space region characteristic changes over time
2. The POC will be limited to a single system and not a more complex system of systems
3. The POC will have two hierarchies: system and subsystem
    a. Consideration of higher granularity or more detailed hierarchies such as assemblies, sub-assemblies and components is out of scope
    b. Burden parameters will be scaled for the POC
4. The framework will provide the mathematical formalisms necessary to execute a software development enabling set solutions indifferent to mathematical solver
    a. A current combinatorial tool/solver is available and will be used during the research

b. The framework will allow for the use of other tool/solvers such as genetic algorithms and other high-speed heuristic mathematical tools

5. The scope of research regarding mathematical formalisms is neither focused toward the defense sector nor the commercial, private sector. Information and examples are included from both sectors. This is because PD is similar enough across both sectors. However, the defense sector has a generally unified, systematic PD structure which lends itself to better examples since the commercial sector can choose to accept or reject any of the systems engineering processes that the defense sector follows.

    a. The scope for the POC will be based from a defense sector problem

    b. The time scope is shown below in Figure 4

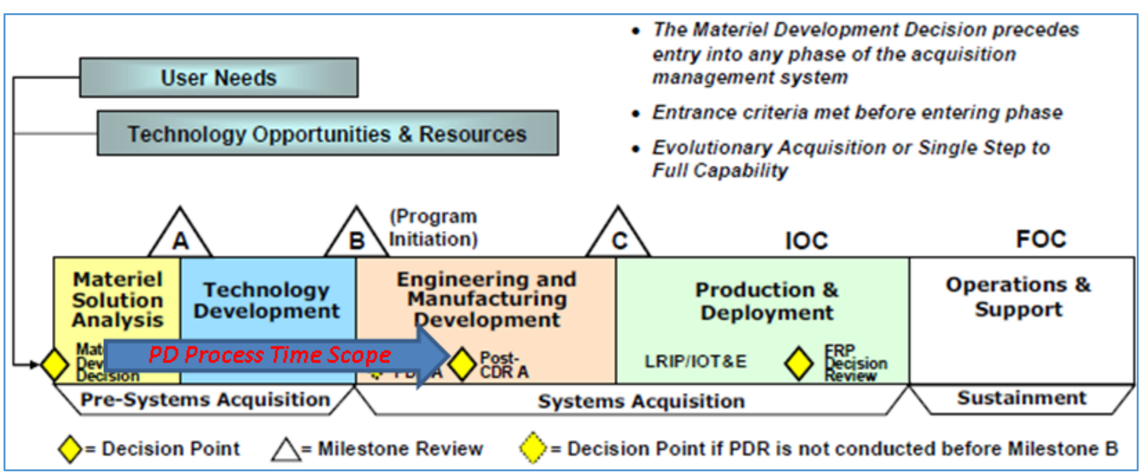

FIGURE 4 – PD PROCESS TIME SCOPE FOR THE DISSERTATION

Research Approach

Development

OBJECTIVE 1

Objective: Develop a rigorous framework extending SBD with formal definition and an analytical treatment of the design space created by system requirements, into feasible regions or "islands". The framework will provide a meaningful organization of the collection of feasible point solutions into regions.

We will develop a formalism that supports set solution region discovery and characterization to be flexible and to address:

1. regions of design space and regions of capability space that is requirements based and supports dynamic, evolutionary modifications of those same requirements
2. different, dynamic and evolving search needs of the design space throughout the PD cycle

We will utilize an optimization solver to both enumerate and define all optimal, feasible and in-feasible system solutions for an example design problem. Once we find all potential system point solutions we will conduct research into the mathematical or state characteristics of these solutions to determine what the actual topology and makeup of set solutions. We are expecting to receive future data sets from both the Tank Automotive Research Development Engineering Command (TARDEC) for a future armored vehicle development program. The purpose will be to implement a medium scale real world problem.

OBJECTIVE 2

Objective: Develop analytic methods to address the sensitivity of the topology of design space (i.e., the organization into regions) to the constraint levels, to understand critical levels of the constraints, and the effect of constraint levels on feasible regions of design space.

We will develop a formalism to manage the set solution space throughout the PD process:

1. Support the gradual necking down of sets
2. Allow programs to proceed without failing cost containment
3. Properly entertain risky, more promising technologies through the PD decision timeline
4. Allow for programs to "fall-back" to safer, within cost, subsystems as risk and uncertainty clear over time, to avoid program cost containment breeches

We will continue to utilize the optimization solver to resolve the trade space as time modifies it, but we will also research the solutions sets themselves to determine how the actual morphing of the constraints and requirements change. This directly impacts the waxing and waning of the set solutions themselves and the fulfillment goal to eventually neck down the set solution to the best production solution.

We will also look at the actual database needs for the design problem itself considering the changes over time. We will consider the state space needs, the metrics and other associate problem constraints, variables and requirements as well.

We will support the basic operations of set-based design and tradespace exploration with mathematical analytical methods:

1. The POC itself, will utilize a current combinatorial optimization algorithm to generate example set solutions
2. We will develop a formalism to allow for multiple analytical methods beyond mathematical optimization
3. This is not restricted to just extending to genetic algorithms, but to any heuristic based mathematical solver

The use of continuous optimization solving and database structuring to maintain the set solution movements will be researched to define the eventual framework itself. This objective will support set solution management for design programs as they move through design time.

OBJECTIVE 3

Objective: Extend the SBD framework to incorporate uncertainty in subsystem/component availability.

We will develop a formalism that addresses the uncertainty associated with both the evolution inherent to the PD cycle and the inclusion of risk as a direct parameter modifying the set solutions:

1. The PD process typically starts with requirements that carry unknowns especially regarding cost and technical fulfillment
2. Risk is a combination of technical fulfillment, cost and schedule unknown impacts

3. Uncertainty and risk heavily interplay with the actual fulfillment of programs and a lack of considering them adds a brittleness to the progression of the design itself

No research will be done to develop a metric or measuring system for either risk or uncertainty in general. We will take a black box approach in assuming that the framework will support analytical risk and uncertainty scoring. This is a different and out of scope effort for this dissertation. However, we will treat and test the usage of uncertainty and risk metrics as to how they impact set solutions as with the other metrics and mathematical elements listed and described in Objective 2.

OBJECTIVE 4

Objective: Develop an approach to collapse a feasible region of design space into one or more characteristic point designs that are most resilient with respect to changes in the constraints, i.e., such that changes in constraints tend not to change the characteristic point(s) for the region.

We will continue to utilize the optimization solver and database tools above to store and understand the developed design information for the set solutions. We will research the means to accurately describe the representative or characteristic region point (s). Potential exists for utilizing closest Euclidian distance, most shared sub-systems or other metrics. The actual technique and validation is part of this research.

DEMONSTRATION AND EVALUATION

The research will develop, in the Proof-of-Concept (POC) example problem, validation for the objectives. As stated above, we are expecting expanded real data from a customer, to further validate our findings and move the research into enterprise. It is expected that an expanded POC for this larger problem set will be created and documented. The POC example problem itself, will research to determine what is needed for future framework expansion both academically and commercially. It is anticipated that the partner data providers will help with the actual validation and assessment of the framework itself but that is all out of scope for the dissertation effort.

The POC example problem will be used to identify computational requirements for the usage of future larger data sets versus the developed framework conceptualization and formalisms. It is also expected that partners will support in validating the accompanying SBD topology in the framework itself, post-dissertation.

Appendix A includes a detailed example of how the framework would work with a small sized, but near real-world automotive upgrade program for a generic Main Battle Tank (MBT). This example explains the framework in a context where designs are the kernel data elements i.e. designs are the points in a set solution. This is the easiest and most visual mechanism to understanding how set value significantly differs from specific point value. However, when the algorithmic structures are created in the framework, we will utilize options as the kernel data elements for solution creation. This presents no issues as every design is a unique set options and every set of options describes one or more designs, i.e. a design set, which are supported by the option set. Thus, options and designs can be expressed uniquely and a simple mathematical transform is all that's required to know which designs are represented by options and vice versa.

CHAPTER 4: MATHEMATICAL FRAMEWORK AND EXAMPLE PROBLEM

SBD is a proven qualitative process associated with organizational conference room decision making. This is not disparaging, but rather is encouraging in that SBD has already proven itself in decision making. Therefore, designing a quantitative form of SBD that keeps the richness of SBD facing and agilely dealing, with uncertainty in PD, is unique and novel.

EXTERNALITIES AND APPROACH

Many systems, especially military systems, have protracted development lifecycles. During development, various external factors that influence design decisions often change. The challenge is to develop a system that ends up being cost-effective and is cost effective to develop, despite changes in the externalities during development. A development process that meets these challenges is resilient with respect to changes in the externalities.

The externalities we are considering are: (1) the relative values of system performance, system burden, and unit production cost, and (2) the development cost, time and uncertainties of candidate technologies/ options. The external factors have a known value at any point in time, but their final value, when the development is over and the system enters production, are unknown until the end. They are indeed "random" variables. Traditional point design treats the externalities as "deterministic". As a result, reactions to changes can incur greater costs and/or performance compromises than if the development program had considered potential variability of the externalities.

The approach we are pursuing is Set-Based Design (SBD). SBD carries a (redundant) set of options and alternatives forward during development, incrementally winnowing and adjusting the set as development proceeds and in response to changes in the externalities, leading to a final point design for production. The principle of SBD is to keep a broad range of options under consideration "as long as possible" to provide resiliency to changes in the externalities.

FRAMEWORK OBJECTIVES

Our research objectives are:

1. To give a rigorous formulation to the principle of Set-Based Design,
2. To give a rigorous analytical approach to make set adjustment decisions,
3. To give a rigorous analytical formulation of when to make set adjustment decisions.

EXPECTED NATURE AND SIGNIFICANCE OF THE RESULTS

We expect that set solutions can be valued to not only include the values of the set's point solutions, but also the value of the set with respect to reducing design uncertainty and increasing process resilience for PD programs. A small example framework problem is constructed to show how a SBD problem can be developed. This example problem utilizes a Markov Decision Process (MDP) with a Dynamic Programming (DP) backward propagation algorithm to optimize the SBD Contribution-to-Design. This "Contribution-to-Design" is stochastically determined and is expected to yield a more resilient, and potentially, a more intuitive design process through time. Finally, we compare the example solution to a traditional single point design scheme. This provides insights into future development to quantify set value versus point value to extend quantitative optimization into the current SBD qualitative processes.

FRAMEWORK FORMULATION

The high-level framework process uses a stochastic process to define the value, i.e. *Contribution-to-Design*, of developing multiple options, in a set from PD milestone/epoch to epoch. Contribution-to-Design is a combination of system performance, production cost, development time and cost. We treat the Contribution-to-Design as a black box treatment (allowing flexibility of application) to seed the values required to develop a Markov Decision Process. This is standard stochastic automata with utilities that presume the "memoryless" property, where actions taken in a state depend only on that state and not prior history. We then recursively solve the problem as a Dynamic Programming model utilizing Bellman's Equation with no discount to determine the optimal action (Bellman 1956). It is worth noting that this is neither direct discrete optimization of a design's characteristics, nor Pareto combinatorial optimization that yields a non-dominated set of point solutions. The three methods will be compared using example problem data.

THE DECISION PROBLEM

We assume a system consists of several subsystems. Each subsystem has alternative technologies and design options to meet the system requirements. A final point design consists of exactly one choice of an option for each subsystem. In the general case, not all the subsystems will be present in the final design, so the option of "none" is a realization option for each subsystem.

During development, technologies and design options are selected or rejected, subsystems are designed, and integrated in a task network organization leading to complete prototype production and testing. Subsystem design and integration take time and incurs cost, and the time and costs depend on the technology and design option choices. The end system performance, production cost, and the development time and cost all depend on: (1) which subsystem technologies and design options are chosen, and (2) when the selection/rejection decisions are made.

During development, information regarding the distribution of the external factors changes. Over time, as more data become available, estimates of the means change and uncertainty generally decreases – although it is possible that uncertainty can increase.

The decision problem is to select or reject subsystem options at the appropriate milestone or epoch to achieve "best value", the Contribution-to-Design at the end of the development program, despite adapting to changes in the externalities.

For the illustrative example below, we will assume that we only have three total system designs: D1, D2 and D3 available from allowable combinations of subsystem option. These three designs enumerate into seven "Set Solutions": SS1, SS2 … SS7. Additionally, we also show the possibility of skipping design work.

In Figure 5, the PD milestones/Epochs are arranged in time order, such that Epoch A is first and Epoch Ω is last. The set solutions are all the possible combinations of the three designs and are shown up to Epoch Ω. At Epoch Ω, the last epoch, the final design must be selected, so only the designs are shown. This is a visual representation of the Set-Based Design Process for considering which designs to develop. The colored arcs below represent one (yellow) or two (red) designs from not developing a design in the previous epoch. In most programs, these are possible but unlikely. For example, SS1 in B going to SS2 in Γ requires catching up all the work for D2 missed before B.

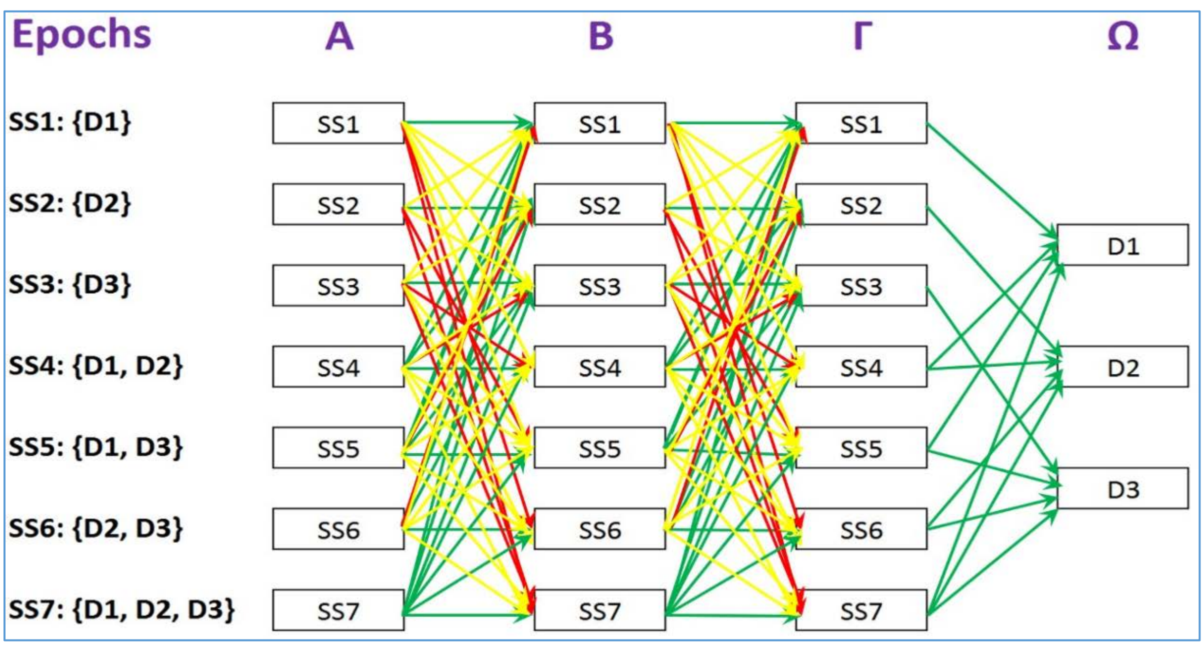

FIGURE 5 – EXAMPLE PROBLEM – TOTAL SET SOLUTIONS

DECISION SITUATIONS AND DESIGN DECISION STRATEGIES

There are four macro decision situations the framework supports:

1. When new information regarding the distributions of the externalities becomes available,
2. When an option enters the critical path, i.e., if the design and integration of the option does not begin immediately, but has begun later, it increases the total development time,
3. When an option reaches the point at which keeping it under consideration, development or integration begins to incur costs at a higher rate, and
4. When new information indicates that a new option is no longer feasible.

Situation (a) involves reconsideration of the entire set-solution. Situations (b) and (c) involve only the decision to keep or reject the specific option. Situation (d) requires the problem to be restructured without the option by removing any system design choices that included the option.

Design decision strategies for a point solution are restricted to either modifying the current point solution or replacing it with a new point solution. For set solutions, the choices are more resilient. One can choose to neck-down (i.e., reduce the set size), open-up, or completely modify the set solution.

The following are examples of design decision strategy changes:

- Neck-Down Example: Epoch A: SS7 to Epoch B: SS6. D2 dropped at Epoch B.
- Open-Up Example: Epoch A: SS3 to Epoch B: SS5. D1 added at Epoch B with recovery costs.

- Complete Modify Example: Epoch A: SS4 to Epoch B: SS3. D1 and D2 dropped and D3 added at Epoch B with recovery costs.

FRAMEWORK DIMENSIONS

- Epochs aka Milestones (E) – Discrete Time points from A to Ω where:
    - Ω = total number of epochs (Ω =3 in the illustrative example):
    - t is the epoch index
    - Design Data is updated, reviewed and then used to make forward programmatic decisions from $A, B, \Gamma \ldots t+1, \ldots \Omega$
- System Designs (Δ) – the allowable individual system designs possible from the system trade space from 1 to D where:
- D = total number of allowable system designs (D=3 in the illustrative example):
    - d is the design index
    - A design must include exactly one option for every subsystem or technology under consideration in the trade space
    - Each System Design, Δ_d is a unique combination of technology options of exactly one per subsystem and is both a point solution and a singleton set solution
    - Other set solutions are the combinations of multiple system designs (see Figure 1)
- Subsystems or Technology types (SS) – The discrete technology subsets that are required to frame a complete system choice where:

- I = total number of technology subsets or subsystems (I =3 in the illustrative example)
- i is the technology subset or subsystem index

- Options (O) – for every technology subset or subsystem there are a set of options available where:
  - J_i = total number of options per i^{th} subsystem ((J_1, J_2, J_3) = (2,2,2) in the illustrative example)
  - j is the i^{th} subsystem option sub-index
  - K = total number of options in the trade space (K =6 in the illustrative example)
  - k is the iterative vector transformation index taken from of the (i,j) option pairings

TOTAL DEVELOPMENT COSTS

The framework calculates Development Costs for every option epoch to epoch. Recovery Costs are calculated for options not developed in prior phases. Finally, System Integration costs are calculated for all phases. The framework considers the following:

1. Options may share certain costs with other options,
2. Reduce or Increase System Integration costs dependent on the shared development,
3. Recovery Costs are calculated for all options where the timeline permits option recovery,
4. There are no recovery costs if the new Set Solution is the same as the old,

5. Recovery costs are lower if the new Set Solution shared development with the old, and
6. Recovery costs are higher if development needs to be made up.

CONTRIBUTION-TO-DESIGN (CTD) VALUE

We assume that a PD program will consider an initial set solution at the beginning and then will potentially modify that set solution at the different program milestones/epochs based on new information as time progresses and uncertainty clears. The Contribution-to-Design (CTD) is calculated for each set solution at each epoch. The framework supports the natural PD cycle where subsystem options are typically developed somewhat independently early and then require more system integration later. The following Table 9 shows the relationships between the framework dimensions as PD advances through the epochs and phases.

Epochs	**Key Dimensions**	**Value Calculations**	**Development Costs (DC)**
A	Options	Set Solution CTD	Options Dev (A to B)
B to (Ω - 1)	Options	Set Solution CTD	Options Dev + Recovery Costs (B to B+1)
Ω - 1	Options	Set Solution CTD	Options Dev + SI Costs (Ω-1 to Ω) *** Final Design Set
Ω	Designs	Point Design Value	Options SI Costs (Ω to TP) for the Optimal Design (Δ*)
Test/Production (TP)	Final Design	NA	NA
Phases			**Cost Control Equations for Phase (E^t to E^{t+1})**
E^A to E^B			$\sum_{k=1}^{K} DC_k$ <= Phase Development Budget
E^B … to … $E^{\Omega-1}$			$\sum_{k=1}^{K} DC_k$ <= Phase Development Budget
$E^{\Omega-1}$ to E^{Ω}			$\sum_{k=1}^{K} DC_k + \sum_{d=1}^{D} \Delta_d\, SI$<= Phase Development Budget
E^{Ω} to E^{TP}			Final Design SI Costs <= Pre-Test Development Budget

TABLE 9 – FRAMEWORK DIMENSIONAL RELATIONSHIPS

In Table 9, we show what the key dimensions, values and costs are that the framework must track and calculate, to data populate the Markov Decision Process algorithm. We also show, what the required cost balance equations are for each phase of work, i.e. epoch to epoch. At

Epoch Ω, we down select to the final design. At Ω, the option development is complete, so the point design values are all that remains. Specifically, this is the CTD for the individual subsystem option of the single designs themselves. It is also assumed that only SI costs remain at this point in the process.

General Control Equations: (1) Reject options development that exceeds the phase budgets, and (2) Select exactly one Design at Epoch *Ω.*

THE CONTRIBUTION-TO-DESIGN FUNCTION: OPTION, SUBSYSTEM AND SET

CTD DESCRIPTION AND DISCUSSION

The Contribution-to-Design (CTD) for an option includes both the performance/burden to target difference of the technology/option and the expected variance of the developing technology/option. The framework formulation measures the expected confidence of the specific technology/option to meet Design Readiness Levels during the phases. The Design Readiness Level is an expansion of the DOD's Technical and Material Readiness Levels. Figure 9, in Appendix A shows the formal structure for the nine DOD Technical Readiness Levels. In the context of the framework, the *Design Readiness Level* reflects the maturity of the design relative to each of the required target levels in a program. Without loss of generality, in the illustrative example, we employ three general levels for design readiness: 1-Least Ready; 2-Somewhat Ready; and 3-Fully Ready.

The general expectation for Design Readiness Level is that, if an option is invested in for development during the next phase, it's readiness should improve, or be no worse, than at the previous epoch. Tracking the Design Readiness Level throughout the program is synonymous to the design moving toward or exceeding the requirement. However, in no case will the framework restrict this. The framework fully allows modeling decline in readiness. There are many real-world cases where readiness estimations were wrong. Modeling risk and uncertainty for readiness is key to injecting realism into estimating readiness improvement. A whole additional area of uncertainty is associated with the potential of changing requirements. Program offices should stochastically model variance in epochal readiness movement for both vagaries in design fulfillment and the potential for requirement shift. The framework imposes no constraint to model the targets stochastically for requirement variance and we explicitly model the designs for development variance.

As time progresses, variance should shrink and the probability of achieving the target value should either increase for a converging design, or decrease for a diverging design. However, the framework will allow any stochastic modeling of the parameters. The work done between Epochs should refine the estimate of the mean and reduce the variance.

While the estimated confidence is based on estimates, the framework allows for continuous updating as uncertainty in the estimates lessens over time. With the inclusion of variance, uncertainty can be properly modeled. Specifically, probabilities should be updated at each epoch as the program learns more about each of the options and can refine the probability estimate as the design crystalizes. Additionally, this approach has significant computational advantages since it permits meta-heuristic optimization techniques. We calculate CTD (s) for: options, subsystem and set solution.

The Contribution-to-Design is specifically the weighted value of an option, subsystem or set solution for a given Design Readiness Level at an epoch. The ability to transition from epoch to epoch is stochastic and is based on the forecasted options' ability to change state.

CTD OPTION FORMULATION

This is the Contribution-to-Design formulation for an option at a ***specific*** Epoch looking forward:

CTD = $\sum_1^n \omega_i \cdot P[Z_i]$ (1)

where P is the probability lookup of the Z_i value (defined below), ω_i is the weight of the externality (performance or burden parameter), and i is the index for all n externalities.

$\sum_1^n \omega_i = 1$ (2)

$Z_i = \{(\mu_i - L_i)/\sigma_i\}$ where: (3)

L = Performance or Burden Requirement Target Value

$\{\mu, \sigma\}$ = System or Sub-system externality probability distribution

In Table 10, we show an example of Option 1's individual Contribution-to-Design calculations for meeting Readiness Levels 1 and 2. In this example we "weight" the three metrics (performance, physical weight burden and AUPC burden). The Contribution-to-Design is then a weighted value of the three probability measures.

Epoch A Option 1	Performance	Weight Burden	AUPC Burden	CTD
Weight	0.3	0.3	0.4	
Current Mean	350	350	47000	
Current SD	50	100	10000	
DRL 1 Target	250	400	45000	
DRL 2 Target	300	390	44000	
P (X > DRL 1 Tgt)	0.9772	0.6915	0.4208	0.6689
P (X > DRL 2 Tgt)	0.8414	0.6554	0.3821	0.6019

TABLE 10 – INDIVIDUAL CONTRIBUTION-TO-DESIGN CALCULATION

CTD SUBSYSTEM FORMULATION

For Set Solutions with more than one option per subsystem, it is expected that the multiple options reduce total design uncertainty should a single option fail to meet targets.

This is the cornerstone of SBD and has been effectively utilized for over twenty years, even if not quantitatively proven. For the framework, we assume that the multiple option designs are independent since we only calculate the probabilities of exceeding targets of the readiness levels. Given independence, the Contribution-to-Design formulation for multiple options in a subsystem is:

$$CTD_{i,RL_t \rightarrow RL_{t+1}} = 1 - \prod_{j=1}^{m_i} \left\{ 1 - \sum_{RL_t=1}^{3} \{1\}_{x_{i,j,t,RL_t}} Pr_{i,j}\{RL_t \rightarrow RL_{t+1}\} \, CTD_{i,j,RL_t \rightarrow RL_{t+1}} \right\}$$

$\{1\}_{x_{i,j,\Omega,RL_t}}$: 1 if option j of subsystem i is selected and its current state in Ω is RL_t, 0 otherwise. (4)

where CTD_i is the subsystem CTD for the i^{th} subsystem. j is the index for the option members, from 1 to m, where m is the total number of options in the same subsystem that are in the set solution.

For the example problem, we will use the data from Table 3 for the third subsystem in Epoch B moving forward to Design Readiness Level of 3. We assume the set solution has options 5 and 6:

CTD (Subsystem 3, Epoch B to Epoch Γ, DRL 3) = $\{1-((1-0.7)\cdot(1-0.1))\}$ = $\{0.73\}$.

CTD Set Solution Formulation

The total CTD for the set solution must then consider the individual subsystem CTD's at the epochs. Depending on program situation, it may be important to weight the subsystem CTD's differently. For example, an automotive program may value its engine subsystem higher than its entertainment system for its sports car while it reverses that weight value for its minivan. The CTD for the entire set solution at an Epoch moving to given Design Readiness Level during the next phase is:

$$CTD = \sum_1^I \{Wt_i \cdot CTD_i\} \tag{5}$$

where I = the number of subsystems, i is the subsystem index, and Wt is the relative weight of the subsystems value.

For the example problem, we will use data from Table 11 at Epoch B for Design Readiness Level 3 and an arbitrary weight vector of {0.4, 0.4, 0.2} for the subsystems. The example set here includes all options except for Option 1. Note: Options 1 and 2 are for a Subsystem A, Options 3 and 4 Subsystem B and Options 5 and 6 Subsystem C in this example.

CTD (Set Solution, Epoch B, DRL 3) = $((0.4\cdot 0.9)+(0.4\cdot 0.73)+(0.2\cdot 0.73))$ = 0.81.

Subsystem	A		B		C		Subsystem	A		B		C	
Epoch A	**1**	**2**	**3**	**4**	**5**	**6**	**Epoch B**	**1**	**2**	**3**	**4**	**5**	**6**
Action	1	1	1	1	1	1	**Action**	0	1	1	1	1	1
State DRL	1	1	1	2	1	2	**State DRL**	1	2	2	2	2	2
P(DRL => 1)	1	1	1	1	1	1	**P(DRL => 1)**	1	1	1	1	1	1
P(DRL => 2)	0.6	0.7	0.5	1	0.7	1	**P(DRL => 2)**	1	1	1	1	1	1
P(DRL => 3)	0.3	0.2	0.2	0	0.3	0	**P(DRL => 3)**	0.2	0.9	0.7	0.1	0.7	0.1

Table 11 – Example Action, State and Transition Matrix

In Table 11, we show examples of the elements for the required action and state spaces to enable the algorithm. The current Design Readiness Levels (DRL) are in Row 3, and Rows 4 to 6 show the probability of the options moving up, if invested in for the phase development to the next epoch. Any probability distribution function is allowable, as we are only concerned with the likelihood of state changes. The DRL's estimates can come from any probability distribution the engineering team would use to model the option design likelihood for movement upwards. We also simplistically assume that if the probability for a DRL is equal to 1, it has attained that DRL.

Although not considered in this current framework a future expanded framework may want to consider multiplying the probabilities, instead of taking a weighted average. For the example above, an alternative CTD = 0.9 * 0.73 * 0.73 = 0.48. Using a weighted average can hide the fact that one of the elements has a probability of 0 if the other elements have a high probability. This may mean that in that case, the concept is not feasible if there is no probability of success for one metric. Using the product may also eliminate the need to develop weights. However, not all subsystems are equal so we developed the CTD to have a weighting structure.

VALUE FUNCTION: A MODIFIED CTD IS THE MDP VALUE FOR THE DP ALGORITHM SOLVE

The value function is the core of the MDP/DP model build and algorithm solve. At the macro level, we begin at the end, Epoch Ω, knowing that we must down select to a single integrated prototype design. So, the MDP Network ends at Epoch Ω. We accept that there may be some modification, rework and additional SI during testing. The value function at Epoch Ω becomes the simple weighted multi-attribute utility value associated to each design. With a design down select at this point, we must select a single point design to complete.

A key insight of the research earlier, was that we only needed to solve the actions of the options themselves and not the designs. The designs under development prior to full system integration, are an extension of the option developments prior to the final epochal decision to design down select. At Epoch Ω, we have taken the optimal set of actions over time, so the design down select needs only consider the remaining system integration required for optimal design.

MDP(s) include: (1) a set of possible world spaces $\{S\}$, (2) a set of possible actions $\{A\}$, (3) a real valued reward function $R(S, A)$ and a description T of each action's effects in each state. For the framework, this is a weighted selection of all the options that considers not only the performance and burdens of each of the options, but also their likelihood to meet their performance requirements and burden budgets.

The Dynamic Programming (DP) back solve is initialized by calculating modified CTD values to seed the MDP Network and its reward function R, beginning with the last epoch just before the system integration. These modified CTD values are the MDP Value Function which we term Contribution-to-Design Value Function (CTDV). We calculate the CTDV at each state, to be a "black box" value from which we can determine a value change, the reward R, from state to state. At a high level, we strip the probabilities of the set CTD and use them as the transition probabilities for the set solution to either change state(s) or remain in the current state at the next epoch. The value portion of the CTD logic is modified and used to create the value of each MDP state. This creates a set of stochastic automata with utilities for DP solving. Furthermore, it must be understood that the CTD(s) for options, subsystems and set solutions were all created utilizing forward logic. The recursive DP solve requires backwards logic and a modification of the CTD(s) to execute a solve to determine optimal investment set actions. This process is discussed further in following sections and a detailed explanation of the algorithmic process and the data used are in Appendix B. Additionally, the specific explanation and how we calculate the CTDV black box data is in Appendix B, Section B.2.

During the development phases between the epochs, we consider budgetary controls to not exceed the phase budget, which is standard DOD policy. We consider every possible action that does not violate the budget. The recursive DP solves for the optimal action. We utilize Bellman's stochastic balance equations to solve for the optimal initial action. The actual value optimized recursively is the Expected Value of all CTDV improvements. This modified optimization approach allows us to focus on which options are invested in for the optimal actions determined at each Epoch. We further assume the memoryless Markov Property: the effects of an action taken in a state depends only on that state and not on the prior history and we apply no discount factor.

THE RESEARCH EXAMPLE PROBLEM PROOF OF CONCEPT

This example problem has a significant level of data associated with it. This section shows the high-level data for the Proof of Concept. The detailed data is in Appendix B. Appendix B also includes the major steps for the development of the software utilizing Excel. We utilized Excel Macros/Visual Basic to instantiate the working framework Proof of Concept.

Figure 6 shows the goals hierarchy and value weights for the example problem. There are two subsystems and three metrics for the problem itself. This data applies for our framework's core SBD solution, the single point optimum solution and the combinatorial single-point based optima solutions.

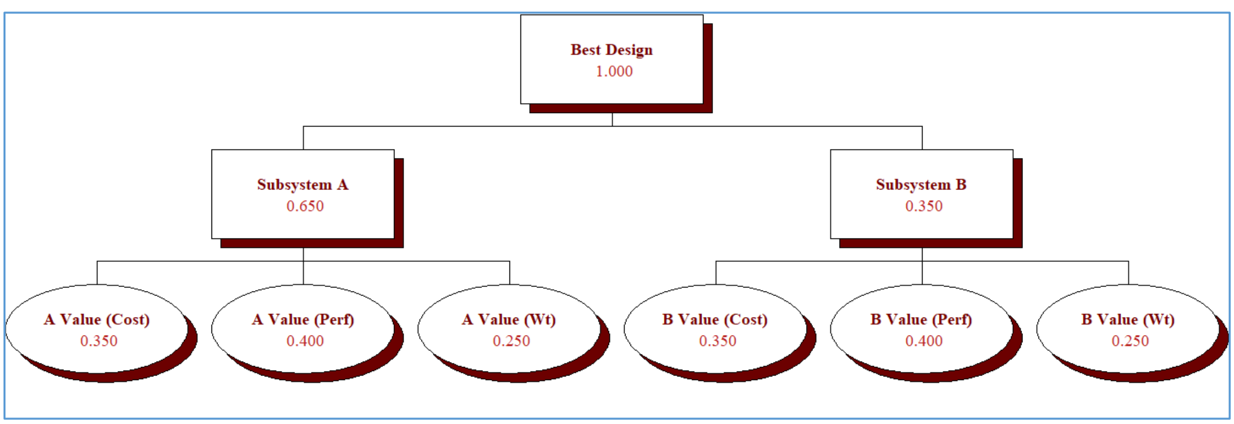

FIGURE 6 PROBLEM GOALS HIERARCHY

FRAMEWORK STATE AND ACTION SPACES

The State Space is the set of all possible states for the problem, which is every possible combination. The individual state's dimensions are carried in a vector of: K (total number of options) ^ (number of parameters/metrics) x epoch. The non-epoch cells hold the Design

Readiness Levels that can vary, but for the example we use exactly three levels per option/metric combination. The following Table 12 below shows the extreme origin and end states.

Options	A1			A2			B1			B2		
Metrics	P	W	C	P	W	C	P	W	C	P	W	C
$State_A$	1	1	1	1	1	1	1	1	1	1	1	1
Options	**A1**			**A2**			**B1**			**B2**		
Metrics	**P**	**W**	**C**	**P**	**W**	**C**	**P**	**W**	**C**	**P**	**W**	**C**
$State_\Omega$	3	3	3	3	3	3	3	3	3	3	3	3

TABLE 12 – NOTIONAL ORIGIN AND END STATES

The state space contains four possible designs. The possible maximum size of the example state space is the product of all option Design Readiness Levels (3^{12}) and the number of Epochs (4) or 2,125,764.

The MDP Network for our example, is sparser since we do not start or end at the notional origin or end points and we do not have state to state arcs for every possibility. The actual example problem state space is less than 1,000. The origin points for all seven dissertation models are in Appendix B. Other sparsity occurs, because all the arcs connect points with only adjacent epochs, except for our skip investment epoch model in Appendix B. And in that case, we only allow a skip of one epoch investment action. In the extreme sense, why would one carry an option that would skip most of the phase development, especially since a minimum of system integration cost would be required? Furthermore, an option not requiring investment would not need to be in the model.

The Action Space is a matrix of decisions as whether to invest (1) or not invest (0) in an option's development for the next Epoch.

Actions	A1	A2	B1	B2
a_1	0	0	0	0
a_2	0	0	0	1
a_3	0	0	1	0
a_4	0	0	1	1
a_5	0	1	0	0
a_6	0	1	0	1
a_7	0	1	1	0
a_8	0	1	1	1
a_9	1	0	0	0
a_{10}	1	0	0	1
a_{11}	1	0	1	0
a_{12}	1	0	1	1
a_{13}	1	1	0	0
a_{14}	1	1	0	1
a_{15}	1	1	1	0
a_{16}	1	1	1	1

TABLE 13 – EXAMPLE PROBLEM ACTION SPACE

In Table 13, the Action Space is a matrix of all possible action combinations that represents whether an option is invested in (1) or not invested in (0) for the next phase's development. This is the Action Matrix! Additionally, the Action Space can be expanded to cover skipped investments. A skipped investment would be marked by a (2) and the Action Space would then be a maximum in size of 81 possible actions. This is a rather rare occurrence in the real world, so it would be easy enough to just add the small set of skip actions for computational purposes.

State	Opt Index				Epoch	Pred
Index	A1	A2	B1	B2	In	Node
1	13	5	13	3	A	Origin
1	13	5	13	3	B	1
2	13	5	13	5	B	1
3	13	5	13	11	B	1
6	13	5	14	3	B	1
7	13	5	14	5	B	1
8	13	5	14	11	B	1
21	13	14	13	3	B	1
22	13	14	13	5	B	1
23	13	14	13	11	B	1
26	13	14	14	3	B	1
27	13	14	14	5	B	1
28	13	14	14	11	B	1
81	14	5	13	3	B	1
82	14	5	13	5	B	1
83	14	5	13	11	B	1
86	14	5	14	3	B	1
87	14	5	14	5	B	1
88	14	5	14	11	B	1
101	14	14	13	3	B	1
102	14	14	13	5	B	1
103	14	14	13	11	B	1
106	14	14	14	3	B	1
107	14	14	14	5	B	1
108	14	14	14	11	B	1

TABLE 14 – TRANSITION MATRIX EXAMPLE

The Transition arcs are stochastic. Forward looking, a transition from one state to another has a value change (reward) from that previous to the future state and there is a probability for that state change. Most the example problem models' transitions are two choices of either move up one or remain the same Design Readiness Level for the individual options, although we did include some other transitions to test the model's robustness. In real life, we need to be prepared for state improvement, no state change and the potential for multiple changes. The framework supports these transitions.

The actual set solution transitions, typically 16 or more in number, are shown in Table 14. The Rewards as stated earlier, are the state to state changes in the CTD. In the Table 14, row 1 is the state node at Epoch A. The remaining rows are the state nodes in Epoch B that a transition arc must be connected to from the prior epoch. The predecessor column shows the parent state node from where the state node in the row came from.

MDP WITH DP SOLVE

Once the MDP is seeded with the CTDV values, we can conduct the DP recursion. First, we calculate from Epoch Γ to Epoch Ω the Expected Values of the CTDV's for every possible action, because each transition arc has an associated probability for the possible actions. Once that is accomplished, we have the optimal action and its Expected Value calculated for every Epoch Γ state to its children Epoch Ω states. It is important to understand, that there may be multiple transition arcs with their value and probability of occurrence coming into a state node. Each transition arc is created "forward looking", i.e. from previous state to future state.

The DP Backsolve is concerned with all transition arcs associated to a specific action into the state node. Thus, for example say, Action 16 would have a State A transition to State B likelihood of 0.6 and a State A transition likelihood to State C of 0.4. If the CTDV is respectively 0.8 and 0.9 at States B and C than the Expected Value at State A for Action 16 and only Action 16 = ((0.8*0.6) + (0.9*0.4)) = 0.84. If Action 16 produces the best expected value than the node State A would be marked as having an Optimal Action 16 with an Earned Value of 0.84. That earned value would be passed backwards as the DP algorithm solves from Epoch Ω to Epoch A. The next paragraph is a quick explanation of the usage of DP in our framework.

In terms of optimization, dynamic programming DP refers to decision simplification by breaking decision steps over time into a sequence. This is done by defining our CDPV to all system states from epochal time A through Ω. The definition of our state value is the CTDV tied to the system index, which represents and knows the complete set of specific option DRL(s) and build-up to the set CTDV. The CTDV at earlier epochs are found by working backwards, using a recursive relationship. In our framework, we utilize the Bellman equation. See Appendix B, Section B.2 for the exact calculations which are normalized with a penalty function structure for a given state not being the best possible state. Since the CTDV is known for every state, the gains in CTDV are easily calculated and balanced with Bellman equations at each epoch. The stochasticity of the model is maintained as we also have calculated the transitional state to state calculations. Finally, the CTDV at the initial state of the system is the value of the optimal set of actions. This process allows us to recover the optimal values of the action decisions, one by one, by tracking back the calculations performed at Ω to A.

For the example problem, this corresponds to measuring approximately 2,000 arcs to determine the Optimal Actions and Expected Values for the 108 Epoch Γ states. We continued recursively to solve the Epoch B to Epoch Γ (approximately 500) arcs to find the optimal actions for the 24 Epoch B states and we pass the previous best Expected Value for those Actions. We then repeat the same process to calculate the Optimal Action from the MDP Networks origin node at Epoch A. Table 15 below, is a small subset of the more than 2,500 Expected Value/Optimal Action sets of calculations. Node and State are used interchangeably and are the same. Transitions and Arcs are also used interchangeably and are the same as well. When either term, node or arc is used, we are emphasizing the network aspect and algorithmic solution.

Arc Number	Epoch In	Node In	Node Out	Epoch Out	Γ_Ω Best Action	Γ_Ω CTD EV	B_Γ Best Action	B_Γ CTD EV	A_B Best Action	A_B CTD EV
Origin				A					a12	0.9927
1	A	1	1	B			a12	0.9850		
6	A	1	8	B			a13	0.9898		
7	A	1	21	B			a12	0.9866		
17	A	1	87	B			a8	0.9915		
24	A	1	108	B			a12	0.9927		
25	B	1	1	Γ	a7	0.9365				
28	B	1	6	Γ	a6	0.9442				
31	B	1	21	Γ	a11	0.9614				
36	B	1	28	Γ	a10	0.9786				
37	B	1	81	Γ	a7	0.9614				
45	B	1	103	Γ	a4	0.9786				

TABLE 15 – BEST ACTION AND EXPECTED VALUE EXAMPLE

SENSITIVITY ANALYSIS OF THE FRAMEWORK MODEL

Multiple Scenario Models were created to conduct sensitivity analysis on the proposed Framework Model. Figure 7 below shows the variant models and the path of model creation to conduct the sensitivity analysis. We considered budget, network origin point, metric weights and sub-system weights as parameters to conduct sensitivity analysis. All detailed data supporting the tables and figures for this chapter are found in Appendix B.

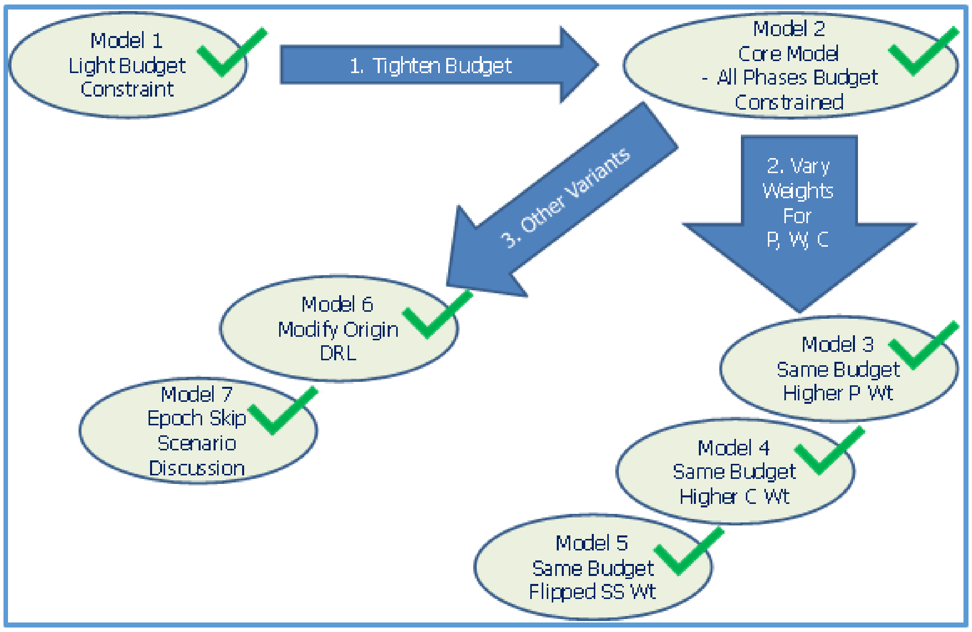

FIGURE 7 – VARIANT MODELS FOR SENSITIVITY ANALYSIS

Table 16 below shows the optimal actions as found from the Dynamic Programming recursive solution.

	A to B	B to Γ	Γ to Ω	Budget - $M
Model 1	a16	a16	a13	72
Model 2	a12	a12	a11	53
Model 3	a8	a8	a7	46
Model 4	a12	a12	a11	53
Model 5	a12	a12	a11	53
Model 6	a12	a12	a11	53
Model 7	a4	a53	a11	51

TABLE 16 – MODEL OPTIMAL ACTIONS

Model 1 had a lightly constrained budget and correctly picked all options where it could. In Model 2 the budget was tightened to specifically force selections between options and it did so correctly. In model 3, we adjusted the performance weight much higher and the model went for the development of option A2 as it became more attractive than A1. In Model 4, we adjusted the cost weight higher and it went back to the same actions in Model 2, although the EV numbers were different. If the cost weight would have been forced even higher, a change in actions would have occurred. In Model 5, we flipped the subsystem weights. Although the numbers did change, it was not enough to alter the actions. Model 6 modified the network origin and had over 90% different arcs. The same modeling structure was applied as in Model 2, so that actions mimicked Model 2. However, the numbers for the EV were completely different, as the state locations were vastly different. Model 7 covered a skip and recovery. In this version, we modeled a later start but with more attractive metrics and better budget. The model did take the skip and recovery for this unusual scenario. Generally, development is required to reduce uncertainty, but the framework can handle budget skips.

SUPPORTING FUNCTIONS NEEDED TO IMPLEMENT THE MODEL

Each externality or parameter/metric must include a target value and a random value (RV) distribution of the technology/option to determine the confidence of the individual design sets (singletons) and then ultimately the multiple design sets. That includes: performance and burden metrics.

Each design must calculate the development costs from epoch to epoch and the recovery costs from the previous epoch to catch up if the set solution did not that option previously. Additionally, reductions and increases associated with shared development and SI costs must be calculated.

Finally, although not shown here in the example problem, the weights of the externalities themselves are also RV's. This framework can be extended with a Monte Carlo simulation. The simulation would create a data set of random externality weights from the weight RV's. Each one of these could then be solved individually, to see the impact on the Set Solutions from epoch to epoch.

CHAPTER 5: COMPARATIVE ANALYSIS OF THE THREE ANALYTICAL METHODS

GENERAL RESULTS

We cannot completely compare the three methods since the framework method employs a DP recursion to calculate optimal actions vs. the single point and combinatorial forward models. However, we can consider what the forward expected CTD would be for all actions when executing the methods going forward. Developing the individual designs from A to Ω is a pure set of a_{11} actions for Design 1, a_{10} for Design 2, a_7 for Design 3, and a_6 for Design 4. The optimal set of actions, recursively solved in the framework solution are: a_{12}, a_{12}, and a_{11}. See Table 17 below for the comparisons. The SBD solution shows a higher EV.

	A to B	B to Γ	Γ to Ω	System EV	Budget - $M
Design 1	a_{11}	a_{11}	a_{11}	0.953602	47
Design 2	a_{10}	a_{10}	a_{10}	0.946763	43
Design 3	a_7	a_7	a_7	0.947511	40
Design 4	a_6	a_6	a_6	0.940672	36
SBD Framework	a_{12}	a_{12}	a_{11}	0.963733	58

TABLE 17 – FORWARD CTD EXPECTED VALUES FOR COMPARING ANALYTICAL METHODS

For this simple problem, we can use the straight forward Multi-Attribute Utility model to determine both the discrete single point optimal design and to also show the full combinatorial optima solution. Figure 8 shows that D1 is the single best design as it has the best total weighted utility. By way of example, the Figure 8 calculations are done by taking the weight vectors from

Figure 6 and multiplying them with the metric value vectors in Table 18. These values are the design values for the single point optimization. See the D1 Value calculation below:

D1 Value = ((0.65*((0.4*0.93175)+(0.25*0.3)+(0.35*0.125))) + (0.35*((0.4*0.1495)+(0.25*0.636)+(0.35*0.125)))) = 0.4113

Key point: these are all point design calculations! The calculations are all made at Epoch A for both single point optimization solutions. This clearly shows the difference in approach between set and point solutions. Point solution algorithms find a single target and then react to design uncertainty and requirement shifts, as they occur. Our set solution framework uses a stochastic process to potentially select multiple options in a region of the design space to mitigate both design and requirement uncertainty.

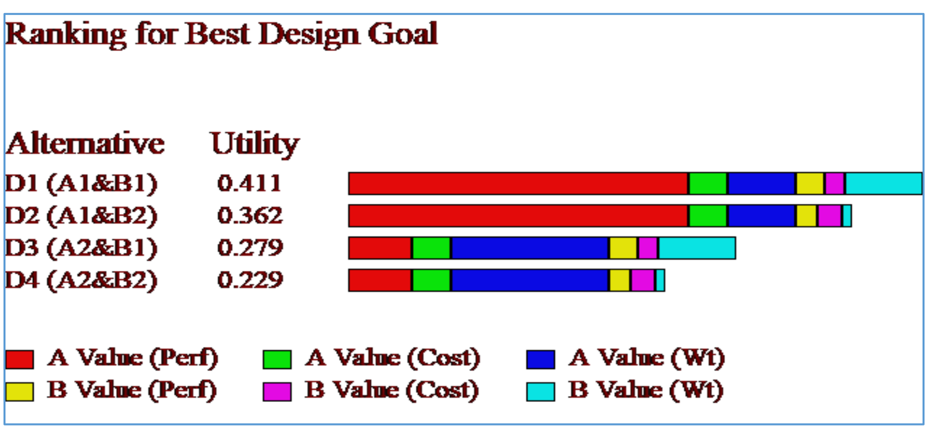

FIGURE 8 – RANKING FOR BEST DESIGN

Table 18 shows the CTD Based Expected Values that come from the main problem. These Expected Values assume that the Designs would follow the typical PD process for point based design, i.e. the options of the design are developed continuously. Additionally, we show the CTD values at the metric level so we can determine if there is any pareto dominance between the designs.

Designs	A Values			B Values		
	Perf	Weight	Cost	Perf	Weight	Cost
D1 (A1&B1)	0.93175	0.30000	0.12500	0.14950	0.63600	0.12500
D2 (A1&B2)	0.93175	0.30000	0.12500	0.11700	0.09200	0.14600
D3 (A2&B1)	0.17750	0.69200	0.12500	0.14950	0.63600	0.12500
D4 (A2&B2)	0.17750	0.69200	0.12500	0.11700	0.09200	0.14600

TABLE 18 – CTD BASED EXPECTED VALUES (DECONSTRUCTED DOWN TO METRIC)

Green represents the maximum metric value. Although D1 almost dominates D2, and D3 almost dominates D4, the B2 option is strongest in cost value, which keeps D2 and D4 as combinatorial optima. Thus, the combinatorial solution shows no complete pareto dominance for any design. The program team would be left to decide it's best single point design solution from the pareto optima to take forward.

The process for defining the optimum or optima in both single point approaches, focuses on selecting a design initially and then developing it. The framework approach rather determines an optimal action to initiate a set solution from an initial optimal action. That action is based stochastically, to develop a set of options that reduce developmental risk and allow for uncertainty in the process.

From a pure quantitative sense, the SBD Framework finds the best expected value. Of note, Action a_{12} is the same as jointly executing a_{10} and a_{11}, which are the two best, single point design approaches, until dropping down to a_{11} in the final phase. This approach reduces the risk of carrying two designs through the first two phases until uncertainty is reduced.

Perhaps the biggest issue regarding the use of SBD is the potentially higher cost associated with it. For this example, the SBD Framework approach may cost an extra $11M to execute. That is the immediate trade-off you take with SBD. However, further factors support the SBD Framework yielding superior design results. The first, is that Production costs are at least one order of magnitude higher than R&D costs and Lifecycle costs are typically a magnitude higher than Production costs. Given that, budget increases for higher SBD R&D costs makes very good sense.

DEPARTMENT OF DEFENSE (DOD) CONSIDERATIONS

The U.S. DoD recognizes developmental risk and often awards two identical design contracts, in the hope that two contractors are better than one to reduce their program developmental risk. They presume independence between the contractors in the development, even though they are both developing the same options. Thus, the typical action would be to pay $94M for two contractors which is $36M more in the example problem. However, contractors working on different options, is inherently more independent than two contractors working on the same options. While their thought processes may differ, both contractors are subject to the same physics, chemistry, math and engineering issues when developing the same design. Further study on how independent designs are, coming from like contracts, is more than warranted for future research.

COMMERCIAL ENTERPRISE CONSIDERATIONS

Although the DoD Product Development process is more deliberate and is under generally more oversight, Set-Based Design can yield significant improvements to commercial Product Development. Commercial Product Development has the general structure of budgetary milestones, but varies dependent on corporate rules. Engineering and Product Development reviews also have similarities. Thus, both defense and commercial sectors share similar epochal needs. Furthermore, although the DoD has a very formalized structure for its product requirements, commercial enterprises almost universally develop and maintain vast databases of requirements and specs for their Product Development.

The greatest difference between the sectors' versions of Product Development is perhaps in time and lifecycle. It is generally accepted that the defense sector has a longer Product Development time span. It is more correct to say that the defense sector has a more deliberate time span. It also needs to be understood, that the defense sector's time span for Product Development can be and will be dramatically shortened during times of conflict to rush needed technical breakthroughs into production.

Assuming that the commercial sector does have a shorter Product Development time span, this Set-Based Design framework provides the same important advantages. One could even say, given that Toyota initially developed Set-Based Design, that this framework may have more applicability to the commercial sector. If so, there are several supporting work streams required to commercialize this framework for say a General Motors' application. We will discuss that in Chapter 6, but it is important to note that a commercial enterprise's internal data, particularly that of failed PD programs is even more important to retain and utilize. This is because, that data will provide past historical refence to how well targets were met and if off, by how much. The ability to accurately describe the effects of uncertainty to engineering estimates, is crucial. Edison himself correlated his thousands of failures to find a workable light source, as a sequence of required steps before finding the correct answer. Enterprises pay a lot for their design data, so they should hold on to it, as its value for future stochastic estimates is truly invaluable.

CHAPTER 6: FUTURE RESEARCH AND NEXT STEPS

REVIEW OF RESEARCH OBJECTIVES ACCOMPLISHMENT

Objective 1

The analytical framework developed a topology that created formal definitions for dimensional control of set solutions. The system requirements became target values for performance, burden and costs. The unique multi-dimensional separations between the test options and their targets allowed us to define mathematically feasible regions in the trade space itself. The organization of the collection of feasible point solutions into regions which are set point solutions allowed us to than compare those regions or sets throughout the epochal movement of the Product Development process through time.

The regions or sets comparative values for similarity and difference from each other are known by the looking at and comparing the set's Contribution-to-Design values and the individual point parameters that form the basis of the set calculations. It is an expectation that every time data changes the framework solution can be immediately updated. This capability is a sign and marker of process resiliency.

OBJECTIVE 2

The analytic methods to address the sensitivity of the topology of design space (i.e., the organization into regions) to the constraint levels and the understanding of the constraints, and the effect of constraint levels on feasible regions of design space was accomplished. The sensitivity analysis capability provided with the Markov Decision Process and Dynamic Programming solution structure allows for testing sensitivities for metrics, weights and structure. The test problem sensitivities themselves are shown and discussed in both Chapter 5 and Appendix B.

OBJECTIVE 3

The SBD framework incorporated uncertainty in both subsystem and option component availabilities. Although, uncertainty was directly modelled for estimation of how well the options could meet design targets for performance, burdens and cost, we also showed that the targets themselves could easily be stochastic estimates themselves. Two stochastic estimates, if their probability density functions can be estimated, just become a new single stochastic function. The moments of value and variance for the framework, follow the rules of logic for stochastic mathematics and estimation of uncertainty and chaos.

OBJECTIVE 4

Our framework approach and the sensitivity analysis of the test problem, showed that characteristic point designs that were resilient with respect to changes in the constraints, appeared again and again in the optimal investment actions by epoch per set solution. It should be noted that this result was for a relatively small trade space. Further verification of this mathematical attribute needs verification with larger data sets, i.e. a larger group of options, subsystems and metrics.

HOW INDEPENDENT IS DESIGN SPACE?

This question is the most important one going forward into future research. The research in this dissertation leveraged the assumption of option design independence. Although, a case can certainly be made to defend this as a general expectation, it may also be specifically wrong in some cases. Does, the current developmental knowledge presuppose PD efforts to be of one form and not another? If so, does that impact the design space by having many, similar option choices? Does the developmental process itself, formulaically cause a certain type of option to be created, more than ones that perhaps don't have the same general technology structure?

For instance, the recent breakthrough of true electric powered cars. Combustion powered engines required more cost and more material to raise both performance and reliability. Each engine type, diesel vs. gasoline, turbo vs. no turbo, etc would have variations of power to weight and weight to reliability curves, but yet they followed similar predictable patterns. The electrical engine, once batteries became sufficiently powerful that hybrid engines were not required to beat a 200-mile range, breaks the curves completely. The reduced part count associated with a pure electrical drive train has a very different reliability structure, torque performance and even power to weight curve. The trade space including electric powertrains is very different than one with traditional combustion-based powertrains.

Since our research assumed option independence, further research to validate it is necessary. However, the general structure of the framework itself would require little to no modification. The primary stochastic nature of the framework structure is inviolate regarding change in its indices structure, system hierarchy and epochal nature. The calculation of probabilities done in a dependent structure can occur separately or with a minor modification of the framework itself.

EXPANDING RESILIENCE THROUGH ADDING DSM TO THE FRAMEWORK

Our framework process, which calculates the probability density functions and values for the Contribution-to-Design function, needs efficiency improvements to handle large scale models. Use of Design Structure Matrices (DSM) in parallel with this framework can further enhance the scalability of the framework. Key linkage between the Set-Based Design process and the usage of Process Design Structures Matrix to reduce design complexity and support design process resiliency was postulated for Navy ship design(Doerry 2009).

POST-DISSERTATION RESEARCH WITH A LARGE MODEL

Although, the maximum possible trade space expansion is geometric in proportion to the number of metrics and epochs we have seen in this research, the real problem size growth appears to be potentially linear and no worse than exponential. The utilization of Dynamic Programming allows the recursive structure of the Markov Decision Process to be preserved and then applied to avoid unnecessary recalculations, with its usage of the Bellman Balance Equations. This assertion needs testing with future research.

Additionally, other known optimization methods may prove to be even more efficient than using Dynamic Programming. A simple approach that we rejected early in the research, was to describe the value of all possible state to state paths for every unique design, could merit reconsideration. This is because the problem could be reduced to a simple, but very large single optimization. We chose not to take this approach, as we focused upon the need to accurately describe what set solution value is and how it could be described quantitatively.

The primary "*raison d'être*" for Set-Based Design was to allow for the unquantifiable reasoning of design uncertainty that generations of developers were very aware of but rarely considered. However, what if a reasonable simulacra of set value is now available and the focus is changed to optimal budget actions, instead of point design refinement over time? If that is the case, any optimization method that properly describes and calculates set value should be considered.

The level of problem complexity was defined in our research effort by the need to both create an explainable test problem and to also allow for quickly modifying our test algorithm as we developed it. Excel, as a tool with its Visual Basic Macro language worked well in this environment, but it also created a situation where its row, column and sheet structure required a complicated approach to solve single problems. Each different model required a separate, new workbook with an array of multiple sheets, each performing a small set of steps. The ability to manipulate and modify is an Excel strength, but the requirement to copy and paste process steps over and over, just to repeat slightly different problem solutions, is a weakness. The small problem of two options per subsystem, two subsystems, three metrics per option and four epochs was easily handled. It is anticipated that having the repeatable steps, of the total Markov Decision Process problem creation with Dynamic Programming recursive back solve algorithm, coded in something as simple as MATLAB would create immediate efficiencies. We are expecting to do this going forward with a post-doctoral research project that would enable an enhanced solution framework structure. We anticipate that a MATLAB analytical based structure with relational database software environment is a potential intermediate solution, which can easily handle up to 50 options, over 10 subsystems with 5 to 10 metrics per option, and 5 or more epochs. The software is itself not the primary constraint to the problem size. Rather, it is expected that we will need to research further things before developing a commercial application. This would include: real-world data sourcing for the problem, verification and validation of mathematical relationships discovered in the current effort and specific algorithm choices affecting solutions.

In the long term, using state-of-the-art optimization algorithms and relational databases, it may be assumed that the current ability to considering billions of unique point system solutions is similarly possible in the set solution context. This dissertation focused on the creation of the mathematics to describe set value quantitatively. Given validation that our proposed set value structure has general merit, any optimization algorithm may be potentially modified for Set-Based Design action optimization.

ANALYTICAL EXTENSIONS

COMMERCIAL CONSIDERATIONS

While Return-On-Investment is more identifiable with profit and loss in the commercial sector, the defense sector also needs to understand whether a development gives an anticipated return.

Return-On-Investment is a key for commercial Product Development decisions. Return-On-Investment calculations are supported by the framework, since optimality is viewed from an application of allowable budget actions to select options in the face of uncertainty. Thus, no extension of the framework itself is required to conduct sensitivity analysis of the budget for chosen option development at the epochs. However, the actual business mechanics of conducting Return-On-Investment calculations for budget decisions is not enabled in the framework. If an enterprise has its own Return-On-Investment software and process, it can be fed directly from the framework solutions and data sources. This could be accomplished in a shared data warehouse, structured query environment or with simple copy/pastes to Return-On-Investment spreadsheet models. Additional Return-On-Investment considerations as to both cost and profit throughout the enterprise's value chain could follow either mechanisms. A truly elegant approach would be to directly attach the framework as a module to the corporate Enterprise Resource Planning (ERP) system. This would allow for integrated data and its maintenance.

Knowledge Management

As stated multiple times, both data quality and usage of historical data to form usable stochastic data for set solutions is incredibly important. Set-Based Design in its current qualitative structure does not rely heavily on data accuracy, as its purpose was to choose option development sets to overcome data uncertainty. However, to employ a quantitative Set-Based Design framework, one must understand and know the specific data and its quality to rely on the set solution calculations.

Moving to a resilient quantitative process will require enterprises to have a change in how they view their Product Development failures. Successes are typically oversold and failures are ignored. This creates not only a dearth in total data, but also skews the possible design trajectories into false paths, where only success data is considered. As stated in the literature review concerning programs, performance attainment is generally over estimated and uncertainty towards burden elements and cost are generally under estimated. If past failure data is disregarded in calculating estimates, then aren't those estimates at best suspect?

Knowledge Management needs two changes. The first is simply the maintenance of design failure data. The second is more profound and quite cultural to the enterprise. Edison's embrace of failure itself as a value to the process for finding solutions, needs to become a cornerstone in engineering analytics. The extension of immediate profit = the only good, and immediate loss = the only bad, creates a long-term loss and misapplication every time an enterprise initiates its next Product Development program. Knowledge Management, where the acceptance and understanding of short term failures, **IS** value for future Product Development. This needs to become the norm.

ANALYTICAL SOLVING EXTENSIONS

Utilizing the knowledge of where the origin and final possible states not only reduces the actual problem size from theoretical maximum, but also allows us to consider what in optimization parlance is termed a "hot-start" basis. Since Dynamic Programming is concerned with keeping the recursive relationship to reduce unnecessary recalculation, it's usage of the hot start basis, is not as dramatic as with launching a simplex or interior point algorithm. It is anticipated that maintaining the Markov Decision Process in the framework to create a mathematical network problem, would be useful for other optimization algorithms. Not only would the state space be accurately derived from the theoretical maximum, but it also conveniently provides a true origin for the network. By using the value of the origin, we could create a hot basis start for both simplex or interior point algorithms.

An analytical extension not considered in the research is whether the actual maturity of the options and their sequence of selection in the set solutions themselves creates issues. Future research should consider, if there are those effects. Is it possible there may be underlying auto correlation occurring? Is the sequence of investment actions against the available options different for more mature options than less mature options?

Although, we asserted in our research that options and designs are readily interchangeable and that we only need to solve option solution sets, is that true? Our framework assumed that system integration costs are mostly in the latter epochs and we only considered them in the final two epochs. Could more immature options require earlier and heavier system integration? If so, is there a pattern that would require additional modelling and possible modification of the framework? This is an additional research question for the future.

CHAPTER 7: CONCLUSIONS

The literature review documented the current state of Set-Based Design as a resilient Product Development process that has proven qualitative results over the past two decades for many Product Development programs. Quantitative processes with design optimization traditionally focus on point solutions. Combinatorial optimization has increased insight into design uncertainty, but still only provide more sophisticated point solutions. Recently, (Avigad, G. 2007) introduced the need to have a Set-Based Design quantitative solution structure that balances both optimality in a point solution sense with variability to capture set solutions in clustered set solutions. Resilience in the Product Development process is supported by utilizing Set-Based Design to create multi-design sets that increase confidence to attain meeting design target requirements.

The initial mathematical framework proposed in this paper shows a method to combine performance and burden information from competing design elements into a Contribution-to-Design Function that can be optimized. Specifically, Contribution-to-Design is an optimality structure formulating Set-Based Design value, which is not point design optimization in the current vernacular. The value is directly associated with maintaining design sets, to directly increase design resilience in the face of design uncertainty. A cost structure that details developmental costs, design recovery costs and system integration costs from epoch to epoch is proposed. By updating the cost structures and performance estimates at each epoch a design fluidity supports set adjustments at epochal decision points. This also supports the decisions as to when to modify the sets. Key linkage between the Set-Based Design process and the usage of

Process Design Structures Matrices to support design process resiliency was postulated for Navy ship design (Doerry 2009). Further research expanding the framework to integrate Design Structures Matrices would allow scalability.

The research was novel in the sense that this is the first quantitative framework to solve a Set-Based Design Product Development problem. Additionally, little research has been applied to providing a value structure to set solutions themselves. Further research supporting the findings in a larger scale problem is needed. Additionally, data quality and knowledge management need both research and application to create a viable commercial application of this research.

APPENDIX A: REAL-WORLD EXAMPLE SCENARIO FOR SET-BASED DESIGN

Enclosed in this appendix is a current example typical to main battle tank programs. Tanks need to balance engine power, running gear and suspension subsystems to provide mobility. Unmet force protection requirements to protect the crew, continue to add weight and/or cost to the tank system through the armor subsystem itself or new exotic subsystems such as active protection. The engine subsystem may need to be changed to support new force protection technology – heavier armor packs in this scenario.

In this appendix, we develop all of the high-level thinking associated with considering set solutions at the design level. This appendix is intended to provide insight to the design process using set solutions. The framework presented in this dissertation mathematically focused on leveraging unique subsystem option choices which is the best way to analytically solve what option set action is optimal. This appendix focuses on the total design and looks at the option combinations to create unique designs and refers to the set solutions as sets of designs. This is easier to understand as a system design is always trying to improve multiple subsystems as the program progresses but also to develop the required system integration of those subsystems. Both viewpoints are quite accurate. The options define the unique designs and the designs are formed from unique option combinations.

Shown below is general background on a subset of three generic subsystems, their interactions and their considerations that we will express visually.

A.1 ENGINE OR POWER SUBSYSTEM

1. Keep the current gas turbine engine
    a. No Risk
    b. May not meet the automotive mobility requirements
2. Switch to a current, modern diesel engine
    a. Low to Medium Risk due to uncertainty over system integration and packaging concerns
    b. Known performance improvements
    c. Fuel Tank subsystem and Air Filter subsystem will be required
3. Develop a high energy density diesel engine
    a. High Risk solution – even if a contractor would promise delivery to spec, unforeseen issues such as new packaging, greater system integration, unforeseen material issues, new supply, etc. could easily negate the best intentions
    b. High appeal for a breakthrough change that would significantly improve the automotive performance while fulfilling all system requirements
    c. Significant potential for a cost overrun and program breech if not met

A.2 RUNNING GEAR SUBSYSTEM

1. Keep the current track
    a. No risk
    b. Higher maintenance costs, but known
    c. More rugged, but heavier in weight

2. Consider a wheeled
    a. Tractive power is less
    b. Extremely inexpensive and lighter weight
    c. Ruggedness has been improved over the years, but it may still not be enough for a MBT

A.3 SUSPENSION SUBSYSTEM

1. MacPherson
    a. Currently only wheeled applications
    b. Versatile, well know
2. Hydro-pneumatic Suspension Unit (HSU)
    a. Been around for a while and there are both US and German examples
    b. Could be used for a wheeled running gear subsystem, but the program office only wants to consider it for a tracked running gear subsystem
    c. Key advantage is weight and the reliability has increased over time
    d. Minor integration issues are foreseen
3. Torsion Bars
    a. No risk
    b. Heavy, in both weight and capability
    c. Inexpensive and it is a known quantity for well over 50 years
    d. Lower in cost than the HSU option

A.4 COST AND MATURATION CONSIDERATIONS

Typically, in the past defense programs have tended to go with technically superior point solutions like the new High Energy Diesel (HED) and the HSU. However, if cost or risk/uncertainty were viewed as overwhelming, more traditional choices would go into the design. Regardless, these decisions would be made for a single point system design that would only have flexibility to modify the subsystems for integration and perhaps minor system modification. Unknowns in cost and maturity, if covered at all, would have a generic total program risk budget to support modifications as unknowns become known over time. In the set solution scenario, some or all the subsystem options are carried forward. This means that all the other subsystem options that would be required to change for all the engine options need to be carried forward as well. That specifically means that the set solution size will potentially grow geometrically based on the number of other impacted subsystems. The total solution set would describe a set region in the total state space of the entire system design.

Now let's consider the time dimension of the program itself and the impact on the set solution. Figure 9 below, shows how defense considers technical maturation or readiness going from concept to implementation. Each of the Technical Readiness Levels (TRL) are expected to be fulfilled or completed before allowing the technology and specifically the system to pass the milestone decisions. Thus, if the HED does not meet the technology fulfillment goals, and regardless of whether lean or other improved development methods are applied, there will be a program breech. The value of the set solution is obvious. The program may have to spend extra money to keep all four options moving forward, but there is no hard "go – no go" decision anymore. Additionally, if the HED does not meet expectations, it can continue under development and then be deployed in an engineering change later when the problems are resolved. Historically, defense programs are cancelled if the point solution fails. This meant that great technologies were often thrown away and then started from scratch years or decades later at great cost.

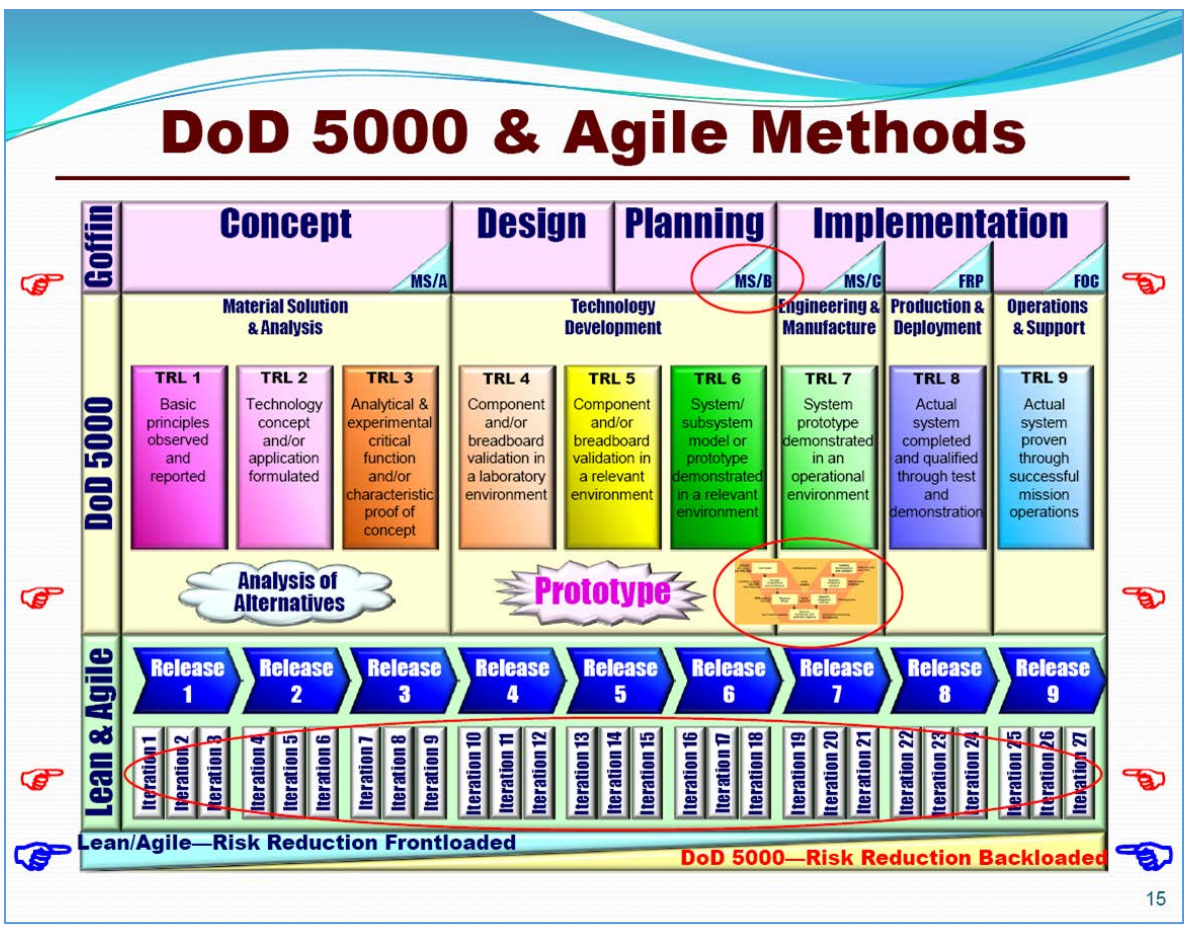

FIGURE 9 - REQUIRED TECHNOLOGY READINESS FOR THE DEFENSE SECTOR PD PROCESS

A.5 SET SOLUTION WALK THROUGH

STEP 1: INITIAL SET SOLUTION

Figure 10 below shows all 18 points of the complete state space and all of the options available regardless as to current feasibility or optimality.

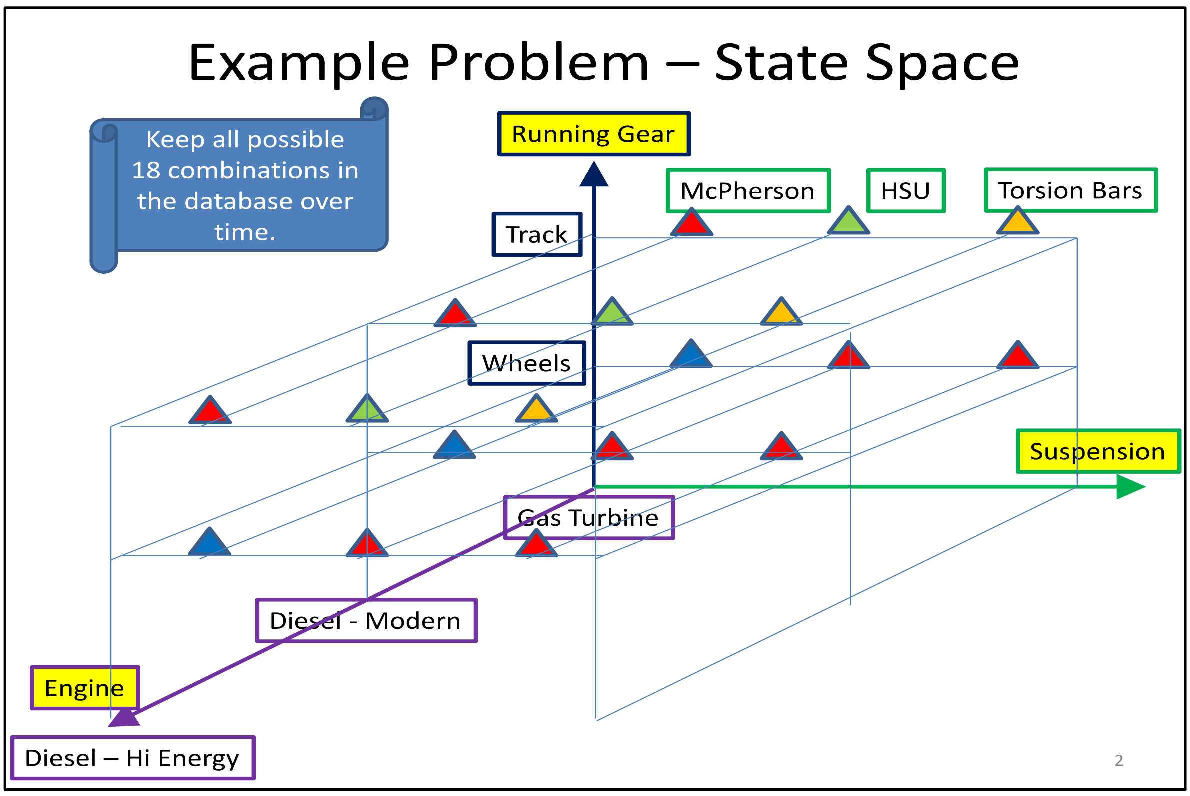

FIGURE 10 – EXAMPLE PROBLEM – FULL STATE SPACE

STEP 2: INFEASIBILITIES NOT SHOWN

Figure 11 shows the removal, from view only of infeasible interactions between running gear and suspension choices.

In SBD, the infeasible choices are maintained in the data so that if the design needs to reconsider infeasibilities that may change over time.

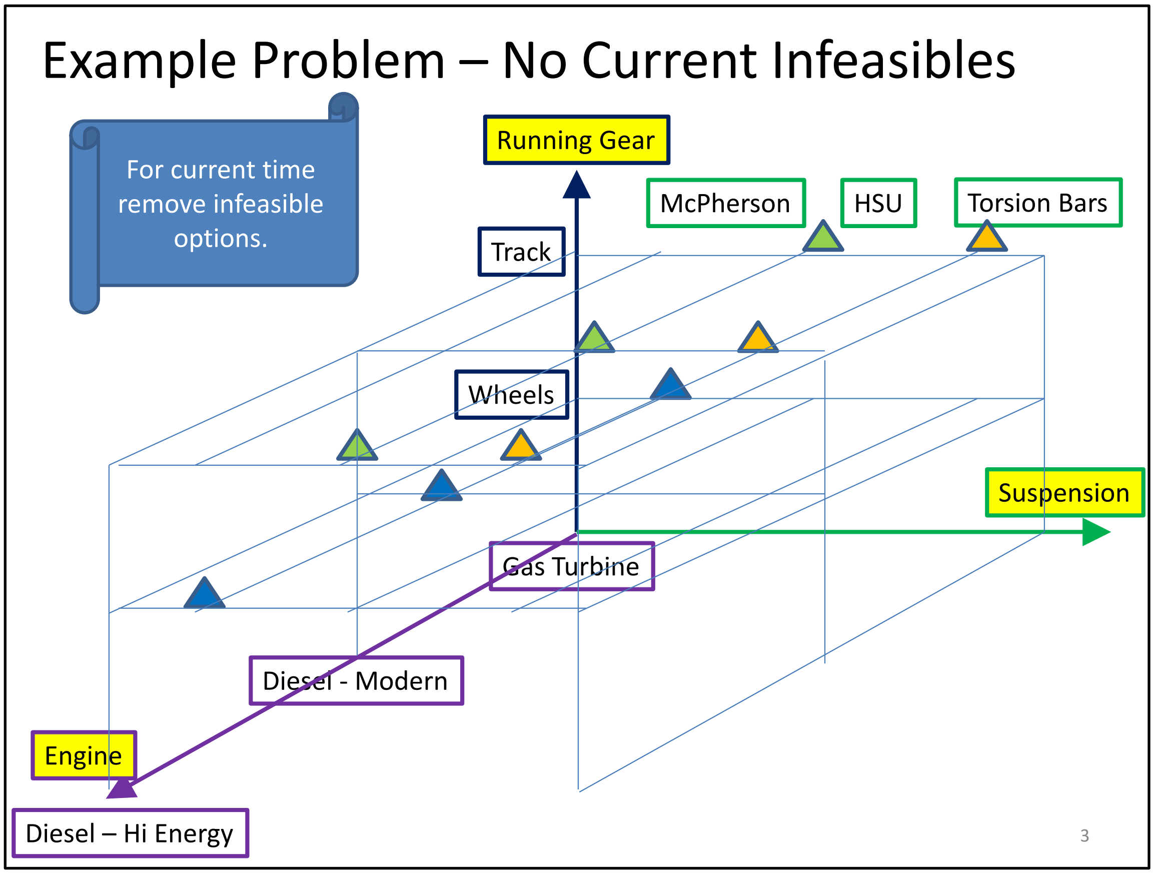

Figure 11 – Example Problem – Infeasible(s) Removed

Step 3: Partition into Convex Set Solutions

We have used the technology difference between track and wheels to define two set solutions. Both are viable now and both are convex sets. Convexity means simply in this example that inside each set solution all subsystem options can be traded. Figure 12 below shows the feasible "two-member" set solution.

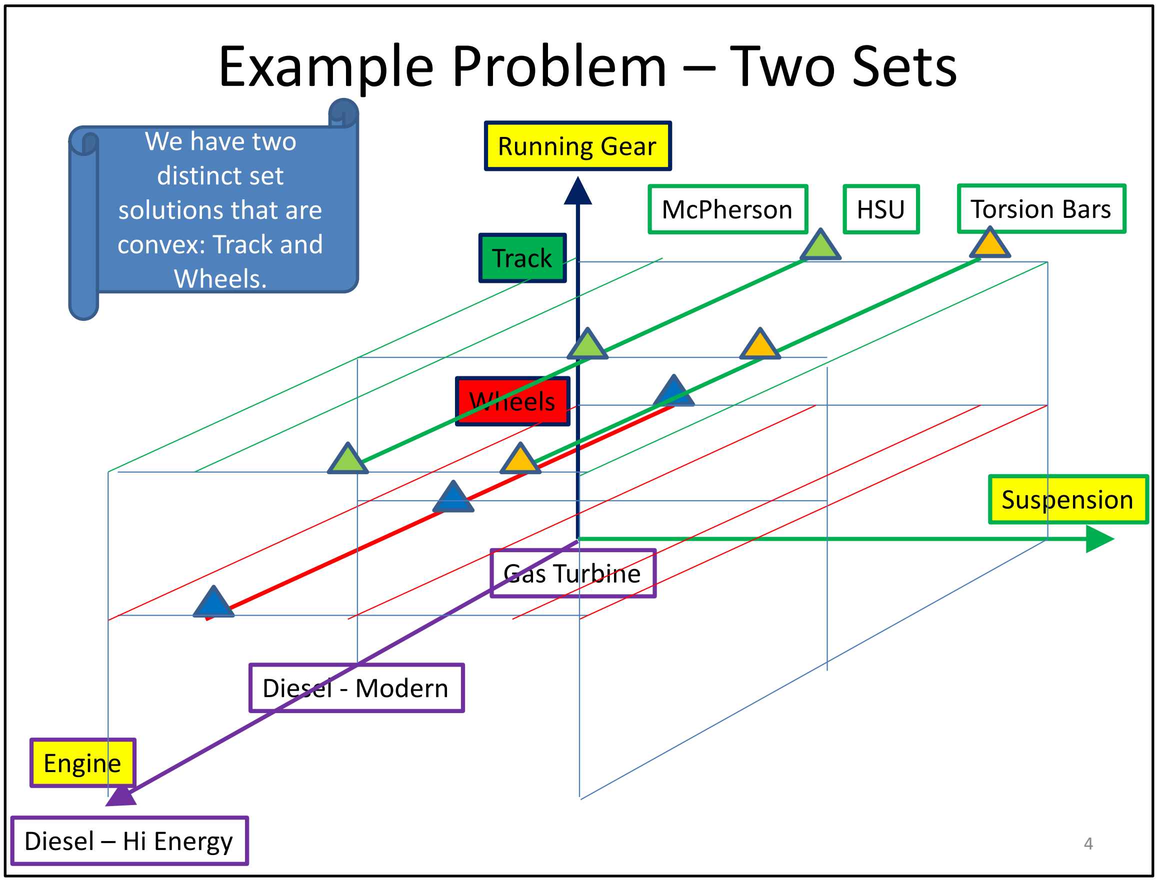

Figure 12 – Example Problem – Two Set Solutions

Step 4: The First Requirement and Impact

We now show how a requirement can remove a set solution. The requirement shown below in Figure 13 will temporarily remove one of the set solutions, the wheeled set from current consideration.

In this case the Wheels solution cannot yet master climbing a 1 Meter vertical face.

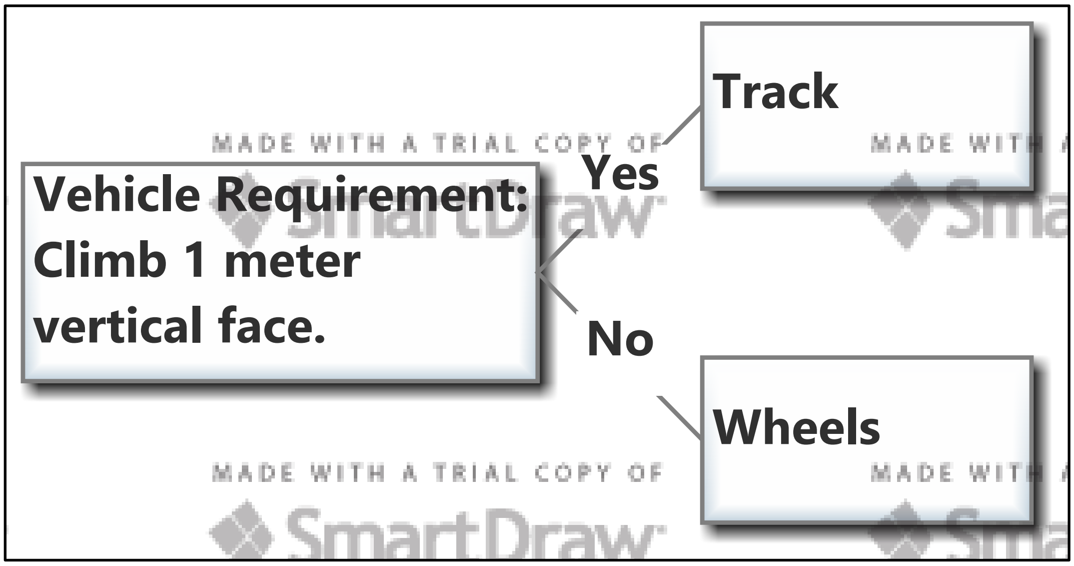

FIGURE 13 – APPLICATION OF REQUIREMENT ONE

However, it may remain in the design database for future consideration as technology maturation and uncertainty unfold. This is an essential task for all defense program offices, i.e. the maintenance of design improvements for their products that may have a fifty-year total lifecycle before complete new design and replacement. The M1 Main Battle Tank, was fielded in the early 1980's and is expected to be in the force until the 2030's or longer.

For our example, there remain six possible point solutions inside the set solution that are currently viable and pass Requirement 1 shown below in Figure 14.

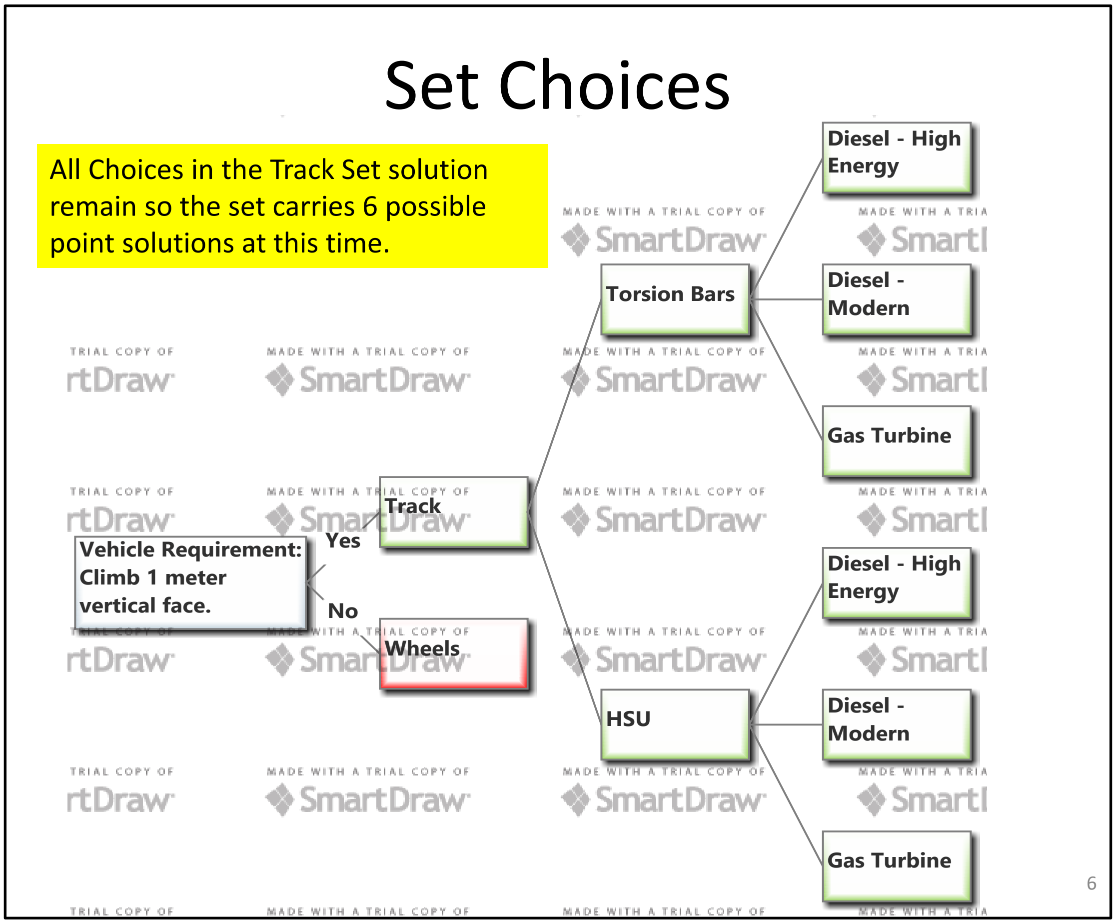

FIGURE 14 – EXAMPLE PROBLEM – REMAINING SET SOLUTION

The remaining trade space for the solution set is shown below in Figure 15. It shows what the requirement does to reducing trade space in the example and by extension what requirements in general do to all trade spaces.

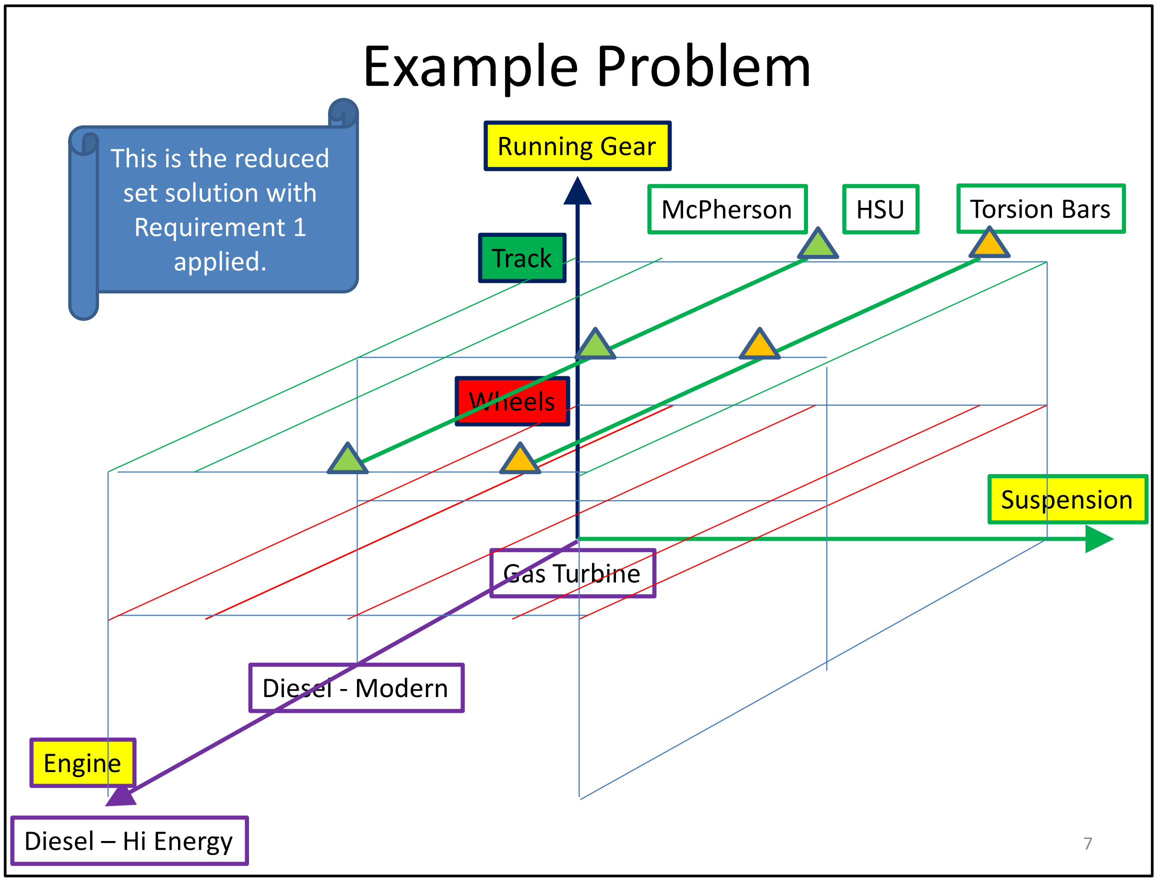

FIGURE 15 – REMAINING TRADE SPACE AFTER REQUIREMENT ONE IS APPLIED

STEP 5: THE SECOND REQUIREMENT AND IMPACT

All Solutions Remain Available in the Set!

Performance and Burden data will be considered to help determine the most central or potential incumbent choice for the design. This is very different from point logic. Now in the example, we show this below in Figure 16, where the 2nd requirement is applied to the trade space. Additionally, the program office would need to apply budget to the entire set solution going forward.

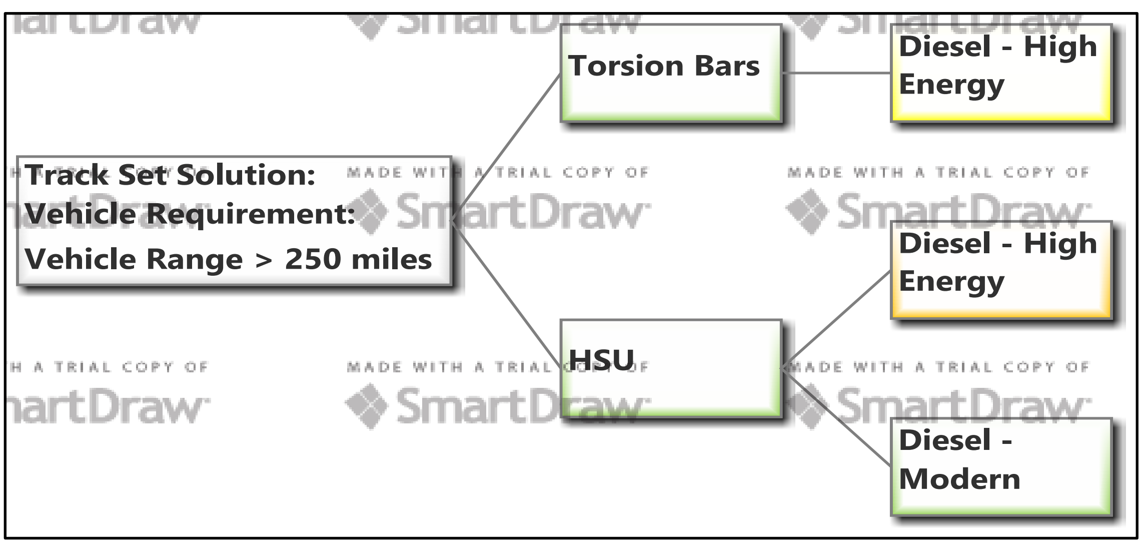

FIGURE 16 – APPLICATION OF REQUIREMENT TWO

The following points are considerations in the example as we apply the 2nd requirement to the trade space itself. Figure 17 below shows these considerations graphically.

1. Torsion Bars are simple but weigh more so they require the High Energy Diesel (higher risk engine development)

2. HSU (s) are slightly more complex but lighter weight. The Modern Diesel will require some weight margin. The Gas Turbine would have required too much weight margin.
3. The best performance option is the HSU with High Energy Diesel but it also carries the greatest risk.
4. The HSU with Modern Diesel is average total risk with better fuel than most other options.

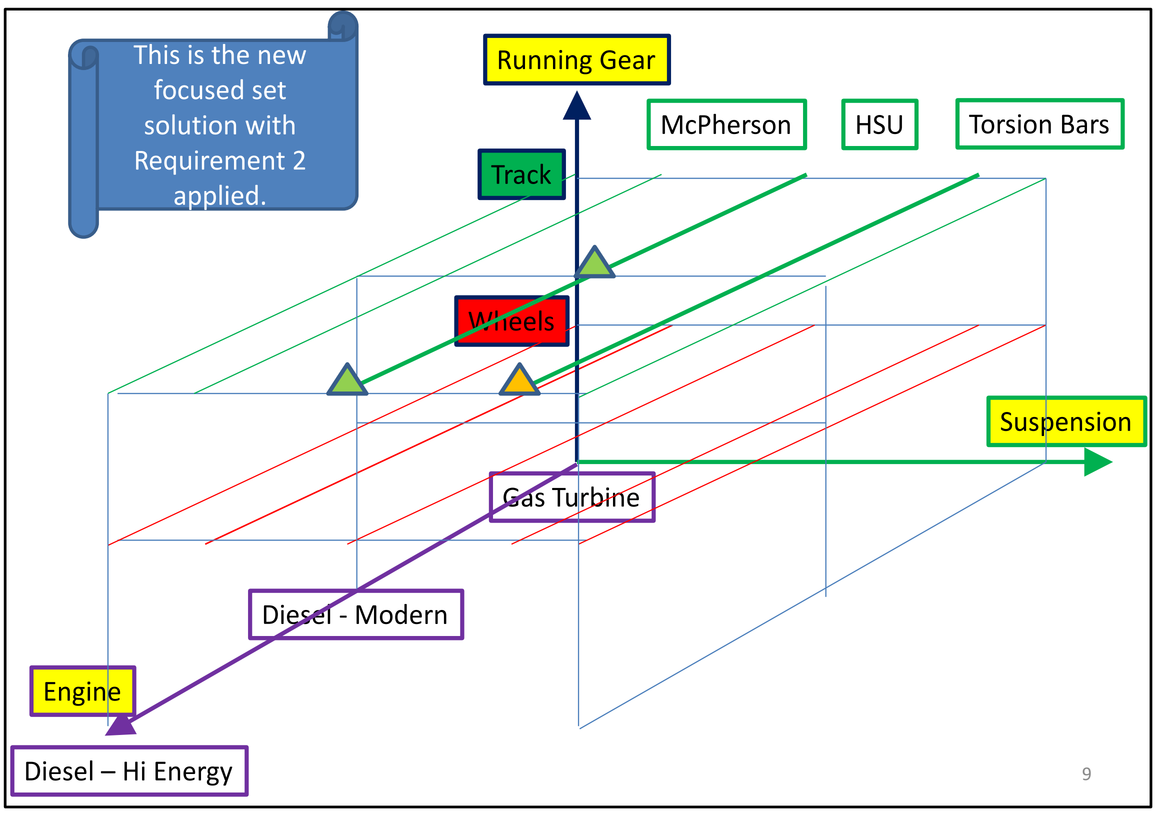

FIGURE 17 - INTERIM NECKED DOWN SET SOLUTION

APPENDIX B: DATA SET AND MODELING EXPLANATIONS

B.1 MACRO PROCESSES

Our macro process for the framework is to:

1. Determine the Contribution-to-Design for each possible state. That requires us to create an indexing structure for all states. Initially, we create option indices where we can do the CTD calculations at the option level. Then from the option indices, we create a mapping to the total state space which includes all combinations of the option. This was not required, but rather, it made the matrix manipulations easier to maintain and easier to understand. Finally, we can build up the CTD calculations for the subsystems and individual set solutions.
2. Create a Markov Decision Process model for the design problem. Given that we will take the CTD values at each state, we have created a suitable black box structure to provide values to the model. We also need to calculate the stochastic movements of all set solutions from state to state. This is the transition matrix. The action space is whether we will invest in an option.
3. The final sub-process is for us to use Dynamic Programming to recursively back solve the model to determine the optimal actions for options investment. We use the Bellman Equations to determine the optimal investment actions at each state space node. After traversing the transition arcs recursively, we can avoid calculating arcs not on the allowable action paths.

Snapshots from the individual models are shown in all the following tables and figures in the appendix. All models are available from the author as open source software. Please email any request to stephen.rapp@wayne.edu.

B.2 DEVELOP CTD

As stated in Chapter 4, the Contribution-to-Design Value Function (CTDV) is described as the value of the option, subsystem or set at a given Design Readiness Level (DRL).

We weight the individual option DRLs for both subsystem and parameters. In the case of the examples in the dissertation that includes: performance, weight (physical) and cost (AUPC). We use a penalty function for the core value and normalize all values. The CTDV itself for each node becomes a block box value.

Step 1: Assign Model Parameters and Weights

For all eight models, we utilize the weights to modify the option CTDs to become CTDVs. We utilize the budget to control allowable investment actions. Shown below in Table 19 are the weights for Models 1, 2, 6 and 7. Models 3, 4 and 5 modify the weights to conduct sensitivity analysis on the parameters and are shown later.

Long Name	Short Name	Values
Performance Weight	PWt	0.4
Weight (Physical) Weight	WWt	0.25
Cost (AUPC) Weight	CWt	0.35
Subsystem A Weight	A_Wt	0.65
Subsystem B Weight	B_Wt	0.35

TABLE 19 – PARAMETER WEIGHTS – MODELS 1, 6 AND 7

STEP 2: BUILD THE OPTION INDICES

Shown below in Table 20 are the option indices for Models 1, 2, 6 and 7. Models 3, 4 and 5 modify the weights to conduct sensitivity analysis on the parameters and are shown later.

The following is the individual calculation for the Option CTD at each option index node: CTD (Option) = 1 – $(((3\text{-PV})^3*\text{PWt})+((3\text{WV})^3*\text{WWt})+((3\text{CV})^3*\text{CWt}))/8$. The detail for CTD formulation is back in Chapter 4. This formulation yields both a weighted and normalized number. Three DRL levels were chosen for the example problem. This was deemed to be complex enough to determine model effectiveness and in keeping the example problem reasonable in size. Remember than the maximum state space size given 4 options over 4 epochs would be: 27 (number of state spaces in option space) $^{4\ (\text{Number of options})}$ * 4 (Number of epochs) = 2,125,764.

Option Index	Performance DRL Value (PV)	Weight DRL Value (WV)	Cost DRL Value (CV)	Option Value CTD
1	1	1	1	0.00
2	1	1	2	0.31
3	1	1	3	0.35
4	1	2	1	0.22
5	1	2	2	0.53
6	1	2	3	0.57
7	1	3	1	0.25
8	1	3	2	0.56
9	1	3	3	0.60
10	2	1	1	0.35
11	2	1	2	0.66
12	2	1	3	0.70
13	2	2	1	0.57
14	2	2	2	0.88
15	2	2	3	0.92
16	2	3	1	0.60
17	2	3	2	0.91
18	2	3	3	0.95
19	3	1	1	0.40
20	3	1	2	0.71
21	3	1	3	0.75
22	3	2	1	0.62
23	3	2	2	0.93
24	3	2	3	0.97
25	3	3	1	0.65
26	3	3	2	0.96
27	3	3	3	1.00

Table 20 – Indices For Models 1, 6 and 7

Step 3: Extend the Option Indices to Create the Total System State Space

In all the following matrices, where the natural total or system state space is indexed, the numbers range from 1 to 531,441 and you need to know which epoch (A, B, Γ or Ω) you are in. We also reduce the system state space down to only show the actual nodes that will be in MDP model itself in a reduced system state space index. We do this, by determining all the potential paths each option could take through development. Typically, this is sparse compared to the total possibilities, particularly when we anchor the origin point. The origin point for all Models except for Model 6 is below in Table 21.

Options	A1			A2			B1			B2		
Metrics	P	W	C	P	W	C	P	W	C	P	W	C
$State_A$	1	1	1	1	1	1	1	1	1	1	1	1
Options	A1			A2			B1			B2		
Metrics	P	W	C	P	W	C	P	W	C	P	W	C
$State_\Omega$	3	3	3	3	3	3	3	3	3	3	3	3

Table 21 – Origin Point and Max Theoretical End Point

Also of note, for the example, the option indices are the same. Had we chosen different allowable DRLs for each option or different parameter weights, we would have different option indices.

Shown below in Table 22 is an example of the state space as marked by the four option indices, the corresponding system index and the reduced system index that is the state space control for the MDP to be developed. We show a smaller subset of the indices instead of the complete set. All indices are needed to correctly construct and maintain the MDP when it is built and the DP Backsolve coming from Epochs Ω to A.

Red_Sys_Ind	Sys_Ind	A1	A2	B1	B2
1	239439	13	5	13	3
2	239441	13	5	13	5
3	239447	13	5	13	11
4	239450	13	5	13	14
5	239452	13	5	13	16
6	239466	13	5	14	3
7	239468	13	5	14	5
8	239474	13	5	14	11
9	239477	13	5	14	14
10	239479	13	5	14	16
11	239547	13	5	17	3
12	239549	13	5	17	5
13	239555	13	5	17	11
315	510748	26	26	17	16
316	510978	26	26	26	3
317	510980	26	26	26	5
318	510986	26	26	26	11
319	510989	26	26	26	14
320	510991	26	26	26	16

TABLE 22 – REDUCED SYSTEM, SYSTEM AND OPTIONS INDICES

With the option, system and reduced system indices created, all value i.e. CTD, calculations can be completed. As explained in Chapter 4, if there are more than one option per subsystem being developed, the CTD calculations assume the option developments are independent regarding the total trade space. We extend this stochastic logic to calculate the subsystem CTD, as if the options' CTDs were probabilities to improve. A subsystem CTD value in this case is 1 – (the product of each option's (1 - CTD Option Value)). This is also specifically why we term CTD to being a black box process. Furthermore, we make these subsystem calculations for every node. If an option is not being invested in, it still retains its current option CTD value and it may be developed later or even down-selected “as-is” at Epoch Ω. This occurs in the real world, specifically when a technology option is developed and it does not make the expected performance or exceeds either reduced cost or weight targets.

Further research, in the independence of option development and if this black box approach provides quality data, would be beneficial. The rest of the CTD calculations for each set solution can now occur with applying the subsystem weights.

B.3 BUILD MDP

With a complete reduced system index, we know which nodes can be used for the MDP state space. We consider each node may exist at every epoch but the actual state space for the MDP is the specific allowable combinations of the options possible states at valid epochs.

STEP 1: DETERMINE MDP NETWORK’S STATE SPACE

Table 23 below shows all the elements in the Excel models to create and control the MDP state space and the snapshot of the action space and transitions. The far right three columns shows the entire potential action space for Option A1 in this case. The origin is shown on the far left. The On/Off columns are turned on or off as needed by DP solver sheets. Color:

1. Green marks what actions are currently on and the MDP network path for those values. In this case no investment in Option A1 was done at all. The CTD value is the option’s value at origin with 100% likelihood.

2. Red marks the paths that are off given the investment choices.

3. The other color choices were purely to help see the data.

A1	State at A			On/Off AB	States at B			$P\{State_B\}$	Index	On/Off BΓ	States at Γ			$P\{State_{\Gamma\|B}\}$	Index	On/Off ΓΩ	States at Ω			$P\{State_{\Omega\|\Gamma}\}$	Index	P{Path}	V{Opt}	Invest Decision		
Yes = 1 and No = 0				0						0						0								AB	BΓ	ΓΩ
13	2	2	1	0	2	2	2	0.8	14	0	3	2	2	0.5	23	0	3	3	2	0.5	26	0.000	0.956	1	1	1
				0	2	2	2	0.8	14	0	3	2	2	0.5	23	0	3	2	2	0.5	23	0.000	0.925	1	1	1
				0	2	2	2	0.8	14	0	2	2	2	0.5	14	0	3	2	2	0.9	23	0.000	0.925	1	1	1
				0	2	2	2	0.8	14	0	2	2	2	0.5	14	0	2	2	2	0.1	14	0.000	0.875	1	1	1
				0	2	2	1	0.2	13	0	2	2	2	0.9	14	0	3	2	2	0.9	23	0.000	0.925	1	1	1
				0	2	2	1	0.2	13	0	2	2	2	0.9	14	0	2	2	2	0.1	14	0.000	0.875	1	1	1
				0	2	2	1	0.2	13	0	2	2	1	0.1	13	0	2	2	2	1	14	0.000	0.875	1	1	1
				0	2	2	2	0.8	14	0	3	2	2	0.5	23	1	3	2	2	1	23	0.000	0.925	1	1	0
				0	2	2	2	0.8	14	0	2	2	2	0.5	14	1	2	2	2	1	14	0.000	0.875	1	1	0
				0	2	2	1	0.2	13	0	2	2	2	0.9	14	1	2	2	2	1	14	0.000	0.875	1	1	0
				0	2	2	1	0.2	13	0	2	2	1	0.1	13	1	2	2	1	1	13	0.000	0.569	1	1	0
				0	2	2	2	0.8	14	1	2	2	2	1	14	1	2	2	2	1	14	0.000	0.875	1	0	0
				0	2	2	1	0.2	13	1	2	2	1	1	13	1	2	2	1	1	13	0.000	0.569	1	0	0
				1	2	2	1	1	13	0	3	2	2	0.2	23	1	3	2	2	1	23	0.000	0.925	0	1	0
				1	2	2	1	1	13	0	2	2	2	0.7	14	1	2	2	2	1	14	0.000	0.875	0	1	0
				1	2	2	1	1	13	0	2	2	1	0.1	13	1	2	2	1	1	13	0.000	0.569	0	1	0
				1	2	2	1	1	13	0	3	2	2	0.2	23	0	3	3	2	0.5	26	0.000	0.956	0	1	1
				1	2	2	1	1	13	0	3	2	2	0.2	23	0	3	2	2	0.5	23	0.000	0.925	0	1	1
				1	2	2	1	1	13	0	2	2	2	0.7	14	0	3	2	2	0.9	23	0.000	0.925	0	1	1
				1	2	2	1	1	13	0	2	2	2	0.7	14	0	2	2	2	0.1	14	0.000	0.875	0	1	1
				1	2	2	1	1	13	0	2	2	1	0.1	13	0	2	2	2	1	14	0.000	0.875	0	1	1
				1	2	2	1	1	13	1	2	2	1	1	13	1	2	2	1	1	13	1.000	0.569	0	0	0

TABLE 23 – TOTAL VIEW OF OPTION A1'S MDP NETWORK (STATE, ACTION AND TRANSITION)

The following Tables 24 through 27 are the individual options' total possible MDP Network paths. The probabilities are specific epoch to epoch. Each row is unique for all actions.

On/Off AB	States at B			$P\{State_B\}$	Index	On/Off BΓ	States at Γ			$P\{State_{\Gamma\|B}\}$	Index	On/Off ΓΩ	States at Ω			$P\{State_{\Omega\|\Gamma}\}$	Index
0						0						0					
0	2	2	2	0.8	14	0	3	2	2	0.5	23	0	3	3	2	0.5	26
0	2	2	2	0.8	14	0	3	2	2	0.5	23	0	3	2	2	0.5	23
0	2	2	2	0.8	14	0	2	2	2	0.5	14	0	3	2	2	0.9	23
0	2	2	2	0.8	14	0	2	2	2	0.5	14	0	2	2	2	0.1	14
0	2	2	1	0.2	13	0	2	2	2	0.9	14	0	3	2	2	0.9	23
0	2	2	1	0.2	13	0	2	2	2	0.9	14	0	2	2	2	0.1	14
0	2	2	1	0.2	13	0	2	2	1	0.1	13	0	2	2	2	1	14
0	2	2	2	0.8	14	0	3	2	2	0.5	23	1	3	2	2	1	23
0	2	2	2	0.8	14	0	2	2	2	0.5	14	1	2	2	2	1	14
0	2	2	1	0.2	13	0	2	2	2	0.9	14	1	2	2	2	1	14
0	2	2	1	0.2	13	0	2	2	1	0.1	13	1	2	2	1	1	13
0	2	2	2	0.8	14	1	2	2	2	1	14	1	2	2	2	1	14
0	2	2	1	0.2	13	1	2	2	1	1	13	1	2	2	1	1	13
1	2	2	1	1	13	0	3	2	2	0.2	23	1	3	2	2	1	23
1	2	2	1	1	13	0	2	2	2	0.7	14	1	2	2	2	1	14
1	2	2	1	1	13	0	2	2	1	0.1	13	1	2	2	1	1	13
1	2	2	1	1	13	0	3	2	2	0.2	23	0	3	3	2	0.5	26
1	2	2	1	1	13	0	3	2	2	0.2	23	0	3	2	2	0.5	23
1	2	2	1	1	13	0	2	2	2	0.7	14	0	3	2	2	0.9	23
1	2	2	1	1	13	0	2	2	2	0.7	14	0	2	2	2	0.1	14
1	2	2	1	1	13	0	2	2	1	0.1	13	0	2	2	2	1	14
1	2	2	1	1	13	1	2	2	1	1	13	1	2	2	1	1	13

TABLE 24 - OPTION A1 MDP

On/Off AB	States at B			$P\{State_B\}$	Index	On/Off BΓ	States at Γ			$P\{State_{\Gamma\|B}\}$	Index	On/Off ΓΩ	States at Ω			$P\{State_{\Omega\|\Gamma}\}$	Index
1						1						1					
1	2	2	2	0.4	14	1	2	3	2	0.5	17	1	3	3	2	0.3	26
1	2	2	2	0.4	14	1	2	3	2	0.5	17	1	2	3	2	0.7	17
1	2	2	2	0.4	14	1	2	2	2	0.5	14	1	2	3	2	0.8	17
1	2	2	2	0.4	14	1	2	2	2	0.5	14	1	2	2	2	0.2	14
1	1	2	2	0.6	5	1	2	2	2	0.6	14	1	2	3	2	0.8	17
1	1	2	2	0.6	5	1	2	2	2	0.6	14	1	2	2	2	0.2	14
1	1	2	2	0.6	5	1	1	2	2	0.4	5	1	2	2	2	1	14
1	2	2	2	0.4	14	1	2	3	2	0.5	17	0	2	3	2	1	17
1	2	2	2	0.4	14	1	2	2	2	0.5	14	0	2	2	2	1	14
1	1	2	2	0.6	5	1	2	2	2	0.6	14	0	2	2	2	1	14
1	1	2	2	0.6	5	1	1	2	2	0.4	5	0	1	2	2	1	5
1	2	2	2	0.4	14	0	2	2	2	1	14	0	2	2	2	1	14
1	1	2	2	0.6	5	0	1	2	2	1	5	0	1	2	2	1	5
0	1	2	2	1	5	0	2	3	2	0.1	17	0	2	3	2	1	17
0	1	2	2	1	5	0	2	2	2	0.8	14	0	2	2	2	1	14
0	1	2	2	1	5	0	1	2	2	0.1	5	0	1	2	2	1	5
0	1	2	2	1	5	0	2	3	2	0.1	17	0	3	3	2	0.3	26
0	1	2	2	1	5	0	2	3	2	0.1	17	0	2	3	2	0.7	17
0	1	2	2	1	5	0	2	2	2	0.8	14	0	2	3	2	0.8	17
0	1	2	2	1	5	0	2	2	2	0.8	14	0	2	2	2	0.2	14
0	1	2	2	1	5	0	1	2	2	0.1	5	0	2	2	2	1	14
0	1	2	2	1	5	0	1	2	2	1	5	1	1	2	2	1	5

Table 25 - Option A2 MDP

On/Off AB	States at B			$P\{State_B\}$	Index	On/Off BΓ	States at Γ			$P\{State_{\Gamma\|B}\}$	Index	On/Off ΓΩ	States at Ω			$P\{State_{\Omega\|\Gamma}\}$	Index
0						0						0					
0	2	2	2	0.7	14	0	2	3	2	0.2	17	0	3	3	2	0.2	26
0	2	2	2	0.7	14	0	2	3	2	0.2	17	0	2	3	2	0.8	17
0	2	2	2	0.7	14	0	2	2	2	0.8	14	0	2	3	2	0.6	17
0	2	2	2	0.7	14	0	2	2	2	0.8	14	0	2	2	2	0.4	14
0	2	2	1	0.3	13	0	2	2	2	0.6	14	0	2	3	2	0.6	17
0	2	2	1	0.3	13	0	2	2	2	0.6	14	0	2	2	2	0.4	14
0	2	2	1	0.3	13	0	2	2	1	0.4	13	0	2	2	2	1	14
0	2	2	2	0.7	14	0	2	3	2	0.2	17	1	2	3	2	1	17
0	2	2	2	0.7	14	0	2	2	2	0.8	14	1	2	2	2	1	14
0	2	2	1	0.3	13	0	2	2	2	0.6	14	1	2	2	2	1	14
0	2	2	1	0.3	13	0	2	2	1	0.4	13	1	2	2	1	1	13
0	2	2	2	0.7	14	1	2	2	2	1	14	1	2	2	2	1	14
0	2	2	1	0.3	13	1	2	2	1	1	13	1	2	2	1	1	13
1	2	2	1	1	13	0	2	2	2	0.8	14	1	2	2	2	1	14
1	2	2	1	1	13	0	2	2	1	0.2	13	1	2	2	1	1	13
1	2	2	1	1	13	0	2	2	2	0.8	14	0	2	3	2	0.6	17
1	2	2	1	1	13	0	2	2	2	0.8	14	0	2	2	2	0.4	14
1	2	2	1	1	13	0	2	2	1	0.2	13	0	2	2	2	1	14
1	2	2	1	1	13	1	2	2	1	1	13	1	2	2	1	1	13

Table 26 - Option B1 MDP

On/Off AB	States at B			$P\{State_B\}$	Index	On/Off BΓ	States at Γ			$P\{State_{\Gamma\|B}\}$	Index	On/Off ΓΩ	States at Ω			$P\{State_{\Omega\|\Gamma}\}$	Index
0						0						0					
0	2	1	2	0.4	11	0	2	2	2	0.1	14	0	2	2	3	0.1	15
0	2	1	2	0.4	11	0	2	2	2	0.1	14	0	2	2	2	0.9	14
0	2	1	2	0.4	11	0	2	1	2	0.9	11	0	2	2	2	0.4	14
0	2	1	2	0.4	11	0	2	1	2	0.9	11	0	2	1	2	0.6	11
0	1	2	2	0.4	5	0	2	2	2	0.5	14	0	2	2	3	0.1	15
0	1	2	2	0.4	5	0	2	2	2	0.5	14	0	2	2	2	0.9	14
0	1	2	2	0.4	5	0	1	2	2	0.5	5	0	2	2	2	0.8	14
0	1	2	2	0.4	5	0	1	2	2	0.5	5	0	1	2	2	0.2	5
0	1	1	3	0.2	3	0	2	1	2	0.4	11	0	2	2	2	0.4	14
0	1	1	3	0.2	3	0	2	1	2	0.4	11	0	2	1	2	0.6	11
0	1	1	3	0.2	3	0	1	2	2	0.6	5	0	2	2	2	0.8	14
0	1	1	3	0.2	3	0	1	2	2	0.6	5	0	1	2	2	0.2	5
0	2	1	2	0.4	11	0	2	2	2	0.1	14	1	2	2	2	1	14
0	2	1	2	0.4	11	0	2	1	2	0.9	11	1	2	1	2	1	11
0	1	2	2	0.4	5	0	2	2	2	0.2	14	1	2	2	2	1	14
0	1	2	2	0.4	5	0	1	2	2	0.8	5	1	1	2	2	1	5
0	1	1	3	0.2	3	0	2	1	2	0.4	11	1	2	1	2	1	11
0	1	1	3	0.2	3	0	1	2	2	0.6	5	1	1	2	2	1	5
0	2	1	2	0.4	11	1	2	1	2	1	11	1	2	1	2	1	11
0	1	2	2	0.4	5	1	1	2	2	1	5	1	1	2	2	1	5
0	1	1	3	0.2	3	1	1	1	3	1	3	1	1	1	3	1	3
1	1	1	3	1	3	0	2	1	2	0.3	11	1	2	1	2	1	11
1	1	1	3	1	3	0	1	2	2	0.5	5	1	1	2	2	1	5
1	1	1	3	1	3	0	1	1	3	0.2	3	1	1	1	3	1	3
1	1	1	3	1	3	0	2	1	2	0.3	11	0	2	2	2	0.4	14
1	1	1	3	1	3	0	2	1	2	0.3	11	0	2	1	2	0.6	11
1	1	1	3	1	3	0	1	2	2	0.5	5	0	2	2	2	0.8	14
1	1	1	3	1	3	0	1	2	2	0.5	5	0	1	2	2	0.2	5
1	1	1	3	1	3	0	1	1	3	0.2	3	0	1	2	2	1	5
1	1	1	3	1	3	1	1	1	3	1	3	1	1	1	3	1	3

TABLE 27 - OPTION B2 MDP

STEP 2: ADD ACTION SPACE

Table 23 above showed the action space tied to the MDP Network for Option A1. The Action Space itself is controlled from the primary control sheet. Table 24 below shows the possible Action Space for the model. Red represents actions that are outside of the phase budget and not available for the DP problem solve. Green represents what is available. For Model 1, we only had a slightly tighter budget for the final phase to ensure a budget restriction would control the solution. All other models had realistic and varying budget controls which we used for both primary quantitative and sensitivity analyses of the models. The second B to Γ column is used for Model 7 where we allowed an epochal investment skip.

Actions	A1	A2	B1	B2	A to B		B to Γ		B to Γ	Γ to Ω
a_1	0	0	0	0	1		1			1
a_2	0	0	0	1	1		1			1
a_3	0	0	1	0	1		1			1
a_4	0	0	1	1	1		1			1
a_5	0	1	0	0	1		1			1
a_6	0	1	0	1	1		1			1
a_7	0	1	1	0	1		1			1
a_8	0	1	1	1	1		1			1
a_9	1	0	0	0	1		1			1
a_{10}	1	0	0	1	1		1			1
a_{11}	1	0	1	0	1		1			1
a_{12}	1	0	1	1	1		1			1
a_{13}	1	1	0	0	1		1			1
a_{14}	1	1	0	1	1		1			0
a_{15}	1	1	1	0	1		1			0
a_{16}	1	1	1	1	1	XX	1	XX		0

TABLE 28 - MODEL 1 ACTION SPACE

On the same control sheet for the model, we also maintain the Black Box calculator and forward looking Expected Value given those actions. See Figure xx below. This is different form the actual Expected Value of a given action when solving recursively which will be discussed below in the DP Backsolve section.

Black Box Calculator						
Phase	Act_Choice	A1	A2	B1	B2	EV(System)
A to B	a16	1	1	1	1	
B to Γ	a16	1	1	1	1	
Γ to Ω	a12	1	0	1	1	
EV(Option)		0.93	0.80	0.89	0.80	
EV(Subsystem)		A EV	0.99	B EV	0.98	0.9829

Table 29 - Black Box (Forward Look)

Step 3: Add The Transitions

Tables 23 through 27 all show the transition likelihood moving from state to state for the specific options. We feed those tables from a Transition Probability Matrix. Table 30 below shows the likelihood for state to state movement for Model 1. This is specifically for Epoch B to Epoch Γ.

OPT	IN		OUT		P{X}
	EPOCH	NODE	EPOCH	NODE	
A1	**B**	13	**Γ**	13	0.1
A1	**B**	13	**Γ**	14	0.9
A1	**B**	14	**Γ**	14	0.5
A1	**B**	14	**Γ**	23	0.5
A2	**B**	5	**Γ**	5	0.4
A2	**B**	5	**Γ**	14	0.6
A2	**B**	14	**Γ**	14	0.5
A2	**B**	14	**Γ**	17	0.5
B1	**B**	13	**Γ**	13	0.4
B1	**B**	13	**Γ**	14	0.6
B1	**B**	14	**Γ**	14	0.8
B1	**B**	14	**Γ**	17	0.2
B2	**B**	3	**Γ**	5	0.6
B2	**B**	3	**Γ**	11	0.4
B2	**B**	5	**Γ**	5	0.5
B2	**B**	5	**Γ**	14	0.5
B2	**B**	11	**Γ**	11	0.9
B2	**B**	11	**Γ**	14	0.1

TABLE 30 - EPOCH B TO EPOCH Γ TRANSITION PROBABILITIES

Before conducting the DP Backsolve, we build the actual MDP from Epochs A through Ω. Those sheets also contain the DP Backsolve calculations are shown in the next section.

B.4 DP BACKSOLVE

STEP 1: COMPLETE MDP BUILD

Table 31 below shows Model 1's allowable MDP nodes from Epoch A to Epoch B.

State	Opt Index				Epoch	Pred
Index	A1	A2	B1	B2	In	Node
1	13	5	13	3	A	Origin
1	13	5	13	3	B	1
2	13	5	13	5	B	1
3	13	5	13	11	B	1
6	13	5	14	3	B	1
7	13	5	14	5	B	1
8	13	5	14	11	B	1
21	13	14	13	3	B	1
22	13	14	13	5	B	1
23	13	14	13	11	B	1
26	13	14	14	3	B	1
27	13	14	14	5	B	1
28	13	14	14	11	B	1
81	14	5	13	3	B	1
82	14	5	13	5	B	1
83	14	5	13	11	B	1
86	14	5	14	3	B	1
87	14	5	14	5	B	1
88	14	5	14	11	B	1
101	14	14	13	3	B	1
102	14	14	13	5	B	1
103	14	14	13	11	B	1
106	14	14	14	3	B	1
107	14	14	14	5	B	1
108	14	14	14	11	B	1

TABLE 31 - MODEL 1 MDP - EPOCH A TO EPOCH B

Table 32 shows how we track the next step to build going from Epoch B to Epoch Γ. Given Excel's structure, this was a manual process. Much of the spreadsheets are automated, but the actual builds of the MDP's and their corresponding DP back solves have manual elements. It is expected that the first extension of this algorithm would be code it as a software package. Excel was useful for the testing and analysis in the dissertation effort, but is highly inefficient as a software solution. Table 32 below shows the in and out structure to both build and track the

MDP itself and then to provide control for the back solve. Additionally, the system index and reduced system index came from these tables. The indices were cleaned separately in a separate sheet and then copied back into the DP solve sheets.

Yellow row 1 shows all the possible option combinations for going from Epoch A to Epoch B. For example, Option A1, at Epoch A can go from Option A1 index 13 to either Option A1 indices 13 or 14 at Epoch B. Green rows are all the possible combinations of transitions coming from Epoch A to Epoch B. Each green row also contains the possible combinations for then going from Epoch B to Epoch Γ. At this point, the Node/Arc structure of the MDP is too big to show efficiently. Table 33 below just shows the very first set of transitional arcs Epoch B to Epoch Γ. There are 24 sets in Model 1 to cover all the Epoch B to Epoch Γ transitional arcs.

Sys_Red	System	Opt Index In				Epoch	Pred	Opt Index Out				Opt Index Out				Opt Index Out			
Index	Index	A1	A2	B1	B2	In	Node	A1	A2	B1	B2	A1	A2	B1	B2	A1	A2	B1	B2
1	239439	13	5	13	3	A	Orig	13	5	13	3	14	14	14	5				11
1	239439	13	5	13	3	B	1	13	5	13	3	14	14	14	5				11
2	239441	13	5	13	5	B	1	13	5	13	5	14	14	14	14				
3	239447	13	5	13	11	B	1	13	5	13	11	14	14	14	14				
6	239466	13	5	14	3	B	1	13	5	14	3	14	14	17	5				11
7	239468	13	5	14	5	B	1	13	5	14	5	14	14	17	14				
8	239474	13	5	14	11	B	1	13	5	14	11	14	14	17	14				
21	246000	13	14	13	3	B	1	13	14	13	3	14	17	14	5				11
22	246002	13	14	13	5	B	1	13	14	13	5	14	17	14	14				
23	246008	13	14	13	11	B	1	13	14	13	11	14	17	14	14				
26	246027	13	14	14	3	B	1	13	14	14	3	14	17	17	5				11
27	246029	13	14	14	5	B	1	13	14	14	5	14	17	17	14				
28	246035	13	14	14	11	B	1	13	14	14	11	14	17	17	14				
81	259122	14	5	13	3	B	1	14	5	13	3	23	14	14	5				11
82	259124	14	5	13	5	B	1	14	5	13	5	23	14	14	14				
83	259130	14	5	13	11	B	1	14	5	13	11	23	14	14	14				
86	259149	14	5	14	3	B	1	14	5	14	3	23	14	17	5				11
87	259151	14	5	14	5	B	1	14	5	14	5	23	14	17	14				
88	259157	14	5	14	11	B	1	14	5	14	11	23	14	17	14				
101	265683	14	14	13	3	B	1	14	14	13	3	23	17	14	5				11
102	265685	14	14	13	5	B	1	14	14	13	5	23	17	14	14				
103	265691	14	14	13	11	B	1	14	14	13	11	23	17	14	14				
106	265710	14	14	14	3	B	1	14	14	14	3	23	17	17	5				11
107	265712	14	14	14	5	B	1	14	14	14	5	23	17	17	14				
108	265718	14	14	14	11	B	1	14	14	14	11	23	17	17	14				

TABLE 32 - EPOCH A TO EPOCH B TO EPOCH Γ TRANSITION ARCS

Sys_Red	System	Opt Index In				Epoch	Pred
Index	Index	A1	A2	B1	B2	In	Node
1	239439	13	5	13	3	Γ	1
2	239441	13	5	13	5	Γ	1
3	239447	13	5	13	11	Γ	1
6	239466	13	5	14	3	Γ	1
7	239468	13	5	14	5	Γ	1
8	239474	13	5	14	11	Γ	1
21	246000	13	14	13	3	Γ	1
22	246002	13	14	13	5	Γ	1
23	246008	13	14	13	11	Γ	1
26	246027	13	14	14	3	Γ	1
27	246029	13	14	14	5	Γ	1
28	246035	13	14	14	11	Γ	1
81	259122	14	5	13	3	Γ	1
82	259124	14	5	13	5	Γ	1
83	259130	14	5	13	11	Γ	1
86	259149	14	5	14	3	Γ	1
87	259151	14	5	14	5	Γ	1
88	259157	14	5	14	11	Γ	1
101	265683	14	14	13	3	Γ	1
102	265685	14	14	13	5	Γ	1
103	265691	14	14	13	11	Γ	1
106	265710	14	14	14	3	Γ	1
107	265712	14	14	14	5	Γ	1
108	265718	14	14	14	11	Γ	1

Table 33 - Transition Arcs From Node 1 at Epoch B to Epoch Γ

Step 2 also contains this expanding structure of laying out the complete MDP but also includes the initial recursive DP step from Epoch Ω to Epoch Γ.

Step 2: Finish MDP Build and Conduct Initial Recursive DP Step

Table 34 below shows multiple things going from left to right:

1. The far left third is final set of transitional arcs for the first node from Epoch Γ to Epoch Ω,
2. Matrix controls to multiply the correct values for actions,
3. Values for the options, subsystems and specific set solutions at the Epoch Ω end nodes and
4. A control structure to calculate the correct Expected Values for determination of the optimal actions going recursively.

SysRed	System	Opt Index In				Epoch	Pred	Opt Index Stay				Opt Index Up			
Index	Index	A1	A2	B1	B2	In	Node	A1	A2	B1	B2	A1	A2	B1	B2
1	239439	13	5	13	3	Γ		13	5	13	3	14	14	14	5
1	239439	13	5	13	3	Ω		1	1	1	1	0	0	0	0
2	239441	13	5	13	5	Ω		1	1	1	0	0	0	0	1
6	239466	13	5	14	3	Ω		1	1	0	1	0	0	1	0
7	239468	13	5	14	5	Ω		1	1	0	0	0	0	1	1
21	246000	13	14	13	3	Ω		1	0	1	1	0	1	0	0
22	246002	13	14	13	5	Ω		1	0	1	0	0	1	0	1
26	246027	13	14	14	3	Ω		1	0	0	1	0	1	1	0
27	246029	13	14	14	5	Ω		1	0	0	0	0	1	1	1
81	259122	14	5	13	3	Ω		0	1	1	1	1	0	0	0
82	259124	14	5	13	5	Ω		0	1	1	0	1	0	0	1
86	259149	14	5	14	3	Ω		0	1	0	1	1	0	1	0
87	259151	14	5	14	5	Ω		0	1	0	0	1	0	1	1
101	265683	14	14	13	3	Ω		0	0	1	1	1	1	0	0
102	265685	14	14	13	5	Ω		0	0	1	0	1	1	0	1
106	265710	14	14	14	3	Ω		0	0	0	1	1	1	1	0
107	265712	14	14	14	5	Ω		0	0	0	0	1	1	1	1

SysRed								Action	Index Actions				Invest P(Stay)				Invest P(Up)		
Index	V{X}	V{A1}	V{A2}	V{A}	V{B1}	V{B2}	V{B}	Γ to Ω	A1	A2	B1	B2	A1	A2	B1	B2	A1	A2	B1
1	**0.77**	0.57	0.53	0.80	0.57	0.35	0.72	a16	1	1	1	1	0.00	0.00	0.00	0.00	1.00	1.00	1.00
1	**0.77**	0.57	0.53	0.80	0.57	0.35	0.72	0.00	0.00	0.00	0.00	0.00	a_1	0.77	a_{13}				
2	**0.80**	0.57	0.53	0.80	0.57	0.53	0.80	0.00	0.00	0.00	0.00	1.00	a_2	0.80	a_{12}				
6	**0.84**	0.57	0.53	0.80	0.88	0.35	0.92	0.00	0.00	0.00	1.00	0.00	a_3	0.84	a_{12}				
7	**0.85**	0.57	0.53	0.80	0.88	0.53	0.94	0.00	0.00	0.00	1.00	1.00	a_4	0.85	a_{12}				
21	**0.87**	0.57	0.88	0.95	0.57	0.35	0.72	0.00	0.00	1.00	0.00	0.00	a_5	0.87	a_{13}				
22	**0.89**	0.57	0.88	0.95	0.57	0.53	0.80	0.00	0.00	1.00	0.00	1.00	a_6	0.89	a_8				
26	**0.94**	0.57	0.88	0.95	0.88	0.35	0.92	0.00	0.00	1.00	1.00	0.00	a_7	0.94	a_8				
27	**0.94**	0.57	0.88	0.95	0.88	0.53	0.94	0.00	0.00	1.00	1.00	1.00	a_8	0.94					
81	**0.86**	0.88	0.53	0.94	0.57	0.35	0.72	0.00	1.00	0.00	0.00	0.00	a_9	0.86	a_{13}				
82	**0.89**	0.88	0.53	0.94	0.57	0.53	0.80	0.00	1.00	0.00	0.00	1.00	a_{10}	0.89	a_{12}				
86	**0.93**	0.88	0.53	0.94	0.88	0.35	0.92	0.00	1.00	0.00	1.00	0.00	a_{11}	0.93	a_{12}				
87	**0.94**	0.88	0.53	0.94	0.88	0.53	0.94	0.00	1.00	0.00	1.00	1.00	a_{12}	0.94					
101	**0.89**	0.88	0.88	0.98	0.57	0.35	0.72	0.00	1.00	1.00	0.00	0.00	a_{13}	0.89					
102	**0.92**	0.88	0.88	0.98	0.57	0.53	0.80	0.00	1.00	1.00	0.00	1.00	a_{14}	INF					
106	**0.96**	0.88	0.88	0.98	0.88	0.35	0.92	0.00	1.00	1.00	1.00	0.00	a_{15}	INF					
107	**0.97**	0.88	0.88	0.98	0.88	0.53	0.94	1.00	1.00	1.00	1.00	1.00	a_{16}	INF					
								1.00	**Action Value**			**0.97**	**a8**	**0.94**	**Max Action Value**				

TABLE 34 - FINAL MDP BUILD AND FIRST DP STEP

Table 35 below is a closeup of the left half of Table 34 for discussion. This is the first set of node-arcs from Epoch Γ to Epoch Ω. There are 108 sets in the total model. As per the previous steps, the first row represents the start node and the remaining rows represent the arcs to all the final nodes. The Option Index Stay to Up Section represents all the possible combinations for the unique option sets for their movements. For example, in the third row, that unique set will not change the state of options A1, A2 and B1. B2 will move from Option State 3 to Option State

5. The 0's and 1's allow us to correctly conduct the Expected Values for the actions. The value rows are the CTD calculations are unique for the specific state nodes.

SysRed	System	Opt Index In				Epoch	Pred	Opt Index Stay				Opt Index Up										
Index	Index	A1	A2	B1	B2	In	Node	A1	A2	B1	B2	A1	A2	B1	B2	V{X}	V{A1}	V{A2}	V{A}	V{B1}	V{B2}	V{B}
1	239439	13	5	13	3	Γ		13	5	13	3	14	14	14	5	**0.77**	0.57	0.53	0.80	0.57	0.35	0.72
1	239439	13	5	13	3	Ω		1	1	1	1	0	0	0	0	**0.77**	0.57	0.53	0.80	0.57	0.35	0.72
2	239441	13	5	13	5	Ω		1	1	1	0	0	0	0	1	**0.80**	0.57	0.53	0.80	0.57	0.53	0.80
6	239466	13	5	14	3	Ω		1	1	0	1	0	0	1	0	**0.84**	0.57	0.53	0.80	0.88	0.35	0.92
7	239468	13	5	14	5	Ω		1	1	0	0	0	0	1	1	**0.85**	0.57	0.53	0.80	0.88	0.53	0.94
21	246000	13	14	13	3	Ω		1	0	1	1	0	1	0	0	**0.87**	0.57	0.88	0.95	0.57	0.35	0.72
22	246002	13	14	13	5	Ω		1	0	1	0	0	1	0	1	**0.89**	0.57	0.88	0.95	0.57	0.53	0.80
26	246027	13	14	14	3	Ω		1	0	0	1	0	1	1	0	**0.94**	0.57	0.88	0.95	0.88	0.35	0.92
27	246029	13	14	14	5	Ω		1	0	0	0	0	1	1	1	**0.94**	0.57	0.88	0.95	0.88	0.53	0.94
81	259122	14	5	13	3	Ω		0	1	1	1	1	0	0	0	**0.86**	0.88	0.53	0.94	0.57	0.35	0.72
82	259124	14	5	13	5	Ω		0	1	1	0	1	0	0	1	**0.89**	0.88	0.53	0.94	0.57	0.53	0.80
86	259149	14	5	14	3	Ω		0	1	0	1	1	0	1	0	**0.93**	0.88	0.53	0.94	0.88	0.35	0.92
87	259151	14	5	14	5	Ω		0	1	0	0	1	0	1	1	**0.94**	0.88	0.53	0.94	0.88	0.53	0.94
101	265683	14	14	13	3	Ω		0	0	1	1	1	1	0	0	**0.89**	0.88	0.88	0.98	0.57	0.35	0.72
102	265685	14	14	13	5	Ω		0	0	1	0	1	1	0	1	**0.92**	0.88	0.88	0.98	0.57	0.53	0.80
106	265710	14	14	14	3	Ω		0	0	0	1	1	1	1	0	**0.96**	0.88	0.88	0.98	0.88	0.35	0.92
107	265712	14	14	14	5	Ω		0	0	0	0	1	1	1	1	**0.97**	0.88	0.88	0.98	0.88	0.53	0.94

TABLE 35 - CLOSEUP OF 1ST BACKSOLVE STEP - ARCS AND VALUES

Table 36 below is a closeup of the left half of Table 34 for discussion. Table 36 shows where the calculations and the controls are to solve the Action Expected Values.

Action	Index Actions				Invest P(Stay)				Invest P(Up)				Invest (ALL) P(Up)			
Γ to Ω	A1	A2	B1	B2	A1	A2	B1	B2	A1	A2	B1	B2	A1	A2	B1	B2
a16	1	1	1	1	0.00	0.00	0.00	0.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
0.00	0.00	0.00	0.00	0.00	a_{1}	0.77	a_{13}									
0.00	0.00	0.00	0.00	1.00	a_{2}	0.80	a_{12}									
0.00	0.00	0.00	1.00	0.00	a_{3}	0.84	a_{12}									
0.00	0.00	0.00	1.00	1.00	a_{4}	0.85	a_{12}									
0.00	0.00	1.00	0.00	0.00	a_{5}	0.87	a_{13}									
0.00	0.00	1.00	0.00	1.00	a_{6}	0.89	a_{8}									
0.00	0.00	1.00	1.00	0.00	a_{7}	0.94	a_{8}									
0.00	0.00	1.00	1.00	1.00	a_{8}	0.94										
0.00	1.00	0.00	0.00	0.00	a_{9}	0.86	a_{13}									
0.00	1.00	0.00	0.00	1.00	a_{10}	0.89	a_{12}									
0.00	1.00	0.00	1.00	0.00	a_{11}	0.93	a_{12}									
0.00	1.00	0.00	1.00	1.00	a_{12}	0.94										
0.00	1.00	1.00	0.00	0.00	a_{13}	0.89										
0.00	1.00	1.00	0.00	1.00	a_{14}	INF										
0.00	1.00	1.00	1.00	0.00	a_{15}	INF										
1.00	1.00	1.00	1.00	1.00	a_{16}	INF										
1.00	**Action Value**			**0.97**	**a8**	**0.94**	**Max Action Value**									

TABLE 36 - CLOSEUP OF 1ST BACKSOLVE - ACTION EXPECTED VALUES

There are automated and manual steps to determine the Expected Values. The white, yellow and green cells underneath the Action Invest section are all pasted data from the automated calculations conducted in the left third, the Action Values which are the Expected Values. In the example, we know that Actions 1 through 13 are allowable and Actions 14 through 16 are over budget, so we mark them as Infeasible. We also know that Actions 8, 12 and 13 dominate the yellow actions because they are under budget and contain the options as subsets in their set solutions. Thus, we only need to solve for Actions 8, 12 and 13. We did solve all the actions in yellow to just check. The green colored data in the Invest P(Stay) B1 column show which action dominates. Refer to Table 28 for verification. For example, Action 12 yields an EV = 0.94. Action 11, which is dominated by Action 12 yields an EV = 0.93.

All the action calculations are automated. The upper left corner is a pull-down menu. When you select the action where you want to determine its EV, it automatically conducts all the matrices manipulations and calculates the action EV. The third row is another set of controls to ensure that the correct CTDs and their likelihood of occurrences are calculated correctly.

Step 3: Store Best Expected Values and Continue Recursive DP Solve

Table 37 below shows the ordered Top 30 out of the 108 Epoch Γ nodes that were solved. The best Action EV is Action 13 with value: 0.9927. Since Model 1 only had restricted budget in its Final Phase, Action 16 would be the optimal action for the first two development phases from Epoch A to Epoch B and Epoch B to Epoch Γ. The remainder of this step will be explained in the next section with Model 2.

Red Index	Index	A1	A2	B1	B2	Epoch	Action	CTD EV
214	445136	23	17	17	14	Γ	a13	0.9929
214	445136	23	17	17	14	Γ	a13	0.9929
194	442949	23	14	17	14	Γ	a13	0.9920
134	267989	14	17	17	14	Γ	a13	0.9918
209	445055	23	17	14	14	Γ	a13	0.9915
174	436388	23	5	17	14	Γ	a13	0.9911
114	265802	14	14	17	14	Γ	a13	0.9907
189	442868	23	14	14	14	Γ	a13	0.9907
212	445127	23	17	17	5	Γ	a12	0.9907
129	267908	14	17	14	14	Γ	a13	0.9904
204	445028	23	17	13	14	Γ	a12	0.9897
169	436307	23	5	14	14	Γ	a13	0.9897
54	248306	13	17	17	14	Γ	a13	0.9895
192	442940	23	14	17	5	Γ	a12	0.9895
132	267980	14	17	17	5	Γ	a12	0.9894
94	259241	14	5	17	14	Γ	a13	0.9894
109	265721	14	14	14	14	Γ	a13	0.9893
207	445046	23	17	14	5	Γ	a12	0.9891
213	445133	23	17	17	11	Γ	a12	0.9889
184	442841	23	14	13	14	Γ	a12	0.9885
124	267881	14	17	13	14	Γ	a12	0.9885
172	436379	23	5	17	5	Γ	a8	0.9882
49	248225	13	17	14	14	Γ	a13	0.9881
89	259160	14	5	14	14	Γ	a13	0.9880
187	442859	23	14	14	5	Γ	a12	0.9879
127	267899	14	17	14	5	Γ	a12	0.9879
202	445019	23	17	13	5	Γ	a12	0.9879
112	265793	14	14	17	5	Γ	a12	0.9878
34	246119	13	14	17	14	Γ	a13	0.9878
193	442946	23	14	17	11	Γ	a12	0.9877

Table 37 - Epoch Ω to Epoch Γ Best Earned Values

B.5 HIGH-LEVEL DATA ALL MODELS

MODEL 2 DATA

Model 2 data differs from Model 1 in having restricted budgets for all phases. A future research idea would be to have a general budget for all phases. There is promise in considering a looser structure for phase budgeting, to see if set solution selection could improve. However, our research scope focused on the current DoD budgeting structure and similar tending commercial large program structures that break down budgets into phases. Altering those budgets requires executive approval. Our sensitivity analysis covers considering budget changes with Model 2. Table 38 below shows the modified budget for Models 2 through 6.

Costs	Phase	A1	A2	B1	B2	Budget - $M
Dev	A to B	14	6	4	2	20
Dev (continue)	B to Γ	8	9	5	4	20
Dev and SI	Γ to Ω	10	10	6	5	16
SI	Ω to TP	5	4	3	3	8
Reserve						9
Total Budget						73

TABLE 38 - MODELS 2 THROUGH 6 BUDGET

Table 39 below shows how the modified budget impacts the allowable actions for the model. Red is over budget and infeasible. Green is feasible. The three Dom columns mark actions that must be solved with double x's or if an action is dominated by another action.

Actions	A1	A2	B1	B2		A to B	Dom	B to Γ	Dom	Γ to Ω	Dom
a1	0	0	0	0		1	All	1	All	1	All
a2	0	0	0	1		1	a12	1	a12	1	a10
a3	0	0	1	0		1	a12	1	a12	1	a11
a4	0	0	1	1		1	a8	1	a12	1	XX
a5	0	1	0	0		1	a13	1	a13	1	a7
a6	0	1	0	1		1	a8	1	a8	1	XX
a7	0	1	1	0		1	a8	1	a8	1	XX
a8	0	1	1	1		1	XX	1	XX	0	
a9	1	0	0	0		1	a13	1	a13	1	a12
a10	1	0	0	1		1	XX	1	XX	1	XX
a11	1	0	1	0		1	XX	1	XX	1	XX
a12	1	0	1	1		1	XX	1	XX	0	
a13	1	1	0	0		1	XX	1	XX	0	
a14	1	1	0	1		0		0		0	
a15	1	1	1	0		0		0		0	
a16	1	1	1	1		0		0		0	

TABLE 39 - MODEL 2 ACTION SPACE AND ACTIONS

MODEL 2 DP SOLVE

Since the budget in the earlier phases of Model 2 make complete option development infeasible, we will show the remaining steps of how the DP solve is conducted back to Epoch A when all the phases cannot be solved by inspection. Additionally, we will also show that the model needs to be pruned once the optimal action at Epoch A is determined as any options not developed from Epoch A forward can be considered for development after Epoch B.

Table 40 below shows the Top 30 EV's from the Model 2 Epoch Ω to Epoch Γ Solve.

Index	In	Action	CTD EV
214	Γ	a11	0.9927
209	Γ	a11	0.9917
194	Γ	a11	0.9915
134	Γ	a11	0.9915
204	Γ	a11	0.9909
189	Γ	a11	0.9905
129	Γ	a11	0.9905
174	Γ	a7	0.9902
212	Γ	a10	0.9900
114	Γ	a11	0.9898
184	Γ	a11	0.9897
124	Γ	a11	0.9897
169	Γ	a7	0.9893
109	Γ	a11	0.9889
192	Γ	a10	0.9888
132	Γ	a10	0.9887
54	Γ	a11	0.9887
164	Γ	a7	0.9884
207	Γ	a4	0.9882
89	Γ	a4	0.9880
104	Γ	a11	0.9880
213	Γ	a10	0.9880
49	Γ	a11	0.9877
172	Γ	a6	0.9875
112	Γ	a10	0.9871
44	Γ	a11	0.9869
202	Γ	a4	0.9869
193	Γ	a10	0.9868
133	Γ	a10	0.9867
187	Γ	a4	0.9867

TABLE 40 - MODEL 2 EPOCH Ω TO EPOCH Γ DP SOLVE

Table 41 below shows the first set solve going from Epoch Γ to Epoch B. The same structure as in the previous solve is maintained. However, the EV calculations are used only to determine the optimal action from Epoch B to Epoch Γ. The EV's that are carried back to the next recursion are the best EV's from the initial solve. Refer to any textbook on DP. There is a total of 24 sets that were solved in Model 2. Note that for Action 12, the model yields a EV of 0.92. This is of course the best EV, but without the next phase development from Epoch Γ to Epoch Ω. When we look at the Best EV from Epoch Ω, which is shown in the far right column we can actually expect an EV of 0.985.

SysRed	System	Opt Index In				Epoch	Pred	Invest P(Stay)				Invest P(Up)				Invest (ALL) P(Up)				Best EV
Index	Index	A1	A2	B1	B2	In	Node	A1	A2	B1	B2	A1	A2	B1	B2	A1	A2	B1	B2	At Node Γ
1	239439	13	5	13	3	B	1	1.00	1.00	1.00	1.00	0.00	0.00	0.00	0.00	0.90	0.60	0.60	1.00	
1	239439	13	5	13	3	Γ	1													0.9365
2	239441	13	5	13	5	Γ	1													0.9442
3	239447	13	5	13	11	Γ	1													0.9499
6	239466	13	5	14	3	Γ	1													0.9442
7	239468	13	5	14	5	Γ	1													0.9564
8	239474	13	5	14	11	Γ	1													0.9538
21	246000	13	14	13	3	Γ	1													0.9614
22	246002	13	14	13	5	Γ	1													0.9691
23	246008	13	14	13	11	Γ	1	a_{1}	DOM	a_{16}										0.9748
26	246027	13	14	14	3	Γ	1	a_{2}	DOM	a_{16}										0.9691
27	246029	13	14	14	5	Γ	1	a_{3}	DOM	a_{16}										0.9813
28	246035	13	14	14	11	Γ	1	a_{4}	DOM	a_{16}										0.9786
81	259122	14	5	13	3	Γ	1	a_{5}	DOM	a_{16}										0.9614
82	259124	14	5	13	5	Γ	1	a_{6}	DOM	a_{16}										0.9691
83	259130	14	5	13	11	Γ	1	a_{7}	DOM	a_{16}										0.9748
86	259149	14	5	14	3	Γ	1	a_{8}	0.89	a_{16}										0.9691
87	259151	14	5	14	5	Γ	1	a_{9}	DOM	a_{16}										0.9813
88	259157	14	5	14	11	Γ	1	a_{10}	0.89	a_{16}										0.9786
101	265683	14	14	13	3	Γ	1	a_{11}	0.90	a_{16}										0.9691
102	265685	14	14	13	5	Γ	1	a_{12}	0.92	a_{16}										0.9813
103	265691	14	14	13	11	Γ	1	a_{13}	0.88	a_{16}										0.9786
106	265710	14	14	14	3	Γ	1	a_{14}	INF	a_{16}										0.9727
107	265712	14	14	14	5	Γ	1	a_{15}	INF	a_{16}										0.9850
108	265718	14	14	14	11	Γ	1	a_{16}	INF											0.9823
								a12	0.92	Max Action Value										0.9850

TABLE 41 - DP SOLVE FOR EPOCH G TO EPOCH B (SET 1 OF 24)

Once we solve all the 24 sets and find their best EV from the previous solve, we have completed the solve to Epoch B, and we know the best action to take from Epoch B to Epoch Γ. Table 42 below shows the 24 state node solutions at Epoch B.

Index	In	Action	Best EV at Node Γ
1	B	a12	0.9850
2	B	a12	0.9889
3	B	a12	0.9889
6	B	a13	0.9871
7	B	a13	0.9898
8	B	a13	0.9898
21	B	a12	0.9866
22	B	a12	0.9905
23	B	a12	0.9905
26	B	a12	0.9887
27	B	a12	0.9915
28	B	a12	0.9915
81	B	a8	0.9867
82	B	a8	0.9905
83	B	a8	0.9905
86	B	a8	0.9888
87	B	a8	0.9915
88	B	a13	0.9915
101	B	a12	0.9882
102	B	a12	0.9917
103	B	a12	0.9917
106	B	a12	0.9900
107	B	a12	0.9927
108	B	a12	0.9927

TABLE 42 - DP SOLVE FROM EPOCH Γ TO EPOCH B (ALL SETS)

Finally, we repeat the same process to solve to Epoch A. Table 43 below shows the complete solve to the Origin Point at Epoch A. There is of course only one set here.

SysRed	System	Opt Index In				Epoch	Pred	Opt Index Stay				Opt Index Up											Action	P{B2|	Index Actions				Invest P(Stay)				Invest P(Up)				Invest (ALL) P(Up)				Best EV
Index	Index	A1	A2	B1	B2	In	Node	A1	A2	B1	B2	A1	A2	B1	B2	V{X}	V{A1}	V{A2}	V{A}	V{B1}	V{B2}	V{B}	A to B	Action}	A1	A2	B1	B2	A1	A2	B1	B2	A1	A2	B1	B2	A1	A2	B1	B2	At Node Γ
1	239439	13	5	13	3	A	Orig	13	5	13	3	14	14	14	5|11	0.77	0.57	0.53	0.80	0.57	0.35	0.72	a13		1	1	0	0	0.10	0.40	1.00	1.00	0.90	0.60	0.00	0.00	0.90	0.60	0.60	1.00	
1	239439	13	5	13	3	B	1	1	1	1	1	0	0	0	0	0.77	0.57	0.53	0.80	0.57	0.35	0.72	0.04	1.00	0.10	0.40	1.00	1.00													0.9365
2	239441	13	5	13	5	B	1	1	1	1	0	0	0	0	1	0.80	0.57	0.53	0.80	0.57	0.53	0.80	0.00	0.60	0.10	0.40	1.00	0.00													0.9442
3	239447	13	5	13	11	B	1	1	1	1	0	0	0	0	1	0.81	0.57	0.53	0.80	0.57	0.66	0.85	0.00	0.40	0.10	0.40	1.00	0.00													0.9499
6	239466	13	5	14	3	B	1	1	1	0	1	0	0	1	0	0.84	0.57	0.53	0.80	0.88	0.35	0.92	0.00	1.00	0.10	0.40	0.00	1.00													0.9442
7	239468	13	5	14	5	B	1	1	1	0	0	0	0	1	1	0.85	0.57	0.53	0.80	0.88	0.53	0.94	0.00	0.60	0.10	0.40	0.00	0.00													0.9564
8	239474	13	5	14	11	B	1	1	1	0	0	0	0	1	1	0.85	0.57	0.53	0.80	0.88	0.66	0.96	0.00	0.40	0.10	0.40	0.00	0.00													0.9538
21	246000	13	14	13	3	B	1	1	0	1	1	0	1	0	0	0.87	0.57	0.88	0.95	0.57	0.35	0.72	0.06	1.00	0.10	0.60	1.00	1.00													0.9614
22	246002	13	14	13	5	B	1	1	0	1	0	0	1	0	1	0.89	0.57	0.88	0.95	0.57	0.53	0.80	0.00	0.60	0.10	0.60	1.00	0.00													0.9691
23	246008	13	14	13	11	B	1	1	0	1	0	0	1	0	1	0.91	0.57	0.88	0.95	0.57	0.66	0.85	0.00	0.40	0.10	0.60	1.00	0.00	a_1	DOM	a_{16}										0.9748
26	246027	13	14	14	3	B	1	1	0	0	1	0	1	1	0	0.94	0.57	0.88	0.95	0.88	0.35	0.92	0.00	1.00	0.10	0.60	0.00	1.00	a_2	DOM	a_{16}										0.9691
27	246029	13	14	14	5	B	1	1	0	0	0	0	1	1	1	0.94	0.57	0.88	0.95	0.88	0.53	0.94	0.00	0.60	0.10	0.60	0.00	0.00	a_3	DOM	a_{16}										0.9813
28	246035	13	14	14	11	B	1	1	0	0	0	0	1	1	1	0.95	0.57	0.88	0.95	0.88	0.66	0.96	0.00	0.40	0.10	0.60	0.00	0.00	a_4	DOM	a_{16}										0.9786
81	259122	14	5	13	3	B	1	0	1	1	1	1	0	0	0	0.86	0.88	0.53	0.94	0.57	0.35	0.72	0.36	1.00	0.90	0.40	1.00	1.00	a_5	DOM	a_{16}										0.9614
82	259124	14	5	13	5	B	1	0	1	1	0	1	0	0	1	0.89	0.88	0.53	0.94	0.57	0.53	0.80	0.00	0.60	0.90	0.40	1.00	0.00	a_6	DOM	a_{16}										0.9691
83	259130	14	5	13	11	B	1	0	1	1	0	1	0	0	1	0.91	0.88	0.53	0.94	0.57	0.66	0.85	0.00	0.40	0.90	0.40	1.00	0.00	a_7	DOM	a_{16}										0.9748
86	259149	14	5	14	3	B	1	0	1	0	1	1	0	1	0	0.93	0.88	0.53	0.94	0.88	0.35	0.92	0.00	1.00	0.90	0.40	0.00	1.00	a_8	0.89	a_{16}										0.9691
87	259151	14	5	14	5	B	1	0	1	0	0	1	0	1	1	0.94	0.88	0.53	0.94	0.88	0.53	0.94	0.00	0.60	0.90	0.40	0.00	0.00	a_9	DOM	a_{16}										0.9813
88	259157	14	5	14	11	B	1	0	1	0	0	1	0	1	1	0.95	0.88	0.53	0.94	0.88	0.66	0.96	0.00	0.40	0.90	0.40	0.00	0.00	a_{10}	0.89	a_{16}										0.9786
101	265683	14	14	13	3	B	1	0	0	1	1	1	1	0	0	0.89	0.88	0.88	0.98	0.57	0.35	0.72	0.54	1.00	0.90	0.60	1.00	1.00	a_{11}	0.90	a_{16}										0.9691
102	265685	14	14	13	5	B	1	0	0	1	0	1	1	0	1	0.92	0.88	0.88	0.98	0.57	0.53	0.80	0.00	0.60	0.90	0.60	1.00	0.00	a_{12}	0.92	a_{16}										0.9813
103	265691	14	14	13	11	B	1	0	0	1	0	1	1	0	1	0.94	0.88	0.88	0.98	0.57	0.66	0.85	0.00	0.40	0.90	0.60	1.00	0.00	a_{13}	0.88	a_{16}										0.9786
106	265710	14	14	14	3	B	1	0	0	0	1	1	1	1	0	0.96	0.88	0.88	0.98	0.88	0.35	0.92	0.00	1.00	0.90	0.60	0.00	1.00	a_{14}	INF	a_{16}										0.9727
107	265712	14	14	14	5	B	1	0	0	0	0	1	1	1	1	0.97	0.88	0.88	0.98	0.88	0.53	0.94	0.00	0.60	0.90	0.60	0.00	0.00	a_{15}	INF	a_{16}										0.9850
108	265718	14	14	14	11	B	1	0	0	0	0	1	1	1	1	0.97	0.88	0.88	0.98	0.88	0.66	0.96	0.00	0.40	0.90	0.60	0.00	0.00	a_{16}	INF											0.9823
																							1.00		Action Value			0.88	a12	0.92	Max Action Value										0.9850

TABLE 43 - EPOCH B TO EPOCH A SOLVE

Table 44 below is a synopsis of all optimal actions for Model 2.

Arc Number	Epoch In	Node In	Node Out	Epoch Out	Γ_Ω Best Action	Γ_Ω CTD EV	B_Γ Best Action	B_Γ CTD EV	A_B Best Action	A_B CTD EV
Origin				A					a12	0.9927
23	A	1	107	B			a12	0.9927		
24	A	1	108	B			a12	0.9927		
472	B	108	214	Γ	a11	0.9927				

TABLE 44 - MODEL 2 TOTAL DP SOLVE

Once we have the complete solve done, we need one final cleanup step. Action 12 at Epoch A means that Option A2 will not be developed. So, we prune all possible, previously non-dominated, in budget actions that include Option A2 and then go through a complete resolve to verify that the solution remains correct.

Table 45 below shows the cleaned-up re-solve.

Arc Number	Epoch In	Node In	Node Out	Epoch Out	Γ_Ω Best Action	Γ_Ω CTD EV	B_Γ Best Action	B_Γ CTD EV	A_B Best Action	A_B CTD EV
Origin				A					a12	0.9927
23	A	1	107	B			a12	0.9927		
24	A	1	108	B			a12	0.9927		
472	B	108	214	Γ	a11	0.9927				

TABLE 45 – MODEL 2 TOTAL DP SOLVE (PRUNED RESOLVE)

OTHER MODELS UNIQUE DATA

Table 46 below shows the modified weights for Models 3, 4 and 5. Models 3 and 4 varied the parameter weights. Model 5 shares the same parameter weights with Model 2, but varies the subsystem weights. Comparative analysis is found in Chapter 5.

PWt	0.8	PWt	0.1	PWt	0.4
WWt	0.1	WWt	0.2	WWt	0.25
CWt	0.1	CWt	0.7	CWt	0.35
A_Wt	0.65	A_Wt	0.65	A_Wt	0.35
B_Wt	0.35	B_Wt	0.35	B_Wt	0.65

TABLE 46 - MODIFIED WEIGHTS FOR MODELS 3, 4, AND 5

Table 47 below shows the modified origin point.

Options	A1			A2			B1			B2		
Metrics	P	W	C	P	W	C	P	W	C	P	W	C
$State_A$	2	1	1	2	1	2	2	1	1	1	1	3

TABLE 47 - MODEL 6 (MODIFIED ORIGIN)

Table 48 below shows the modified budget for Model 7. Model 7 allows for skipping an investment from Epoch A to Epoch B.

Costs	Phase	A1	A2	B1	B2	Budget - $M
Dev	A to B	14	6	4	2	20
Dev (continue)	B to Γ	8	9	5	4	20
Dev (catch up)	B to Γ	20	21	12	8	
Dev and SI	Γ to Ω	10	10	6	5	16
SI	Ω to TP	5	4	3	3	8
Reserve						9
Total Budget						73

TABLE 48 – MODEL 7 MODIFIED BUDGET

BIBLIOGRAPHY

Azadian, F. (2011). "Dynamic Routing of Time-Sensitive Air Cargo Using Real-Time Information." Transportation Research **48**: 18.

Azizian, N. (2009). A Comprehensive Review and Analysis of Maturity Assessment Approaches for Improved Decision Support to Achieve Efficient Defense Acquisition. World Congress on Engineering and Computer Science, San Francisco, CA.

Bassler, D. (2011). Risk-Driven Design - Integrating Risk Management as an Intrinsic Part of Product

Development. . PhD Technical, Technical University of Munich.

Batini, C. (2009). "Methodologies for Data Quality Assessment and Improvement." ACM Computing Surveys **41**(3): 52.

Bellman, R. (1956). "Dynamic Programming and Lagrange Multipliers." Proceedings of the National Academy of Sciences **42**(10): 3.

Bertsimas, D. (2004). "The Price of Robustness." Operations Research **52**(1): 19.

Bradford, R. (2011). What Happened to FCS? An Organizational Case Study. Master of Strategic Studies Degree Strategy Research Project, U. S. Army War College.

Daskilewicz, M. (2012). "Rave: A Computational Framework to Facilitate Research in Design Decision Support." Journal of Computing and Information Science in Engineering **12**(June 2012): 9.

Doerry, N. (2009). Using the Design Structure Matrix to Plan Complex Design Projects. ASNE Intelligent Ships Symposium ASNE Intelligent Ships Symposium. Philadelphia, PA, ASNE**:** 15.

Dremont, N. (2013). Maturity Integrated in a Meta Model of Knowledge to Help Decision Making in Preliminary Collaborative Design of Mechanical Systems. PhD Dissertation, L'Universite de Technologie de Compiegne

Fiksel, J. (2003). "Designing Resilient, Sustainable Systems." Environmental Science & Technology **37**(23): 10.

Fortunato, S. (2010). "Community Detections in Graphs." Science Direct Physics Reports **486**(Spring 2010): 99.

Goel, T. (2006). "Response Surface Approximation of Pareto Optimal Front in Multi-Objective Optimization." Computer Methods in Applied Mechanics and Engineering **196**(Fall 2006): 15.

Gray, A. (2011). Applied Set-Based Communications and Negotiation System. Human System Integration Symposium, Vienna, VA, American Society of Naval Engineers.

Gray, A. (2011). Enhancement of Set-Based Design Practices Via Introduction of Uncertainty Through the Use of Interval Type-2 Modeling and General Type-2 Fuzzy Logic Agent Based Methods. PhD Dissertation, University of Michigan.

Gries, M. (2004). "Methods for Evaluating and Covering the Design Space During Early Design Development." Integration: the VLSI Journal **38**: 53.

Holland, J. (2013). Engineered Resilient Systems: A DoD Science and Technology Priority Area. U. S. A. C. o. Engineers. Washington, DC, Department of Defense**:** 20.

Liker, J. (1999). "Toyota's Principles of Set-Based Concurrent Engineering." Sloan Management Review **40.2**(Winter 1999): 29.

Madhavan, K. (2008). An Industrial Trial of a Set-Based Approach to Collaborative Design. International Design Engineering Technical Conferences, New York, NY, ASME.

Martin, B. (2012). Arleigh Burke Destroyers: Additional Analysis and Oversight Required to Support the Navy's Future Surface Combatant Plans. G. Audit. Washington, DC, U. S. Government Accounting Office.

McKenney, T. (2012). Determining the Influence of Variables for Functional Design Groups in the Set-Based Design Process. ASNE Day 2012. Arlington, VA, American Society of Naval Engineers. **Proceedings:** 12.

Mebane, W. (2011). "Set-Based Design and the Ship to Shore Connector." Naval Engineers Journal **2011**(3): 14.

Miller (1993). Concurrent Engineering Design. Dearborn, Michigan, Society of Manufacturing.

Oehman, J. (2012). Lean Enablers for Managing Engineering Programs. NDIA Program Management Systems Committee. Arlington, VA, Lean Advancement Initiative, MIT**:** 43.

Ratiu, D. (2004). Using History Information to Improve Design Flaws Detection. European Conference on Software Maintenance and Reengineering, Tampere, Finland.

Singer, D. (2009). "What Is Set-Based Design?" Naval Engineers Journal(4): 13.

Spero, E. (2014). "Tradespace Exploration for the Engineering of Resilient Systems." Procedia Computer Science **28**: 10.

Thangaraj, C. (2010). "Rapid Design Space Exploration using Legacy Design Data and Technology Scaling Trend." Integration: the VLSI Journal **43**: 18.

Unknown (2012). Ship Design Manager and Systems Integration Manager Manual. D. f. I. O. a. Reports. Arlington, VA, Naval Sea Systems Command. **February 2012:** 366.

Wang, R. (1995). "A Framework for Analysis of Data Quality Research." IEEE Transactions on Knowledge and Data Engineering **7**(4): 18.

Zhou, Q. (2011). "Application of Data Maturity in Product Development Process Control." Applied Mexhanics and Materials **58**: 6.

ABSTRACT

PRODUCT DEVELOPMENT RESILIENCE THROUGH SET-BASED DESIGN

by

STEPHEN HORTON RAPP

August 2017

Advisor (s): Dr. Ratna Babu Chinnam and Dr. Gary Witus

Major: Industrial Engineering

Degree: Doctor of Philosophy

Often during a system Product Development program external factors or requirements change, forcing system design change. This uncertainty adversely affects program outcome, adding to development time and cost, production cost, and compromise to system performance. We present a development approach that minimizes the impacts, by considering the possibility of changes in the external factors and the implications of mid-course design changes. The approach considers the set of alternative designs and the burdens of a mid-course change from one design to another in determining the relative value of a specific design. The approach considers and plans parallel development of alternative designs with progressive selection of options, including time-versus-cost tradeoffs and the impact change-costs. The approach includes a framework of the development process that addresses design and integration lead-times, and their relationship to the time-order of design decisions, and the time-dependent burden of design changes.

The framework includes all the mathematical dimensions to define the problem, the time-based epochal structure, modeling parameters and a unique Set-Based Design value structure. The framework uses the Markov Decision Process to create a structured set solution graph and the required action space, state space and transition matrix. The framework's stochastic based value structure, termed Contribution-to-Design, is used to "black box" seed the graph with values for a Dynamic Programming algorithm to conduct a recursive back solve. The Bellman Equation based Dynamic Program determines optimal actions for the design set solutions at each epoch. Multiple models are developed to conduct sensitivity analysis on the parameters of the framework.

AUTOBIOGRAPHICAL STATEMENT

Stephen Rapp was born in Huntingdon, PA, USA. He was appointed to the U. S. Naval Academy graduating with a Bachelor of Science in Engineering and Finance. He became a Marine Officer specializing in armor and later space defense. He supported combat operations in Beirut and Grenada and participated in combat during Operations Desert Storm and Iraqi Freedom. He retired from the Marines as a Major. He attended the Naval Postgraduate School graduating with a Master of Science in Operations Research, specializing in Engineering Optimization. His thesis, a discrete network optimization for rapid mobilization won the MORS Award for "the thesis most likely to improve national defense" and later the INFORMS, Koopman Prize.

In business, Steve pursued a career as a project and program manager in multiple companies, first in classified space programs and then consulting for large data mining and warehousing for IBM. Then at CSC Consulting as a Principal, and Cap Gemini as a Director and Vice President, he worked and managed change programs for Dupont, AIG, AT&T, GM and Ford. Later, at General Dynamics he worked and managed multiple vehicle programs until retiring to pursue his PhD full time at Wayne State University, where he is currently on an ASEE fellowship as a SMART Scholar. He is also on the faculty, teaching Decision and Risk Analysis. He will become a senior scientist, post-graduation, at the U. S. Army Tank-Automotive Research, Development and Engineering Command.

He spends much time in Colorado and California with his wife, Polly and their four grown children: Lydia, Joshua, Caleb and Christina and soon to be grandchildren.