DESIGN FOR AFFORDABILITY IN DEFENSE AND AEROSPACE SYSTEMS USING TRADESPACE-BASED METHODS

by

Marcus Shihong Wu

B.Eng. Electrical and Electronics Engineering
Imperial College London, 2012

Submitted to the Department of Aeronautics and Astronautics and Engineering Systems Division
in Partial Fulfillment of the Requirements for the Degrees of

Master of Science in Aeronautics and Astronautics
Master of Science in Technology and Policy

at the

Massachusetts Institute of Technology

June 2014

© 2014 Massachusetts Institute of Technology. All rights reserved.

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

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

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

Certified by.....
Daniel E. Hastings
Cecil and Ida Green Education Professor of Engineering Systems and Aeronautics and Astronautics
Academic Advisor

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

Accepted by.....
Dava J. Newman
Professor of Aeronautics and Astronautics and Engineering Systems
Director, Technology and Policy Program

DESIGN FOR AFFORDABILITY IN DEFENSE AND AEROSPACE SYSTEMS USING TRADESPACE-BASED METHODS

by

Marcus Shihong Wu

Submitted to the Department of Aeronautics and Astronautics and Engineering Systems Division on 22 May 2014 in Partial Fulfillment of the Requirements for the Degrees of Master of Science in Aeronautics and Astronautics, and Master of Science in Technology and Policy

ABSTRACT

Program failures have plagued the defense and aerospace industry for decades, as unanticipated cost and schedule overruns have rendered the development of systems ineffective in terms of time and cost considerations. This raises the need to holistically include performance, cost and schedule considerations during the early-phase design of systems to perform valuable tradeoffs that derive more feasible and affordable solutions. This paradigm is the design for affordability.

This design for affordability conundrum is targeted at defense and aerospace systems, which have complex mission requirements and stakeholder involvement that are susceptible to changes and perturbations over time. Without a systematic framework, the design for affordability process can potentially become cognitively challenging to system architects and lead to unsatisfactory results. To resolve affordability, it can first be defined as the property of becoming or remaining feasible relative to resource needs and resource constraints over time. Affordability can then be treated as an ility that drives the design of more affordable yet technically sound architectures.

Tradespace-based methods are introduced to drive affordability and incorporate these holistic considerations into the design process. They facilitate the systematic and disciplined search for affordable solutions to the system, program and portfolio of interest. Multi-Attribute Tradespace Exploration (MATE), Epoch-Era Analysis (EEA) and the Multi-Attribute Expense (MAE) function were modified for affordability analysis. Their feasibility was demonstrated through application to two design case studies. Results from both case studies demonstrated the dynamic tradeoffs among performance, cost and schedule parameters. Tradespace-based methods can thus be applied to the progressive design of systems, programs and portfolios using either a bottom-up or top-down approach to deliver affordable solutions in these cases.

Affordability is not only an engineering problem; it is also a policy and management problem. Therefore, affordability can be approached through perspectives beyond engineering design. New policies and refined management practices can be used alongside tradespace-based methods for affordability analysis to ensure the continued delivery of affordable systems for the future.

Thesis Supervisor: Dr. Donna H. Rhodes

Title: Principal Research Scientist and Senior Lecturer, Engineering Systems
Director, Systems Engineering Advancement Research Initiative

ACKNOWLEDGEMENTS

"Consider it pure joy, my brothers, whenever you face trials of many kinds, because you know that the testing of your faith develops perseverance. Perseverance must finish its work so that you may be mature and complete, not lacking anything." - James 1:2-4

This quote from the Holy Bible summed up not only my entire experience in writing this thesis, but also the time I have spent in MIT and the United States. 2 years. It has only been two years. As I lay back in my chair and typed away at this section, I took a quick glance at the digital clock on my screen. It was 4:58am. All was dark and quiet around me, yet all felt surreal. I could feel nothing more but leaps of emancipation that were growing stronger and stronger through the night. Deep down, I knew the source of that irrepressible sense of liberation. It was the sweet taste of victory. It was the sweet taste of success. It was the glowing enlightenment of knowledge. It was the warmth of completeness. It was the hope for better days ahead. It was, everything.

There has never been a dull moment on my journey. Each day brings new challenges, and I am expected to deliver new solutions the next day. “Taking a drink from the fire hose” is still an understatement of the unique learning experience I received at MIT. Since Day 1, I had set my sights on learning as much as I could and getting the best experience possible out of this prestigious technical institution. At first, I wanted to take a more encompassing approach to learning about the world. That led me to the Technology and Policy Program, which I always consider my home base. I wanted to learn about engineering topics beyond the scope of mathematics and programming, and see how the skills I have acquired initially as an Electrical Engineering student could help shape the world in a way I wanted it to be.

Then, I decided to do more and I applied to the Department of Aeronautics and Astronautics to pursue a second, concurrent masters. I have always been interested in space systems engineering but never had the opportunity to pursue further. Simply being at MIT gave me that chance and I was elated that I finally got to pursue what I always dreamt of. At the same time, I also became a Research Assistant at the SEAri, the only laboratory I enquired about and gave me a chance prior to my enrollment in MIT. By the end of 2012, I was on the way to completing two masters and doing my research. All was fine, or so it seemed. But by the turn of 2013, MIT showed its true face, and that hit me really hard. In Spring 2013, I learnt never to take 5 graduate classes at the same time again, and never to take on so many major design projects in one sitting.

However, the horror story was just beginning. Despite my asphyxiating class and research schedule, I felt motivated or even compelled to take on the job of President of the MIT Singapore Students Society. It was a rewarding experience to treat my fellow countrymen here in Boston with servitude and smiles, but it was physically and mentally draining. If this looked bad, I only got myself into worse situations. I again felt motivated and compelled to take on the job of Chairperson of the Ashdown House Social Committee. The job scope was no different than that for the Singapore Society, but just for an entirely different crowd. 2 masters programs, 2 leadership roles, and 1 major research commitment. Woe (Wu) is me indeed.

But then, the sun began to shine brightly on me. In the midst of all this work and responsibilities, I met a girl. A really great girl called Ruth Choi, from a setting and a background that I never imagined to be associated with. She began to show me that there were indeed greater things out there, not in engineering, but in life. And I knew that she was the girl of my dreams. To cap off all the interesting things I had experienced or subjected myself to, I declared her to be the love of my life. We took our blossoming romance through the summer and winter of 2013 and shared many happy memories along the East Coast. By Spring 2014, I proposed to her and voila, we were engaged and looking forward to getting happily married that July.

It has been an incredible journey with so many life-changing twists and turns that brought me through a range of emotions for days on end. By Fall 2013, schoolwork got less intense, but research work began escalating. My job as President and Social Chair also kicked into full force. Graduation was in sights, but many major hurdles still lay ahead. It was evident that I never learnt from the lessons of the past, and I chose to take a very intensive math class instead of the other easier options that were made available to me. I needed the class to graduate, I needed to run my simulation models, I needed to write my thesis, I needed to finish my jobs of President and Social Chair, and I needed to plan for the wedding. And above all that, I wanted to be with Ruth.

Fast forward to present day, it may be peculiar as to why I have gone such a long way in describing my journey and not yet thank a single person in this section titled "Acknowledgements". But I do this to say that this journey is not simply about the things that happened to me. It is about the people who I have met along the way and it is these people that define what my journey really is about. These people have been significant to me in my two years and I am who I am today because of them. Without them, I would not have had the strength to become more mature and complete in the face of trials and tribulations. Without them, I would not have learnt to persevere and grit my teeth through hard times. Without them, I would never have learnt to keep faith in the things that meant most to me. Without them, I would not have found joy in whatever I was doing.

The first person I will like to truly thank will be my research advisor **Dr. Adam Ross**. Adam is an exceptional scientist and his meticulous and intelligent work ethos is exemplar to all. Rarely does an error or even a little blip slip by without his attention, and that drives me to continuously strive towards clarity and near perfection in my work. With his guidance and teachings here in SEAri, I have mastered many important systems engineering concepts and reshaped the way I see the world. Now, I see affordability in everything and I even operationalized the principles of dynamic performance, cost and schedule tradeoffs in wedding planning. An affordable location that provided a wide variety of food, drink and entertainment was found and affordable solutions to the wedding dress, gifts, invitations and photography were derived. Multi-stakeholder negotiation with Ruth, my parents and future in-laws was also conducted. Almost every aspect of the wedding is on track and within budget after I had time and cost as independent variables in my wedding planning design problem. Now, I feel ready for the real world. Of course, systems engineering is not as trivial as that and I have encountered a lot more complex design problems. But from my experience here in SEAri and the work ethics I hope that I have inherited from Adam, I want to be able to take the world by its horns and get things done the right way.

The next person I will like to thank will be my other advisor **Dr. Donna Rhodes**. Donna is the face of SEAri in many ways and she was the one who corresponded with me in my first email exchanges with SEAri. Together with Adam, she brought into me the world of new-age systems engineering and I was never happier to have been able to acquire the very skills that this laboratory was built upon. She has provided me with valuable advice throughout my time here and it has been a pleasure and a fruitful learning experience to be working with her and Adam.

I will then like to thank my academic advisor **Professor Dan Hastings**, who always brings me positivity in his words whenever he shares his experiences in space systems design and engineering. I will not forget how he stepped into my 16.89 Space Systems Engineering class just seconds before I was to present my tradespace analysis segment. He enjoyed the work I produced and that meant a lot to me at that time.

I cannot sign off my time here in SEAri without sincerely thanking my wonderful lab mates who are also my fellow graduate students, beginning with those who have graduated. In order of seniority, I will first like to thank **Nirav Shah**. My encounter with Nirav was a short one as he was gone by the winter of 2012. However, his knowledge of systems engineering was to me well above any other student in SEAri and he could answer almost every question with aplomb. Many other students will also share the same sentiments as me. I recently met him at CSER 2014, and he was still the same: full of wit, full of passion, and full of knowledge. Without him, many of my questions or misconceptions would have remained unaddressed.

I will also like to thank **Paul Grogan**, who very recently passed his dissertation defense. Paul has provided me and everyone else in the lab with new perspectives on gaming simulation and multi-stakeholder interactions. I am also part of the SIRG research group with him and my encounters with him often left me in awe of his work throughput rate. He has programmed many simulations from scratch and conducted many experiments, which I truly enjoyed and learnt a lot.

For those who are still currently in SEAri, it has been fun sharing every "SEArious" moment with you. I will like to thank **Matthew Fitzgerald**, for being the big brother of the student group who takes care of almost everything in the lab, ranging from technical troubleshooting of ViSLab and the computers, as well as watering the plants. I value Matt for his valuable opinions on everything and it is evident that he has thought through the questions that I am just starting to discover. He was the first to give me a student's perspective of SEAri and I enjoyed his immense contributions to the group.

Next, I will like to thank my fellow batchmates, **Nicola Ricci** and **Michael Schaffner**. I managed to psyche Nico into applying for TPP and I am glad it is a decision we both do not live to regret. Sharing the office space with Nico was a valuable experience, as we often discussed each other's work. I got to learn a lot more about options and he got to learn a lot more about affordability. We also got to discuss a lot of TPP homework together in Spring 2013, and never would I have been able to get through that semester without knowing someone who is in the same boat as I am. I also enjoy his signature lines of "Ordinal versus Nominal", "Static versus Dynamic", and "So it is ~".

Michael is my partner in crime when we dissected the topic of affordability over these 2 years. I will not forget how we did our first joint presentation, throwing high-5's as we switched from section to section. We also co-authored a paper together, which essentially threw out the research

questions that directed our subsequent research thrusts and our theses. Michael is also a good coffee buddy and everything seems more cheerful when he is around. Together with Nico, there has never been a dull moment in the lab.

My new lab mates whom I got to know a lot more this year have contributed greatly to my learning as well. **Paul La Tour**, for all his experiences and advice in designing satellite systems. Without his help, I would not have had the inspiration to perform some of the modeling techniques shown in this thesis. He also briefly helped validate my FSS model and results and that left me feeling more reassured of my assumptions. **Michael Curry**, for his technical prowess that I often see evidence of running on the computer screens in SEAri and for his advice of modeling satellite systems. **Ben Putbrese** and **Hunter Zhao**, with whom I shared an office briefly, but with memorable moments when we shared our thoughts on life.

Yeong Li Qian, for being the other awesome Singaporean in SEAri and also the best baker that I ever knew for a friend. Thanks for taking on the role of President of the Singapore Society. It is a challenging task but I would not have asked him to do it if I did not have the confidence that he could make it his.

Other Singaporeans have also been significant here during my time here in Boston and MIT. **Kenneth Loh**, **Tan Siah Hong**, **Ng Sheng Rong**, **Karthick Murugappan**, **Xue Kun**, **An Jingzhi**, **Kang Zi Han**, **Lee Yin Jin**, **Liu Yun**, **Jonathan Teo**, **Ng Huey Jeen**, **Lim Shi Min** and **Joy Chua** are all good friends who have made my transition from Singapore/UK to USA comfortable and enjoyable. Thanks for keeping me sane and happy even without knowing it.

This brings me to my experience in Ashdown House. **Grace Gu**, for being such an awesome friend without whom I would have run out of ideas and initiatives for more social events. Also, simply for just being a great friend and a supportive buddy. **Thomas Mahony** and **Jenny Schloss**, my ex-roommate and his now fiancé, who were great company during my first year in Ashdown. **Christopher Foy**, for all your wittiness and funny moments that never failed to keep me entertained.

And how could I forget my **TPP friends**, many of whom I have shared a laugh with and there were truly memorable experiences that we were all in together. **Brandon Karlow**, for being the friendliest person I know around here and my best man for the wedding. I shared many classes with him and I would not have gone through all of them without his words of advice and encouragement. We are practically good buddies on almost every scale.

Hisham Bedri and **Ekene**, for being such cool dudes that I will enjoy hanging out and talking about anything under the sun. **Jordan Foley**, for helping me out on a number of topics, most notably the naval ship design and for bringing pride to the US Navy.

Erin Leidy, for being such a great companion during classes and never failing to cheer anyone up with her bright red hair and vivacious nature. **Nina Schuchman**, for being such a bubbly friend as well, and together with Brandon and Erin, we form the gang of four that has gone out on countless dinners and drinks together. And also, for being there at some of my lowest moments.

I did not have the luxury of time to know everyone in AeroAstro and get off on a good note with them. But one person I have worked with in some of my classes really stands out. **Giuseppe Cataldo**, whom I worked with in 16.851, 16.89 and 16.895, for being the most reliable and

knowledgeable astronautical engineer and space scientist I know, and for always being an inspiration to me, and for hosting me in DC and at NASA Goddard Space Flight Center.

Ioana Josan-Drinceanu, **Kathleen Voelbel**, and **Dustin Hayhurst**: the Communications Subsystems team in 16.89 that I really enjoyed working in. I never forgot the laughter, the fun, and the pains we went through to get our modular codes working (or so it seemed).

Finally, my **Mum and Dad**, who always love me dearly no matter what happens. It has been tough on them for the last 7 years, during which I have spent most of it away from home due to my time in military service, at Imperial College London in the United Kingdom, and at MIT in the United States. I can still feel the love from them (and my cat) through our regular Skype calls, and we never fail to set up an entertaining chat that we can relish for days. So thank you very much Mum and Dad, without whom I definitely would not be where I am or who I am today. It is only in recent years that I realized how much you have cared for me when I was younger, and I now look back in shame at the moments when I behaved like a spoilt brat or really disappointed them with my bad behavior. Thank you very much for everything.

Now, that I am getting married, I have also become part of Ruth's family. Here in the United States, I have another **Mum and Dad from the Choi family**. Throughout my visits to Philadelphia in the previous summer, Thanksgiving, Christmas, New Year's and Valentine's Day weekend, I felt the family love that I have always been yearning for as long as I have been away from Singapore. Never have I felt so loved by another family before. It definitely reassures me that a lot of love and hospitality do exist in the world out there. Thank you very much for always praying for me, and getting me safely through all my obstacles.

To my beloved **Ruth Choi**, no amount of words can truly express how thankful I am to have you during this period and in my life. You have stood by me all this while and showered my stresses, sorrows and perpetual lack of sleep with your love and grace. Never could I have soldiered through the days without your words and hugs of comfort. You have definitely made me a better person, and it was through you that I rediscovered my faith.

Finally, to God Almighty, my Savior Jesus Christ, and the Holy Spirit, thank you for always watching over me. Do let me through this final hurdle and lead me on your path to Glory. Amen.

Thank you very much.

I dedicate this thesis to all of you.

You all mean very much to me.

BIOGRAPHICAL NOTE

Marcus Shihong Wu was born to a humble family in Singapore, and lived in a cozy government flat tucked away in the only quiet corner of an otherwise bustling city district. He was an only child and did not have many people to look up to except for his parents. However, that was more than sufficient, as his parents sowed the seeds of perseverance, determination and love in him. As a civil servant and a nurse by profession, his parents were doing enough to earn their keep and provided the little family with enough food, sustenance and entertainment.

His mother took a first hand in directing her son's education and before he enrolled in first grade, Marcus was able to read a broad array of children's books in both English and Mandarin. He was also blessed with good memory and had the knack of repeating word for word what other people were saying and the dialogue lines he heard on television. He began to exhibit streaks of perfectionism, as he would spend hours writing the same Chinese characters until their form was consistent over an entire page. He also liked the arrangement of objects in certain ways, often symmetric or monotonic, and would make all effort to get them the way he wants.

Apart from these traits, he was a shy boy who did not enjoy going to the playground and mixing with other kids. He was not the typical boy, as he did not enjoy games very much and often liked to be alone reading his comic books. However, he began to find joy in sports, with soccer and basketball being his favorites. Although he never exhibited much dexterity in any of them, he always enjoyed watching or simply being associated with them.

Despite a promising start, Marcus did not do too well in his first two years of school. He was not fantastic in his classes, especially English and Mathematics. However, something unexplained dawned upon him in the third year and from out of nowhere, Marcus topped his class that was ranked close to the bottom. To his surprise, he found out that he even topped his school. He never looked back since and he would go on year after year to achieve stellar academic results. He eventually graduated top of his elementary school.

He later enrolled in a prestigious high school called Raffles Institution, which has a rich tradition of grooming many government officials, military leaders, and scientists in the country. It was there that he really began to excel. Apart from excelling in his studies, he also became a leader in his co-curricular activities. It was during his adolescent years that he began to come out of his shell and started making himself heard. He rose to prominence with a close to perfect score in the GCE O Level Examinations and everything seemed on track for him to succeed.

However, Marcus got too involved in his sideline activities and began to neglect his schoolwork at a critical period of time. Towards the end of high school, his grades dipped significantly and it was only a miracle that he managed to finish within the top 10% of the school for the GCE A Level Examinations. There were over a hundred people ahead of him and they appeared more well-rounded candidates than him in both academics and leadership. Receiving scholarships were typically regarded as the pinnacle of a local education experience, and the most well rounded students would receive the most prestigious scholarships.

Marcus was struggling at this point. He was not on the radar of many prominent scholarship agencies and his chances worsened with his not-so-ideal experiences in the military. Nonetheless, he put on a brave front and applied for a number of scholarships. One night after a day of intense military training, he was surprised to receive a text stating that the Defense Science and Technology Agency (DSTA) of Singapore had offered him an overseas scholarship. He was elated and he promptly seized the chance to become a DSTA scholar and pursue life overseas for a few years. However, there was a little problem. He had to study Electrical and Electronics Engineering (EEE), instead of his preferred major – Aeronautical Engineering. Nonetheless, he went ahead with applying to several UK and US colleges and received a number of positive responses. In the end, he decided to pursue EEE at Imperial College London. He did what any other naïve and excited 19-year-old boy would do – he chose London, not engineering.

As someone who chose to major in a subject not for academic interest but purely to seek entertainment outside Singapore, Marcus struggled in his first two years at college. He was still doing well and came in the top 5% for those two years, but there were major incidents in his budding academic journey that shook his confidence. He found great difficulty in programming and had a harder time understanding algorithms as compared to everyone else. Also, he spent a year on a project, which was eventually declared a failure and left with a poor grade being the embodiment of his disappointment. However, the seeds of perseverance and determination began to grow, and Marcus did his utmost best to overcome his weaknesses and failures.

In his senior year at Imperial College London, Marcus finally understood why he chose EEE. He began to enjoy all his classes and loved how all the topics were interconnected. Although he excelled again in his final set of undergraduate classes, his big break came through his yearlong capstone project. He had the fortune of being assigned to one of the best supervisors, who took him deep into a journey of memristive devices and quantum mechanics. Marcus, when left on his own, surprisingly began to flourish. He began applying his new knowledge to the development of analogue electronics and bio-inspired devices. He started developing several software tools to analyze the non-linear behaviors of these devices and unknowingly became very adept at programming. He spent day and night working on this project, and he was rewarded with the top prize for best undergraduate project. For the obstacles he had encountered before, this achievement was his crowning moment and Marcus once again found himself at his best.

In between, he applied to several US colleges for graduate school. Without harboring a single strain of hope, he was pleasantly surprised to hear that he was accepted into MIT. He was to enroll in the Technology and Policy Program, and was about to step away from the world of nanodevices to confronting broader engineering challenges. He also managed to enroll in the Department of Aeronautics and Astronautics to pursue a second, concurrent Masters. Currently, he hopes to be able to complete all class and thesis requirements in order to graduate. After which, he is planning to get married in Philadelphia to the love of his life - Ruth Choi.

Marcus will be commencing his work with DSTA back in Singapore by August and he hopes to be able to apply all he has learnt at MIT to his career and his life. Despite his overly packed schedule, Marcus still finds time to follow his favorite team Manchester United, watch his favorite TV shows, listen to music, and spend time with people who matter to him most!

TABLE OF CONTENTS

Abstract **3**

Acknowledgements **5**

Biographical Note **11**

List of Figures **17**

List of Tables **23**

1 Introduction **25**

1.1 Motivation: The Ariane VI Conundrum 25

1.2 Prevalence of Cost and Schedule Overruns 27

1.3 Considering Performance, Cost and Schedule for Affordability 30

1.4 Finding the Best Approach to Affordability 31

1.5 Affordability through Better Systems Engineering 32

1.6 Research Questions 35

1.7 Research Methodology and Thesis Outline 36

2 Literature Review of Affordability **37**

2.1 Motivation 37

2.2 The Meaning of Affordability 37

2.2.1 Lexicographic Analysis 37

2.2.2 Common Usage and Application 39

2.2.3 Significance in the Defense and Aerospace Industry 41

2.3 Affordability in the Context of Engineering Design 45

2.3.1 Affordability and the Goal of Design 45

2.3.2 The Operating Climate for Affordability 47

2.4 Current Ways of Understanding Affordability 54

2.4.1 Working Definitions 54

2.4.2 The Affordability Triangle 56

2.4.3 Should-Cost Review and Statistical Methods 60

2.4.4 Sand Chart Tool 63

2.4.5 Cost and Time as Independent Variables 65

2.4.6 Effective Portfolio Management 67

2.5 Limitations of Existing Affordability Methods 69

2.5.1 Lack of Time Centricity 69

2.5.2 Failure to Recognize Complexities of Scale 70

2.5.3 Lack of Cost Breakdown Structures 70

2.5.4 Treating Affordability as a Constraint 71
2.5.5 Lack of Value-Centric Perspective 71
2.6 Summary 71
3 Tradespace-based Methods For Affordability Analysis 73
3.1 Motivation 73
3.2 The Tradespace Exploration Paradigm 74
3.3 Capturing Value through Tradespace Exploration 76
3.3.1 Value Creation 77
3.3.2 Multi-Attribute Utility Theory for Value-Centric Design 77
3.4 Multi-Attribute Expense to replace 'Cost' 79
3.5 Multi-Attribute Tradespace Exploration for Affordability Studies 80
3.6 Affordability as an Ility 83
3.7 Affordability in Systems of Systems 84
3.8 From System to Program to Portfolio and Back 85
3.9 Constraint Levels for Utility and Expense 90
3.10 Epoch-Era Analysis for Affordability 92
3.11 Demonstration of Tradespace-based Methods in the Affordability Analysis for a Space Tug system and program 95
3.11.1 Demonstration of System Level Analysis 95
3.11.2 Demonstration of Program Level Analysis 98
3.11.3 Single-Epoch Affordability Analysis for a Space Tug Program 100
3.11.4 Multi-Epoch and Single-Era Analysis for a Space Tug program 101
3.12 Summary 104
4 Federated Satellite Systems System Analysis 107
4.1 The Federated Satellite Systems Paradigm 108
4.2 FSS Architectural Assumptions 109
4.3 System Definition 116
4.4 System Level Design Variables and Epoch Variables 120
4.5 Hypothetical FSS Development Missions 124
4.6 Epoch Construction and Sequencing for System, Program and Portfolio Level Analysis 126
4.7 System Level Design-to-Value Mapping 129
4.8 Generation of System Tradespaces and Design Identification 132
4.9 Lognormal Distributions and Confidence Intervals 136
4.10 Discussion of System Level Results 147

4.11 Differences in using MAE and Cost 148
4.12 Summary of System Level Analysis 149
5 Federated Satellite Systems Program Analysis 151
5.1 Program Definition 151
5.2 Program Level Design Variables 154
5.3 Program Level Design-to-Value Mapping 157
5.4 Generation of Program Tradespaces and Design Identification 159
5.5 Expense and Utility Trajectories for Constellation Designs 169
5.6 Discussion of Program Level Analysis Results 176
6 Federated Satellite Systems Portfolio Analysis 181
6.1 Portfolio Definition 181
6.2 Determining Weights and Preferences for Portfolio Analysis 184
6.3 Portfolio Cost, Performance, and Satellite Profiles 189
6.4 Portfolio Analysis Results 204
6.5 FSS Development Strategy 206
7 Integrating Tradespace-Based Methods with Industry Practices & Policies for Affordability 211
7.1 Motivation 211
7.2 Benefits and Concerns of Using Tradespace-Based Methods for Affordability Analysis 213
7.2.1 Benefit 1: Systematic, Disciplined and Convenient Approach to Design 213
7.2.2 Benefit 2: Layered and Scalable Methodology for Error and Complexity Reduction 215
7.2.3 Benefit 3: Overcoming the Limitations of Existing Methods 217
7.2.4 Drawback 1: Inherent Subjectivity of Stakeholder Preferences and Multi-Attribute Utility 220
7.2.5 Drawback 2: Variations in Model Fidelity and Tractability 225
7.2.6 Drawback 3: Merits and Drawbacks of Qualitative and Quantitative Measures 226
7.2.7 Drawback 4: Problems with Model Validation 227
7.2.8 Drawback 5: Availability of Expertise 228
7.2.9 Drawback 6: Availability of Decision Makers 229
7.2.10 Drawback 7: Computational and Schedule Constraints 229
7.2.11 Summary of Implementation Issues 230
7.3 Applying Tradespace-based Methods in Defense and Aerospace Acquisition 231
7.3.1 US Department of Defense Acquisition Framework 231
7.3.2 NASA Project Life Cycle and Trade Study Framework 235

7.3.3 Affordability Engineering Framework with Tradespace-based Methods 237
7.4 Affordability through Effective Policies and Management 243
7.4.1 Implementing New Systems Engineering and Acquisition Policies 243
7.4.2 Managing Multiple Stakeholders 244
7.4.3 Better Communication and Work Flow Organization 244
7.4.4 Integrating Practices from Industry and Academia 244
7.4.5 Continuously Educate and Train the Acquisition Workforce 245
7.5 Summary 245
8 Conclusion & Future Work 247
8.1 General Conclusion 247
8.2 Research Contributions 247
8.3 Future Work 256
8.3.1 Enhancement to the FSS case study 256
8.3.3 Applying the top-down approach 257
8.3.4 Enhance the formulation and aggregation of expenses 257
8.3.5 Trading Affordability with other Ilities 258
8.3.6 Multi-Era Analysis 258
8.3.7 The Concept of Product-Service Systems 258
8.3.7 Affordability, Profitability and Sustainability 259
8.3.8 Extending the Scope of Affordability 259
8.4 Epilogue 261
References 263
Appendix 1: System Tradespace Model 271
Appendix 2: Program Tradespace Model 287
Appendix 3: Portfolio Level Analysis 305

LIST OF FIGURES

Figure 1-1: Comparison of the Ariane V (left) and Ariane VI (right) (ESA, 2013) 25

Figure 1-2: Percentage of Programs Meeting Total Acquisition Cost Growth Targets (GAO, 2013) 27

Figure 1-3:The Army's Comanche armed reconnaissance helicopter (left), the Navy's DDG-1000 next-generation surface combatant (middle), and the Air Force's TSAT (right). 28

Figure 1-4: James Webb Space Telescope (GAO, 2013 - JWST) 29

Figure 1-5: Percent cost growth and percent schedule growth plot for various Earth and Space Science missions conducted by NASA. The cost overruns for some missions are breakdown into the 3 phases during which they were committed. (NRC, 2010) 30

Figure 1-6: Problems in trading off performance, cost and schedule (Adapted from Alleman, 2010) 31

Figure 1-7: Commitment, system-specific knowledge, and cost over system lifecycle (Blanchard and Fabrycky, 2006) 33

Figure 2-1: Classification of affordability articles based on sector (Bankole, 2011) 40

Figure 2-2: The goal of design (Ross, 2006). 46

Figure 2-3: F-35 Joint Strike Fighter (left) and Ballistic Missile Defense System (right). 47

Figure 2-4: Number of significant and critical Nunn-McCurdy breaches from 1997 -2012 (DoD, 2013) 48

Figure 2-5: Historical changes in Total Defense Budget Authority from 1948-2012 and forecasted reduction from 2012-2020 (Plummer – Washington Post, 2013). 50

Figure 2-6: Challenges in cost and schedule estimation and ways to mitigate them (GAO, 2009). 52

Figure 2-7: DoD-recommended affordability assessment method (GAO, 2009) 53

Figure 2-8: Recommended Cost Estimating Process (GAO, 2009) 53

Figure 2-9: The Affordability Triangle (Tuttle and Bobinis, 2012). 56

Figure 2-10: Determining affordable solution by trading off cost and performance after fixing capabilities and schedule as decision criteria (Tuttle and Bobinis, 2012). 58

Figure 2-11: Lifecycle cost composition of a typical defense acquisition program (Tuttle and Bobinis, 2012). 59

Figure 2-12: Breakdown of TOC into subsystem costs and identifying areas for improvement (Booz Allen Hamilton, 2011). 60

Figure 2-13: The difference between should-cost and traditional cost estimating methods (AT Kearney, 2011). 61

Figure 2-14: Interval cost estimation process (Blue indicates main path, Orange indicates optional elements, Green indicates data sources) and its generated results (Kroshl and Pandolfini, 2000). 62

Figure 2-15: Affordability methodology used by the Aerospace Corporation (Emmons, 2010) ... 63

Figure 2-16: Sand chart as affordability analysis results from a notional NASA program (Emmons, 2010). 64

Figure 2-17: "Penalties" are applied to fit costs to budget in the Aerospace Corporation methodology and Sand Chart Tool (Emmons, 2010). 65

Figure 2-18: Time As an Independent Variable (TAIV) for Fielding Capability (Patterson, 2013). 66

Figure 2-19: TAIV reveals trade space dimensions (Patterson, 2013). 67

Figure 2-20: Wave Model Relation to Portfolio Evaluation (Davendralingam and DeLaurentis, 2013) 68

Figure 2-21: Archetypal Node (System) Behaviors (Davendralingam and DeLaurentis, 2013) ... 69

Figure 3-1: Four classes of trades: (1) Local point solutions (2) Frontier subset solutions (3) Frontier solution sets (4) Full tradespace exploration (Based on Ross and Hastings, 2005). 75

Figure 3-2: Sample Tradespace (Ross and Hastings, 2005). 76

Figure 3-3: Multi-Attribute Tradespace Exploration (Ross and Hastings, 2005). 81

Figure 3-4: Data flow for the Tradespace Exploration process in Affordability Analysis (adapted from Ross and Hastings, 2005). 81

Figure 3-5: Flow chart for conducting the tradespace exploration process from system to portfolio. 89

Figure 3-6: Narrowing the affordable solution space using external constraint levels for a fixed context. 92

Figure 3-7: Epoch-Era Analysis (Ross and Rhodes, 2012). 93

Figure 3-8: Modified EEA for affordability analysis using a resource-centric approach. 94

Figure 3-9: A Space Tug system. 95

Figure 3-10: Tradespaces for the Space Tug system with added expense considerations change across different epochs as a result of changing preferences, mission needs and context. 97

Figure 3-11: Tradespace for a Space Tug program in Epoch 1. 6 designs along the Pareto front were chosen and labeled A to F (Wu, Ross and Rhodes, 2014). 100

Figure 3-12: (a) Number of epochs in affordable solution region for every design; (b) Number of epochs above minimum utility level for every design (Wu, Ross and Rhodes, 2014). 102

Figure 3-13: EEA with expense considerations in a single era (Wu, Ross and Rhodes, 2014). .. 103

Figure 3-14: EEA with utility considerations in a single era (Wu, Ross and Rhodes, 2014). 103

Figure 4-1: FSS Earth Observation Support Infrastructure Example in STK© Visual Representation (Golkar, 2013) 109

Figure 4-2: System Tradespace for Phase 1 – Epoch 1 – Mission 0 in Context 0-0 133

Figure 4-3: System Tradespace for Phase 2 – Epoch 2 – Mission 1 in Context 0-0 133

Figure 4-4: System Tradespace for Phase 3 – Epoch 3 – Mission 2 in Context 0-0 134

Figure 4-5: System Tradespace for Phase 4 – Epoch 10 – Mission 3 in Context 0-1 134

Figure 4-6: System Tradespace for Phase 5 – Epoch 17 – Mission 4 in Context 1-0 135

Figure 4-7: System Tradespace for Phase 6 – Epoch 24 – Mission 5 in Context 1-1 135

Figure 4-8: System Tradespace for Phase 7 – Epoch 19 – Mission 0 in Context 1-1 136

Figure 4-9: Risk-adjusted probability distribution for cost and time elements (NATO OTAN, 2007) 137

Figure 4-10: Lognormal distributions with different variances (Wikipedia). 138

Figure 4-11: Epoch 1 Pareto front design points with 99% MAE confidence intervals 140

Figure 4-12: Epoch 2 Pareto front design points with 99% MAE confidence intervals 141

Figure 4-13: Epoch 3 Pareto front design points with 99% MAE confidence intervals 142

Figure 4-14: Epoch 10 Pareto front design points with 99% MAE confidence intervals 143

Figure 4-15: Epoch 17 Pareto front design points with 99% MAE confidence intervals 144

Figure 4-16: Epoch 24 Pareto front design points with 99% MAE confidence intervals 145

Figure 4-17: Epoch 19 Pareto front design points with 99% MAE confidence intervals 146

Figure 4-18: Comparing tradespaces for Epoch when MAU is plotted against MAE (left) and when MAU is plotted against total cost (right). 148

Figure 5-1: System Tradespace for Phase 1 – Epoch 1 – Mission 0 in Context 0-0 160

Figure 5-2: System Tradespace for Phase 2 – Epoch 2 – Mission 1 in Context 0-0 161

Figure 5-3: System Tradespace for Phase 3 – Epoch 3 – Mission 2 in Context 0-0 162

Figure 5-4: System Tradespace for Phase 4 – Epoch 10 – Mission 3 in Context 0-1 163

Figure 5-5: System Tradespace for Phase 5 – Epoch 17 – Mission 4 in Context 1-0 164

Figure 5-6: System Tradespace for Phase 6 – Epoch 24 – Mission 5 in Context 1-1 165

Figure 5-7: System Tradespace for Phase 7 – Epoch 19 – Mission 0 in Context 1-1 166

Figure 5-8: Demonstrating transformation to *a posteriori* constraint levels to determine *a posteriori* affordable solution region 167

Figure 5-9: Expense trajectory plot, Utility trajectory plot, and Histogram of number of epochs in affordable solution region for each design for Constellation 1 – Epoch 1: Mission 0 in Context 0-0 170

Figure 5-10: Expense trajectory plot, Utility trajectory plot, and Histogram of number of epochs in affordable solution region for each design for Constellation 2 – Epoch 2: Mission 1 in Context 0-0 171

Figure 5-11: Expense trajectory plot, Utility trajectory plot, and Histogram of number of epochs in affordable solution region for each design for Constellation 3 – Epoch 3: Mission 2 in Context 0-0 172

Figure 5-12: Expense trajectory plot, Utility trajectory plot, and Histogram of number of epochs in affordable solution region for each design for Constellation 4 – Epoch 10: Mission 3 in Context 0-1 173

Figure 5-13: Expense trajectory plot, Utility trajectory plot, and Histogram of number of epochs in affordable solution region for each design for Constellation 5 – Epoch 17: Mission 4 in Context 1-0 174

Figure 5-14: Expense trajectory plot, Utility trajectory plot, and Histogram of number of epochs in affordable solution region for each design for Constellation 6 – Epoch 24: Mission 5 in Context 1-1 175

Figure 5-15: Expense trajectory plot, Utility trajectory plot, and Histogram of number of epochs in affordable solution region for each design for Constellation 7 – Epoch 19: Mission 0 in Context 1-1 176

Figure 6-1: Weight assignments to portfolio performance and expense attributes 186

Figure 6-2: SAU curves for portfolio performance attributes in Single Era Analysis 187

Figure 6-3: SAE curves for portfolio expense attributes in Single Era Analysis 188

Figure 6-4: Portfolio tradespace with selected designs for further analysis (Design 43 [1,2,2,2,1,1,1] – Green, Design 19 [1,1,2,2,1,1,1] – Cyan, Design 1 – Black [1,1,1,1,1,1,1]) 189

Figure 6-5: Cost and performance profiles of satellite constellations 1-2 of Portfolio 43 190

Figure 6-6: Cost and performance profiles of satellite constellations 3-7 of Portfolio 43 191

Figure 6-7: Satellite and performance profiles of satellite constellations 1-4 of Portfolio 43 193

Figure 6-8: Satellite and performance profiles of satellite constellations 5-7 of Portfolio 43 194

Figure 6-9: Cost and performance profile of FSS Portfolio 43 195

Figure 6-10: Satellite and performance profile of FSS Portfolio 43 195

Figure 6-11: Difference in profiles for Constellation 3 when not operated and operated for legacy 197

Figure 6-12: Cost and performance profile of Portfolio 43 when Constellation 3 is operated for legacy 198

Figure 6-13: Satellite and performance profile of Portfolio 43 when Constellation 3 is operated for legacy 199

Figure 6-14: Cost and performance profile of Portfolio 1 201

Figure 6-15: Satellite and performance profile of Portfolio 1 201

Figure 6-16: Cost and performance profile of Portfolio 19 203

Figure 6-17: Satellite and performance profile of Portfolio 19 203

Figure 7-1: Milestones A, B, C and the 5 phases of a typical program acquisition timeline (DAG, 2013). 232

Figure 7-2: NASA Project Life Cycle (NASA, 2012) 236

Figure 7-3: NASA Trade Study Process (NASA, 2012) 237

Figure 7-4: Affordability Engineering Framework (MITRE, 2013) 238

Figure 7-5: AEF Step 2 – Affordability Evaluation (MITRE, 2013) 239

Figure 7-6: AEF Step 3 – Tradeoff Analyses (MITRE, 2013) 241

Figure 7-7: AEF Program Trigger Points (MITRE, 2013) 242

Figure 8-1: Data flow for the Tradespace Exploration process in Affordability Analysis (adapted from Ross and Hastings, 2005). 249

Figure 8-2: Narrowing the affordable solution space using external constraint levels for a fixed context. 250

Figure 8-3: FSS Case Study: Epoch 1 Pareto front design points with 99% MAE confidence intervals 250

Figure 8-4: FSS case study: Expense trajectory plot, Utility trajectory plot, and Histogram of number of epochs in affordable solution region for each design for Constellation 1 – Epoch 1: Mission 0 in Context 0-0 251

Figure 8-5: Cost and performance profile of Portfolio 19 252

Figure 8-6: Satellite and performance profile of Portfolio 19 253

Figure 8-1: Flow chart for conducting the tradespace exploration process from system to portfolio 254

Figure 8-8: Links between supplier sustainability, manufacturer profitability and customer affordability (Bankole, 2011) 259

Figure 8-9: Factors influencing Affordability (Bankole, 2011) 260

LIST OF TABLES

Table 1-1: Areas of weakness causing cost growth in programs. Adapted from (IDA, 2009). 32
Table 2-1: Collection of definitions of Affordability 56
Table 3-1: Design Variables for a Space Tug System 96
Table 3-2: Performance Attributes for a Space Tug System 96
Table 3-3: Performance and Resource Expense Attributes for a Space Tug Program 99
Table 3-4: Performance and Resource Attributes for Designs A to F in Epoch 1. 101
Table 3-5: Performance and Resource Constraints for a Set of Epochs (sequenced as an Era) (Epochs 1,5,6,13,14) 101
Table 4-1: Launch cost and payload capacity for selected current launch vehicles. 112
Table 4-2: System Level Performance and Expense Attributes 117
Table 4-3: System Level Design Variables 120
Table 4-4: System, Program and Portfolio Level Context Variables 123
Table 4-5: Possible development missions for the FSS 124
Table 4-6: Summary of Hypothetical FSS Development Mission Sequence 129
Table 4-7: DVM for Satellite System 130
Table 4-8: Assumed σ values for different contexts 139
Table 5-1: Program Level Performance and Expense Attributes 152
Table 5-2: Program Level Design Variables 155
Table 5-3: DVM for Satellite Constellation Program 157
Table 5-11: Summary table of the characteristics of program designs that have been identified as affordable 178
Table 6-1: Portfolio Level Performance and Expense Attributes 182
Table 6-2: Program Constellation Profile of FSS Portfolio 19 204
Table 6-3: Satellite System Profile of FSS Portfolio 19 205

1 INTRODUCTION

1.1 Motivation: The Ariane VI Conundrum

In July 2013, the European Space Agency (ESA) announced that it had selected a basic design for a new launch vehicle, the *Ariane VI*, which will be powered by two solid-fuelled lower stages and incorporate the liquid-fuelled upper-stage currently being developed as an upgrade for the existing Ariane V vehicle (BBC, 2013). The basic design was chosen after conducting a series of trade studies for six months and feasibility work approved by ESA member states.

However, despite being the newest member of the Ariane family, the Ariane VI will have less lifting capacity than the Ariane V and it will only be able to carry a payload of 3 - 6.5 tons to the high orbits occupied by telecoms satellites. The latest variant of the Ariane V, however, will actually be able to carry up to 11.5 tons after its upgrade. Furthermore, the Ariane VI will only be able to launch just one spacecraft at a time, not the two that are routinely lifted by the Ariane V.

The Ariane VI seems to have several performance limitations, which may deeply impact the nature of future ESA space missions. Such a development will appear confounding at first sight, as it seems that the latest class of launch vehicles is taking a step backwards in terms of performance. Designing and launching the Ariane VI already consumes substantial time and monetary resources from ESA and such expenditures would typically be justified under traditional standards only if there were a significant improvement in performance levels. However, this was not observed in the Ariane VI. This poses several questions as to how and why this might have happened, and whether it was accidental or intentional.



Figure 1-1: Comparison of the Ariane V (left) and Ariane VI (right) (ESA, 2013)

ESA representatives offered their explanations for this new twist in design and they stated that the *primary driver for the new configuration is the quest to reduce costs of manufacture and operation*. The Ariane V, despite being highly reliable and successful, is priced above its competition and will probably not be sustainable for future space missions. As a result, ESA representatives feared that global demand for Ariane launch vehicles would decrease over time unless *a cheaper approach* is adopted (BBC, 2013). Therefore, ESA has set a target to try to produce and launch the Ariane VI for no more than about 70m euros (£60m/$90m). This is much cheaper than the current launch cost of the Ariane V, which is estimated to be around 145m euros (£60m /$200m) in 2013 (Parabolic Arc, 2013).

However, reducing costs is not a straightforward task. Producing a new launch vehicle would involve the collaboration of many ESA member states and there is much to be done during early-phase design to ensure that the Ariane VI can operate reliably. Alain Charmeau, the CEO of Astrium Space Transportation, which leads the Ariane industrial consortium, expressed his concern regarding the challenge of adopting a cheaper approach and reducing costs:

"Astrium now has to capture the ball that has been sent to us today by the agency."
"We will have to make Ariane 6 a very competitive launcher. It's really a complete change in Europe. It's the first time ever that we will try to develop a rocket thinking about production price and not just performance."

Through producing the Ariane VI, ESA hopes to achieve more economic returns by reducing the scale of the production consortium spread across the European continent, and by including fewer, less complex components in the build itself. ESA also envisions that once the Ariane VI enters service and proves its reliable performance, it can replace both the more expensive Ariane V and the Russian medium-class Soyuz launcher. If development is approved by 2014, the Ariane VI could make its first flight by 2021-22. Eventually, operating just the Ariane VI alone can help ESA fulfill a wide range of customer needs and provide substantial profits in the future. Therefore, the Ariane VI may pale in comparison in terms of performance, but it is projected to make space transportation more affordable in the future.

The Ariane VI conundrum serves to highlight several important issues. It appears that performance has no longer become the top priority during the design process. It is clear that the definition of "better" is no longer confined to *performance* levels of the system. In order to remain competitive in today's defense and aerospace industry, cheaper approaches must be explored and there has to be more emphasis on price or *cost* in addition to performance.

Apart from cost, there is also the notion of adhering to the development *schedule*, as every year delayed equates to another year of operating the more expensive Ariane V. There is now a need to place more priority on *cost and schedule* throughout the design process and determine which aspects would allow new designs to remain attractive and competitive. Antonio Fabrizi, Director of ESA launchers, offered his opinion of how this could be done:

"We don't reduce the costs via technologies; there are no breakthrough technologies that help us to make revolutionary launchers that can provide performance at low cost"

It appears to suggest that other ways must be devised and deployed to allow the conduct of *tradeoffs between performance, cost and schedule* during the design process.

1.2 Prevalence of Cost and Schedule Overruns

The approach taken by ESA in designing the Ariane VI is motivated by the need to reduce the cost and time spent developing the system. It is just one of the many design cases where cost and schedule are explicitly taken into consideration during design at the expense of performance levels. Previously, emphasis was placed mostly on maximizing performance and less attention was given to managing cost and schedule attributes. As a result, many cost and schedule overruns are experienced during system and program development. This predicament has been observed to be prevalent throughout the US defense and aerospace industry for decades.

In 2012, nearly half of the US Department of Defense's (DoD) 96 largest acquisition programs (GAO, 2012) have failed to meet the cost growth and schedule standards that were established to identify troubled defense programs (Schwartz, 2010, 2013). In fact, Figure 1-2 shows that the number of programs that met these criteria have been decreasing in recent years. This is an alarming trend as it indicates a reduced buying power for the military.

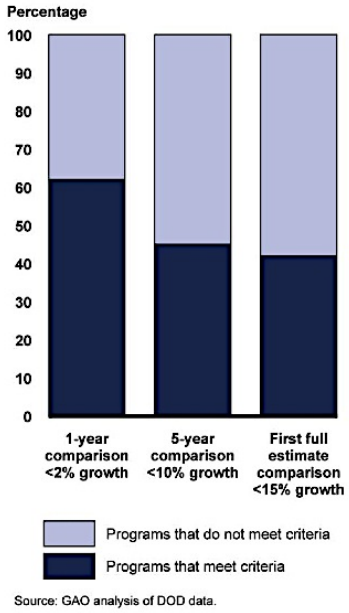


Figure 1-2: Percentage of Programs Meeting Total Acquisition Cost Growth Targets (GAO, 2013)

It was further reported that the total acquisition cost of DoD's Fiscal Year 2011 portfolio of 96 major defense acquisition programs grew by more than $74.4 billion, or 5%, in the past year. About $31.1 billion of that amount can be attributed to factors such as inefficiencies in production, $29.6 billion to quantity changes, and $13.7 billion to research and development cost growth (GAO, 2013). DOD's largest weapon system acquisition program – the Joint Strike Fighter program – accounted for most of the cost growth (GAO, 2013). However, it is just one of the many programs to experience management and execution problems as a result of cost and schedule overruns. Despite active reductions in weapon unit quantities and reduced performance expectations, the cost overruns on such Major Defense Acquisition Programs (MDAPs) have grown to more than $300 billion over original program estimates. These overruns have led to delays in program developments and even cancellations.

Other notable defense programs that experienced cost and schedule overruns were the Army's Comanche armed reconnaissance helicopter, the Navy's DDG-1000 next-generation surface combatant, and the Air Force's Transformational Satellite Communications System (TSAT) (Cancian, 2010). The Comanche program commenced in 1982, but increasing unit costs resulted in a 10-year delay in schedule and its eventual cancellation in 2004. The $6.9 billion initially allocated for the procurement of 120 Comanche helicopters over 5 years could have been directed towards upgrading 350 AH-64 attack helicopters to deliver greater warfighter capability, but was instead used to purchase 800 other helicopters (Cancian, 2010).

Similarly, the DDG-1000 program was cancelled in 2009 due to high costs and mission limitations, and funds were instead used to procure additional units of the older DDG-51 model. Unnecessary expenditures and schedule delays could be averted if the Navy initially decided to purchase 13 units of the DDG-51 class for its $23 billion investment in only 3 DDG-1000 units. TSAT was also cancelled in 2009 due to rising costs and schedule slips. The Air Force might have used the $3.5 billion initial investment in TSAT to purchase 7 units of the existing Advanced Extremely High Frequency (AEHF) satellites to avoid gaps in coverage (Cancian, 2010).

The failure to deliver these defense systems as a result of cost and schedule overruns can seriously compromise the US military's warfighting capabilities. These high-profile failures have therefore accentuated *the need to reduce cost overruns and schedule delays*.



Figure 1-3:The Army's Comanche armed reconnaissance helicopter (left), the Navy's DDG-1000 next-generation surface combatant (middle), and the Air Force's TSAT (right).

These failures are also abundant in the aerospace industry, with the most notable being the James Webb Space Telescope (JWST). Referred to as the "Next Generation Space Telescope" and a top priority in the National Aeronautics and Space Administration (NASA) science decadal survey, JWST is a large deployable, infrared-optimized space telescope that has been designed to succeed the Hubble Space Telescope. JWST is to conduct a 5-year mission to find the first stars and trace the evolution of galaxies from their beginning to their current formation. However, like the defense programs described, the development of JWST also experienced significant increases to project costs and schedule delays (GAO, 2013).

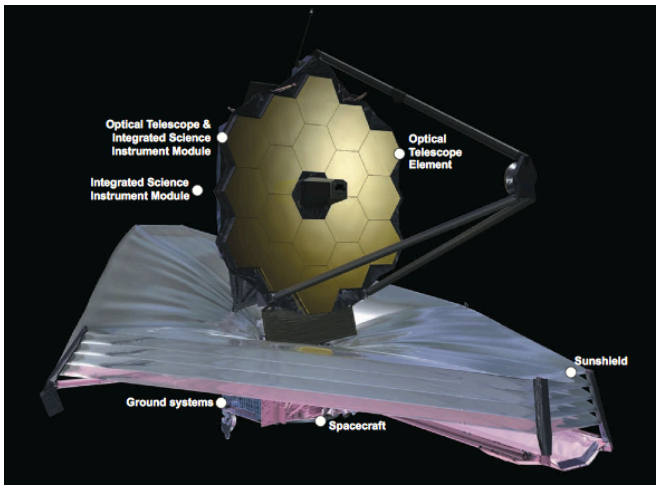


Figure 1-4: James Webb Space Telescope (GAO, 2013 - JWST)

Prior to being approved for development, cost estimates of the JWST project ranged from $1 billion to $3.5 billion with expected launch dates ranging from 2007 to 2011. In March 2005, NASA increased the JWST's lifecycle cost estimate to $4.5 billion and slipped the launch date to 2013. It was found that the cost growth was due to a 1-year schedule slippage, which was caused by a delay in the decision to use the Ariane V launch vehicle. Further schedule slippages followed due to budget profile limitations in later fiscal years. Further cost growth was later incurred as a result of changes to requirements and architectural decisions. Most recently, NASA announced that the project has been re-baselined at $8.835 billion - a 78% increase to the project's lifecycle cost from the initial baseline - and would be launched in October 2018 - a delay of 52 months. (GAO, 2013 – JWST)

Cost growth and schedule growth in other NASA Earth and Space Science missions conducted were also significant. Shown in Figure 1-5, a number of missions have experienced at least a 40% increase in both percent cost growth and percent schedule growth (NRC, 2010). The percent cost growths for some missions were also broken down across the phases during which they were incurred. These phases are "Start to PDR", "PDR to CDR" and "CDR to Launch". At NASA, the Preliminary Design Review (PDR) is typically conducted to demonstrate that the preliminary design meets all system requirements with acceptable risk and within the cost and schedule constraints. The Critical Design Review (CDR) occurs later and it serves to demonstrate that the design has matured sufficiently to support proceeding with full-scale fabrication, assembly, integration, and test. From Figure 1-5, it appears that majority of the cost growths occurred from "CDR to Launch". This trend is important as it highlights that most cost growths occur later in development and it may suggest that little is done during the earlier phases to prevent the occurrence of these growths. Therefore, it is imperative to *consider cost and schedule parameters on top of performance levels right at the very beginning of design*. Holistic considerations of these elements at program inception can potentially reduce the overruns of overruns in future.

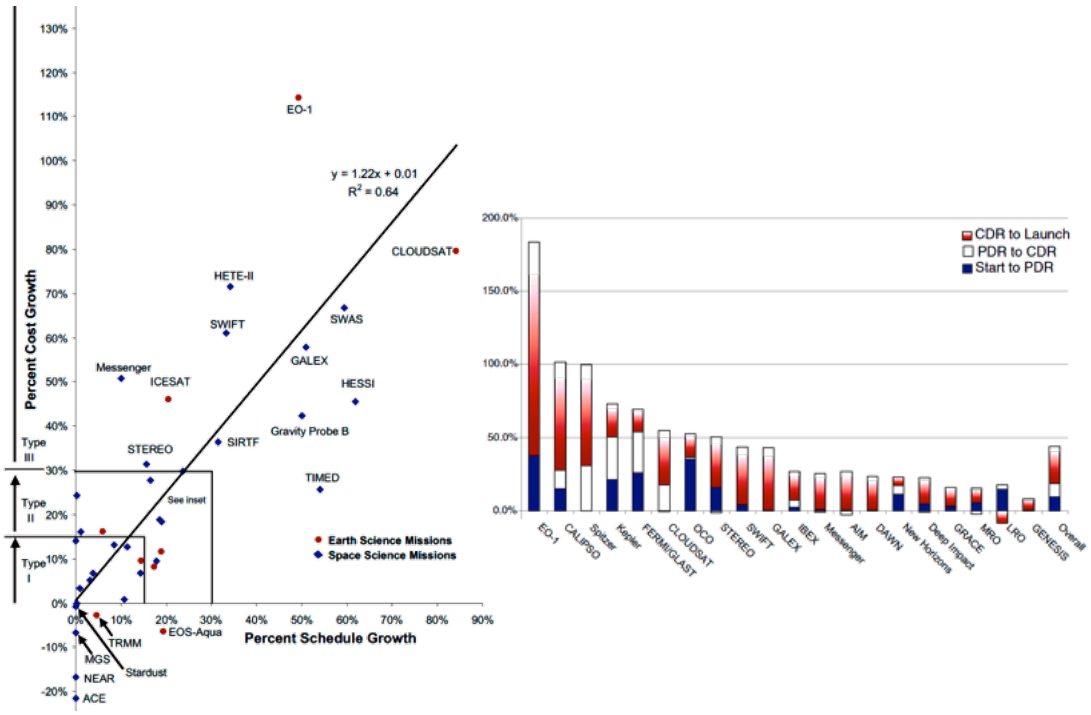


Figure 1-5: Percent cost growth and percent schedule growth plot for various Earth and Space Science missions conducted by NASA. The cost overruns for some missions are breakdown into the 3 phases during which they were committed. (NRC, 2010)

1.3 Considering Performance, Cost and Schedule for Affordability

Across all cases of program management failures observed in the defense and aerospace industry, unanticipated cost and schedule overruns have rendered systems and programs *unaffordable*. Project developments are thus becoming more expensive and longer than initial estimates as a result of both cost and schedule growths over time. The failure to adequately consider cost and schedule at the beginning of design as well as the ineffectual management of tradeoffs among the three key elements have collectively contributed to the current circumstances.

The rudimentary tradeoffs among the three elements are shown in Figure 1-6, which shows how performance, cost and schedule are closely interconnected. When a system costs more or is under performing, system architects are immediately prompted to implement a greater cost margin and spend more to make up for these shortfalls. A similar action is taken if the system costs more or is behind schedule. When a system is now behind schedule or underperforming, system architects will apply a schedule margin and extend the development schedule. With more states that the system being developed can exist in, many margins are implemented and their cumulative effects will eventually result in cost and schedule overruns. Therefore, there is a need to perform these trades better to prevent such overruns and be able to design systems that remain affordable over time. This is the principle of *affordability*.

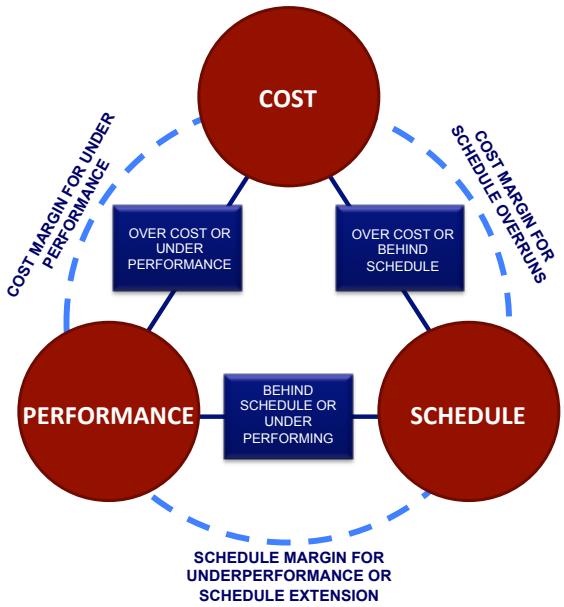


Figure 1-6: Problems in trading off performance, cost and schedule (Adapted from Alleman, 2010)

In response to the prevalence of cost and schedule overruns, the US government issued a new initiative - "*Mandate affordability as a requirement*" (DAU, 2013). Dr. Ashton Carter, then Under Secretary of Defense for the United States, outlined this affordability initiative to improve efficiency in spending to ensure that the country will be able to afford the systems it acquires. Dr. Carter defined affordability as an approach "to manage programs for weapons or information systems without exceeding our available resources". If programs were not designed with the notion of affordability, more time, effort and money would be wasted on cancelled programs. In the face of budgetary and mission uncertainties, affordability is needed more than ever to achieve the optimal balance of performance, cost and schedule elements. Therefore, affordability has now become a design requirement. It now remains to find out the most preferred and suitable approach to implementing affordability.

1.4 Finding the Best Approach to Affordability

To find the best approaches to affordability, the problem has to be tackled at its roots. Over the years, many investigations were conducted to identify the major causes of cost and schedule growth in defense and aerospace programs. A 2009 report by the Institute for Defense Analyses (IDA) narrowed the causes to two main categories: weaknesses in management visibility, direction and oversight; and weaknesses in initial program definition and costing (IDA, 2009). The first broad category covers a *general lack of discipline in management*, and this may imply lax or inappropriate implementation of policies, excessive reliance on unproven management theories and acquisition strategies, and poor contractor selection processes. The second broad category encompasses *failures in systems design and early-phase planning*, as well as unrealistic cost estimates. Specific causes within this category may include failure in eliciting or anticipating stakeholder requirements, usage of immature technologies, shortfalls in systems engineering methods, as well as inefficiencies resulting from schedule compression and concurrency.

IDA further reviewed a number of programs and discovered their cost growths can be attributed to an overlap of several weaknesses from both categories. The weaknesses underlying the cost growths for 11 selected programs are shown in Table 1-1. Tabulating the frequency of occurrence for each weakness reveals that *the lack of appropriate systems engineering methods is a common underlying factor of cost growth in major acquisition programs*. Other dominant factors include failures in the requirements process, as well as schedule compression and concurrency. As such, these results indicate that much can be done to *enhance the overall systems engineering framework used for designing and managing a program*. Therefore, systems engineering can be the main platform upon which affordability can potentially be implemented and designed for. *Applying more advanced systems engineering methods thus constitute the best approach to affordability*.

Table 1-1: Areas of weakness causing cost growth in programs. Adapted from (IDA, 2009).

Defense Program	**Weaknesses in top management activities**			**Weaknesses in initial program definition and costing**			
	Lax or inappropriate policies	*Unproven theories and strategies*	*Poor contractor selection*	*Failures in requirements process*	*Immature technology*	***Lack of good SE methods***	*Schedule compression concurrency*
ARH			X	X	X	**X**	X
EFV		X	X	X		**X**	X
FCS	X		X	X	X	**X**	
Global	X	X	X	X		**X**	X
JASSM		X	X	X		**X**	X
JSF				X		**X**	X
JTRS	X			X	X	**X**	
LCS		X	X	X		**X**	X
LPD-17		X	X		X	**X**	X
SBIRS	X	X	X	X	X	**X**	X
WIN-T	X				X	**X**	

1.5 Affordability through Better Systems Engineering

Affordability can be implemented in the defense and aerospace industry through better systems engineering methods. Systems engineering is a discipline that has been cultivated from the very same industry and it has evolved as systems become increasingly complex. Nowadays, defense and aerospace systems are composed of a myriad of interacting subsystems that independently and collectively must satisfy a complex set of performance requirements. These requirements can change over the system development cycle and can even evolve during system operation to satisfy new challenges and mission requirements. Therefore, the design, development and operation of such systems can be an arduous task.

Systems engineering emerged from these needs to manage complexities and it provides system designers and engineers the capability to ensure that the design process proceeds smoothly and that the system can fulfill most or if not all requirements within budgetary constraints. Therefore,

systems engineering is a methodical, disciplined approach for the design, realization, technical management, operations, and retirement of a system (NASA, 2012). It is a way of looking at the "big picture" when making technical decisions and achieving stakeholder functional, physical, and operational performance requirements in the intended use environment over the planned life of the systems (NASA, 2012).

To address affordability issues using systems engineering methods, it is first important to note that large commitments of technology applications, system configuration and system performance characteristics, obligation of resources, and potential lifecycle cost all occur at the early stages of a program. The system design conducted during these early stages is known as *early-phase design*. Referring to Figure 1-7, it is at early-phase design when decisions on conceptual and preliminary design are made and they will have a great impact on the cost of activities later on. However, system-specific knowledge is often limited during early-phase design, but decisions will still have to be made to further development progress. The defense and aerospace industry has been applying a variety of systems engineering methods over the years and they have seen many success, most notably the Apollo program. However, the industry has also been riddled with many program failures and this could indicate the need for novel solutions today to resolve this problem. Therefore, *better systems engineering methods may be required for affordability*.

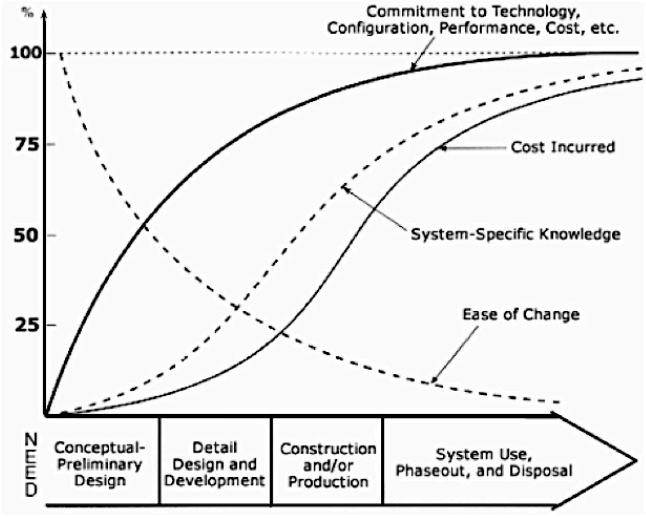


Figure 1-7: Commitment, system-specific knowledge, and cost over system lifecycle (Blanchard and Fabrycky, 2006)

Therefore, this paves the way for *advanced systems engineering* methods to be used. Advanced Systems Engineering is "a branch of engineering that concentrates on *design and application of the whole as distinct from the parts*... looking at the problem in its entirety, taking into account all the facets and variables and relating the *social* to the *technical* aspects" (Booton and Ramo, 1984). Therefore, a more *holistic* approach can be taken to tackle the affordability problem, which has its roots in all facets of performance, cost and schedule considerations. Also, it is clear that defense and aerospace systems are not simply physical products. They are *sociotechnical* systems that warrant attention to its social, technical, political and economic aspects through their design and development.

It is through this justification that methods from the *Systems Engineering Advancement Research Initiative (SEAri)* at the Massachusetts Institute of Technology (MIT) can be operationalized for affordability purposes. SEAri is a MIT research lab that is affiliated with both the Engineering Systems Division (ESD) and the Department of Aeronautics and Astronautics. It aims to "*advance the theories, methods, and effective practice of systems engineering applied to complex socio-technical systems through collaborative research*". Therefore, SEAri is uniquely positioned for interdisciplinary research in advancing systems engineering to meet contemporary challenges of complex socio-technical systems (Ross and Rhodes, 2008a).

Through integrating SEAri constructs and applying SEAri methods, advanced systems engineering techniques has the potential for being effective in ensuring affordability in defense and aerospace systems. The underlying goal for systems engineering may be reduced simply to *maximizing experienced, and therefore perceived, system success by stakeholders*. Part of the success of the system can be defined narrowly, in terms of minimizing costs, improving scheduling efficiencies, or meeting performance requirements, or it can be defined more broadly by maximizing the net benefit experienced by stakeholders through interactions with the system, while meeting or exceeding expectations (Ross and Rhodes, 2008a). Through these advanced systems engineering methods, design and leveraging decisions made early in development can then be improved in terms of efficiency and reliability.

MIT SEAri has developed a research agenda that spans over various important aspects of systems engineering. First and foremost, it aims to develop methods for value robustness through concept exploration, architecting and design using a dynamic perspective for the purpose of realizing systems, products, and services that deliver sustained value to stakeholders in a changing world (Ross and Rhodes, 2008a; Ross, Rhodes and Hastings, 2008; Ross, Rhodes and Hastings, 2009). Next, it seeks to enhance sociotechnical decision making through developing multi-disciplinary representations and analysis techniques, and adopts an economics-based view of systems engineering to achieve measurable and predictable outcomes while delivering value to stakeholders (Ross and Hastings, 2005; Richards, Viscito, Ross and Hastings, 2008). In addition, it also aims to achieve more effective systems engineering practice in the context of the system and the characteristics of the associated enterprise (Rhodes, Ross and Nightingale, 2009; Mikaelian et al., 2011). Finally, it focuses on developing prescriptive strategic guidance to inform the development of policies and procedures for systems engineering practice (Broniatowski and Weigel, 2008; Szajnfarber and Weigel, 2009). This makes MIT SEAri well positioned to approach affordability through a systems engineering perspective.

Therefore, it is in this motivation that the author of this thesis attempts to address the affordability problem encountered in the defense and aerospace industry through constructs and methods developed by MIT SEAri.

1.6 Research Questions

The aim of this thesis is to analyze current concepts and practices associated with the design of affordable systems, and attempt to address the affordability problem in engineering design through advanced systems engineering methods. Research for this thesis was guided by four principle questions outlined below:

1. *What is affordability in the context of defense and aerospace systems?*
2. *How can affordability be incorporated in early-phase design?*
3. *What are the issues associated with considering affordability in early-phase design?*
4. *How can affordability concepts be propagated through the defense and aerospace industry?*

The *first question* seeks to direct the initial thrust of the research towards finding out what affordability means with respect to the design and operation of engineering systems. Given the problems currently faced by the defense and aerospace engineering communities, it will be of interest to find out how affordability issues relate specifically to the engineering design of complex systems in the industry. It is also of interest to find out what has been done and what has yet to be done to consider affordability in design.

The *second question* then builds upon the work done in the first research thrust and it will form the biggest research thrust in this thesis. It aims to determine how affordability can be qualitatively and quantitatively incorporated into the engineering design process. Affordability warrants the increased consideration of cost and schedule parameters into the design process. It is then important to find out which design solutions are affordable or unaffordable.

To ensure balanced tradeoffs among performance, cost and schedule, there exist many solutions for system design. However, not all of them are equal in terms of benefits, risks and cost. Therefore, the second research thrust also seeks to introduce and apply advanced systems engineering methods for affordability purposes. In addition, it is also of interest to find out what methodologies or heuristics can be applied in the search for affordable design solutions.

There are a wide variety of engineering design problems and some sociotechnical systems are greater in scale than others. Given the inherent complexity of these systems, there is potential for confusion to occur in the use of systems engineering methods as well as the people applying them. Therefore, it is also of interest in this research thrust to determine if the advanced systems engineering methods introduced can be conducted in a progressive and disciplined manner. The manner in which these methods are applied may also vary according to the size and complexity of the system being designed.

The *third question* then seeks to determine the potential benefits and problems with the methods introduced to address the affordability problem. Methods are never perfect and it is important to know how to apply them in ways that maximize their strengths and mitigate their weaknesses.

The *fourth question* directs the final research thrust and it seeks to determine ways beyond the technical approaches of systems engineering to ensure the design of affordable systems. Affordability is not only an engineering technical problem but also a management problem. Therefore, it is of interest to find ways in which policies, frameworks and management practices can be refined in order to realize affordability throughout the defense and aerospace industry.

1.7 Research Methodology and Thesis Outline

In this research, a literature review was first conducted to collate and analyze different takes on affordability by various academic, industrial and government institutions. This was done to determine what has already been done in affordability research and what the knowledge gaps that may exist are. A new definition of affordability was then proposed for the purposes of this research. Methods and constructs developed in MIT SEAri were then modified and introduced as potential solutions to address the knowledge gaps and limitations in considering affordability during early-phase design. These methods were applied to a Space Tug system and program case study to demonstrate the feasibility of these methods.

The author of this thesis was also a member of the Strategic Innovation Research Group (SIRG), a space systems engineering research consortium comprising student and faculty members from MIT and the Skoltech Institute of Science and Technology. As part of SIRG's research agenda, additional research was conducted in the design and development of a new space systems concept: Federated Satellite Systems (FSS). The FSS was used as the primary case study in this thesis, where conceptual and computerized models were formulated to facilitate its early-phase design process. The methods introduced were applied to derive affordable solutions for a single satellite, a satellite constellation, and a portfolio of satellite constellations. After the application of the methods to these case studies, the methods were then analyzed in terms of their benefits, risks and cost. This allows potential users of these methods to remain aware of these issues and apply them appropriately to obtain best results.

In order to determine how considerations of affordability can be propagated throughout industry, prominent acquisition frameworks were analyzed to assess areas in which the systems engineering methods introduced can be applied for maximal effect. Finally, other strategies for implementing affordability were also discussed so that they can potentially be used concurrently with systems engineering methods to design affordable systems in future.

The chapters in this thesis are organized according to the key research activities described:

1. *Introduction*
2. *Literature Review of Affordability*
3. *Tradespace-based Methods for Affordability Analysis*
4. *Federated Satellite Systems – System Analysis*
5. *Federated Satellite Systems – Program Analysis*
6. *Federated Satellite Systems – Portfolio Analysis*
7. *Integrating Tradespace-based Methods with Industry Practices and Policies for Affordability*
8. *Conclusions and Future Work*

Affordability is an emergent concept whose importance has grown considerably within the defense and aerospace industry as a result of persistent cost and schedule overruns. To bridge the knowledge gaps identified in the current state of affordability studies, the research in this thesis thus aims to enhance affordability considerations in early-phase design so that more feasible and affordable systems can be delivered within the defense and aerospace industry.

2 LITERATURE REVIEW OF AFFORDABILITY

2.1 Motivation

This chapter aims to provide a comprehensive review of current affordability research and practices in both academia and industry. A repertoire of journal articles, conference papers, industry reports, government documents, theses and books were reviewed in order to obtain a holistic understanding of the approaches taken to integrate the emerging concept of affordability into existing system engineering frameworks and acquisition practices. The chapter begins with a lexicographic analysis of the term *affordability* and the tracking of its increasing relevance and usage in daily applications to the engineering of complex sociotechnical systems. In coherence with the overall theme of this thesis, the scope of affordability usage and application will be narrowed down to the defense and aerospace industry, where its significance in the design of complex systems will be explained in further detail. The evolution of affordability concepts in the defense and aerospace industry will then be described, beginning with the motivating factors that lead to its necessity and the corresponding paradigms for understanding affordability. The chapter ends with evaluating the limitations of current research and practice, which establishes the motivation for newer and more advanced methods that can bridge the identified gaps in affordability studies.

2.2 The Meaning of Affordability

In this section, a lexicographic analysis of affordability will be performed. Its common usage and application as well as its relevance to the defense and aerospace industry will be assessed.

2.2.1 Lexicographic Analysis

Affordability, a portmanteau of the words '*afford*' and '*ability*', has been colloquially defined as '*the ability to afford*' by all who engage in the universal transactions of time and/or monetary resources in return for desired products and/or services. According to Merriam-Webster (2014), the word 'afford' can take on two generalized meanings: *to manage to bear without serious detriment and to be able to bear the cost of; or to make available, give forth, or to provide naturally or inevitably*. The word 'ability' also has two broad definitions: *the quality or state or being able to perform or execute; or the natural aptitude or acquired proficiency*. The general concept of affordability can thus be easily understood through combining any pair of lexicographic definitions for these words. In this lexicographic analysis, affordability will be defined through the examination of its root words.

Central to these early conceptions of affordability is the notion of 'ability', which can be interpreted as an enabling characteristic or feature inherent to a product or a service that appeals to either the buyer or seller side of the transaction. This 'ability' may be imbued into the product or service in ways such that customers can 'afford to buy' and suppliers can 'afford to sell', thereby establishing the potential to initiate more successful transactions in the future. More often than not, this 'ability' can reduce the amount of time and/or monetary resources that are usually incurred in such a transaction. While it is most desirable when this 'ability' is naturally occurring

or emergent over time, it is more common that this 'ability' has to be engineered or designed into the product and/or service. *It is thus of both economic and academic interest to be able to design for this 'ability' in the product and/or service.*

Over the course of history, the words '*afford to*' have been used by numerous individuals, groups, businesses and governments under various circumstances. Most common extensions of these words include '*afford to buy*', '*afford to sell*', '*afford to wait*' and '*afford to lose*'. These notions of affordability can range from the relatively simple to the relatively complex, depending on the perspectives of those experiencing these circumstances. A simple everyday scenario can include a group of students choosing between two campus restaurants selling similar cuisines at different prices, of which the cheaper restaurant has a shorter queue.

Individuals within the student group may be debating between what they can 'afford to buy' with their individual budgets and whether each of them can 'afford to wait' in the queue for the cheaper food. Restaurant owners will also be concerned by this dilemma. The owner selling the more expensive dishes will begin to think about whether he can 'afford to sell' at lower prices to attract more customers, while the other owner will be pondering over whether he can 'afford to lose' more customers because of the waiting time. To remedy this situation, the former can possibly look into procuring cheaper ingredients in order to sell his dishes at more competitive prices, while the latter can explore how he can reduce the preparation time of his dishes without compromising much on the quality. In this fictional scenario, the key elements of being able to 'afford to' are exemplified in both the customer and supplier sides.

This simple scenario can be extrapolated to more complex government projects such as the design of a new reusable launch vehicle that has the potential to lower the cost of space transportation. The government may submit a request for tender and allow several aerospace companies to propose their design and estimated cost. After gathering all the proposals, the government may evaluate them based on what they can 'afford to buy' with their current budget, how long they can 'afford to wait' depending on their future mission needs and the estimated production times for the launch vehicles. Different divisions within the government may have varying preferences for launch vehicles, as each division may desire to use them for different payloads and scientific purposes. The companies, being the sellers, have different interests and they are possibly interested in how many vehicles and at what price they can 'afford to sell' in order to generate substantial profit. Concurrently, the companies are also concerned about how much they can 'afford to lose' in terms of long-term profits should they not win the tender or should their proposed design not meet actual mission requirements.

Parallels can be drawn from both scenarios, where the students and the government divisions form the *customer* base while the restaurant owners and the aerospace companies form the *supplier* base. The elements that allow the customer and supplier bases to converge in their respective transactions are the products – the food and the new launch vehicle. In the first case, the quality and preparation method of the food will have to be adjusted in order to increase its appeal to the greater student community, while in the second case, the lifting capacity, number of launches, cost and the production time per launch vehicle can be traded off in order to satisfy the requirements of all the government divisions.

The actions taken to perform these adjustments thus provide the newly modified product and/or service with the critical 'ability' such that customers can now 'afford to buy' and 'afford to wait'. As such, the success of any simple or complex transactional scenario is pivoted on the engineering and design of the product and/or service in a manner such that time and monetary resources can be reduced while meeting desired requirements and maintaining sustainability of the customer base. *There is thus greater responsibility on the supplier base to assess the customer base and fully cater the design of a product and/or service towards their time and monetary needs while maintaining their own set of interests.* Given the pervasion of economic principles in every industry, it is no wonder that the notions of the innate 'ability' and being able to 'afford to', and ultimately the concept of affordability, can be found and applied across multiple scenarios.

2.2.2 Common Usage and Application

Affordability is a widespread concept that has been found in virtually every industry and facet of life. As illustrated in the previous section, affordability concepts can exist in the simplest of social settings and everyday scenarios. Rational individuals and small businesses looking to purchase a product or acquire a service they need are considering their cost and time budgets concurrently in order to pick the most feasible of options made available to them. At an enterprise level, these concepts are incorporated into business and economic models of firms and corporations.

These industrial players may be seeking to explore new ventures or establish mergers, which all necessitate the integration of cost and time related elements into their decision-making processes. At multinational levels, governments and global organizations are always looking to implement strategies and policies that achieve maximum results within the shortest time frame and at the lowest cost possible. Be it defense, transportation, healthcare or finance, affordability concepts are omnipresent, thereby motivating the customer and supplier base to interact continuously and design a product and/or service that can become increasingly suitable for their requirements.

To determine the extent of affordability applications, Bankole (2011) conducted an in-depth review of affordability-related articles appearing in academia, government and industry. *A surprising result of the study was that while the words 'affordability' or 'affordable' appeared extensively in many articles, most of them did not explore affordability as a concept.* This is due to the fact that common definitions for affordability were not provided and existing definitions tend to be industry-specific.

For example, in construction, affordability is "*a measure of whether housing can be afforded by certain groups of households*" (Semple, 2007) while in the public utility sector, it can be "*the ability of customers to pay for utility service billed to them*" (Smyth, 2005). The scope of the definition increases within the defense and aerospace industry, where affordability can be defined as "*the ability to procure a system as the need arises, within a budget, operate at a required performance level; maintain and support it within an allocated life-cycle budget*" (Kroshl and Pandolfini, 2000) or "*the degree to which the life cycle cost of an acquisition program is in consonance with the long-range investment and force structure plans of national defense administrations*" (North Atlantic Treaty Organization, 2007).

While these definitions are vastly different, they all suggest that affordability is concerned with the *comparison of some monetary measure to known levels of customer income, investment or*

budget. For example, the definitions provided within the construction and utility sectors focus on consumer goods and they generally compare the cost of providing housing or utility services to individual customers with the household income while taking account of all basic necessities. *Another notable aspect of these definitions is also the consideration of different customer groups with varying budget preferences.* This was exemplified through phrases like "certain groups of households" and "national defense administrations" that demonstrate categorization of customers.

Another important aspect in these definitions is the concept of time, which may be monthly for individual consumers and annually for business and government customers. Owing to the nature of agreement for the product and/or service provision between supplier and customer, fees for housing and utilities are generally shorter. On the other hand, product and/or service provision for business and government customers, especially in the defense and aerospace industry, take the form of contracts, which have long durations and are often subjected to higher degrees of uncertainty.

In addition to concerns about the expenditure relative to their budgets, customers are also concerned about the availability of the product and/or service when they need it, its capability to provide the required functionality, and overall cost and time effectiveness for its expected life cycle. Hence, customer affordability, be it individual, business or government, focuses on the customer's ability to pay for the product or service provided by the supplier. This is usually affected by the customer's perception of value and the worth of the product offering. In the literature review conducted by Bankole (2011), 'customer affordability' is often referred to as 'affordability', which is why most of the articles reviewed were focused on affordability because the subject is usually explored from the customer's perspective. *Customer affordability is thus a major perspective of affordability and should become the main focus of affordability studies.*

Bankole (2011) then classified the articles according to their industry of origin and the viewpoints expressed by the authors. The results were first classified according to sectors surveyed by the authors in each of the papers. Eight sectors were identified, namely: Aerospace (Defense and Civil), Construction, Energy, Water, Financial, Telephone and Shipping. The results were shown in a pie chart in Figure 2-1, which illustrates *a good majority or 52% of the materials reviewed were within the aerospace sector*. Affordability is a new research concept within the defense and aerospace sector and it is in the process of establishing measurement techniques and improvement guidelines. *Evidenced by such a strong industrial and academic focus on affordability concepts in the defense and aerospace industry, it is thus of interest to find out why and how these concepts are important to the survival and development of this industry.*

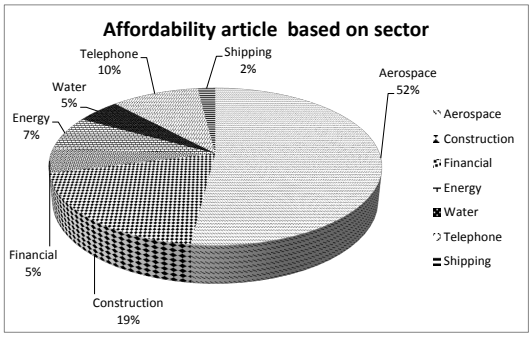


Figure 2-1: Classification of affordability articles based on sector (Bankole, 2011)

2.2.3 Significance in the Defense and Aerospace Industry

The defense and aerospace industry has traditionally been characterized by large-scale, complex engineering systems such as commercial aircraft fleets, missile defense systems, satellite systems and space exploration vehicles. These systems are designed to fulfill broad mission statements and are often subjected to changes and disturbances across their lifecycles. Each of these systems further constitutes multiple subsystems and multiple stages of development that have numerous stakeholders partaking in the formulation of cost, schedule and performance requirements. The architectural development of these systems is also multidisciplinary in nature, where hundreds or even thousands of people and businesses with diverse backgrounds are involved. Defense and aerospace platforms thus comprise a myriad of interacting subsystems that must independently and collectively satisfy a complex set of performance requirements.

2.2.3.1 System vs. Program vs. Portfolio

The defense and aerospace sector is a multi-billion dollar industry that is sustained by the continuous development and evolution of these systems, where the natures of these transactions are often of high costs, high risks and high stakes. The design requirements for these systems can change and evolve throughout their lengthy development cycles to satisfy new challenges and performance needs. These cycles may last 5-10 years for small systems or even stretch over 30-50 years for more complex ones. Depending on the scale and duration, development projects can be classified as a *system*, *program* or *portfolio*. Given the scalability of the work in this industry, the pillars of cost, schedule and performance can vary greatly. As such, considerations for affordability are performed more in the defense and aerospace industry than anywhere else and it is the complex nature of this industry that begets the need for such considerations to be executed in an integrated and systematic manner.

A *project*, in generally, can be defined as the enveloping process that encompasses the socioeconomic and technical considerations in delivering a system or a program or a portfolio (KLR, 2008). Therefore, a *system* is defined to be a combination of interacting elements organized to achieve one or more stated purposes (INCOSE, 2012) while a *program* can be defined as a group of related and interdependent projects managed together to obtain specific benefits and controls that would likely not occur if these projects were managed individually (KLR, 2008). A *portfolio* can be defined as a collection of projects or programs grouped together to facilitate the effective management of efforts to meet strategic business objectives (KLR, 2008). *Both programs and portfolios can be regarded as System of Systems (SoS).*

Program- and portfolio- level affordability can be achieved through either a *top-down* or *bottom-up* approach. A top-down approach entails the application of affordability considerations at the program level such that its effects potentially cascade down to its constituent systems. A bottom-up approach conversely demands the aggregation of system-level affordability for each constituent system in order to establish program-level affordability. Application of either approach may yield different results. Portfolio-level affordability analysis may involve applying affordability considerations across multiple projects, programs, and possibly even portfolios. A portfolio-level affordability study can potentially provide overarching guidance to architecting entire defense capabilities within realistic bounds of cost and time. Similarly, top-down and bottom-up approaches can also be taken to achieve portfolio-level affordability. *Therefore,*

affordability considerations in the defense and aerospace have to be applied at the levels of system, program and portfolio and this can become a very complex process in program management.

2.2.3.2 The Better Buying Power Initiative

The study of affordability has been growing of importance in recent years. After experiencing multiple high-profile failures in program development, the defense industry in particular has sought to undergo several transformations in the next few decades as the DoD seeks to implement numerous initiatives to strengthen their fighting power as well as enhance their business and engineering practices. One of those is the "*Better Buying Power*" (BBP) initiative, which requires the DoD to "do more without more" by reducing low-priority overheads during periods of budgetary decline and use those funds for modernizing warfighting capabilities (DAU, 2010; DoD, 2012).

This initiative has had considerable impact on the defense industry in recent years. Dr. Ashton Carter, then Under Secretary of Defense for Acquisition, Technology, and Logistics, issued the memorandum "*Better Buying Power: Guidance for Obtaining Greater Efficiency and Productivity in Defense Spending*" in 2010 to *target affordability and control cost growth*. Later in 2012, current Under Secretary of Defense Frank Kendall launched the "*Better Buying Power 2.0*" initiative, an update to the original BBP effort. As such, affordability has emerged as a new concept that must be explicitly considered during the system design and architecting phases.

"*Mandate affordability as a requirement*" is the first specific initiative in the first area of the BBP initiatives to "*target affordability and control cost growth*". Within the context of this initiative, mandating affordability means to manage programs for weapons or information systems without exceeding available resources such as funding, schedule, and manpower. Failure to do so will result in wastage of time, effort and money on cancelled defense programs, which will cascade over the years to result in more budgetary uncertainty and unwanted compromise on warfighting capabilities (DAU, 2010). In order to ensure that defense systems, programs and portfolios are delivered on schedule and within budgetary requirements, affordability considerations will play deeply significant roles in the design and architecting of defense systems.

2.2.3.3 Unaffordable System Constructs

The close relationship between the military sector, aerospace sector and the federal government means that the aerospace industry is also experiencing the same cost and schedule problems in their programs and will be subjected to the same policies, initiatives and regulatory frameworks. NASA, who is a major player in the US space industry, has been experiencing cost and schedule inflations in its space exploration programs, where lack of historical data and improper cost estimation techniques have often been cited as the main reasons for incompetence in program management.

In recent years, military and civil space acquisitions have received much criticism for their failure to sufficiently incorporate cost and schedule considerations into their program design and their inability to produce realistic cost and schedule estimates. Complex space systems like the Space Based Infrared System (SBIRS) High and the National Polar-Orbiting Operational Environmental Satellite System (NPOESS) have experienced excessive cost growths, leading to the perception that the space acquisition process is "broken," ultimately eroding the credibility of the space

acquisition community (Allard, 2005; Gourley, 2004; Lee, 2004). In an era where space systems have become increasingly critical to the conduct of military and civil operations in the US, the combination of this dependence and recent difficulties in space systems acquisition has become a major concern for future space program developments.

Evidenced by multiple program delays and cancellations, it can be seen that the root of all these failures in the defense and aerospace industry is the unaffordable nature of the initial program construct. The current ways of setting requirements and acquiring large and complex systems produce programs that are extremely expensive and unrealizable due to high technology requirements. These old ways are not effective, and the space community needs to find fundamentally different ways of doing business. *As such, it is the culmination of recent programmatic failures and acknowledgement of obsolete practices that are driving both the defense and aerospace sectors to place more emphasis on cost and time cutting measures, more specifically taking affordability considerations into their systems architecting process.*

2.2.3.4 Acquisition vs. Procurement

In the practice of affordability concepts in the defense and aerospace industry, the key words that require attention are "*acquisition*" and "*procurement*". The Defense Acquisition University (DAU) defines *acquisition* as the process that "includes design, engineering, test and evaluation, production, and operations and support of defense systems" (DAU, 2010). The term "*defense acquisition*" is thus industry specific and generally applies only to weapon systems and related infrastructure such as software, operation procedures, labor services, maintenance etc.

The word *procurement* is defined as "the act of buying goods and services for the government" (DAG, 2010). However, procurement is often mistaken for acquisition and vice versa. *Procurement is instead, only one of the many functions performed as part of the acquisition process.* For example, much infrastructure required by the US DoD, such as passenger vehicles, office supplies, and waste removal, are regarded as "acts of procurement". However, they are not subjected to the full range of regulatory oversight inherent in the acquisition process for weapons, information technology systems, and supporting services. *Affordability considerations are thus most critical during defense acquisition, as major cost committals are often decided during the early phases of program development and it is easiest to reduce cost and time expenditures at this stage.*

As changes to system requirements occur often, the defense and aerospace industry has adopted the notion of *evolutionary acquisition*, which is a strategy to develop and deliver warfighting capabilities in successive increments in order to meet overall requirements. The process of acquisition by itself consists of different phases, each of which are already occurring in succession and incur a different amount of cost and time. As there is potential for overlapping to occur between constituent acquisition costs and schedule, it becomes very important to distinguish between overall system affordability and the affordability of the individual program increments. This can be extended to portfolio level management, which can consist of multiple programs each of which is composed of multiple systems or increments. This will be explained later in the thesis. Managing acquisition and procurement processes efficiently is thus the ultimate goal of the BBP initiative.

2.2.3.5 Longstanding Problems in the Industry

While mandating affordability as a requirement during acquisition is a relatively new initiative, the need to reduce cost and schedule overruns in defense and aerospace programs has been a longstanding problem within the industry. This has warranted various measures to be taken over the years. Many defense acquisition management and cost analysis groups have been established to identify current problems in business practice as well as propose new methods and frameworks in a bid to reduce cost and schedule overruns in current defense programs. In particular, the DAU identified major problems in the DoD's ability to acquire military capabilities in a timely and affordable manner and they include *technology requirements creep*, *overly optimistic cost and schedule estimates*, and *knowledge gap throughout the defense organization* (DAU, 2010).

Technology requirements creep occurs when requirements for new systems too often reflect the far limits of current technology and it is difficult to predict how and when an advanced technology can become successful for application in a proposed program. Furthermore, such unrealistic technological requirements continue to increase throughout a program's life cycle, and failure to implement such technologies can potentially lead to the cancellation of the program.

Overly optimistic cost and schedule estimates are another contributor to this predicament, as the acquisition process too often encourages overly optimistic estimates in order to ensure approval of proposed programs. Underestimating cost is likely to result in situations where they are too many programs chasing too few dollars, which eventually lead to cost threshold breaches that necessitate program terminations and reporting to Congress (DAU, 2010). Likewise, underestimating program schedules can lead to the planning of too many programs within the same period, leaving little cost and time buffers for accommodating unanticipated delays. Finally, there is a knowledge gap in the industry as the acquisition community still lacks trained personnel in the areas of cost estimation, systems engineering and acquisition management. This may cause problems in the conduct of effective cost and schedule oversight. Affordability practices can thus possibly bridge this gap.

2.2.3.6 Emergence of Value in Engineering

In addition to the identification of major problems, the need for stronger affordability considerations has also prompted numerous practices to be taken as the industry revamps its acquisition and procurement practices. These practices are not simply concerned about cutting overhead costs, but rather providing *value improvement* and enhancing the *value delivery* process. *Maximizing value has thus become the key objective of design and engineering within the defense and aerospace communities.* Some of the value-centered practices include *Value Engineering* (VE) and *Earned Value Management* (EVM) (DAU, 2010).

Conducting VE entails the functional analysis of systems, equipment, facilities, services, and supplies to ensure they achieve their essential functions at the lowest life cycle cost consistent with required performance, reliability, quality, and safety. Apart from scientific techniques, VE also incorporates available technologies as well as the principles of economics and business management into its procedures. Historical data from the application of VE within the DoD has so far demonstrated a positive return on investment from the VE process.

Building on the increasing emphasis on value delivery within the defense industry is the widely embraced framework of EVM. The EVM concept is a more holistic management approach that

provides all levels of management with early visibility into cost and schedule problems once incorporated into program design. EVM is fully embraced by the DoD and NASA acquisition workforce as an inherent part of the acquisition program management value chain and provides accurate and timely insight into cost, schedule and performance of DoD weapons systems and services programs.

As budget declines are expected, there will be increased competition amongst defense and aerospace programs seeking maximal return on value for their committed expenses. It thus becomes essential that government acquisition programs deliver as promised, not only because of their value to their users, but also because every dollar and every time unit spent on one program will mean one less available dollar and one less time unit to fund other programs. *To get better return on value for initial investments, defense and aerospace programs will need higher levels of design knowledge that can provide more complete perspectives to address problems in technology requirements creep, cost estimation and schedule estimation. Therefore, it is this desire for more knowledge during early-phase design that motivates the conduct of affordability studies within the defense and aerospace industry.*

2.3 Affordability in the Context of Engineering Design

The earlier section mentioned the importance of applying affordability principles and considerations in defense and aerospace acquisition. While acquisition comprises an entire lifecycle of activities, affordability is most pertinent to the earliest phase of the process – *design*. The type of design that is of principal concern is that of *engineering design*, where the end goal is to create an artifact, product, system, or process that performs a function or functions to fulfill customer need(s). Conceptualizing, defining, or understanding an artifact, product, or system, in terms of function, is thus a fundamental aspect of engineering design (Pahl and Beitz, 1984; Ullman, 1997; Ulrich and Eppinger, 1995; Hubka et al., 1988; Otto and Wood, 2001).

Building upon this foundation, this section begins by relating affordability with the goal of engineering design. This is followed by an evaluation of the current *context* of defense and aerospace programs, which is characterized by *rising cost and schedule growths* and *shrinking defense budgets*. Another problem plaguing the industry is the *difficulty in cost and schedule estimation practices*, which tend to be overly optimistic. The main participants in the defense and aerospace acquisition process are the US DoD and NASA, who are both strongly interlinked in the promulgation of system design needs and also subjected to heavy influences from the US government. The culmination of these factors, together with the interests of participants in the acquisition process, thus drives the need for affordability in engineering design. Finally, the current state of practice is provided by a brief description of affordability measures and initiatives taken within the defense and aerospace industry.

2.3.1 Affordability and the Goal of Design

From a broader perspective, affordability concepts are the crucial elements in driving towards the goal of engineering and design of the product and/or service. Whether it is a simple food dish, a technically complex reusable launch vehicle, a high-risk investment portfolio or a far-reaching defense or healthcare policy at the national level, they all require significant considerations of time and cost related parameters during the design process in addition to quality and performance

specifications. Ross (2006) describes the goal of design is to "create a *system* that fulfills some *need* while *efficiently* utilizing *resources* within some *context*". In this definition, there are key terms that require further explanation, as they will be used consistently throughout this thesis to complement affordability studies.

As described by Ross (2006), the *system* is "the concept, process or object, that is the product of the creative design process through which a need will be fulfilled". The *need* is then the driving value statement that led to the desire for a system in the first (or other) place. Some needs drive the design process while others are derived from it. The word *efficient* implies that "expenditure of time, money, labor, information, energy, and matter must be done with an eye to avoiding waste in order to improve the chances the *system* will be realized".

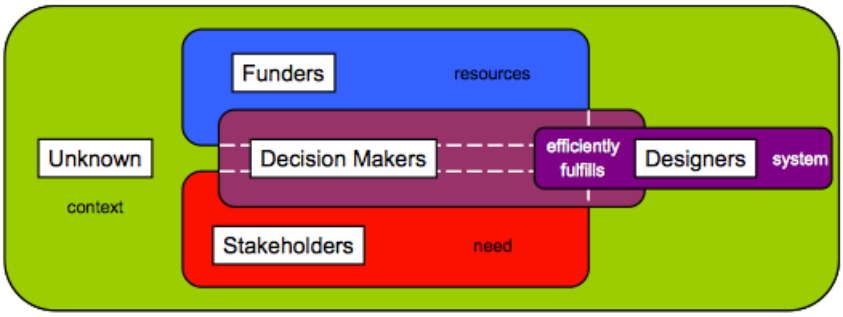


Figure 2-2: The goal of design (Ross, 2006).

The fundamental items and mediating supplies that are expended in the process of realizing the system are collectively known as *resources*, and they typically include currency flows such as money, energy, information, matter, and perhaps time and labor as well. The *context* comprises the constraints and environment that exists at and beyond the *system* boundary. An important aspect of the context is that it is imposed or out of the control of the designer and must be treated as an exogenous variable in the design endeavor.

Figure 2-2, conceptualized by Ross (2006), shows a graphical depiction of the goal and the key terms. The context encompasses the entire process, which illustrates the sphere of influences of the main participants. *Stakeholders* have influence over the definition and evaluation of the needs while *funders* have influence over and allocation of the resources. *Decision makers* act as the gatekeeper of needs and resources and they are empowered to determine whether to pursue a system development effort. Finally, *designers* command influence over the definition of the system, while efficiently utilizing resources and fulfilling needs, as determined by *decision makers*. As such, it is of great interest to determine how affordability concepts impact both the engineering design process as way as their players.

In order to integrate affordability concepts with engineering design, it becomes necessary to know what the current context is and what the current state of practice is. It is also important to identify the system, needs, resources and what is needed to achieve the desired efficiency. Since the *context* is the encompassing element in the goal of engineering design, the next section aims to describe the context - the current operating climate for affordability practices in the defense and aerospace industry. This can help determine how advanced engineering design methods can be better positioned in order to achieve desired results.

2.3.2 The Operating Climate for Affordability

This section addresses the current operating climate for designing affordable systems in the defense and aerospace industry.

2.3.2.1 Rising Cost and Schedule Growths

The trend of rising cost growths has been well documented in both the defense and aerospace sectors. However, due to the close relationship between both sectors, this trend will be described from the perspective of the defense industry, where programmatic failures in cost and schedule management have gained most prominence in recent decades.

Since the end of the Cold War, the US military has undergone numerous modernization phases to enhance its warfighting capabilities and it currently boasts the strongest armed forces in the world through extensive human and monetary investments. Despite numerous successes on the warfront, historical records have shown that many of these transformation programs were plagued by massive cost overruns, schedule delays, failure to anticipate future requirements and ultimately unrealized capabilities (Cordesman and Frederiksen, 2006). These cost and schedule growths, if uncontrolled, may threaten the economic feasibility of future acquisition programs and combat readiness of the US military in the long run.

The recent GAO 2013 report titled "Assessment of Selected Weapons Programs" reported a notable decrease in both size and cost of the defense portfolio, which appears to be buckling the usual inclination towards spiraling costs. This may appear to downplay the severity of the cost and schedule overruns. However, this is largely due to the cancellation of several programs, as well as reduction in procurement quantities for existing programs (GAO, 2013). While this may reduce costs and fulfill budgetary considerations, program cancellations and quantity reductions will eventually lead to reduced buying power and weakened defense capabilities. *Hence, it is more ideal to revise engineering methods instead of cancelling programs, so that the US DoD can maintain the same level of warfighting capability without the need to incur more time and cost in doing so.*

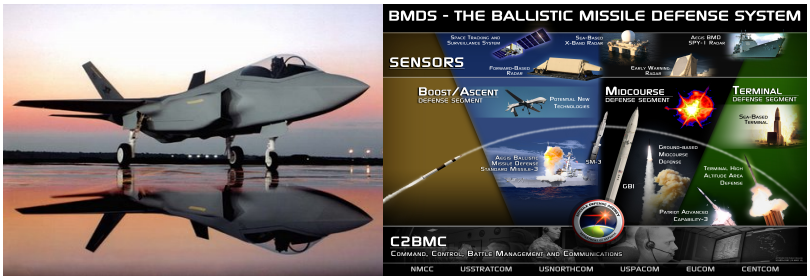


Figure 2-3: F-35 Joint Strike Fighter (left) and Ballistic Missile Defense System (right).

The current defense portfolio is dominated by a number of expensive programs, most notably the F-35 Joint Strike Fighter (JSF) and Ballistic Missile Defense System (BMDS) programs, which are currently troubled by frequent cost and schedule overruns. The F-35 JSF program, which aims to develop a new class of fighter aircrafts with enhanced ground attack, reconnaissance and stealth capabilities, was initially estimated to cost $219.9 billion at its point of inception and span 116 months for the production of 2866 aircrafts. However, its total cost has now grown to $336.1 billion and only 2457 aircrafts have been produced (GAO, 2013). With the inflated unit cost of an

aircraft, stakeholders in the JSF program have significantly reduced buying power. Similarly, the BMDS program is experiencing immense cost growths during its development of a complex system of systems including land-, sea-, and space-based sensors, interceptors and battle management systems (GAO, 2013). With more than $130 billion to be spent through 2017, more is needed to curb potential cost growth for such critical programs and prevent their cancellations or reduction in capabilities.

Large-scale defense programs like the F-35 JSF and BMDS are classified within a separate class known as Major Defense Acquisition Programs (MDAPs). To control their cost growth and ensure the scheduled delivery of capabilities, Congress and policymakers have implemented a framework for consistently monitoring and assessing the economic feasibility of MDAPs. As such, each MDAP is legally obliged to submit a Selected Acquisition Report (SAR) to Congress annually and detail updated MDAP cost, schedule, and technical performance measures (DAG, 2013).

The cost growth of a program or any of its constituent systems can be calculated as the ratio of its corresponding cost estimate in the current SAR to that of a prior SAR. Based on accumulated SAR data, cost growth, along with other metrics, will be derived in an examination process known as the "Nunn-McCurdy" process (DoD, 2013). This is performed to determine the feasibility of a program for further development by assessing whether these metrics exceed *lower or higher thresholds*, or meet other criteria specified by Congress.

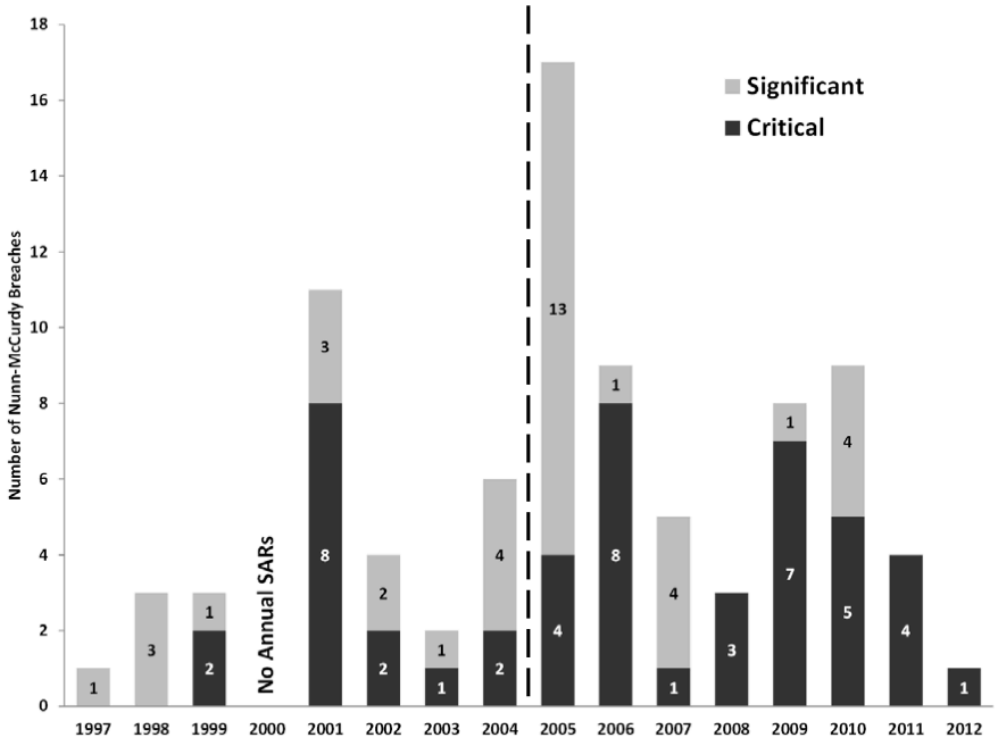


Figure 2-4: Number of significant and critical Nunn-McCurdy breaches from 1997 -2012 (DoD, 2013)

A program is economically unsound if it *breaches* the Nunn-McCurdy process. This breach can be *significant* or *critical* (DAG, 2013). A "significant" breach is a breach of the lower threshold, which can be a 30% cost growth from the original baseline or 15% from the current baseline reported in the prior SAR. A "critical" breach signifies an even higher cost growth that demands immediate cost control measures. This breach can be a result of either exceeding the upper threshold of 50% cost growth from the original baseline or 25% from the current baseline. The number of Nunn-McCurdy breaches since 1997 is illustrated in Figure 2-4, and it also includes programs that have been breached multiple times significantly or critically over consecutive years.

The large spike in 2005 to 13 significant and 4 critical breaches was due to more stringent reporting requirements for the Nunn-McCurdy process. From 1997 to 2012, there have been a total of 86 breaches, which constitute 31% of all MDAPs commencing from 1997 onwards that have experience cost overruns (DoD, 2013). Despite a decrease in the number of breaches in recent years, it is most likely due to the cancellation of programs and reduction capabilities, rather than a marked improvement in acquisition management.

Aggregating data for the entire 2012 defense portfolio, the GAO 2013 report showed that program costs for research and development, procurement and acquisition have escalated significantly. For research and development, there has been a 49% increase in cost since its first full estimate for the entire portfolio. *Total procurement cost and total acquisition cost have also increased by 35% and 38% respectively since then* (DoD, 2013). In addition to cost growth, schedule overruns were also rife. Majority of defense acquisition programs took far longer to develop capabilities than initially forecasted. The average change in delivering capabilities has been an increase in duration by 27 months, which is approximately a 37% increase over initial schedule estimates (DoD, 2013). *As such, a combination of cost and schedule overruns has been responsible for the number of breaches in lower and upper thresholds in recent years and even driven several MDAPs to the brink of cancellation.*

2.3.2.2 Shrinking Budgets

In addition to cost and schedule growth of weapons systems, shrinking defense budgets over the years may also render many acquisition programs unaffordable in future. Figure 2-5 illustrates the historical changes in total budget authority allocated to the defense industry from 1948 to 2012, as well as a forecast to beyond 2020. Owing to a series of caps on discretionary spending and sequester cuts, a 31% decrease in budget from its highest peak in 2010 is expected in the near future (DoD Comptroller, 2013). As the US military continues to operate under austere sociopolitical conditions, shrinking defense budgets, coupled with growing costs in weapons systems, cancelled programs and reduced production quantities, can severely degrade combat readiness levels. This motivates the maximized usage of shrinking defense dollar to obtain the most effective defense capabilities.

Recent annual budgets have shifted in scope and focus as they attempt to reduce acquisition costs, make better usage of resources and achieve better buying power (DoD Comptroller, 2013). The budgets of 2010 and 2011 were primarily the termination of weapons programs that experienced high cost and schedule overruns, while the budgets of 2012 and 2013 have shifted to refining defense business operations. These budgets aim to achieve more lean acquisition programs with

reduced overhead and support costs. Most significant among these refinements is the implementation of the BBP initiative (Carter, 2010a), which aims to restore affordability through pursuing greater efficiencies and responsiveness in acquisition.

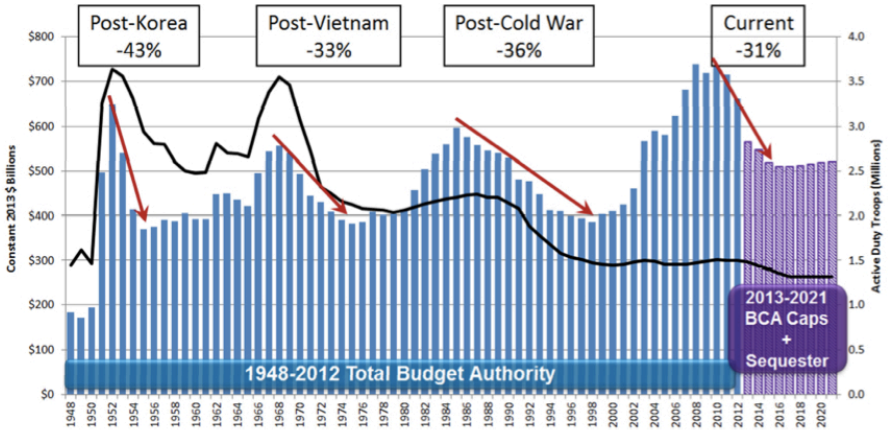


Figure 2-5: Historical changes in Total Defense Budget Authority from 1948-2012 and forecasted reduction from 2012-2020 (Plummer – Washington Post, 2013).

The relevance of the BBP initiative to the defense industry was described earlier. In the face of shrinking budgets, the BBP initiative offers guidance to the acquisition community for obtaining greater efficiency and productivity in defense spending. Apart from recommending strategy-driven changes in labor force structure and modernization, it emphasizes the more disciplined use of resources. Central to the streamlining of business operations and adhering to budget guidelines is the principle of *targeting affordability and controlling cost growth* in acquisition programs.

2.3.2.3 Difficulties in Cost and Schedule Estimation

Given how cost and schedule growths dominate the current context for developing defense and aerospace systems, estimating cost and schedule to the best possible accuracy is critical in the process of designing for affordability as they greatly determine the development status of defense and aerospace programs. These estimates help support decisions such as allocating the annual budget to various programs, commencing or terminating a particular program development, evaluating resource requirements at key decision points, and developing performance measurement baselines. Having a realistic estimate of projected costs and schedule during early-phase engineering design facilitates effective resource allocation and it increases the probability of a program's success.

Cost and schedule estimation is closely related to the design for affordability. *Whether a program is affordable or not depends on the quality of its cost and schedule estimate.* Affordability analysis can hence validate whether a program's acquisition strategy has an adequate budget for its planned resources. In addition, decision makers should also consider affordability at each decision point during a program lifecycle. It is important to know the program cost and schedule at particular intervals during the acquisition process in order to ensure that adequate funding is available to execute the program according to plan. As such, cost and schedule estimation are critical activities to affordability analysis.

However, developing reliable cost and schedule estimates is difficult. Recent failures in program management highlight that programs cost more and run longer than expected and deliver results that do not satisfy all requirements. One of the reasons for this predicament is the *use of unfounded assumptions about technologies* during the cost and schedule estimation process. This can be illustrated by an example in the military space sector, where the GAO reviewed six DoD space system acquisition programs. In five of the six programs, officials and cost estimators assumed critical technologies would be mature and available during cost and schedule estimation. This assumption was made and program development commenced despite an incomplete understanding of how long the programs would run or how much it would cost to ensure that the technologies could work as intended.

After the programs began, and as their development continued, the technology issues ended up being more complex than initially believed. This would eventually result in the cost and schedule overruns currently experienced. An example is the National Polar-orbiting Operational Satellite System (NPOESS), to which the DoD and the US Department of Commerce committed funds for developing and producing satellites before the technology was mature. When the program ran into financial difficulties, it was found that only 1 of its 14 critical technologies was mature at program initiation (GAO, 2013). The availability of mature technology in the future is just one of the many assumptions made during cost estimation. Many other assumptions are made during the estimation process, and they are often poorly defined with no supporting documentation for validation and verification.

Apart from unfounded assumptions about technology, another obstacle to cost estimation is the *lack of quality in historical databases to develop reliable cost and schedule estimating relationship*. Due to complexity of program development, it is often impossible to collect all the data needed to develop quantitative relationships that can better predict future cost and schedule. This problem is further compounded when the industry often relies on individuals without proper cost analysis skills to perform estimates in order meet a pressing need. In addition, limited budget and time during program formulation can constrain participation in the cost and schedule estimation process, thereby reducing the accuracy at which trade-offs, sensitivity, and even uncertainty analyses are performed.

Many cost estimating challenges can also be traced to *over-optimism*. Many defense and aerospace programs have suffered immense difficulty because the organizations architecting these programs have too often encouraged goals that are unattainable. This is mainly because organizations have been overly fixated on the benefits and not managing the risks properly. Through making more unfounded assumptions, they have unintentionally created more risks and continue strengthening a myopic belief that their programs will proceed successfully. The best way to combat such optimism is to *build more risk into plans during early-phase design*. This ensures that the organizations involved are aware of possible changes in scope, schedule delays, or other elements of risk. Another way that can be taken counter this optimism is to adopt the "honest broker" approach, which requires external organizations to address and understand the actual risks a program faces.

GAO (2009) also states that *program stability* presents another challenge to cost and schedule estimation. If the contractor has knowledge of the program budget, the contractor is pressured into presenting a cost estimate that fits the budget instead of providing a realistic estimate. Such

budget decisions can drive program schedules and procurement quantities. If development funding is reduced, the schedule can stretch and costs can increase. Also, a reduction in production funding can decrease the number of quantities of the system to be produced. Applying economic principles, unit procurement costs will increase as a result.

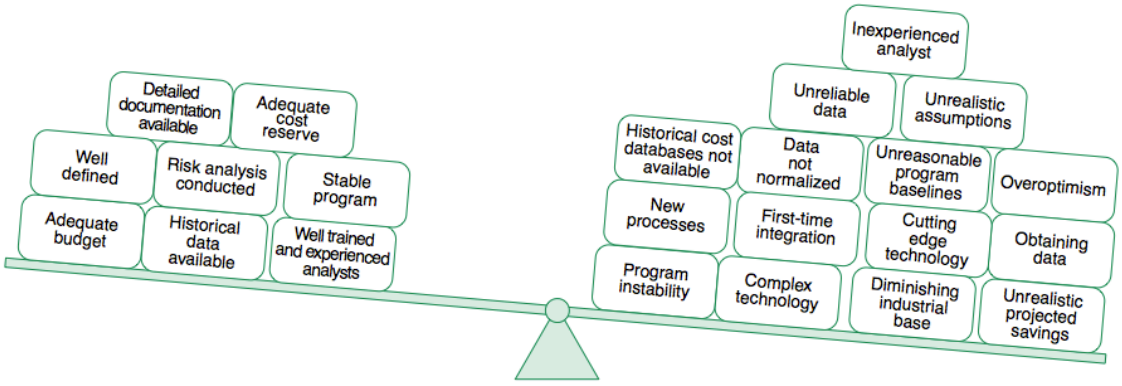


Figure 2-6: Challenges in cost and schedule estimation and ways to mitigate them (GAO, 2009).

According to GAO (2009), developing a good cost estimate requires stable program requirements, access to detailed documentation and historical data, well-trained and experienced cost analysts, a risk and uncertainty analysis, the identification of a range of confidence levels, and adequate contingency and management reserves. Figure 2-6 summarizes all the challenges described and some of the ways to mitigate them. Such is the uncertain nature of cost and schedule estimation that it is almost impossible to get truly predictive values. However, the best that can be possibly done during affordability analysis is to get *reasonable* estimates of program cost and schedule.

In addition to more reliable cost and schedule estimates, GAO (2009) also recommends conducting an *affordability assessment* to address program requirements and uncertainties throughout its lifecycle. This can give parameter estimates that help decision makers understand that *not all programs require the same type of funding profile*. Defense and aerospace programs can differ in terms of component commodities that require various outlays of funding and are affected by different cost drivers (GAO, 2009).

As such, some programs may cost less for research and development, but require a lot more resources for operations, maintenance and support. GAO (2009) recommends the use of stacked area charts or sand charts that plot program funding and expenditures together in order to give decision makers a high-level analysis of the portfolio and the resources they will need in future. This can be illustrated using Figure 2-7, which shows seven programs labeled A to G plotted against time, with the corresponding amount of resources needed to support their goals.

In Figure 2-7, it can be seen that funding needs are relatively stable in fiscal years 1–12, but there arises an increasing need for additional funding from fiscal year 12 to fiscal year 16. This is referred to as a *bow-wave*, which signifies an impending spike in the requirement for additional funds (GAO, 2009). Whether these funds will be available will determine which programs remain within the portfolio. Using such a chart can facilitate the conduct of an affordability assessment at the agency level and not at individual program level. This enables a more holistic assessment of the entire portfolio and should become a critical component of affordability assessments.

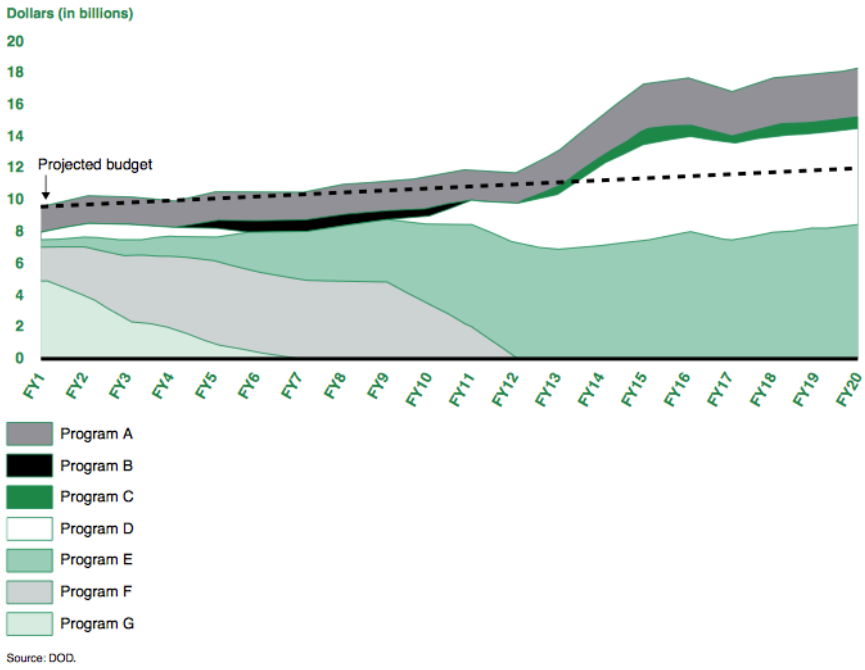


Figure 2-7: DoD-recommended affordability assessment method (GAO, 2009)

GAO (2009) also recommended a 12-step estimating process to ensure that decisions made are based on credible cost estimates. This 12-step process addresses best practices, starting by defining the program's purpose, developing the estimating plan, defining the program's characteristics, determining the estimating approach, identifying ground rules and assumptions, obtaining data, developing the point estimate, conducting sensitivity analysis, performing a risk or uncertainty analysis, documenting the estimate, presenting it to management for approval, and updating it to reflect actual costs and changes. The same process can be used for schedule estimation. As such, following this process can potentially ensure that realistic cost and schedule estimates are developed and help decision makers determine whether the program is affordable within the portfolio plan.

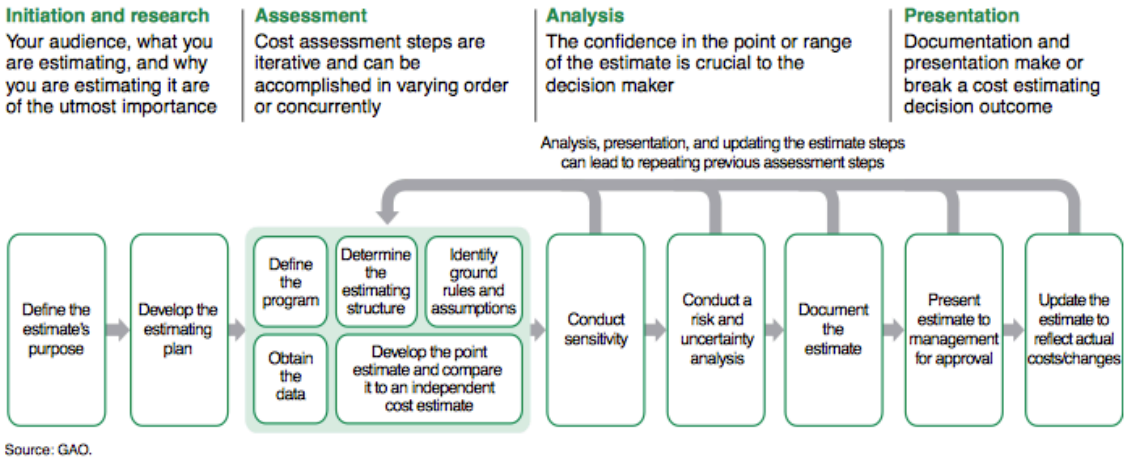


Figure 2-8: Recommended Cost Estimating Process (GAO, 2009)

Rising cost and schedule growths, shrinking budgets, and uncertainties in estimation and mitigation techniques have characterized the context the defense and aerospace industry is currently operating in. Cost and schedule overruns have contributed greatly to recent failures in program management, necessitating the need to place more emphasis on affordability considerations during the early phases of engineering design. It is within the very same context that designing for affordability has to be conducted. While this may prove to be a very challenging endeavor, industry players and the government have not turned a blind eye to the ailing acquisition process. Many remedying frameworks apart from more holistic cost and schedule estimating processes have been implemented to curb this unhealthy trend. More methods for affordability assessment are necessary in order to obtain insights that help decision makers determine whether a program remains affordable within a portfolio or not.

2.4 Current Ways of Understanding Affordability

Since affordability can be aligned to the goal of engineering design, the integration of affordability studies and engineering design methods can be known as the paradigm of *designing for affordability*. This has been growing of importance recently owing to rising acquisition costs in the defense and aerospace industry and the need to "*mandate affordability as a requirement*". Over the years, the DoD, NASA and various engineering groups have attempted to integrate affordability considerations of time and cost with systems design methods to reduce cost and schedule overruns in their acquisition programs. This has galvanized the discipline of affordability studies as an imperative element in systems architecting within the defense and aerospace industry. Affordability studies have since been conducted under different tenets and quantitative frameworks to varying degrees of effectiveness. Therefore, this section explores the current depth of affordability studies through a collection of definitions as well as describing the prominent frameworks that have been applied to better include cost and schedule considerations during engineering design.

2.4.1 Working Definitions

Different people and organizations have studied affordability over the years, and many definitions have been proposed to facilitate the understanding of cost, schedule and performance parameters across all industries. Table 1 is an extensive collection of definitions found in a variety of books, journals and publications. Some definitions are more general, while some are more industry specific. Evidenced by these definitions, apart from the words "cost", "schedule" and "performance", the common themes surrounding affordability include *value*, *systems engineering*, *attributes*, *constraints*, *mission needs*, *risks*, *lifecycle*, *budget* and *strategy*. A comprehensive affordability assessment is thus expected to encompass all these themes and facilitate decision-making based on cost, schedule and performance attributes.

Source (Year)	Definitions
Borky, Bauhaus Jr and Brown (1998)	Affordability [is the] ability of an alternative to fit into an executable program within reasonable budget projections.
Mavris and DeLaurentis (1998)	Affordability may be viewed as a measure of value balancing the product's effectiveness [including capability (performance), reliability, maintainability, safety, and other such system attributes] against its associated [lifecycle] cost and risk, for a given schedule.
Redman and Stratton (2001)	Affordability is that characteristic of a product or service that enables a customer to • Procure it when they need it • Use it to meet their performance requirements at a level of quality that they demand • Use it whenever they need it over the expected life span of the product or service • Procure it for a reasonable cost that falls within their budget for all needed products or services • Components shall plan programs consistent with the DOD Strategic Plan, and based on realistic projections of likely funding available in the future years. From the contractor's point of view, affordability is that characteristic of a product or services that: • Makes it available when the customer initially needs it • Enables it to meet customers' performance requirements at a level of quality they demand • Makes it available whenever customers need it during its expected life span (life-cycle) • Allow customers to fit it into their budget for all competing products or service.
Bever and Collofello (2002)	The Office of Naval Research (OM) has further defined the definition of affordability as a characteristic of a product or service which enables consumers to purchase a product when needed, use the product to meet their performance and quality demands, use anytime over the life span of the product, and at the same time, purchase a product at a reasonable cost.
Emmons (2010)	For NASA's purpose, Program Affordability equates to developing a long-term strategy that will meet critical objectives while remaining within NASA's budget.
Mallory (2011)	"...Affordability is a Systems Engineering process used during all phases of the product life cycle where cost is balanced with performance and schedule to define and deliver best value solutions to the customer."
Carter (2010)	The 2010 Carter memorandum defines affordability as "conducting a program at a cost constrained by the maximum resources the Department can allocate for that capability".

Source (Year)	Definitions
Defense Acquisition Guidebook (2013)	The Defense Acquisition Guidebook defines affordability as "the degree to which the life-cycle cost of an acquisition program is in consonance with the long-range modernization, force structure and manpower plans of the individual DoD Components, as well as for the Department as a whole"
Herald (2011)	Affordability is not so much a single definition as: A trade space of domain-relevant affordability attributes Stakeholder-relevant Value metrics for each attribute
INCOSE Affordability Working Group (2011)	Affordability is the balance of system performance, cost and schedule constraints over the system life while satisfying mission needs in concert with strategic investment and organizational needs. Design for affordability is the systems engineering practice of balancing system performance and risk with cost and schedule constraints over the system life satisfying system operational needs in concert with strategic investment and evolving stakeholder value.
NDIA Affordability Working Group (2011)	Affordability is the practice of ensuring program success through the balancing of system performance (KPPs), total ownership cost, and schedule constraints while satisfying mission needs in concert with long-range investment, and force structure plans of the DOD
MITRE (2012)	Affordability: Ability to fund desired investment. Solutions are affordable if they can be deployed in sufficient quantity to meet mission needs within the (likely) available budget.

Table 2-1: Collection of definitions of Affordability

2.4.2 The Affordability Triangle

Due to the variety of themes related to affordability, there are many different lenses through which one can understand its relationship and impact on engineering design. One way of understanding affordability is through the Affordability Triangle shown in Figure 2-9. Conceptualized by Tuttle and Bobinis (2012), it depicts the relationship among Capabilities, Performance, Schedule and Budget. The triangle shows capabilities form the baseline of any acquisition process and it is important to first establish the military need and identify how it fits within the existing defense portfolio. After determining the required capability, the affordability decision criteria are then based on the secondary elements of Performance, Budget and Schedule.

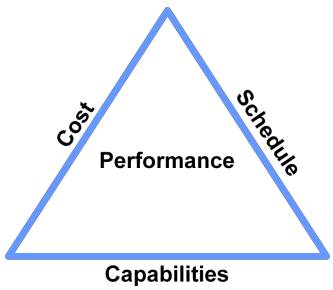


Figure 2-9: The Affordability Triangle (Tuttle and Bobinis, 2012).

These elements form the main components of affordability and establishing a framework based on them ensures its compliance with common definitions listed in Section 2.4.1. The balance of performance, budget and schedule considerations thus constitutes a standard engineering trade study. To transform this process to a system affordability trade study, Tuttle and Bobinis (2012) recommends *the extension of the system's time horizon, as well as the inclusion of all cost elements and program increments*. This is a clear indication that *time is a critical component to a complete affordability assessment*. Based on current definitions of affordability and the elements contained within the triangle, they proposed the extraction of the following affordability components:

1) **Required Capabilities**
   a) This is the first step of the affordability trade study and it begins with the identification of the required capabilities and the time phasing for inclusion of the capabilities. Time phasing means determining when to include certain capabilities and when to retire them during the life cycle.
2) **Required Performance**
   a) The next step is to identify and specify the required Measures of Effectiveness (MOEs) for each of the capabilities. These MOEs are measurement *attributes* commonly used to provide quantifiable benchmarks against which the system concept and implementation can be compared.
   b) Define the time phasing for achieving the MOEs.
   c) After the MOEs, identify and specify Measures of Supportability (MOSs), which are measurement *attributes* used to determine the require amount of *resource* support to sustain system operations.
   d) Define *time phasing* for achieving the MOSs.
3) **Budget**
   a) Identify the budget elements to include in the affordability evaluation.
   b) Determine the *Time-phased* budget for each of the budget elements or as the total budget

This approach can be viable for affordability practice and its strengths lie in the elicitation of requirements, capabilities and performance, their quantification through measurement attributes, and the use of time phasing for both the budgets as well as capabilities.

Tuttle and Bobinis (2012) then illustrated the purpose of these affordability components through an example depicted in Figure 2-10. To begin, one or more of the affordability elements of capabilities, performance, schedule or budget is designated as the decision criteria that will be used to perform engineering trade studies or decision-making. The remaining affordability elements that are not designated as decision criteria will become specified *constraints*.

In this notional example, capabilities and schedule have been fixed as constants. This results in a relatively straightforward tradeoff between cost and performance, as either can become the decision criteria while the other becomes the constraint. The maximum budget and the minimum performance thresholds are identified and they are reflected as horizontal and vertical lines on the cost and performance axes respectively. Design solutions below the maximum budget line are considered affordable within the context of the definition provided by the 2010 Carter memorandum. Solutions to the right of the minimum performance line will at least satisfy all stakeholder requirements and they are considered as technically compliant solutions.

The green rectangle on the graph is formed by the two threshold lines and thus represents the region containing the affordable solutions that meet the minimum performance requirements and within the maximum budget.

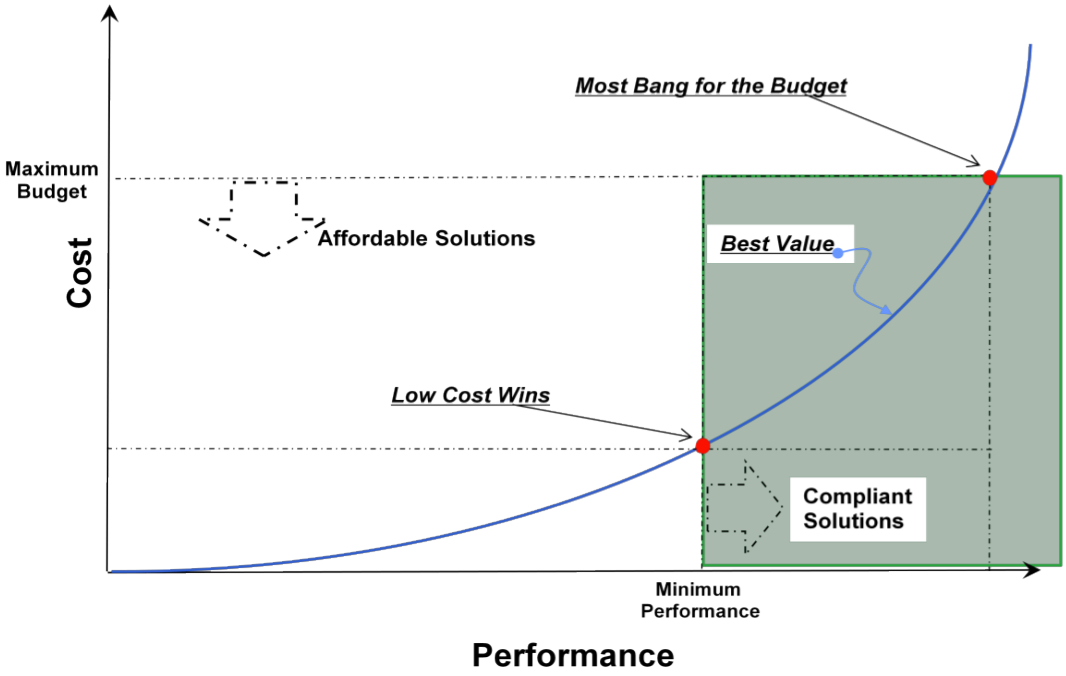


Figure 2-10: Determining affordable solution by trading off cost and performance after fixing capabilities and schedule as decision criteria (Tuttle and Bobinis, 2012).

The blue curve on the graph is another notional construct that connects all solutions that have the "best value". These solutions are considered *Pareto optimal*, as they provide the best possible performance achievable given the minimum cost. If cost is set as the decision criteria, the "Low Cost Wins" solution will be selected since it requires the lowest cost expenditure for meeting the minimum performance requirements. However, if the decision criterion is performance, the "Most Bang for the Budget" solution will be selected. The tradeoff for high performance is that the entire budget would be expended. As such, affordable and Pareto Optimal solutions in this notional example lie along the blue line in the green rectangle.

Through this example, Tuttle and Bobinis (2012) demonstrated that *designating the main affordability elements as either decision criteria or constraints could facilitate the identification of affordable solutions for a system or a program*. They also suggested that system or program affordability trade studies could be performed more accurately *if the budget is time-phased*. While the maximum budget in Figure 2-10 is shown as a single number, the budget can actually be divided and illustrated as an annual budget for a sequence of fiscal years (Tuttle and Bobinis, 2012). With a time-phased budget, it will be easier for system architects and stakeholders to identify the years during which the program is affordable or unaffordable. Similarly, *capabilities and performance can also be time phased*, and their requirements for a particular time frame may not be met in the context of insufficient budget.

To enable the conduct of system or program affordability trade studies, Tuttle and Bobinis (2012) also analyzed the lifecycle cost (LCC) of a typical program and *broke it down into different categories*, namely Research Development Testing and Evaluation (RDT&E) cost, Procurement cost, military construction (MILCON) cost, Operations and Maintenance cost, and finally military personnel (MILPERS) cost. Breaking down acquisition cost in this manner is equivalent to defining the Work Breakdown Structure (WBS), which is a common practice within the defense and aerospace industry. This composition is shown in Figure 2-11 and such a distinction made between different types of cost gives rise to the notion of '*different colors of money*' (Wiggins, 2009) since these categories have their unique purpose and usage.

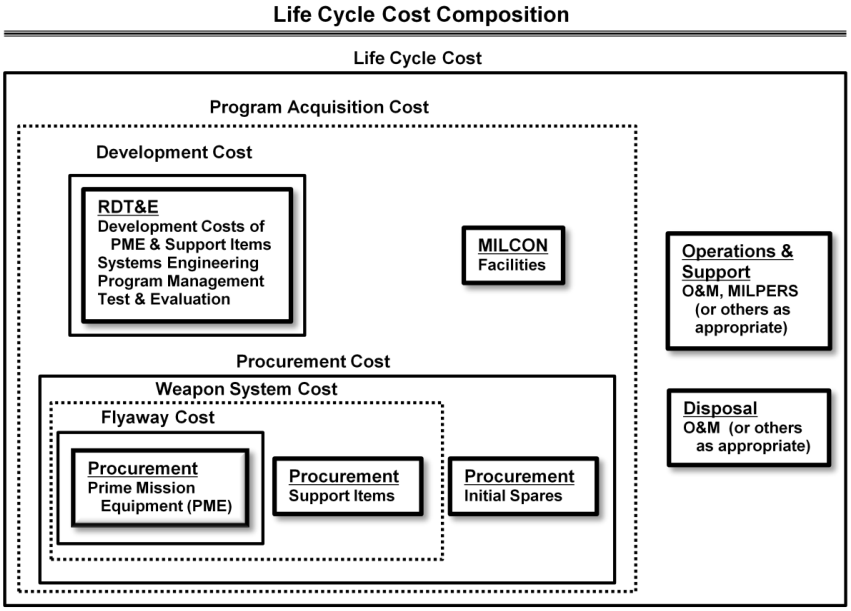


Figure 2-11: Lifecycle cost composition of a typical defense acquisition program (Tuttle and Bobinis, 2012).

With the integration of time phases, *the cost of system can be calculated as the sum of all colors of money across all time increments*. The equation for the system cost can be written as such:

$$System\ Cost\ B_i = \sum_{j=1}^{N} B_{i,j}$$

where

$B_{1,j} = RDT\&E\ Cost\ of\ Increment\ j$

$B_{2,j} = Procurement\ Cost\ of\ Increment\ j$

$B_{3,j} = Milcon\ Cost\ of\ Increment\ j$

$B_{4,j} = Operations\ \&\ Maintenance\ Cost\ of\ Increment\ j$

$B_{5,j} = Military\ Personnel\ Cost\ of\ Increment\ j$

$i = 1\ or\ 5$ for a single budget element or the total budget respectively

$N =$ number of program increments, depending on the number of time phases

This is similar to the approach formulated by consultancy firm Booz Allen Hamilton and illustrated in Figure 2-12, which aims to drive affordability considerations by breaking down total ownership costs (TOC) into their constituent costs (Booz Allen Hamilton, 2011). This is followed by a further breakdown to subsystem cost components, which are then plotted on a graph with three dimensions. The subsystems are first plotted based on their TOC and their 'ease of capture' in terms of room for further cost reduction. The third dimension is given by the size of the point denoting each component, which can represent a 'different color of money' such as Operations and Support cost. As such, areas with potential for cost reductions can be identified quickly and facilitates affordability considerations in the design process.

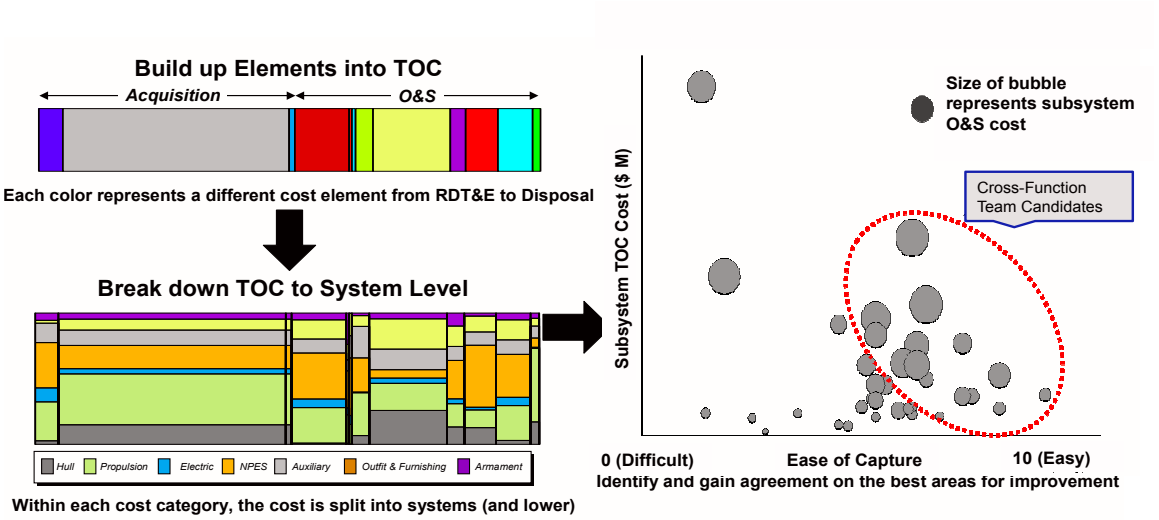


Figure 2-12: Breakdown of TOC into subsystem costs and identifying areas for improvement (Booz Allen Hamilton, 2011).

An understanding of the *cost breakdown* also helps the system architect to identify areas in which overhead and support costs can be reduced in order to mitigate overall program cost growth. This principle is reinforced by the Carter memorandum, which states, "the ability to understand and control future costs from a program's inception is critical to achieving affordability requirements" (Carter, 2010a). As such, breaking down acquisition costs into its constituents can help decision makers identify key areas that drive cost and design systems with the lowest overall cost possible.

2.4.3 Should-Cost Review and Statistical Methods

The DoD has also attempted to achieve major cost savings in acquisition through their own measures, which is the *Should-Cost/Will-Cost* framework that was implemented in accordance to the Carter Memorandum. It is also sometimes known as the *Should-Cost Review (SCR)*.

SCR is a tool targeted at program managers to control all costs throughout the lifecycle, and consistently aims to lower cost wherever and whenever possible. Setting a goal of "*how much a program should cost*" helps to set a target cost at program inception and reiterate the need to keep cost below the target. Through prudent cost control, SCR can constraint stakeholder requirements to more realistic boundaries and ensure that the program unit cost and sustainment cost of the final product are within budget.

Cost estimates from SCR have leverage on lowering the costs of contracts and incentives when the DoD collaborates with the defense industry. Based on Figure 2-13, Should-Cost motivates program managers to have a detailed understanding of cost drivers across the value chain, establish cost elements from the subsystem, monitor expenditures for audit readiness, and encourages the conduct of lean business operations (AT Kearney, 2011). As such, SCR drives the DoD to find all ways possible in the management process to lower cost figures below traditional metrics such as independent Will-Cost estimates, which are often based historical data and inflated due to uncertainty and changing requirements. Avenues for lowering costs usually include the reduction of program overhead and unproductive organizational processes. Since cost reductions are performed from a management perspective, it does not compromise the rigorous design process and systems engineering practice typically demanded of defense systems or programs.

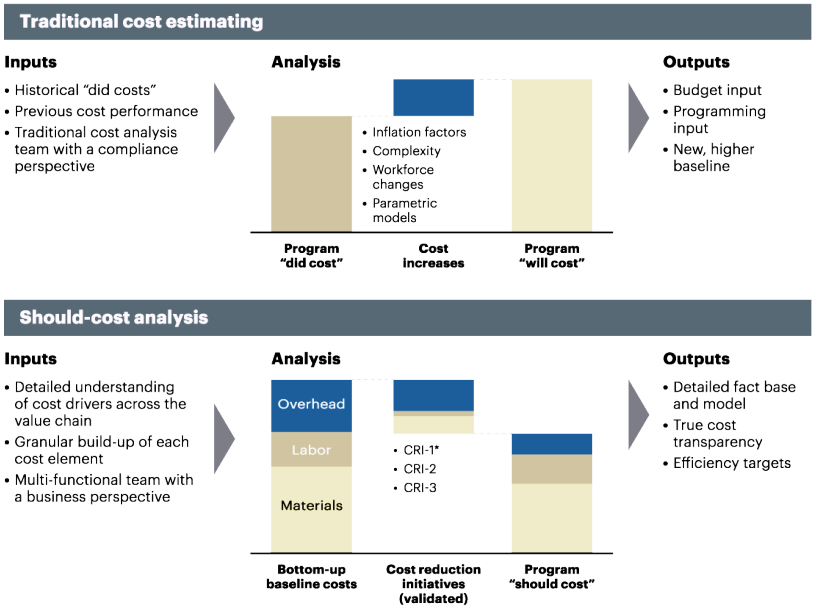


Figure 2-13: The difference between should-cost and traditional cost estimating methods (AT Kearney, 2011).

Apart from SCR, the DoD has also applied a multitude of quantitative methods to several affordability case studies to complement their decision-making process. Some of these methods include *interval cost estimation* and *system lifecycle analysis*, which were used to perform affordability analysis on a number of programs owned by the Defense Advanced Research Projects Agency (DARPA) (Kroshl and Pandolfini, 2000). Traditionally, cost estimates have been obtained by extrapolating historical data according to parameters such as technical complexity or concept maturity. However, as discussed earlier in the chapter, the quality of historical data is often questionable. Such estimates are often flawed and far from actual figures because they fail to capture uncertainty. Therefore, interval cost estimation can circumvent this problem by deriving cost estimates with associated probabilities. This is illustrated in Figure 2-14, which shows a distribution of average unit flyaway price (AUFP) for a hypersonic cruise missile, as well as its probability of meeting or exceeding a price.

This analysis is conducted by first breaking down the total cost into their constituent costs or "different colors of money" (Wiggins, 2009), followed by developing optimistic, pessimistic and most likely cost estimates based on expert opinion or other relevant parameters. Each cost element is then associated with a probability distribution, which may be modified to reflect uncertainty in cost estimates. Finally, all constituent cost with their assigned probability distributions are aggregated to obtain summary statistics and probability distribution for the total cost. As such, interval cost estimation helps to build up a stochastic cost model that captures uncertainty in a systematic and traceable manner. Results from this process include a probability curve that illustrates the probability of a cost metric meeting or exceeding a price established in the affordability requirement, as well as a distribution of probabilistic cost estimates bounded by pessimistic and optimistic thresholds. This helps system architects and stakeholders to derive realistic cost estimates and determine if the program of interest remains within the affordability requirement.

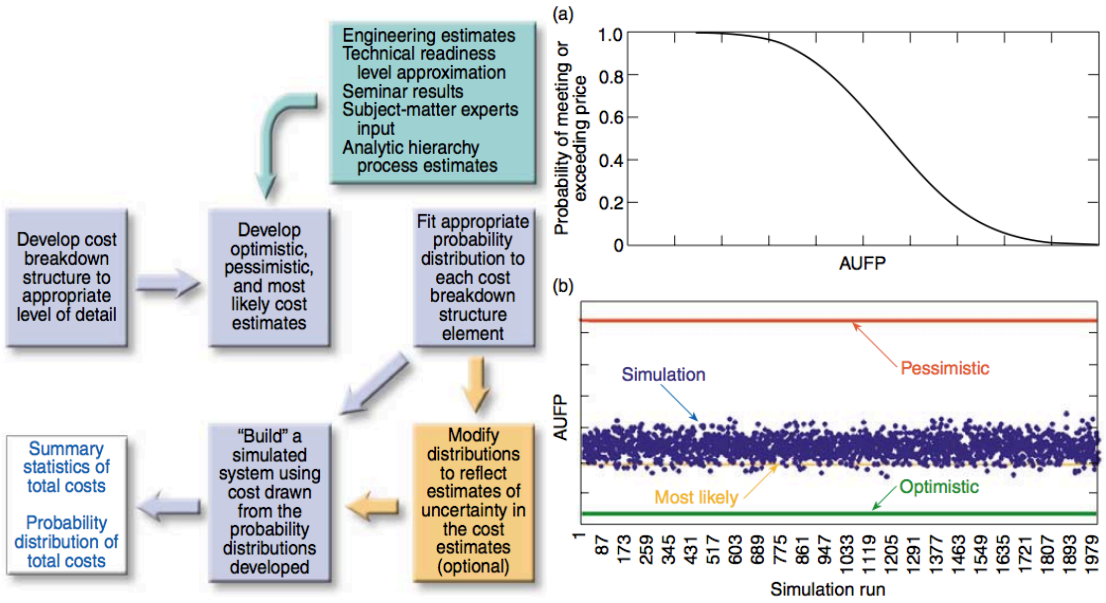


Figure 2-14: Interval cost estimation process (Blue indicates main path, Orange indicates optional elements, Green indicates data sources) and its generated results (Kroshl and Pandolfini, 2000).

System lifecycle analysis is much like the process described by Tuttle and Bobinis (2012), where cost or performance can be set as the decision criteria while the other is left as the constraint. However, system lifecycle analysis uses additional metrics used drawn from engineering economics and financial analysis, and they include net present value (NPV), internal rate of interest (IRR) and learning curve functions. The calculation of these metrics can provide decision makers with useful information during early-phase design to determine the lifecycle cost of a program through different perspectives. These metrics also account for the time value of money, which allows decision makers to consider cost depreciation with each fiscal year and also be able to account for economic inflation.

2.4.4 Sand Chart Tool

The defense sector is not alone in conducting affordability analysis through engineering methods and quantitative methods, as big players in the aerospace sector have come up with their own techniques for cost and schedule accountability in their design process. For example, The Aerospace Corporation has developed its own affordability methodology and NASA uses the *Sand Chart Tool* (SCT) to support their economic assessments of programs and portfolios (Emmons, 2010). The SCT, by Emmons (2010), leverages on the method employed by The Aerospace Corporation that is shown in Figure 2-15.

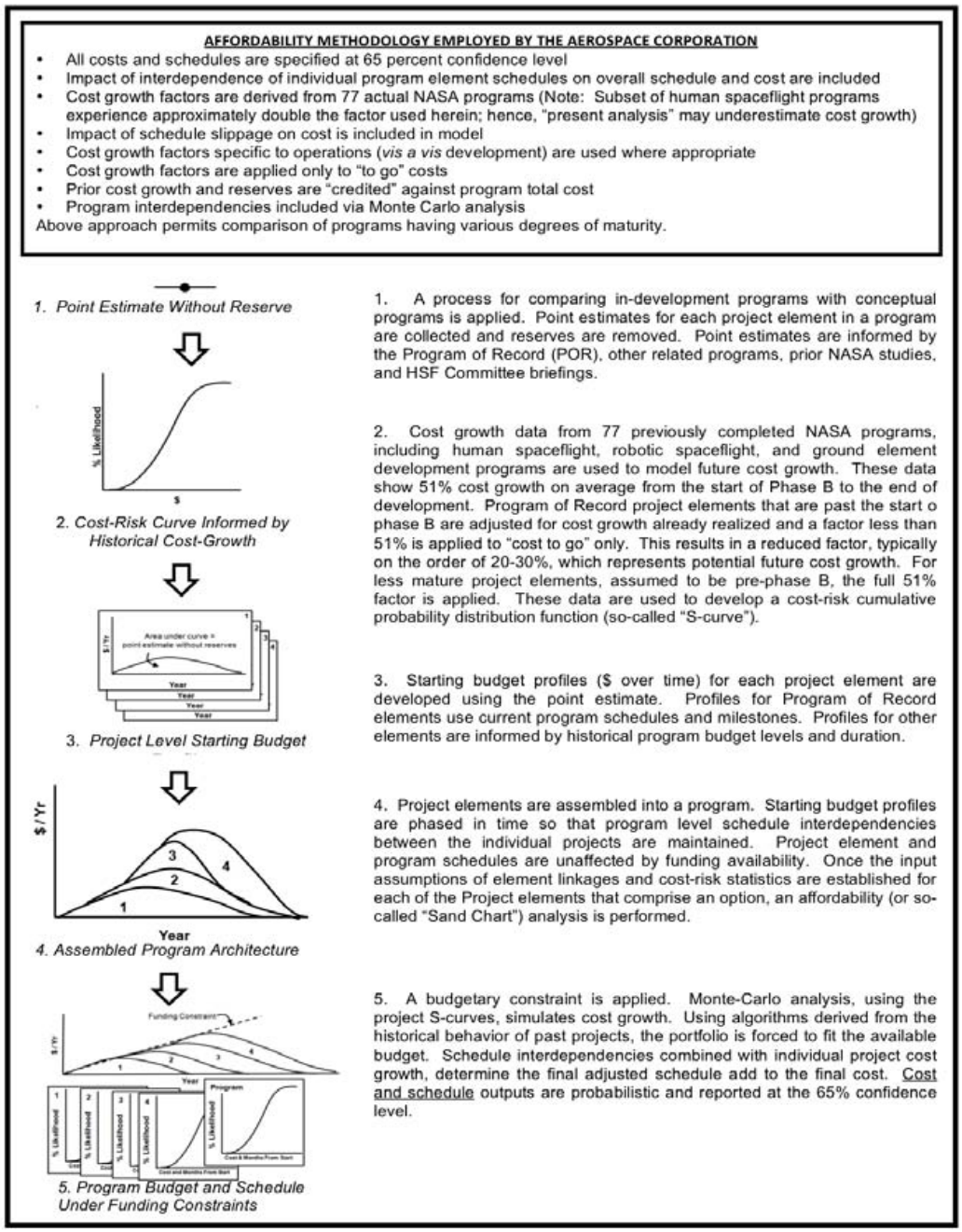


Figure 2-15: Affordability methodology used by the Aerospace Corporation (Emmons, 2010)

In The Aerospace Corporation affordability methodology, the process begins with developing point estimates and cost-risk curves for each project element, followed by a profiling of their corresponding budgets available. The project elements are then assembled into a program and phased in time so that schedule interdependencies are maintained. Affordability analysis can then commence through the use of Sand Charts, which demonstrate how programs can be realized based on program cost, schedule, historical risk using S-curves and budget considerations (Emmons, 2010). Incorporating probabilistic estimates, the sand chart shown in Figure 2-16 presents the change in total cost for constituent systems and 'different colors of money' over two decades for a notional NASA development portfolio. A probabilistic analysis of program cost and schedule can help determine the robustness of a program plan after accounting for cost-growth risks, schedule linkages and other factors. Apart from providing an investment strategy to stay within budget at a specified confidence level, it also provides recommendations as to when systems should commence development or be retired. Two budget lines colored in black and red are also shown and they represent the FY2010 budget and a theoretical "Less Constrained" budget respectively.

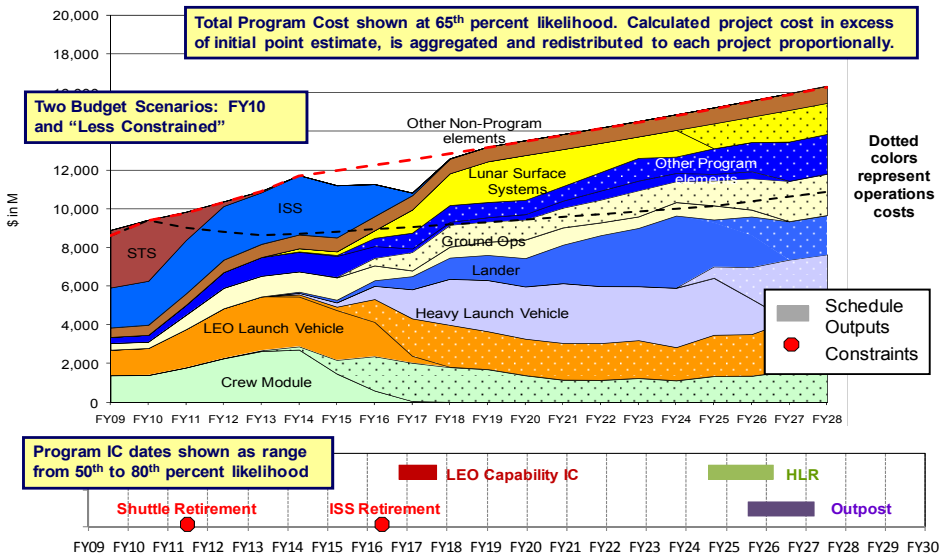


Figure 2-16: Sand chart as affordability analysis results from a notional NASA program (Emmons, 2010).

The sand charts used in this methodology function as convenient visualization tools, and they can facilitate design planning with affordability considerations. In addition, "penalties" can be applied in the analysis of the same notional example in order to reduce expenditure in fiscal years that exceed budget allocations (Emmons, 2010). This is shown in Figure 2-17 and these penalties may range from delaying a start of project to delaying project phase transitions to reduced funding. Incorporation of these penalties will then adjust the program cost to fit within the entire duration of the budget. By performing probabilistic estimates of cost, schedule and performance over time, affordability analysis using The Aerospace Corporation methodology and SCT can help stakeholders implement a feasible and affordable program that satisfy all requirements within an acceptable time period. This may help provide a better understanding of the relationship between the risk of cost growth and available budget. However, it can be seen that much data is required for this method, and it can be both time consuming and computationally demanding.

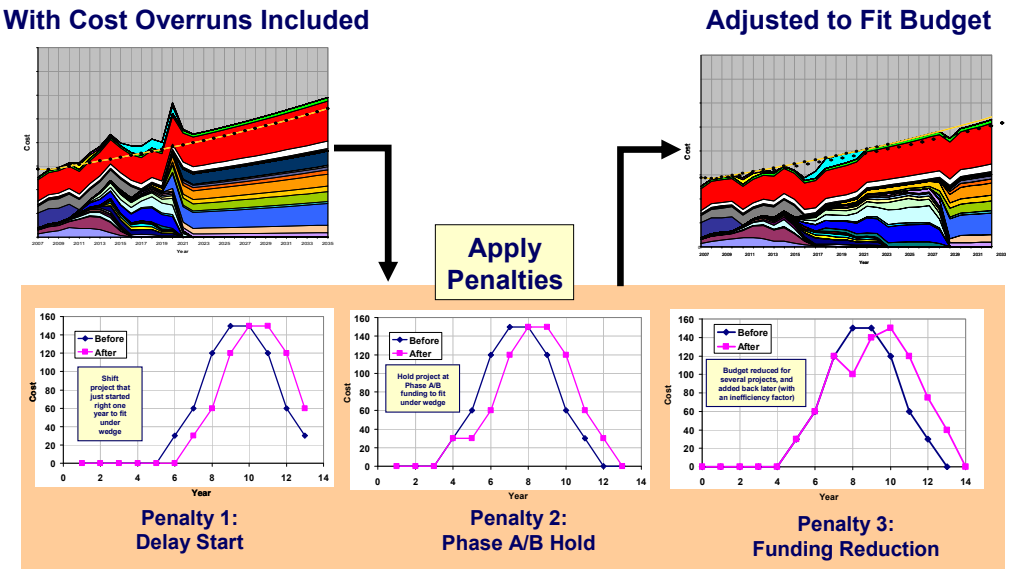


Figure 2-17: "Penalties" are applied to fit costs to budget in the Aerospace Corporation methodology and Sand Chart Tool (Emmons, 2010).

2.4.5 Cost and Time as Independent Variables

Other schools of thought that aim at acquisition reform also exist in the defense and aerospace industry and they should not be overlooked. The most prominent of which is the establishment of *Cost As an Independent Variable (CAIV)*, which emphasizes cost or unit price as the only constant (Higgins, 1997). Cost, schedule and performance have traditionally been the main variables of a program. Although a program can in theory be managed by allowing three parameters to vary in response to program dynamics, it requires a very elaborate management scheme and often yields poor results. Establishing one variable as a constant or independent variable thus allows greater control of the program through manipulation of the other two variables. Therefore, CAIV establishes the affordable price for a system or program and trades off either performance or schedule to meet that price. However, the tradeoffs should not compromise specified warfighter requirements and the system or program should still be delivered within a reasonable timeframe.

If performance were set as the constant, stakeholders would specify many low-level requirements and this would make the design process of a system or program very slow and costly. CAIV eliminates such a scenario by declaring that system requirements be stated in few broad, top-level terms. Given a set of broad requirements, any number of designs can actually be proposed to meet this need. This will promote the generation of many possible solutions that can achieve a specified function and prevents stakeholders from being fixated on a single point design. On top of performing engineering tradeoffs between performance and cost, CAIV also drives increased collaboration among program managers to reduce the need for costly changes in future and reduce program overhead cost. It also promotes the use of incentives to encourage contractors to come up with better value designs.

CAIV has been successfully applied in the corporate sector for decades and its transference to defense applications was much anticipated. CAIV emphasizes that lower cost designs did not necessarily equate to lower quality designs and allows cost reduction at a program level without significant depreciation of the system's value. In fact, it has been reported in the commercial world that lower cost designs are usually simpler and easier to produce. They also provide better performance and reliability since designers are forced to propose creative designs with the greatest capability for a given cost (Higgins, 1997).

The other competing paradigm is *Time As an Independent Variable (TAIV)* and it sets time or schedule as the constant instead (Patterson, 2013). Proposed and trademarked by Patterson (2013) TAIV is a much newer construct and it establishes time as a structured way to determine the limits or boundaries of acquisition programs. While traditional time-based approaches motivates finding of new ways to reduce time spent on activities in progress, TAIV determines at the very beginning what performance or capability is possible based on when the system is required to be operational. The program is then driven towards this boundary and system architects strive to achieve the best possible performance of a system or program within that fixed time frame. The process of applying TAIV is shown in Figure 2-18.

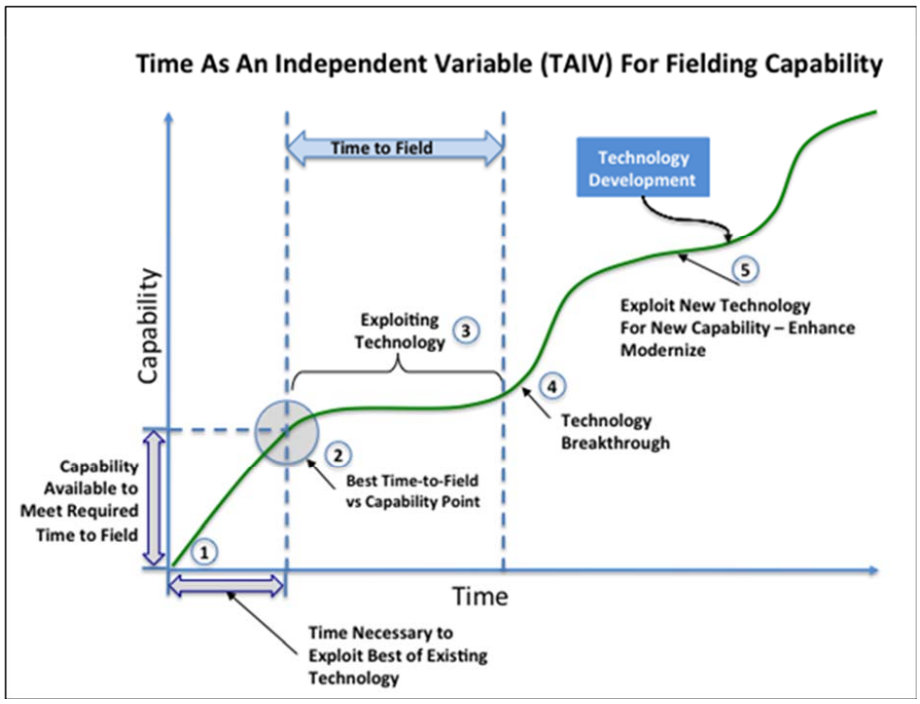


Figure 2-18: Time As an Independent Variable (TAIV) for Fielding Capability (Patterson, 2013).

Patterson uses Figure 2-18 to describe how TAIV can be used for fielding a particular capability. The green line traces the phases that a technology undergoes as it matures over time. At Point 1, there is first a necessity to assess the maturity and relevance of the particular technology that provides a warfighting capability to a system or program of interest. As time elapses from Points 1 to 2, new technology may be developed or adapted from elsewhere, or mature technology can be exploited and transformed into a new capability. Point 2 is thus the best time-to-field as there is the greatest increase in capability for time elapsed.

It is at this point that the maximum amount of capability for the technology available is realized. The time between Points 2 and 3 is when the technology is being used to great effect. During this period, it is not recommended to develop another technology since it does not provide significantly greater capability. Only until there is a new technology breakthrough at Point 4 is it viable to exploit the new technology for enhanced capabilities. As such, the TAIV process demands the fielding of a capability when the underlying technology has its greatest value. New technologies will emerge and TAIV spurs their usage at times when they have the highest values to the system or program.

Patterson (2013) states that the key benefit of TAIV is that it can reveal *tradespace* dimensions as it "reveals the amount of time necessary to meet a required fielding data with the most capability." By setting threshold, best and objective values for desired capability or performance, the times taken to achieve these 3 values can be obtained as shown in Figure 2-19. Therefore, the trade metric in this case is time and the *tradespace* is bounded between the time taken to achieve the threshold value and the latest delivery date. The delivery date is the time taken to arrive at Point 2. Through the application of the *tradespace* concept, TAIV can potentially be used to determine when best to field a capability with a particular technology.

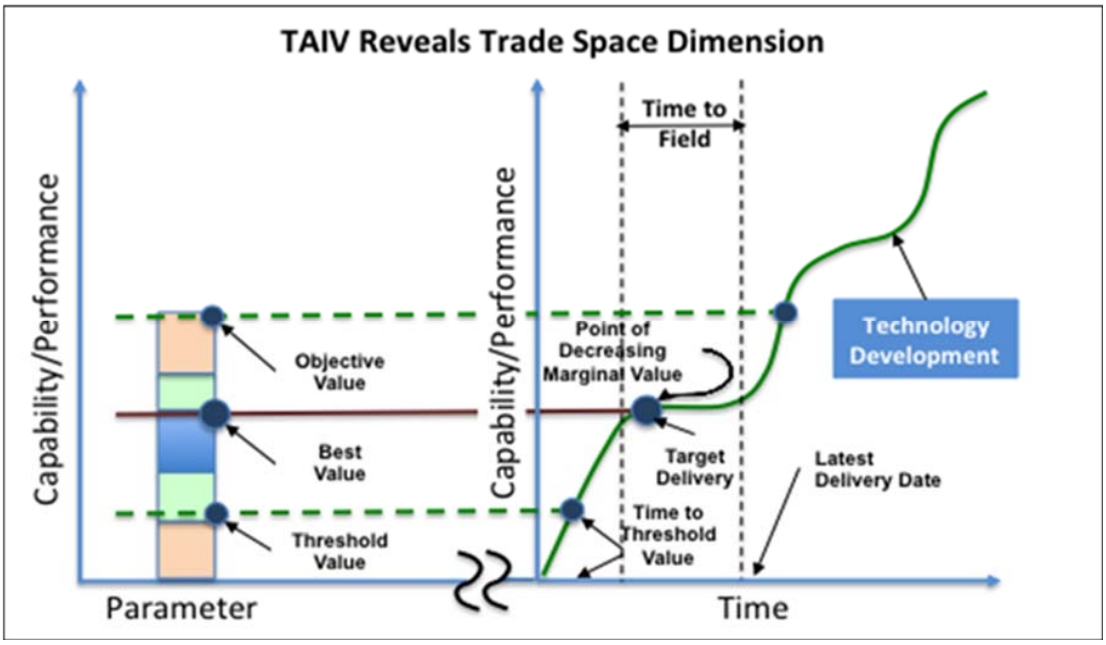


Figure 2-19: TAIV reveals trade space dimensions (Patterson, 2013).

2.4.6 Effective Portfolio Management

One of the most recent analytical frameworks for enabling better-informed decisions in affordability analysis is the use of *robust portfolio optimization* and a *multi-period portfolio management approach*. This framework, established by Davendralingam and DeLaurentis (2013), is built upon principles from financial engineering and operations research. A robust multi-period portfolio management approach to decision-making can ensure the balance of performance of a "portfolio" of systems against potential risks. As mentioned earlier in this chapter, an instance of such a "portfolio" can be a SoS.

Taking a *multi-period* portfolio approach means that time is taken into design consideration, where a SoS can potentially evolve after each period to achieve a different level of required capability. At each period, a capability and risk analysis for the SoS is conducted, thereby providing decision makers valuable information required to develop, implement and integrate the SoS portfolio in order to evolve to the next level. This translates to identifying actions that balance potential gains in SoS capabilities against risks such as cost and schedule growth over a specified time horizon. Figure 2-20 is an abstraction of the evolution of a "portfolio of systems" that constitutes a SoS, as part of the wave model (Dahmann et al., 2011).

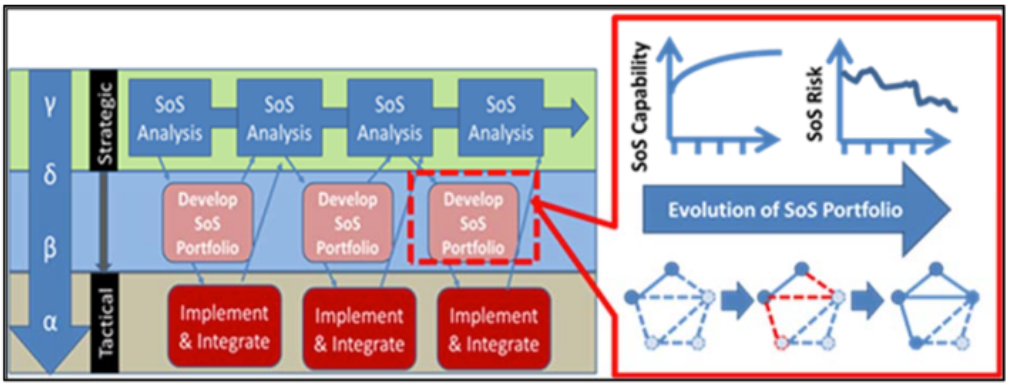


Figure 2-20: Wave Model Relation to Portfolio Evaluation (Davendralingam and DeLaurentis, 2013)

According to Davendralingam and DeLaurentis (2013), the wave model is an extension of DoD guidelines on systems engineering for a SoS that translates SoS systems engineering core elements, interrelationships, and decision-making artifacts to a time-sequenced model representation (Dahmann et al., 2011). This method is thus aligned to specifications within the defense industry. Based on the results from analyzing risk and capability levels, decision makers can then *explore the tradespace for all possible design options* across multiple time periods and determine the modifications and assets required to evolve the SoS forward across successive periods. To identify optimal "portfolios" of systems to be acquired in pursuit of desired SoS capabilities for each period, *robust optimization methods* are used to support these SoS-level acquisition decisions. This ensures that rigorous quantitative analysis is conducted to determine the sequential acquisitions needed to propagate required capabilities while minimizing operational and developmental risks.

Decisions taken in designing the evolution of a SoS can be the *sequential acquisition (and removal)*. As such, decisions made at each time period affect the decision options of future time periods, thus affecting long term performance and risks of the SoS. Davendralingam and DeLaurentis (2013) state that the translation of these sequential decisions to the context of a multi-period investment model requires an adequate description of node (system) *attributes*. Optimization methods are then applied to each period, and the analysis now aims to *maximize node (system) attributes* while subjected to possible *constraints* such as cost and schedule. Aggregating performance attributes as the overall *objective function* and assigning cost and schedule risks as constraints can thus ensure the selection of feasible portfolios that satisfy nodal requirements and minimize cascading risks.

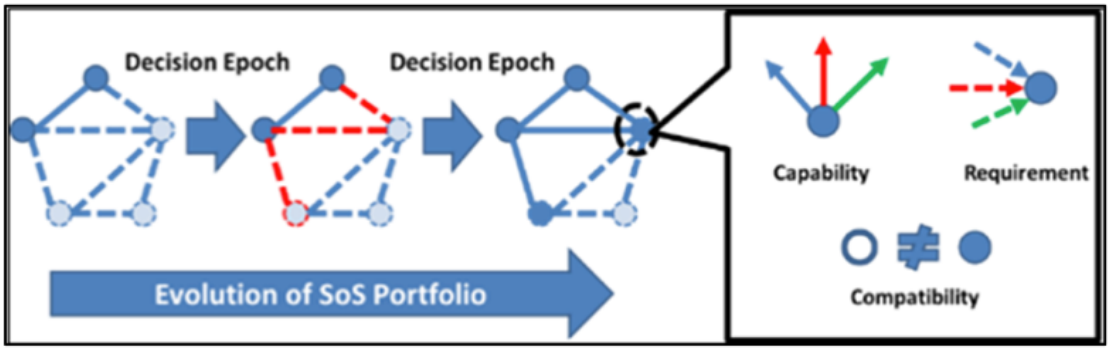


Figure 2-21: Archetypal Node (System) Behaviors (Davendralingam and DeLaurentis, 2013)

Knowing the optimal portfolios for each period can then facilitate *SoS Modeling*. After trading off performance attributes against risks in one time period, decision makers can then find the optimal solution to evolve to in the next time period. Figure 2-21 can illustrate the SoS evolution, where each node represents an independent system, a dashed line represents a possible connection by means of acquisition, and a bold line represents an existing connection by means of acquisition. Across the time periods (decision epochs), modifications are made to the SoS, as new systems are added across successive time periods to produce the new SoS construct. As such, a combination of robust optimization methods and multi-period approach can produce a long-term strategy for evolving a system/program/portfolio to meet affordability objectives. *This can potentially provide a future framework for affordability analysis as it captures the essence of time through the use of multiple periods, the exploration of tradespaces for all possible design options in each period by trading performance attributes against cost/schedule constraints, and the sequential execution of system acquisition or removal for evolving a SoS.*

2.5 Limitations of Existing Affordability Methods

A number of frameworks for understanding and analyzing affordability were described in the earlier sections, where their different perspectives on affordability and associated benefits were also identified. A review of these methods was necessary as it reflected the current state of affordability practice in the defense and aerospace industry. Through the documentation of their pros and cons, future spirals of affordability analysis frameworks can not only become an aggregation of the important benefits of their predecessors, but also an enhancement to their weaknesses in approach. Therefore, drawbacks in these approaches have to be identified so that future methods can take them into consideration. The major drawbacks identified were the *lack of time centricity*, *failure to recognize complexities of scale*, *lack of cost breakdown structures*, *treating affordability as a constraint rather than a requirement*, and finally a *lack of a value-centric perspective*.

2.5.1 Lack of Time Centricity

The lack of time centricity not only refers to the lack of consideration that systems can evolve or subjected to change over their lifecycles, but also the lack of time elements such as schedule during the formulation of early-phase design requirements. For example, the methods proposed by Tuttle and Bobinis (2012) Should-Cost Review and the statistical methods by Kroshl and Pandolfini (2000) have very strong elements of cost consideration, but there is little on how costs

may change and impact the system over time. Other methods such as the Sand Chart Tool, CAIV/TAIV and multi-period portfolio management fare better in accounting for time-related factors, as they do track the evolution of the system over time through the use of multiple time periods and evaluate changes at different points throughout its lifecycle. However, in all the methods reviewed, there has been little emphasis on having program development schedule or any other time-related activities to be taken into consideration during initial engineering design. Schedule or other time-related parameters should be explicitly taken into account in the design process.

2.5.2 Failure to Recognize Complexities of Scale

Development projects in the defense and aerospace industry can be a system, a program, a portfolio, or any general forms of a SoS. Each of these projects has different degrees of complexity involved and the difficulties in managing them cannot be estimated through extrapolation. While most engineers have experience in system design and management, programs and portfolios pose a totally different level of complexity as many sociotechnical factors, which were initially non-obvious at the system level, can become significant to the success of a bigger project. Building upon the principle of time centricity, both programs and portfolios require the consideration of project schedule development, which is the time required to complete every constituent element of the project. However, most traditional methods are more suitable for system-level applications and they are often applied to programs and portfolios without recognizing the change in scale and complexity. Schedules for individual systems may overlap and this affects the overall schedule of the larger program or portfolio. Only methods such as the Aerospace Corporation methodology, Sand Chart Tool and multi-period portfolio management do take into account certain degrees of this complexity as they aggregate results from constituent elements in the project in order to make an informed decision about the SoS. However, they still do not consider schedule or any other time-related parameters into the engineering design process.

2.5.3 Lack of Cost Breakdown Structures

Following the failure to recognize complexities of scale in systems to portfolios is the lack of cost breakdown structures. While it is always simpler to work with just one cost metric, it prevents decision makers from recognizing that there are many different elements of cost during the acquisition process. Acquisition by itself comprises many different activities that may have different funding profiles and some decision makers may be interested in certain cost elements over others. The success of a development project is not dependent on a single cost element, but all of them in entirety. As such, aggregating all these considerations into a single dollar value may over-simplify the design process and masks other cost elements that may become an impediment in the acquisition process. This is where cost breakdown structures, or the notion of "colors of money", are necessary. By acknowledging at the beginning of program/portfolio inception that acquisition contains many different cost elements that require different ways of treatment, the design process can harbor more valuable information that will better facilitate decision-making across the project lifecycle. The affordability method by Tuttle and Bobinis (2012), Should-Cost Review and the Aerospace Corporation methodology are the main frameworks that explicitly consider cost breakdowns in the analysis.

2.5.4 Treating Affordability as a Constraint

Designing for affordability is concerned with the explicit consideration of cost and schedule elements in the design process. However, all the methods reviewed only considered time in the form of system evolution over its lifecycle, but not as an active trade in the design process. Instead, many methods treat affordability as a constraint and do not incorporate fully into their 'design' process, thereby restricting the design space available to designers before conceptual design has even begun. While specifying affordability as a constraint may work, there is a great risk that many more valuable system designs are overlooked as no affordability tradeoffs are performed. In fulfilling the goal of engineering design, there is a need for upfront consideration of affordability prior to design formulation and an active trade among cost, schedule and performance related parameters.

2.5.5 Lack of Value-Centric Perspective

The fifth limitation of existing affordability analysis methods is the lack of a value-centric perspective. However, the defense industry is beginning to emphasize the notion of value through their Value Engineering (VE) and Earned Value Management (EVM) initiatives. As such, affordability analysis should also be aligned to the goals of *value-driven design*. Ross et al. (2010) defined the *creation of value* as "the balancing and increasing the net level of (1) satisfaction, with (2) available resources, while addressing (3) its degree of importance". Therefore, the design of systems in a dynamic operating environment with explicit affordability considerations requires a reformulation of how systems, programs or portfolios can provide value to stakeholders over time. As complex projects have multiple attributes related to cost, schedule and performance, utilizing value as a unifying metric can enable the evaluation of multiple paths for system evolution in order to achieve the same value delivery (Ross, 2006). None of the methods reviewed explicitly considered the notion of value in design. As such, the process of designing for affordability in defense and aerospace systems should also be value-driven.

2.6 Summary

A comprehensive review of the current state of affordability research and practice in industry was conducted in this chapter, beginning with a lexicographic analysis and the charting of its usage in daily applications to the engineering of complex sociotechnical systems. The scope of affordability usage and application was then narrowed down to the defense and aerospace industry, where the construction of complex systems, programs and portfolios is immensely challenging and they necessitate the consideration of cost and time elements in the design and acquisition process.

Current definitions for affordability within the industry were also collected to determine the important themes that overlap. Apart from the pillars of cost, schedule and performance, major themes include value, systems engineering, attributes, constraints, mission needs, risks, lifecycle, budget and strategy. Existing frameworks for understanding affordability and conducting affordability analysis were also described and reviewed in order to identify their advantages and disadvantages. The main drawbacks of the methods reviewed include the lack of time centricity, failure to recognize complexities of scale, lack of cost breakdown structures, treating affordability as a constraint rather than a requirement, and a lack of a value-centric perspective.

Aggregating the state of current practice and these drawbacks only imply that systems are currently not designed with explicit affordability considerations. In order to improve sustainability within the defense and aerospace industry, affordability analysis must build upon the strengths of existing frameworks and reduce the extent of their weaknesses. This can be done by including all the major themes in affordability and employing new methods that can remedy the drawbacks in current practice.

In this chapter, there was a brief mention of how possible designs for a system can be explored using *tradespaces* across *multiple time periods* in order to determine a desired system, program or portfolio evolution strategy that can enhance *value delivery* over its lifecycle. It is in this motivation that the concepts of *tradespaces*, *multiple time periods* and *value delivery over time* are introduced to enhance the state of affordability practice in the defense and aerospace industry.

3 TRADESPACE-BASED METHODS FOR AFFORDABILITY ANALYSIS

3.1 Motivation

Conceptual design formulation and system development are often subjected to multiple revisions that lead to unanticipated delays and changes in technical specifications. As discussed in Chapter 2, the accumulation of these outcomes often leads to rising costs and schedule slippages, which can eventually compromise the success of the system or program in development. With high-profile failures in system, program and portfolio delivery in the last decade, there has been a paradigm shift in approaches to systems architecting and acquisition. Performance is no longer regarded as *sine qua non*, and simulation of complexity in systems to portfolios often require considerations and analysis beyond a single cost attribute. This has necessitated the need to additionally account for multiple cost and schedule parameters elicited from stakeholders during early-phase design. This emerging paradigm in systems engineering is the *design for affordability*, where systems, programs and portfolios are architected to satisfy multiple performance, cost and schedule needs of stakeholders.

Many attempts have been made to propose frameworks for affordability analysis, and integrate them with existing systems engineering methods to generate affordable design solutions. As described in Chapter 2, various quantitative methods have been used alongside numerous visualization tools to quantify affordability during the systems architecting process. However, these processes have their drawbacks and they have been limited to static tradeoffs of systems between performance and costs in current operating environments, or in single point futures. More fundamentally, there is also a lack of a consensual definition and a common set of guiding principles for affordability within the systems engineering community. This gap in knowledge about the meaning and implications of pursuing affordability has resulted in the variety of approaches currently in existence, with few being able to explicitly capture the dynamic elements of the system, program or portfolio and its operating environment over its lifecycle.

To bridge these gaps, a *common definition* and a *common set of principles* for affordability can integrate approaches taken by the government, industry and academia into a concerted effort for reducing overall system or program costs and schedule slippages. However, it is beyond the scope of this thesis to achieve this common consensus on the definition of affordability. Instead, a consistent definition can be used in this thesis in order to establish a foundation for future affordability studies that can apply the same definition.

Given that systems and programs exist in a dynamic and uncertain world, designing for affordability not only necessitates new methods capable of evaluating them across many possible alternative futures, but also a *new philosophy* for treating the affordability paradigm. As such, it is in the interest of this chapter and the overall goal of this thesis to introduce *tradespace exploration as the fundamental method for exploring affordability tradeoffs in a dynamic manner.*

This chapter begins by describing what tradespace exploration is and how it can be used for affordability analysis. Next, a new metric is introduced to the tradespace exploration paradigm to help aggregate cost, schedule and other non-monetary expenditures into a single numeric value and replace the dollar value of a system commonly used in early-phase design. To account for value delivery and sustainment over the lifecycle, a modified framework based on an existing version is then proposed to perform affordability tradeoffs over time and facilitate the search for affordable solutions. Finally, the viability of the tradespace exploration method is demonstrated through an application case study.

3.2 The Tradespace Exploration Paradigm

Tradespace exploration is the model-based investigation of many design alternatives in order to find better design solutions, while avoiding premature fixation on point designs and narrow requirements (Ross and Hastings, 2005). Based on the concepts of trade studies, tradespace exploration leverages computer-based models and simulations to help stakeholders evaluate many potential designs in an efficient manner. As such, tradespace exploration allows a holistic consideration of a broad array of system capabilities and mission utility during early-phase design, instead of being locked too early into requirements and key performance parameters.

By enumerating and evaluating a large number of potential designs, tradespace exploration is most relevant to the design of complex engineering systems with multiple dimensions of benefits and expenses. The design process is often difficult to optimize and key design concepts may not appear intuitive to the designer. Applying tradespace exploration to the design process can circumvent these difficulties, prevent fixation on a single design or a local point solution trade, and explore the broader relationships between potential design concepts and stakeholder preferences. The use of tradespaces instead of simple tradeoffs of several point designs can thus evaluate the costs, benefits, and risks of systems concepts during early-phase design, thereby leading to better lifecycle results.

The tradespace for a system is the space spanned by all possible design alternatives for the system, which are obtained through the complete enumeration of all design variables for the system. Therefore, given a set of design variables, the tradespace is the space of points representing all possible design options, and expanding the tradespace will require either the generation of either new design variables or reconfigurations of existing combinations of variables to produce new points (Ross and Hastings, 2005).

The points on the tradespace can either be *dominated* or *non-dominated*. Non-dominated solutions typically have better tradeoffs between design variables as compared to dominated solutions. They can be identified along the frontier of the tradespace, and the solutions in this set are known as *frontier set solutions*. Further tracing these solutions along the frontier produces the *Pareto Front*, which connects the set of points that are the best for a given metric with all other metrics held constant (Ross and Hastings, 2005). The Pareto Front is thus the tradeoff curve between all system metrics and dominated solutions are those that are not on it.

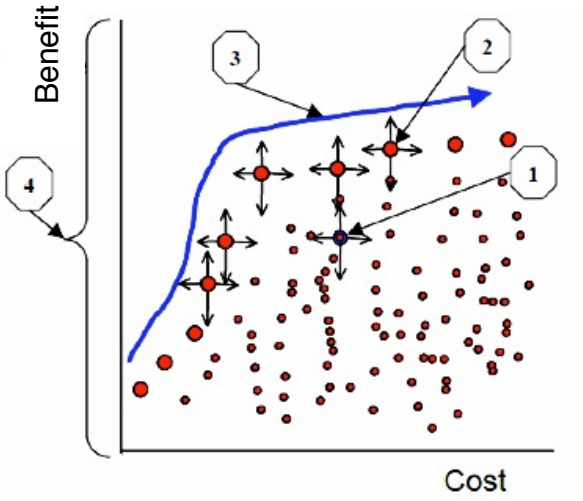


Figure 3-1: Four classes of trades: (1) Local point solutions (2) Frontier subset solutions (3) Frontier solution sets (4) Full tradespace exploration (Based on Ross and Hastings, 2005).

The process of tradespace exploration requires the reduction of the full tradespace containing all possible design alternatives to a smaller set of potential solutions based on the metrics of interest, from which stakeholders may eventually pick the one design most valuable to them. Typically, this smaller set of solutions is along the *Pareto Front*. However, exploring the tradespace fully can prevent premature fixation on a single design and increases the potential value created and delivered to the customers, especially if the metrics for benefits and costs change. Four classes of trades are possible and they are introduced in Figure 3-1 to depict the spectrum of tradespace considerations: (1) *Local point solutions*, (2) *Frontier subset solutions*, (3) *Frontier solution sets*, and (4) *full tradespace exploration*. It is important to understanding all types of trades when exploring all of the concept options (Ross and Hastings, 2005).

Choosing a *local point solution* is the minimalistic approach to a trade study since stakeholders simply choose a point and do not consider others on the tradespace. However, as mentioned previously, fixation on single designs can result in incomplete knowledge of the bigger design problem and stakeholders lose the opportunity to gain knowledge of better value solutions. Instead of finding local point solutions, the *Pareto Frontier subset* solutions can be found and key value tradeoffs among the design points within the subset can be identified. This subset of designs cannot increase utility without increasing costs, and are therefore efficient designs. Exploring designs that are not Pareto optimal can also reveal more information about design concepts and this allows stakeholders to recognize the key value tradeoffs that exist in the tradespace.

Going a step further in the exploration process will entail finding the complete Pareto Frontier, which explicitly identifies the key benefit-cost trade-off among design options. Establishing the Pareto Frontier enables the immediate assessment of new design options in terms of their distance from the "optimal" trade-off curve. Finally, performing a *complete tradespace exploration* will require the analysis of dominated solutions as well as the Pareto Frontier set solutions. Including dominated solutions in the analysis can help capture value metrics that are initially non-obvious and this allows for a more detailed and dynamic analysis of the structure of the tradespace itself (Ross and Hastings, 2005).

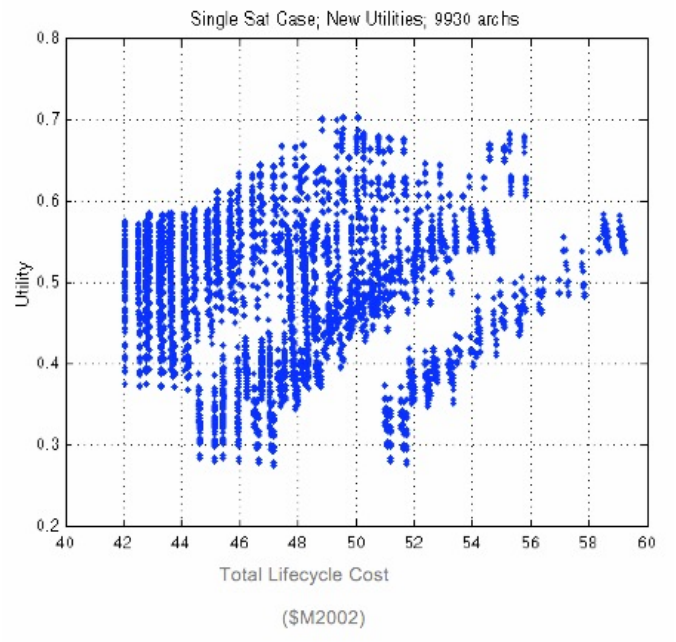


Figure 3-2: Sample Tradespace (Ross and Hastings, 2005).

A sample tradespace is shown in Figure 3-2 for a low altitude space science mission, where each point represents a unique design choice (Ross and Hastings, 2005). The design points on the tradespace were evaluated using a set of models and simulations in terms of lifecycle cost and utility to a science user. Each design point is then a pointer to an array of information regarding that design option, such as the values for the design variables, intermediate variables, system attributes and cost. A tradespace can hence be parameterized by any metrics of interest to the stakeholder. In the case of Figure 3-2, the metrics of utility and cost facilitate the evaluation of design points and highlight critical decision metrics. As such, tradespace exploration can enable a holistic understanding of system design concepts through performing trades on single points, multiple points, or along the Pareto Frontier.

In recent years, tradespace exploration has progressed substantially with a broadening of application areas, development of new metrics, as well as new representations and constructs for considering time and change. There is also the maturation of the process of exploration itself through an effort to codify the tacit knowledge of tradespace exploration researchers (Ross et al., 2010a). The overall outcome of this effort is structured guidance for systematically exploring tradespaces to extract answers to practical questions and to generate other forms of useful knowledge from the data in a tradespace dataset. This structured exploration guidance is a key enabler to the successful use and broad applicability of the tradespace exploration paradigm (Ross et al., 2010b).

3.3 Capturing Value through Tradespace Exploration

Section 2.2.3 discussed the emergence of value in engineering, and how the affordability problem in defense and aerospace systems can potentially be resolved by adopting a value-centric approach towards design. Value-driven design is not simply concerned about cutting overhead

costs and making enterprise operations lean, but about providing value improvement and enhancing the value delivery process in system design and engineering. However, like affordability, there is a lack of consensual definition for the term 'value', which has often been ambiguous. As (Ross, O'Neill, Hastings and Rhodes, 2010) defined the creation of value as "the balancing and increasing the net level of (1) satisfaction, with (2) available resources, while addressing (3) its degree of importance", this definition of value will be used throughout the rest of this thesis to espouse the importance of value centricity in affordability analysis. Therefore, it is the aim of this section to describe how tradespace exploration and related tradespace-based methods can be used to capture this aforementioned value in the design of affordable systems.

3.3.1 Value Creation

In addition to frameworks such as Value Engineering and Earned Value Management, various value-centric design methodologies (VCDMs) have been proposed to integrate value-driven design and traditional engineering system design. VCDMs adopt a technical approach to problem solving and they combine *scientific principles* and *cost-based system models* with a *valuation model*. Having a valuation model that outputs derived system value to stakeholders can then facilitate the promulgation of system selection criteria. Therefore, VCDMs are highly relevant to early-phase design and it can potentially guide stakeholders towards the selection of valuable system designs from amongst a set of candidate system designs.

To explicitly consider "value" in order to drive design, (Ross, O'Neill, Hastings and Rhodes, 2010) discusses the need to align perspectives on "value" with the method used to quantify "value". Creating value to stakeholders is thus the ultimate goal for system design and engineering across industries. Value creation can become very challenging as stakeholders have different preferences and a system attribute deemed valuable by one may not appear as valuable in the eyes of another. The process of value creation then requires the understanding and capturing of customer or user needs, and developing systems that best meet their interpretations and expectations. As nuances of the word "value" are in abundance, it is important to ensure that value is understood in the manner proposed by Ross et al in order to better benefit system design and engineering. One of the VCDMs reviewed by Ross et al is called *Multi-Attribute Utility Theory* (MAUT), which has the potential to extract, create and deliver value to stakeholders in a systematic manner.

3.3.2 Multi-Attribute Utility Theory for Value-Centric Design

MAUT is an extension of *utility theory*, which is a fundamental framework that can be used by decision makers to help quantify the idea of value consistently in the goal of engineering design. Utility theory is based on maximizing the value of a system with respect to a decision maker's objectives. Each decision maker may have multiple objectives that can be broken into a set of attributes. As described earlier in the goal of design, attributes are metrics that measure how well an objective for a system is met. Each attribute will have a definition, units, and range of accepted values, and requires careful consideration between both the designer and decision maker up front (Ross, 2003). Through careful elicitation of stakeholder preferences for an attribute, its range of accepted values are translated to a utility metric ranging from 0 to 1, such that the least acceptable range equates to 0 and the most preferred being 1. Capturing utility values across a range of factor

levels for an attribute then produces a *single attribute utility (SAU) curve* for that particular attribute from one particular stakeholder.

In the design of complex systems, multiple attributes are often of interest to stakeholders and it becomes difficult to select a preferred design by comparing and trading off among these attributes one at a time. Hence, there is a need to aggregate them into a single utility metric that accounts for their combined preferences on all attributes. This leads to *multi-attribute utility* (MAU), which is a measure of aggregate benefit to stakeholders and it can be calculated using the multi-attribute utility function introduced by Keeney and Raiffa (1993). This function is shown in Equation (1).

$$KU(X) + 1 = \prod_{i=1}^{N}[Kk_iU_i(X_i) + 1], \text{ where } K = -1 + \prod_{i=1}^{N}[Kk_i + 1] \quad (1)$$

In this equation, $U(X)$ is the aggregate utility value for the multiple attributes and their respective single attribute utilities, $U_i(X_i)$; k_i is the ith corner point which is a swing weighting factor for the ith attribute X_i; n is the total number of attributes; and K is the normalization constant. $U(X)$ is quantified on a scale from 0 to 1, where a value of 0 and 1 is the least and most desirable respectively. Therefore, the MAUT function provides a means to aggregate n monetary and/or non-monetary benefits produced by a system into a single metric that can then be used to conveniently rank numerous systems aggregately across n attributes.

Applying MAUT to the design of a system with multiple attributes and aggregating them according to varying stakeholder preferences using the above equation produces a single *multiple attribute utility* metric. By having utility instead of designer-specified metrics such as mass or power as one of the parameters on the tradespace, a tradespace plot becomes value-centric. Decision-makers can then compare MAU values between different designs to determine which ones will be better able to fulfill the cost, schedule and performance requirements for early-phase design. Utility is thus a direct reflection of value, rather than an inferred one. Therefore, value, as interpreted, quantified, and represented by MAUT, is the aggregation of non-monetary benefits relative to the monetary cost of obtaining those benefits (Ross et al, 2010).

MAUT is a more appropriate method for valuing engineering systems as it allows the ranking of design alternatives based on multiple sources of non-monetary value under uncertainty (Ross et al, 2010). Usage of the MAUT equation also assumes *mutual utility independence*, a weaker (less restrictive) assumption than mutual additive (preferential) independence, which is often assumed for discounted cash flow methods such as net present value and cost-benefit analysis (Ross et al, 2010). By using utility curves to measure stakeholder preferences for every attribute, MAUT allows for the consideration of substitution and complement affects among multiple attributes (Ross et al, 2010). MAUT can hence allow the design of systems to conform to the desired behavior of stakeholder decision-making more accurately than methods with governing equations that assume mutual additive (preferential) independence.

The motivation to use MAUT is further underscored by the inherent inability of stakeholders to assign a monetary value to an outcome or set of outcomes in the first place. This is reinforced by difficulties in cost estimating. As such, given these advantages of MAUT, its use for value-centric design has been significantly motivated in academia as well as industry for valuation of space systems. MAUT is a very versatile approach for decision making, as it allows for the quantitative aggregation of both monetary and non-monetary stakeholder preferences for, and hence

stakeholder perceived value of, a given system (Ross et al, 2010). This is why MAUT should be used in tradespace exploration, and more specifically in the use of tradespaces for affordability analysis, where there is now a stronger need to explicitly consider stakeholder preferences for additional attributes such as different cost commitments under the cost breakdown structure and development schedule in early-phase design. However, MAUT also has inherent limitations and uninformed usage may compromise the decision making process. Hence, it should be applied in prudent ways that maximize its strengths. More on its drawbacks will be discussed in Chapter 7.

3.4 Multi-Attribute Expense to replace 'Cost'

Designing for affordability requires the additional consideration of both cost and schedule parameters. However, traditional forms of tradespace exploration typically compares utility against cost, which may render them less effective for more holistic considerations of different cost elements across the system lifecycle. The term 'cost' is ambiguous and it can simply refer to the development cost, maintenance cost, support cost, retirement cost, or even the total lifecycle cost, depending on what the stakeholders are interested in evaluating for a particular set of designs. However, in reality, stakeholders and decision-makers may place more emphasis on one or more cost attributes over another due to changes in contexts. This paradigm is more commonly referred to as *'different colors of money'*, where different cost elements are spent with differing degrees of ease depending on the operating contexts and stakeholder preferences towards these cost elements (Higgins, 1997).

For example, budget caps may be placed on individual cost elements in an actual system development and exceeding any of these caps is likely to result in delays or even its cancellation. Stakeholders are thus more wary of their cost commitments and they rationally want to place more emphasis on the first cost element, followed by subsequent cost elements in order to allow system development to progress along its intended timeline. However, summing these cost elements into a single dollar value will remove stakeholder's visibility of these individual budget caps during early-phase design. Also, a single cost metric will also mask stakeholder preferences on different cost attributes that may otherwise augment the tradespace to reveal better-value designs. Using traditional tradespace exploration methods, there is then a chance that preferred design alternatives have satisfactory total cost but contain a cost element that has exceeded its budget cap. However, this breach in cost element budget cap will not be discovered until the formal implementation of the system, by which it has become impediment to progress in system development.

Also, temporal considerations like schedule and other non-monetary factors are often difficult to represent in dollars and even more challenging to account for using traditional tradespace exploration methods. Translating schedule or any other time-related attribute into a dollar value incurs a high degree of subjectivity during stakeholder's assessment of the monetary value of time. This adds a lot of uncertainty during the evaluation of design points on the tradespace. Adding this monetary value for schedule into 'cost' will further mask stakeholders' awareness of development schedule, which is more than often a critical attribute of a system, program or portfolio. Given that schedule delays is becoming commonplace in the defense and aerospace industry, it is necessary to consider schedule or other time-related elements during tradespace exploration.

By considering various expenditures across monetary resources and time, stakeholders can perform dynamic tradeoffs among various cost and schedule attributes on top of performance attributes. These cost and schedule attributes are collectively known as **resource expense attributes**. Therefore, better-value designs can potentially be found if performance and resource attributes are considered holistically. It is then of interest to replace the term 'cost' with a metric that is representative of these aggregated cost and schedule contributes. This can be done through applying MAUT to these attributes. Since all elements of cost and development schedule are forms of expenditure incurred to realize the development of the system, they are called 'expenses' and the aggregated expenses under MAUT is called **multi-attribute expense (MAE)** (Diller, 2002; Nickel, 2010). By applying MAUT, mutual independence is assumed to exist among these expense attributes such that stakeholder preferences for one expense attribute is independent of preferences for another.

To determine the MAE for a system, the MAE function can be used and it is formulated similarly to the MAU function proposed by Keeney and Raiffa (1993), with the utility function replaced by an expense function $E(X)$. $E_i(X_i)$ represents the single attribute expense for different cost elements and time-related attributes that are considered during early-phase design. Using the MAE function thus allows the aggregation of these different types of dollar budgets.

$$KE(X) + 1 = \prod_{i=1}^{N}[Kk_iE_i(X_i) + 1], \text{ where } K = -1 + \prod_{i=1}^{N}[Kk_i + 1] \quad (2)$$

The notion of expense is akin to the notion of negative utility. Quantified on a 0 to 1 scale, an expense level of 1 denotes complete dissatisfaction and an expense level of 0 denotes minimal dissatisfaction. A rational stakeholder will typically demand maximal utility and minimal expense in an ideal design. Like MAU, an MAE function requires careful construction through stakeholder interviews to elicit informed responses and aggregate preferences to capture articulated value.

Since MAE is a dimensionless, non-ratio scale metric, an entity with twice the MAE number over another does not imply that it is twice as expensive in terms of monetary value. Since temporal elements have extensive leverage on the different 'colors' of money, the MAE can be extended to affordability applications in system, program and portfolio design. Instead of simply comparing monetary costs against utility, tradespace exploration can be modified to compare MAE against MAU in order to perform affordability-driven analysis. This gives rise to the method of *multi-attribute tradespace exploration* (MATE) for affordability analysis, which allows for the conduct of dynamic tradeoffs among additionally considered attributes for cost and schedule in system design.

3.5 Multi-Attribute Tradespace Exploration for Affordability Studies

Previous sections discusses the use of aggregating utilities for multiple performance, cost and schedule attributes as well as the use of tradespace exploration to create value in system design for stakeholders. The confluence of these important themes yields **Multi-Attribute Tradespace Exploration (MATE)**, which is a conceptual design methodology that unites multi-attribute utility theory, tradespace exploration, and model-based design to provide a decision-making tool for stakeholders (Ross and Hastings et al., 2004).

MATE typically begins with the identification of needs, followed by the enumeration and finally the evaluation of possible design alternatives. After eliciting the real mission needs and stakeholder preferences, attributes and design variables for the system are then chosen in order to evaluate all possible designs using some measure of costs and utilities. Different design points on the tradespace are then compared against one another to determine the most preferred design that has the best tradeoffs among its attributes relative to stakeholder needs. MATE has been applied in numerous case studies, where tradespaces for new systems are typically parameterized by MAU and a single cost metric. A 48-step description of the MATE process is given in (Ross, 2003) and it can be summarized visually in Figure 3-3.

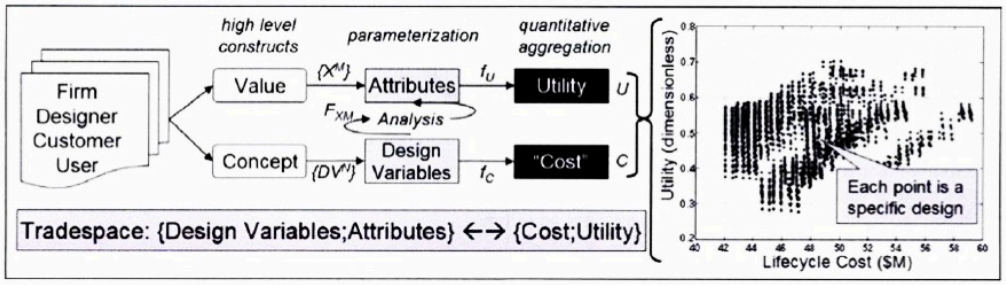


Figure 3-3: Multi-Attribute Tradespace Exploration (Ross and Hastings, 2005).

MATE for affordability analysis, however, is different in terms of the parameterization of the design space. Instead of 'cost', MATE for affordability uses the MAE and design points on the tradespaces are hence compared against one another in terms of their MAE and MAU values. The conduct of MATE is generally similar, but is now inclusive of different cost and schedule considerations at the very beginning. MATE for affordability begins with the elicitation of stakeholder needs that facilitate the establishment of *design variables*, which are factors within the designer's control that will drive the attributes.

There are also *epoch variables*, which are factors that parameterize uncertain potential operating contexts. Design-to-value mapping of performance, as well as cost and schedule parameters, is then conducted. Both design and epoch variables are combined under logical assumptions and scientific principles to produce a tradespace model that will evaluate potential designs in different epochs in terms of performance, cost and schedule attributes. Stakeholder preferences towards individual attributes and swing weights for each attribute in every epoch are then elicited to produce the corresponding SAU curves.

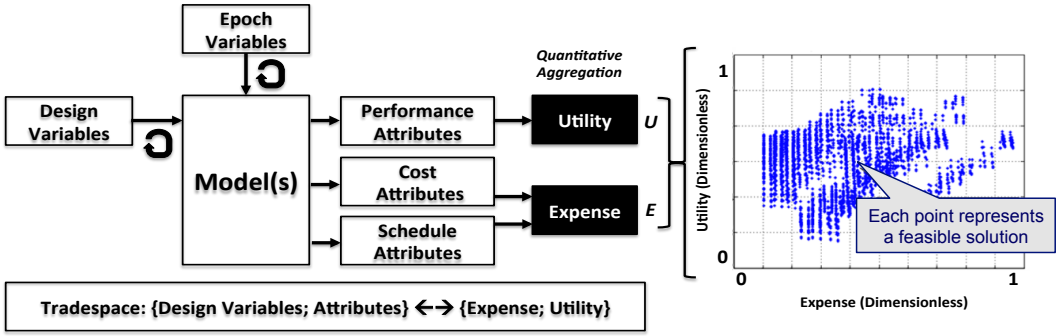


Figure 3-4: Data flow for the Tradespace Exploration process in Affordability Analysis (adapted from Ross and Hastings, 2005).

Under MAUT, each attribute then delivers a unique independent utility and can be combined with other attributes to produce an overall utility for a design. MATE uses the multi-attribute utility (MAU) function to aggregate, which combines different single performance attribute utilities, ranging from 0 to 1, with 0 defined as minimally acceptable and 1 as the point where no further benefit is gained. The MAU function can be a linear weighted sum if the attributes are independently contributing to the aggregate utility. Similarly, each expense attribute can be defined and then combined in the same manner using the MAE function. Applying MAUT then produces MAE and MAU values for all design alternatives across all considered epochs. The flow of data in the steps taken to conduct MATE for affordability analysis is shown in Figure 3-4.

The systems engineering discipline has also been advanced through the use of non-traditional design criteria called "**ilities**", which are system properties that often manifest and determine value after a system is put into initial use (de Weck, Ross and Rhodes, 2012). Incorporating ilities into tradespace exploration may allow for better discrimination between different conceptual designs (McManus et al., 2007). However, (Richards, 2009) stated that representing temporal properties like ilities in a static construct may be inappropriate, and that system attributes need to be perceived as independent, which is counteracting the definition that ilities are defined by attribute performance over time.

To circumvent these challenges, MATE can also be performed *dynamically* by accounting for changes in the tradespace due to changes in needs or contexts. This gives rise to **dynamic MATE**. As such, dynamic MATE can guide decision-makers towards finding good value designs, compare the strengths and weaknesses of selected designs, and determine the tradeoffs required to make selected designs more feasible. Apart from its inherent focus on value-centric design, dynamic MATE can also account for the importance of time in system evolution, the possibility of change or perturbations to the system, and uncertainty across the system lifecycle. Using dynamic MATE can thus address the limitations of current methods for affordability analysis and enable the aggregate consideration of performance and resource attributes in the selection of preferred designs on a tradespace. This allows decision makers to address multiple attributes that may go into a decision on a single convenient platform.

The process of conceptual design is often plagued with many decisions or design choices for a system to achieve mission requirements in different ways. Using other methods such as numerical simulation or optimization to find the most preferred design can make it difficult for decision makers to determine and choose between different design concepts. As such, tradespace exploration can serve as a primary tool for decision makers to evaluate the utility of various concepts before proceeding with a specific design. In particular, to fulfill the aim of value-centric design, MATE can help quantify stakeholder perspectives of value towards multiple design attributes and facilitate comparison of various design points using just two aggregated metrics. MAE is one of the metrics and it is introduced to replace 'cost', which has been commonly used in traditional trade studies to quantify investment into designing and developing a system. To perform affordability analysis, additional cost and schedule attributes have to be considered simultaneously and assumed to be mutually independent of one another in order to justify the usage of MAE for the weighting and aggregation of all resource attributes. By parameterizing the tradespace with aggregated utility and expense metrics, MATE can help decision makers explore

the tradespace more meaningfully and compare design points against one another to determine the most technically sound yet cost effective and schedule effective architectures.

3.6 Affordability as an Ility

Ilities concern wider impacts with respect to time and stakeholders and can better promote the development of successful systems as compared to solely technical criteria. Commonly known ilities such as survivability and evolvability have already been defined in many engineering fields and their inclusion in the design process often leads to desirable outcomes. **Affordability can be treated as an ility that drives the design of more affordable yet technically sound architectures.** With affordability as an ility, advanced systems engineering methods like *tradespace exploration* can be applied in the enumeration, evaluation, identification and selection of affordable designs.

Affordability is defined as the property of becoming or remaining feasible relative to resource needs and resource constraints over time.

This property can be applied to any entity, be it a system or a program or portfolio, with the bigger entities warranting a higher degree of complexity and more attributes and resources for consideration. A **resource** may be defined as the aggregation of cost, schedule and other non-monetary factors necessary for architecting, development and operation. **Resource needs** are the set of resource requirements elicited from stakeholders, and **resource constraints** are the statements of restrictions on these requirements that limit the range of feasible solutions.

Some resource needs include monetary elements like development cost, operations cost, maintenance cost etc. as well as the time investment required to perform research and development. Realistically, rational stakeholders will always prefer zero resources to be expended during the design process, but they also have preferences over increasing levels of resource expenditure. These needs are a direct reflection of stakeholder preferences and they quantify the expenditures required to design a system, program or portfolio that best fits the mission requirements. The aggregate of these resources needs then quantifies the minimum amount of resource expenditure required to achieve at least the desired amount of performance.

Resource constraints, however, are somewhat in countenance to the notion of stakeholder needs as they are restrictions imposed upon the range of resources that could be made available to stakeholders. These restrictions are independent of stakeholder preferences and are often imposed by people or organizations with fixed agendas on reducing cost and schedule overruns, and has no direct interest in the performance outcome of the entity being designed.

Examples of such restrictions include budget caps for every cost element in the design process based on contractual agreements and time required to deliver a particular capability or an entire system. As such, rational stakeholders will definitely prefer to have a greater range on their spending capabilities in order to design the most desirable product, but resource constraints are almost always limiting that range in order to keep solutions within time and monetary budgets.

As these entities and their operating contexts may be dynamic, resource needs and resource constraints may *change over time*. Consequently, architectural solutions for these entities become feasible if they fulfill resource needs and function within the resource constraints for a fixed context. Also, they also need to fulfill performance requirements and deliver a value greater than the minimum required utility level for the same fixed context. As contexts change, these entities may *remain in, enter, or exit the feasible set of solutions*. Affordable solutions are thus those that remain in or enter the feasible set of solutions. Therefore, the general goal of affordability analysis is to *identify solutions that remain feasible throughout or for a large part of the system lifecycle*. Using this operationalization, an affordable solution will be one that is *capable of satisfying changing resource requirements and resource constraints, as well as satisfying performance requirements and performance constraints over the system lifecycle*. Designing for affordability thus aids in the delivery of affordable solutions by *trading off excess performance levels for more cost and schedule margins*.

Designing for an ility can provide extra value to the system. By applying tradespace-based methods like dynamic MATE to the process of designing for affordability, both monetary and non-monetary stakeholder preferences for as well as stakeholder perceived value of a given system could be captured explicitly. Designs can then be compared against one another as using MAUT helps quantify and ranks potential architectures according to their respective benefits and stakeholder preferences uncertainty. When the preferences of a decision maker for a performance or resource attribute change in the same manner as how mission requirements change, the tradespace is altered accordingly to reflect the change and the impact on the tradespace can be rapidly assessed. More importantly, changes to cost and schedule requirements occur ever more frequently than that of performance requirements. Using tradespace-based methods can thus facilitate the investigation of the impacts of such volatile changes more holistically and multi-dimensional tradeoffs among various performance and resource attributes. This can better deliver value that is aligned to the ility being designed for in the system. In the design for affordability, having awareness of and performing tradeoffs among performance, cost and schedule attributes can help stakeholders and decision-makers arrive at design solutions that are both cost-effective and time-effective.

3.7 Affordability in Systems of Systems

The previous chapter discusses the importance of designing for affordability in a system, program and portfolio. A *system*, together with all its socioeconomic and technical considerations taken in its development process, is known as a *project*. A *program*, is of a higher degree of complexity than a project, and it is defined as a group of related and interdependent projects managed together to obtain specific benefits and controls that would likely not occur if these projects were managed individually (KLR, 2008).

Finally, a *portfolio*, which is of the highest complexity and scale, is defined as a collection of projects, or programs or both grouped together to facilitate the effective management of efforts to meet strategic business objectives (KLR, 2008). These projects or programs may or may not be working in concert to achieve some overarching objective or deliver an emergent capability, but their assigned proportions of the overall schedule and budget may affect their individual

performances. As such, whether the elements are loosely or closely interacting, both programs and portfolios can be generally regarded as System of Systems (SoS).

Although the design of SoS is gaining recognition and importance, the elements that constitute a SoS is often unclear and ambiguous (Maier, 1998). DoD (2008) defines a SoS as "a set or arrangement of systems that results when independent and useful systems are integrated into a larger system that delivers unique capabilities. Both individual systems and SoS conform to the accepted definition of a system in that each consists of parts, relationships, and a whole that is greater than the sum of the parts; however, although an SoS is a system, not all systems are SoS."

The concept of SoS is more than just simply applying the definition of "system" to another system, but it has unique characteristics that come about when one combines interacting systems in a way to achieve additional functionality. As such, performance, cost and schedule attributes for a system and SoS are likely to be different, but the attributes of the latter may be a function of the attributes and design variables of the former. Likewise, performance and resource attributes for a system, program and portfolio are going to be different from one another. Program-level attributes and portfolio-level attributes are likely to be functions of the attributes of its constituent systems.

For example, depending on the nature of the SoS being designed, the overall capability of a program may be the product of individual system capabilities; or the cost of developing a program may simply be the sum of its parts; or the time taken to develop the program greatly depends on whether its constituent systems are developed in series or parallel. Portfolio-level attributes may not be derived in as straightforward a manner, as its constituent systems or SoS may not be interacting with one another to produce an emergent capability. Hence, the performance, cost and schedule attributes for a portfolio may be a function of just some or even all of its parts. Given the increased degree of complexity of program and portfolios, careful elicitation of stakeholder needs and preferences is necessary to best design the SoS. Based on the definition of the SoS, there should thus be at least one performance attribute that is representative of the emergent behavior of the program or portfolio.

3.8 From System to Program to Portfolio and Back

In the defense and aerospace industry, there are often many technical requirements for new systems being designed, and the initial investments are high. For example, constructing a missile defense system requires the design and construction of several missile launchers located in several bases, where the launchers can have different target ranges and destructive capabilities in order to counteract a variety of aerial threats. Similarly in the space industry, designing a new Mars Exploration Rover requires the integration of many science modules, with each module having its own capability and possibly interacting with other modules in order to perform a variety of tasks in concert.

Meeting all these requirements by designing a single entity in an aggregated manner is almost impossible due to exorbitant costs, high risks of failure, and overwhelming complexity to the system designer. The most realistic manner of design is often to introduce certain degrees of modularity or redundancy, which necessitates the breakdown of a single entity into multiple smaller entities, or the inclusion of two or more elements that are performing the same or perhaps

different functions in order to achieve greater mission objectives. The act of designing entities that may interact with one another to produce capabilities greater than the sum of its parts is the designing of a SoS, which can be equated to the design of a program or a portfolio of systems. Given that entities designed to meet a set of mission requirements can exist as a system, program or portfolio, the next question to be asked pertains to which a category it belongs. An entity can initially exist as any of the three depending on where decision makers draw the system boundaries. The initial entity should have its own independent performance and resource attributes that are perceived independent of one another. At this juncture, a **bottom-up** or **top-down** approach can be taken in the design for affordability process.

In the **bottom-up** approach, the entity is first classified as a system characterized by its system-level performance and resource expense attributes. Should multiple systems be grouped to produce an emergent capability, the design process has transitioned to a higher level of that for a program. As mentioned earlier, the program will have different performance and resource expense attributes but they are reflective of the new capability. Should multiple programs, either the same or individually different, be grouped together in the design process, it will transition to the next higher level of that for a portfolio. Similarly, portfolio attributes are unique but may or may not be reflective of an emergent capability among multiple and similar/different programs.

The bottom-up approach is the most obvious in industry practice, as systems engineering practitioners often design individual smaller systems first, then validating their cost, schedule and performance attributes before integrating them together as a greater system or SoS. Experience with and knowledge about individual systems are also likely to exist in greater quantities than that at the program or portfolio level given the nature of work breakdown structures for projects in the defense and aerospace industry. As such, the workforce in charge of developing all the smaller entities is also very likely to be a lot bigger than the senior management team at the program or portfolio level. Given such constraints in workforce structure and engineering practice that may very well exist in real-world scenarios, it is thus fundamentally easier to begin the design process at the system level and scale it up to a program and finally a portfolio level.

If the bottom-up approach is taken, decision makers have a greater degree of control and confidence in the performance and resource attributes or requirements for every individual entity. They will be more familiar with the early-phase design and actual construction of every system, thus allowing their preferences and needs to be explicitly captured in the design process. They will be able to know the full range of possible architectures that can exist in a system tradespace and be able to perform single or multiple point trades as desired.

Arriving at a technically satisfactory and affordable solution for every system and then aggregating these solutions at the higher levels is likely to produce affordable solutions for a program or portfolio in design. Furthermore, resource uncertainties at the system level are often lower than at the higher levels and taking the bottom-up approach provides an opportunity to mitigate this uncertainty before it accumulates.

However, results obtained at the portfolio level are highly dependent on the system or program designs picked to establish the portfolio. If only a handful of designs were selected, this limits what the possible portfolio designs that can be obtained at higher levels of aggregation. If selection of designs was conducted in an over-conservative manner, the portfolio results may be affordable but sub-optimal as compared to what is actually possible. However, this could be mitigated using the top-down approach.

The **top-down** approach is conducted by breaking down one or two overarching objectives at the program or portfolio level into many smaller unknowns. This helps decision makers to decompose the problem into more manageable segments. However, commencing the design process at the program or portfolio level incurs a much higher degree of risk and uncertainty, as they often stretch for long periods of time and contain numerous entities with unique levels of uncertainty that have yet to be accounted for.

Furthermore, it becomes difficult to derive results for the system level as little or no information is provided with regards to the design variables or attributes for the smaller constituents. Even if results were obtained, they are unlikely to be the same as the results obtained from a bottom-up approach, where much more information is available for the design of individual systems. There is thus a risk that better-value designs may be lost if system level results are derived from the higher levels. It then appears that the top-down approach is less useful to performing the design process.

However, this does not imply that there is no value in taking the top-down approach. It is actually most vital to systems architecting and master planning activities at the senior management level. Taking either a bottom-up or top-down approach is highly dependent on who the decision makers are, the entity level at which the decision makers exist and have the highest degree of influence, and ultimately the decisions made by them in order to determine the point of entry for initiating the design for affordability process. Should decision makers be at the upper echelons of senior management in the defense and aerospace industry or even the government, they are likely to initiate the design process from the design level in which they have the most authority.

These decision makers are responsible for the directions of research and development in their corporations. Decision makers at different levels can have different sets of preferences and needs. A decision maker at the system level may be most concerned about the performance capability of one system while a decision maker at the program or portfolio level may be interested in the one or more emergent capabilities of several integrated system platforms.

Also, a decision maker at the system level is only concerned about the costs and development time for one system, while a decision maker at a higher level will be concerned about how cost and time elements for different systems may affect one another. Also, decision makers at the program or portfolio level can do things that a system level decision cannot do. They can choose to partition the time and monetary resources available and distribute them according to which systems and programs mean most to them. Therefore, decision makers at different levels exhibit different behaviors. They are incentivized differently and therefore value different criteria in their decisions.

Taking a **two-pronged approach**, where decision makers at the system level adopt the bottom-up approach and decision makers at higher levels adopt the top-down approach, can possibly produce even better results in the design for affordability process. Results from the two approaches can be compared and adjustments can be made to system, program or portfolio level needs, preferences and attributes if necessary. The design for affordability process through tradespace exploration is ultimately an iterative process that requires communication and collaboration among decision makers at all levels to produce affordable solutions to the system, program or portfolio they are designing.

Should results from the two approaches be in conflict, decision makers at all levels should participate in a multi-stakeholder negotiation to make the aforementioned adjustments in order to obtain a consensual agreement. For example, decision makers at the program or portfolio level may have set conservative estimates for the development cost of each system, thereby yielding equally conservative estimate of each system's performance levels. However, decision makers at the system level, who have more information about design at the ground level, may wish to highlight the existence of better value designs that cost slightly more or just as much as the earlier conservative designs.

Decision makers at the program or portfolio level may then agree to pick the better value designs or be less conservative about their cost and performance estimates in order to achieve higher utility. Else decision makers at the system level will comply with the conservative choices made and pick designs with the highest utility level possible. Therefore, having a two-pronged approach at the system-program or program-portfolio level transitions can be helpful to the collaborative search for more affordable solution at all levels.

The focus of affordability analysis in this thesis, however, will be the *bottom-up approach*, where dynamic tradespace exploration is performed first at the system level, then at the program level, and finally at the portfolio level. The tradespace exploration process was described earlier in this chapter and the extrapolated process for designing from system to portfolio is shown in Figure 3-5. The top-down approach is also valuable to design making, but it is beyond the scope of this thesis for further discussion and implementation.

There are three points of entry for affordability analysis using tradespace-based methods when designing from a system to a portfolio. The analysis can hence begin at the system level, program level or portfolio. Regardless of the point of entry, the steps taken to perform the analysis are essentially the same. It typically begins with the identification of key decision makers and depending on the point of entry, they may be engineers and project managers at the system level, and senior management officials at the program or portfolio level. These decision makers are then interviewed, where their mission needs are first carefully elicited. From the needs, the mission concept for the entity of interest is then formulated and this represents the overarching design objective for that level of entry. Based on the mission concept, key performance attributes for the entity are then identified. Concurrently, resource expense attributes such as development cost, operations cost, labor cost and development schedule are also identified.

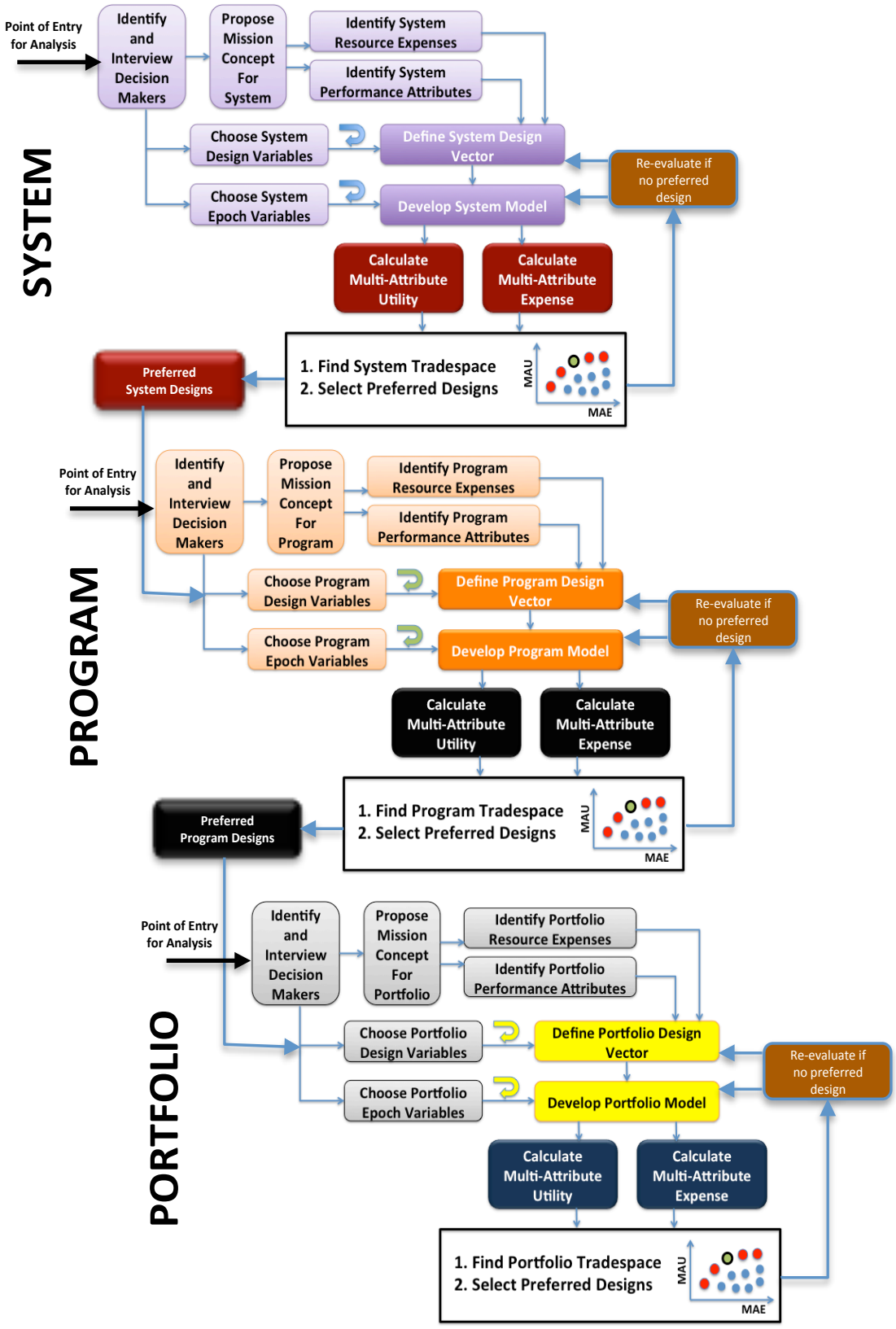


Figure 3-5: Flow chart for conducting the tradespace exploration process from system to portfolio.

The decision makers are also interviewed on their choice of design variables and preferences for each performance and resource expense attribute. This results in the establishment of SAU curves for each of the attributes that reflect how decision makers' perceived values change as the design variables and attributes are varied. In addition, the spaces of all possible contexts or operating environments of the entity are also obtained, thereby deriving the epoch variables that determine the changes in these contexts. Enumerating all design variables and epoch variables then produces all possible design points on a tradespace for each epoch, where each point represents a unique design vector. Interactions also exist among all the design variables, and epoch variables may determine if such interactions or even entire design vectors are feasible across different contexts. Integrating all these variables and their interactions then produces the system, program or portfolio model that eventually produces unique values for performance and resource expense attributes for all possible designs.

Applying MAUT, the MAE and MAU values for each design in an epoch are then calculated and plotted on a tradespace for that entity. Each design point on the tradespace can be compared based on these aggregated metrics. Decision makers can then explore the tradespace to search for preferred designs. One of these ways is identify the Pareto front of the tradespace and select one or more preferred design points in the Pareto frontier set. If there are no preferred designs, a random design can be picked or the tradespace model can be reformulated and re-evaluated. Design variables and epoch variables can be changed or added while preferences and interactions among the design variables can also be modified to augment the tradespace. These steps are performed in an iterative manner until more confidence in the model, and at least a single preferred solution, are obtained.

If the design process entails a transition from system to program or from program to portfolio, the preferred designs found at the lower level will become design variables at the higher level since they will be constituents of the larger entity. At the program or portfolio level, the design variables and epoch variables will be combined with the preferred designs from the lower system or program level to form the new design vector. Program or portfolio level performance and resource expense attributes will hence be indirect functions of system or program level design variables and attributes. The same steps are then applied until one or more preferred design solutions are obtained. Again, if no solutions are obtained, then there is a need to re-evaluate the model derived at the higher level. Therefore, the bottom-up approach to tradespace exploration from system to program to portfolio can ensure that satisfactory utility and expense levels are obtained for designs chosen at every level, thereby producing affordable solutions that are likely to be cost-effective and time-effective across the lifecycle.

3.9 Constraint Levels for Utility and Expense

After establishing the tradespace bounded by MAE and MAU, external imposed restrictions on cost, schedule or performances that are completely independent of stakeholder's preferences can be reflected as **constraint levels**. As a rational stakeholder's true preferences, especially towards expense, are often higher than any externally imposed restrictions such as maximum budget or fixed deadlines, constraints and preferences have to be considered separately. For example, a space systems engineer is only most satisfied when a satellite to be designed has a payload of

30000kg in order to achieve a new scientific capability, but senior management may have set a limit of 20000kg due to safety concerns or implementation of radical cost cutting measures.

This is limit is thus a constraint level on a performance attribute. As such, there may exist feasible satellite designs with a payload capacity above 20000kg but will not be recognized as preferred solutions because of the breach in capacity constraint. Similarly, an engineer will definitely prefer to have an indefinite budget to spend on building the satellite in order to derive the maximum possible capability, but senior management may again set a cap of $20 million for development cost. With that constraint level on monetary expenditure for satellite development, the engineer can only have a lower but still satisfactory level of utility than initially desired.

During the design process, there is a chance that the engineer may know of the preferences of senior management to set caps on performance, cost or schedule attributes and he may adjust his preference levels such that his maximum level of utility for a particular attribute is attained at the value of that cap. However, it is important to elicit the decision maker's true articulated preferences for a certain attribute and distinguish them from any external constraints that may be imposed upon him.

This is because setting a stakeholder's minimum preference level at the constraint value without the use of additional constraint levels will make it difficult to determine what designs have become unaffordable due to changes in resource constraints since they will not be reflected in the tradespace. If decision makers may not be aware of these constraints initially, they then establish their own preferences without knowledge of the environment. It is also likely that even if they set preference levels according to the external constraint, the latter may change due to volatility in mission or budgetary requirements. Therefore, applying constraint level and making a distinction between constraints and preferences enables a realistic depiction of the relationship between stakeholders and their environment.

Shown in Fig 3-6, **constraint levels for minimum utility** and **maximum expense** can represent the minimum required performance levels and maximum budget respectively that are imposed as constraints by external sources. If no external constraints are available, the default value will be an acceptable preference level specified by the stakeholders. These can be calculated by first setting the constraints on individual performance and resource expense attributes. The minimum constraint level for expense can then be obtained by the intersection between the minimum utility constraint level and the design point with the minimum expense on the tradespace. The vertical line through this design is referred to as the **derived minimum expected expense constraint level**.

The **affordable solution region** is then the intersection of the possible solution space and the area bounded by the planes representing the minimum utility, the derived minimum expected expense and the maximum expense constraint levels. The two points at the corners of the affordable solution region are actual evaluated design points in the solution space. An "affordable" solution then will be any solution that falls within the affordable solution region.

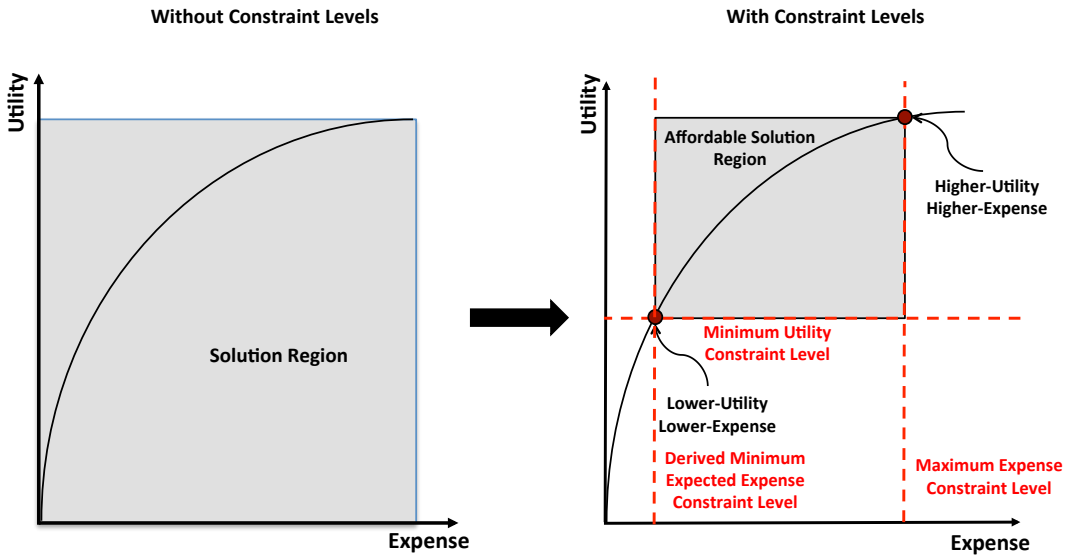


Figure 3-6: Narrowing the affordable solution space using external constraint levels for a fixed context.

3.10 Epoch-Era Analysis for Affordability

The lack of time-centricity was one of the limitations identified in current methods being used to design for affordability. Systems can change over time due to uncertainties in their operating contexts and mission requirements and this is often not accounted for in many forms of affordability studies such as the one performed by Tuttle and Bobinis (2012). Tradespace exploration, if performed simply using a single set of fixed needs and requirements, will yield solutions that are only Pareto optimal under that specific set of conditions and are unlikely to remain optimal when there are changes to either needs or requirements in the future. Such changes are typical with the progression of time and the system to be designed can possibly be operating within different contexts. As such, it is of high importance to be able to design systems that remain Pareto optimal for all or at least a large number of possible operating contexts. However, it is often a challenge to find such a system and a way to generate such a solution is for a Pareto optimal design in one context to change to another solution in another context through the path of lowest resource expenditure.

To do so, different tradespaces over time for a system can be generated and analyzed in the process of dynamic MATE. In the design of engineering systems, the timespan of interest to decision makers and stakeholders will be the **system lifecycle**. The system lifecycle is the core construct that designers use to characterize the phases of a system during its lifespan, from initial concept to end of life (NASA, 2012). System lifecycle processes facilitate the organization of various activities required to design, develop, and operate a system. The system lifecycle is hence comprised of phases that have defined start and end points in time, which are based on the resources available to complete a set of phase activities. As useful it is a construct for framing all system-related activities, the system lifecycle view is only an imposed artificiality for managing system activities, and does not explicitly consider system context as a dynamic variable (Ross and Rhodes, 2008b).

This means that creating timelines during system design does not directly consider the impacts of a diverse set of context changes. Furthermore, just using the system lifecycle alone during the design process does not enable understanding of system value delivery across its lifespan. This is especially more so in affordability analysis, where multiple performance, cost and schedule attributes have to be considered simultaneously. Each of these attributes, as well as its underlying variables, is likely to be affected by changes that occur during the system lifecycle. While variances in performance can be better controlled using a variety of methods, uncertainties in cost and time elements associated with each system lifecycle are much harder to predict and let alone control. Therefore, there is a need for using other methods to consider the temporal view in designing highly complex systems that will remain value robust in the face of changing mission needs and uncertainty.

A way of doing so is **Epoch-Era Analysis (EEA)**, which is an approach for conceptualizing system timelines using natural value centric timescales, wherein the context and expectations define the timescales (Ross and Rhodes, 2008b). EEA discretizes the lifecycle according to impactful changes in the operating environment, stakeholders, or the system itself, through the constructs *epochs* and *eras*, instead of traditional system milestones. The full lifespan of system is referred to as the System Era, which can be decomposed into Epochs. An epoch is a period of time for which the system has fixed context and fixed value expectations. Each epoch is characterized by static constraints, available design concepts, available technology, and articulated attributes. As changes trigger the start of a new epoch, the system may need to transform in order to sustain value, or else it may fail to meet expectations as defined for this new epoch. An era is thus simply an ordered sequence of epochs that describe a potential progression of changing contexts over time.

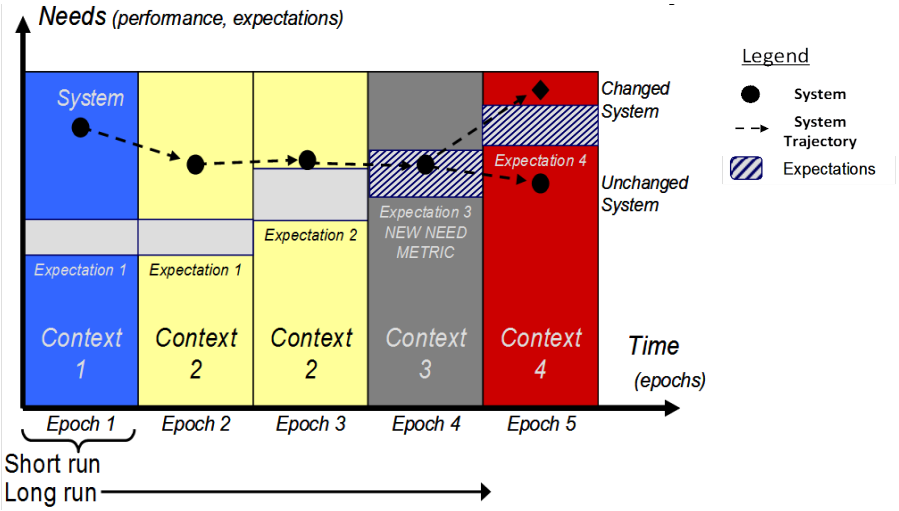


Figure 3-7: Epoch-Era Analysis (Ross and Rhodes, 2008b).

EEA thus provides an intuitive base upon which to perform analysis of value delivery over time for systems, programs and portfolios under the effects of changing contexts and mission requirements. By considering all performance and resource expense attributes in different tradespaces and their corresponding epochs, solutions that can remain technically sound and affordable across all or many epochs can be found. This is especially important when evaluating

large-scale engineering systems because they usually have long lifespans and are expected to cope with a variety of missions. EEA can be used in conjunction with MATE during conceptual system design, allowing for the evaluation and comparison of the value-over-time of many different potential designs across different operating contexts. For affordability analysis, EEA can be modified to assess the temporal progression of a system as resource needs and contexts change so as to adopt a more resource-centric approach to evaluating system design concepts.

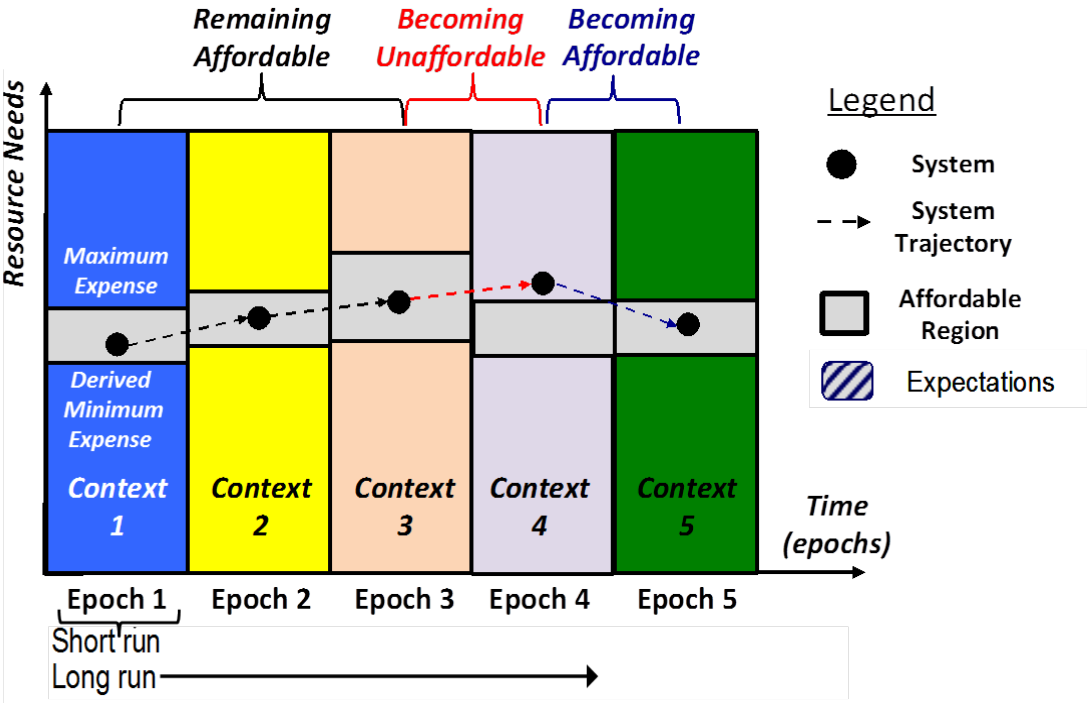


Figure 3-8: Modified EEA for affordability analysis using a resource-centric approach.

Figures 3-7 and 3-8 are the original EEA diagram and the modified version for affordability analysis respectively. In both figures, the vertical columns represent the epochs that are time-ordered to form an era, while different colors of these epochs represent changes in context. Changes to the original EEA are reflected in Figure 3-8, where the vertical axis has been modified to measure resource needs rather than performance needs, and the bounded regions now represent affordable regions instead of expectation levels. The horizontal bands are the constraint levels illustrated in Figure 3-6 and they represent the minimum resource needs and maximum allowance resource needs levels for that epoch. Affordable regions can change independently of one another as shown by the different horizontal bands.

The trajectory of the system over time in Figure 3-8 can be interpreted in the following manner: as the system traverses through the first 3 epochs while staying within the affordable region unique to each epoch, the system is *remaining affordable*. In the transition to Epoch 4, the system has now exceeded the maximum constraint level of the affordable region, thus *becoming unaffordable* by the end of the epoch. Finally, the system transits back to the affordable region in Epoch 5 and is said to be *becoming affordable*. The system state transitions of remaining affordable and becoming affordable are thus illustrated in the EEA diagram modified for affordability analysis.

Therefore, MATE, MAE and EEA can be combined to establish a tradespace-based method for affordability analysis and facilitate the search for design solutions that can remain affordable across a range of alternative futures. By explicitly accounting for cost, schedule and performance requirements over time, the method is able to account for system changes due to shifts and perturbations, manage lifecycle differences between subsystem components, evaluate feedback, and be adaptive to evolving system behaviors. As affordability is a concept evaluated over time, such a method can provide structured options for improvement to enable enhanced design for affordability.

3.11 Demonstration of Tradespace-based Methods in the Affordability Analysis for a Space Tug system and program

This section demonstrates the feasibility of tradespace-based methods in the design for affordability process through application to a Space Tug design case study. Only system and program level analysis were conducted.

3.11.1 Demonstration of System Level Analysis

To demonstrate how affordability analysis can be conducted using these methods, a simple case study involving the design of a Space Tug system and a Space Tug program is now presented. The Space Tug is a single general-purpose space transportation vehicle designed to transfer space systems between orbits.



Figure 3-9: A Space Tug system.

Described by McManus and Schuman (2003), a single Space Tug system is parameterized by three design variables: **manipulator capability**, **propulsion type**, and **fuel mass**. Manipulator capability can be low, medium, high or extreme; propulsion type can be storable bipropellant, cryogenic, electric or nuclear; and propellant mass can be 30, 100, 300, 600, 1200, 3000, 10000 or 30000kg. The design variables are summarized in Table 3-1. Enumerating all possible designs, there are 128 possible designs for a Space Tug system.

Table 3-1: Design Variables for a Space Tug System

Manipulator Capability	Propulsion Type	Propellant Mass
Low	Storable Bipropellant	30
Medium	Cryogenic	100
High	Electric	300
Extreme	Nuclear	600
		1200
		3000
		10000
		30000

Each Space Tug design will be assessed based on three performance attributes: mass capability, transfer speed and delta-V. The mass capability is measured in kilograms and it increases as the manipulator capability is increased from low to extreme. This is expected as the ability to drag spacecraft out of their orbits depends very much on the technology as well as size of the manipulator. However, having a larger manipulator also implies an increase in dry mass as more infrastructures is required on the Space Tug main frame to support the additional capability. The transfer speed is simply the speed of the Space Tug when moving from one orbit to another. The transfer speed is only slow when electric propulsion is used, and fast for the remaining propulsion technologies.

Finally, the maximum delta-V capability is calculated using the Tsiolkovsky rocket equation and it is a function of the specific impulse, initial mass including propellant mass, and final mass after transfer of orbit. The initial mass is simply the sum of the base mass, manipulator mass and propellant mass, while the final mass is equal to the sum of the base mass and manipulator mass only based on the assumption that all the fuel has been consumed to perform the orbital change. The specific impulse is dependent on the propulsion technology used on the satellite, where storable bipropellant gives the lowest value, followed by cryogenic, electric, and finally nuclear with the highest value. The attributes, as well as a summary of the interactions among their underlying variables, are shown in Table 3-2.

Table 3-2: Performance Attributes for a Space Tug System

Mass Capability	Transfer Speed	Delta-V
A function of manipulator capability	A function of propulsion type	A function of propulsion type which affects I_{SP}
Contributes to overall dry mass	Simply defined with only 2 levels: “Slow” (Level 0) or “Fast” (Level 1)	A function of wet mass which is determined by the amount of propellant
		A function of dry mass which comprises the base mass and manipulator mass

To perform affordability analysis, the system cost is broken down into its cost elements and in the case of the Space Tug, the elements are: **development cost**, **launch cost** and **development schedule**. In this demonstration, the development cost is simply a function of the dry mass of the Space Tug and it is calculated to be at $475/kg as recommended by (Wertz and Larsson 2011) in the development cost estimation for small satellites. The baseline schedule of the Space Tug has been determined to be 4, 8, 12, 18 months when built with low, medium, high and extreme manipulator capabilities respectively. This baseline schedule is multiplied by a factor of 1.5, 2.0, 2.5, and 3.5 when the Space Tug is equipped with storable bipropellant, cryogenic, electric and nuclear propulsion modules respectively.

The Space Tug has been designed to operate across 8 different missions as described by several authors, each of which has different levels of preference on each attribute depending on the nature of the mission (Fitzgerald 2012; McManus and Schuman, 2003; Fulcoly et al., 2012). Only one epoch variable, technology level, was chosen for the Space Tug system, and it can either be present or future. A technology level in the future can yield higher delta-V values, lower mass and increased capabilities, giving the same set designs higher values on their attributes. Given 8 different missions that can be conducted using either present or future technology, a total of 16 epochs are possible for the Space Tug system. This means that there are 16 tradespaces to be explored if one wishes to choose designs that remain on or close to the Pareto frontier for all 8 missions and do the same when technology levels transform to that of the future. In this demonstration example, 16 tradespaces were generated and it was observed that all 128 designs were scattered differently in each epoch due to changes in stakeholder preferences and epoch variables. 4 out of the 16 epochs were chosen and shown in Figure 3-10 and they show that the tradespaces do change over time as a result of different preferences on constraints. No further investigation on individual system tradespaces was conducted and program level analysis was demonstrated for this example.

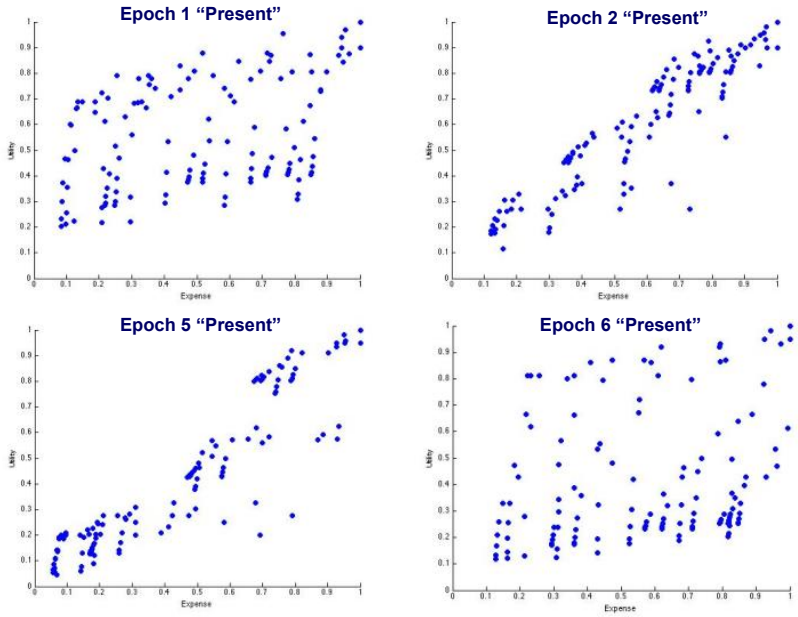


Figure 3-10: Tradespaces for the Space Tug system with added expense considerations change across different epochs as a result of changing preferences, mission needs and context.

3.11.2 Demonstration of Program Level Analysis

Program-level analysis has a higher degree of complexity in order to reflect a broader set of affordability considerations. The analysis serves as a preliminary demonstration of how the additional inclusion of cost and schedule parameters influence the spatial distribution of design points, and how tradespaces can become reflective of performance, cost and schedule considerations of importance to the stakeholders of a complex engineering project.

A hypothetical scenario is first introduced in order to formulate the mission concept for the program and it is described below:

"Due to satellite debris at various orbital altitudes and inclinations, many American satellites have been misaligned from their original orbits and been slightly damaged. As a result of the misalignment, more than 1 pair of satellites may collide into each other within the next 5 years and can potentially produce more debris in orbit. It has been determined in study that a single Space Tug will not be capable of realigning all satellites without incurring any risk of collisions. NASA needs to find a quick, effective but affordable solution to realign these satellites in order to prevent any collision and increase in orbital debris."

Based on the mission concept, a possible solution is to design a Space Tug program consisting of combined development and launch of two (possibly different) systems to achieve a more complex mission. MATE was conducted for the Space Tug program, with the MAE function used to calculate expenses of alternative programs. Epochs were then constructed for EEA.

For the purposes of EEA, 16 different epochs were constructed using 8 different preference sets for the program and one context variable, which is the technology. The epochs are thus constructed in the same way as that for a Space Tug system except that the needs are based on preferences on aggregated or emergent attributes of the program instead of a single spacecraft. No preferred designs were chosen at the system level, hence there are 128 possible designs for each Space Tug in the program. Program level design variables, namely the paired-reliability levels and paired-orbit locations and launch configuration, are also introduced and each variable can have four levels. Therefore, there are a total of 128 x 128 x 4 x 4 = 262,144 possible designs for the Space Tug program.

The Space Tug program has five performance attributes: **program mass capability, program delta-V, program transfer speed, probability of success, and mission time**. A number of simple interactions were introduced in order to produce attribute values that are reflective of emergent but not necessarily aggregated behavior from the sum of its constituent systems. As mass capability of a single vehicle is a function of vehicle manipulator capability, the program mass capability is the lower of the two values for the two Space Tugs, so that the program is able to fulfill at least the minimum requirement or better.

Delta V of a single vehicle is a function of the vehicle mass and specific impulse. Similar to mass capability, the program delta V is also the lower of the two system values. Transfer speed is the measure of how fast a vehicle can transfer between orbits and it can either be fast or slow as a result of propulsion type. With two tugs, the program transfer speed can exist in four combinations: Slow/Slow, Slow/Fast, Fast/Slow, Fast/Fast.

Each tug can have different reliability levels and orbit locations, which are introduced as new design variables to the Space Tug program model. The probability of success is calculated as the product of probabilities for each tug based on **paired-reliability level**, which can be in Low/Low, Low/High, High/Low, High/High configurations. The last performance attribute, mission time, is the duration taken to perform the mission. It can be long or short, which is indirectly dependent on orbit location.

Each vehicle can be orbiting in low earth orbit (LEO) or geostationary earth orbit (GEO). The **paired orbit locations** for the two vehicles can be LEO/LEO, LEO/GEO, GEO/LEO or GEO/GEO. For the launch configuration, if the two vehicles are in the same orbit, they can be launched at the same time on the same launch vehicle and can perform the mission quickly. If they have different orbits and different launch times, only one vehicle can be launched first and this hampers the speed at which the mission can be conducted.

The three expense attributes are: **program development cost (PDC), program launch cost (PLC) and program development schedule (PDS)**. The PDC is simply the sum of development costs for the two vehicles, which is the total cost required to develop the hardware of the Space Tug and is calculated as a function of dry mass. The PLC can either be the sum of launch costs of individual vehicles if they are launched to different orbits on separate launch vehicles, or two-thirds of the sum if they are launched to the same orbit on a single launch vehicle.

The launch cost of a single vehicle is a function of the wet and dry masses of a vehicle. The PDS will be the higher of the development schedules of the two vehicles if they are launched to the same orbit on a single launch vehicle, or the lower of the two if they are launched to different orbits on separate launch vehicles. The development schedule of a Space Tug increases with manipulator capability and complexity of propulsion type.

With five program performance attributes (PA) and three resource expense attributes, the tradespace model for the Space Tug program is a lot more complex than the system. A summary of all the program level attributes is shown in Table 3-3.

Table 3-3: Performance and Resource Expense Attributes for a Space Tug Program

Attributes	Description
(EA-1) Program Development Cost	Sum of development cost of individual Space Tugs
(EA-2) Program Launch Cost	Sum of launch costs of individual Space Tugs if in different orbits, else 2/3 of the value
(EA-3) Program Schedule	Maximum of the schedules of the two Space Tugs if launched to the same orbits, else the minimum
(PA-1) Program Mass Capability	The lower of the 2 Space Tugs in order to guarantee that the other one has higher delta-V
(PA-2) Program Delta-V	The lower of the 2 Space Tugs in order to guarantee that the other one has higher delta-V
(PA-3) Program Transfer Speed	Sum of Speed levels - “Slow-Slow” (Level 0), “Slow-Fast”/”Fast-Slow” (Level 1), “Fast-Fast” (Level 2)
(PA-4) Probability of Success	Probability of 2 Space Tugs being able to perform their missions at the same time
(PA-5) Mission Time	Duration taken to prevent the first predicted collision or multiple collisions predicted to occur at the same time

3.11.3 Single-Epoch Affordability Analysis for a Space Tug Program

A single-epoch affordability analysis was first conducted for the Space Tug program. The MAE and MAU values of all designs in Epoch 1 were calculated and they form the tradespace of the Space Tug program shown in Figure 3-10. To facilitate ease of analysis, six designs along the Pareto front were selected, which are labeled A to F in the direction of increasing expense alongside with their unique color and shape identifiers. Performance and resource attributes of the Space Tug program for these six are shown in Table 3-4.

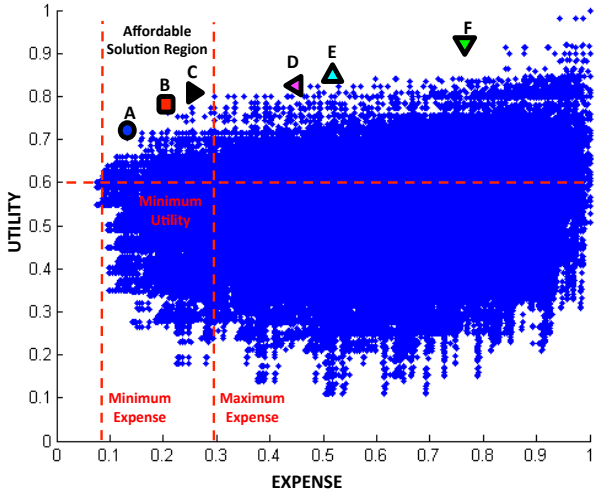


Figure 3-11: Tradespace for a Space Tug program in Epoch 1. 6 designs along the Pareto front were chosen and labeled A to F (Wu, Ross and Rhodes, 2014).

Commonalities among the designs include Fast/Fast program speed, short mission time, low mass program payloads and relatively short development schedules. Not reflected in Table 1 are that all designs are in LEO/LEO orbit and High/High reliability configurations. Large ranges of values for program Delta-V, PDC and PLC are observed as the designs chosen were spaced apart. Constraints for performance and resource attributes, which have values greater than minimum preference levels of stakeholders were established for Epoch 1, yielding values for the constraint levels on minimum utility and maximum expense, as well as the derived minimum expected expense for this particular epoch.

As a first-pass analysis, constraint levels on performance attributes in Epoch 1 are set at values slightly lower than the attributes of Design A (see Table 3-4). Constraint levels on resource attributes are set at a multiplicative factor of 1.5 to 2 times of the values for Design A. The resultant constraint level values for attributes are shown in Table 2. The constraint levels define the affordable solution region for Epoch 1 and are shown in Figure 3-11. Designs A, B and C are affordable solutions within this epoch, while D, E, and F are not (they violate maximum expense constraint). Single-epoch analysis is straightforward after calculating the constraint levels and establishing the affordable solution space.

Table 3-4: Performance and Resource Attributes for Designs A to F in Epoch 1.

	Performance Attributes				*Resource Attributes*					
Design (Number)	Program Payload (*kg*)	Program Speed	Program Delta-V (ms^{-1})	Prob. Success	Mission Time	PDC (*$mil*)	PLC (*$mil*)	PDS (*mths*)	Utility	Expense
A (26836)	300	Fast/Fast	6147	0.96	Short	940.5	376.8	8	0.715	0.131
B (28900)	300	Fast/Fast	8091	0.96	Short	1805	808	8	0.774	0.208
C (59860)	300	Fast/Fast	12645	0.92	Short	2090	764	14	0.800	0.254
D (125908)	1000	Fast/Fast	8910	0.88	Short	3420	1212	28	0.823	0.448
E (127972)	1000	Fast/Fast	16150	0.88	Short	4750	1800	28	0.840	0.517
F (194020)	3000	Fast/Fast	10984	0.86	Short	8550	3080	42	0.915	0.763

3.11.4 Multi-Epoch and Single-Era Analysis for a Space Tug program

As programs operate in dynamic environments over their lifecycle, it is important to find out how the utility and expense of the program changes across multiple epochs. Multi-epoch analysis can be performed to find out how many epochs during which designs remain affordable. Epochs 1, 5, 6, 13 and 14 were chosen for multi-epoch analysis, as the expense preferences were most distinct from one another. Varying constraint levels for performance and resource attributes were chosen for each epoch, giving rise to different utility and expense constraint levels that yield different affordable solution regions. The constraint values and the resultant constraint levels are shown in Table 3-5.

Table 3-5: Performance and Resource Constraints for a Set of Epochs (sequenced as an Era) (Epochs 1,5,6,13,14)

	Performance Constraints					*Resource Constraints*			*Constraint Levels*		
Epoch	Program Payload (*kg*)	Program Speed	Program Delta-V (ms^{-1})	Prob. Success	Mission Time	PDC (*$mil*)	PLC (*$mil*)	PDS (*mths*)	Minimum Utility	Derived Minimum Expense	Maximum Expense
1	300	Fast/Fast	5500	0.95	Short	2000	500	12	0.605	0.087	0.294
5	300	Fast/Fast	7000	0.90	Short	3000	900	12	0.661	0.178	0.349
6	1000	Fast/Fast	4000	0.90	Short	4000	1000	12	0.576	0.293	0.389
13	300	Fast/Fast	6500	0.95	Short	5000	1200	24	0.643	0.079	0.583
14	1000	Fast/Fast	7000	0.90	Short	3000	1600	24	0.681	0.175	0.527

In a simple demonstration of multi-epoch analysis, the expenses of all designs across all epochs are studied. Figure 3-12(a) shows that Designs A, B and C are affordable in most epochs, but only Design C is affordable in all 5 epochs. From Figure 3-12(b), all designs except for A are always above the minimum performance constraints across all epochs.

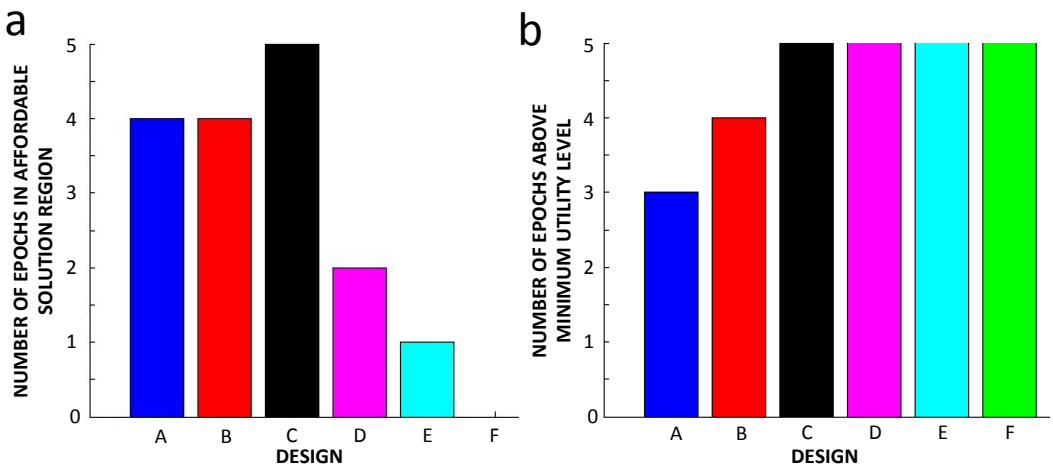


Figure 3-12: (a) Number of epochs in affordable solution region for every design; (b) Number of epochs above minimum utility level for every design (Wu, Ross and Rhodes, 2014).

Multi-epoch analysis becomes single-era analysis when the epochs are viewed as an ordered sequence that fits the program lifecycle. Era analysis requires the tracing of both expense and utility trajectories of designs over the defined era. Tracing the trajectories of the design utilities over the era in Figure 3-13 shows that Designs B to F always remain above the minimum utility constraint levels throughout the era and are thus possible candidates for the final design. However, the value of considering resources in addition to performance comes in tracing the expense trajectories for these designs. Figure 3-12 shows that Designs A and B become unaffordable in the transition to Epoch 8, but becomes affordable again later in Epochs 13 and 14, while Designs D and E have only one instance of being affordable in Epoch 13. Design F is the most expensive and remains unaffordable throughout the era. As such, only Design C remains within the affordable solution regions across all ordered epochs and it is the most affordable solution in this constructed era.

Combining the results from tracing both utility and expense trajectories, it can be seen that Design C (see Table 3-4) has the best tradeoffs among performance, cost and schedule attributes over time. Given its midrange values for all performance and expense attributes, Design C is indeed the most affordable solution that is always above the minimum utility constraint levels as seen in Figure 3-13. Both Space Tugs have '**Low**' mass capability, use '**Nuclear**' propulsion, propellant mass of **3000kg**, in **LEO-LEO** orbit configuration, **high reliability**, and are carried on the **same launch vehicle**. The PDC is **$2.09 billion**, PLC is **$0.764 billion**, and development schedule is at least **14 months**. It remains feasible relative to the resource needs and resource constraints over the era. **Through program-level analysis, the stakeholders can know which combination of design variables to use to design the Space Tug program that will exhibit acceptable levels for performance, cost and schedule attributes.** A portfolio level analysis will not be conducted for this example, but it will entail taking similar procedures and determining new performance and expense attributes for the portfolio. Conducting affordability analysis using tradespace-based methods in the form of MATE and EEA thus facilitates a resource-centric approach in the down-selection and identification of affordable designs.

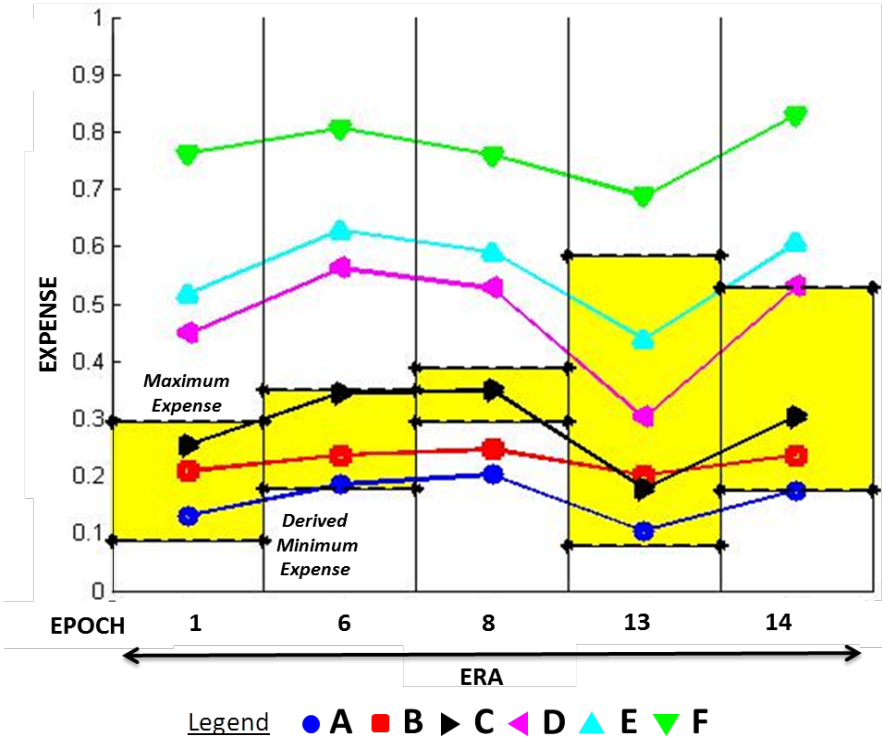


Figure 3-13: EEA with expense considerations in a single era (Wu, Ross and Rhodes, 2014).

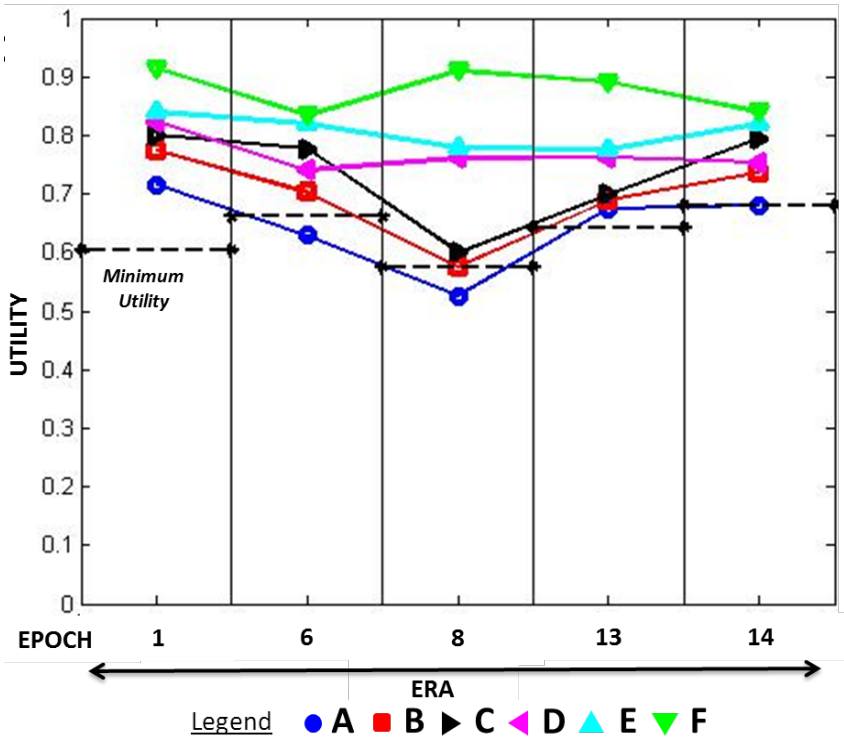


Figure 3-14: EEA with utility considerations in a single era (Wu, Ross and Rhodes, 2014).

3.12 Summary

Affordability is defined as the property of becoming or remaining feasible relative to resource needs and resource constraints over time. Affordability can be treated as an ility that drives the design of more affordable yet technically sound architectures.

To conduct affordability analysis and perform affordability tradeoffs during conceptual design, methods for systems engineering tradeoff analysis are required to demonstrate changes in resource expenses as major decision parameters and times to completion are varied. The minimization of resource expenses, while maintaining or increasing performance specifications across changing contexts over time, motivates the construction of tradespaces with considerations of temporality. Leveraging the increased availability of computation power, affordability analysis can be conducted through *tradespace exploration*, which is the model-based investigation of many design alternatives in order to find better design solutions, while avoiding premature fixation on point designs and narrow requirements.

Tradespace exploration allows a holistic consideration of capabilities and mission utility during early-phase design, instead of being locked too early into requirements and key performance parameters. As tradespace exploration entails the enumeration and evaluation of a large number of potential designs, this method is most relevant to the design of complex engineering systems with multiple dimensions of benefits and expenses, which are often difficult to optimize and rarely intuitive. The use of tradespaces instead of simple tradeoffs of several point designs can thus lead to better lifecycle results for the system or program of interest.

As tradespace exploration enables the promulgation of affordability as an ility, tradespace-based methods are introduced for designing for affordability in systems or programs. With complex engineering systems as the target application, *Multi-Attribute Tradespace Exploration (MATE)* can be used in the value-driven search for affordable designs by aggregating multiple dimensions of benefits into a single utility metric. Tradespaces have been traditionally viewed as two-dimensional plots bounded by the parameters of utility and costs, representing the high level tradeoff of "what you put in" (i.e. cost) and "what you get out" (i.e. utility). Since the design process considers more than just cost, "cost" can be replaced with an aggregate measure for resource expenses to enable affordability analysis.

The *Multi-Attribute Expense (MAE)* function can hence be used to aggregate cost, schedule and other non-monetary factors into a single expense metric. Design points on a tradespace can then be compared using their unique MAE and MAU metrics, which are representative of all performance and resource expense attributes, as well as their corresponding stakeholder preferences. Constraint levels for each attribute are also determined and aggregated to form MAE and MAU constraint levels for the tradespace in each epoch. The area bounded by the maximum MAE, minimum MAU, and derived expected minimum MAE is then the affordable solution region, in which the most affordable solutions are most likely to be located. By searching in the narrowed affordable solution regions in every tradespace space for every epoch, affordable solutions for systems, programs and portfolios can be found.

Finally, to account for how the performance, cost and schedule attributes of a system or program evolve over time across dynamic operating environments, *Epoch-Era Analysis (EEA)* will be used. EEA is a design approach used to clarify the impacts of time and context on the value of the

system or a program, and can be modified and applied to enable affordability analysis over multiple epochs (periods of fixed contexts) and multiple eras (ordered sequences of epochs). Trajectories that track the changes in utility and expense of a system, program or portfolio can be plotted to determine which solutions are fully affordable or partially affordable across a period of time.

The methods introduced were then applied to a simple Space Tug case study, which has few design variables and attributes to consider. Interactions between the design variables, epoch variables and attributes were relatively straightforward, and the program level analysis was conducted simply by putting two Space Tug designs together and determining which combination was the most technically superior and affordable at most times. A preferred program design could be found using these methods, and it has been assessed to be technically adequate and affordable at all times. However, a portfolio level analysis was not conducted in this demonstration example. Expectedly, a program level analysis became more complex with more design variables, performance attributes and expense attributes. This complexity can only increase when the design for affordability process cascades down to the portfolio level in the bottom-up approach.

Systems, programs and portfolios in the real world are much more complex, and there are so many considerations to be taken into account during the design process. As such, a more convenient way to proceed forward from system to portfolio will be to down-select a handful of preferred designs at the end of each level, which will become design variables at the next level of analysis. This narrows down the number of possible designs for systems, programs and portfolios, enabling a more focused and easy path for tradespace exploration. The lessons learnt from this demonstration example and the methods introduced in this chapter will be applied to a case study with a greater degree of complexity.

4 FEDERATED SATELLITE SYSTEMS SYSTEM ANALYSIS

The tradespace-based methods used in the design for affordability process are applied to the design and implementation of Federated Satellite Systems (FSS) (Golkar, 2013). The FSS is a new spacecraft operation paradigm and it comprises many heterogeneous satellite constellations or monolithic satellites in Low-Earth Orbit (LEO) working together to function as a supplier base for in-orbit data storage and processing capacity. The FSS is catered to customer spacecraft in other orbits that do not have direct access to ground stations during critical periods of time in operation and choose to transfer data via inter-satellite links (ISL) to supplier spacecraft for temporary storage or processing in-situ. Such an operation is analogous to having Cloud Computing in space. This can potentially revolutionize the design and operation of future spacecraft, as they can leverage the communications and data handling capabilities of existing spacecraft and operate without dedicated subsystems for these capabilities. This implies the establishment of a new space communications network, reduced overall dry mass, and greater capacity for science payloads that can greatly support human endeavors in space and beyond.

Based on a high-level assessment of its design and mission requirements, the FSS can be seen as a SoS that requires multiple phases of development and launching different satellites in order to achieve its desired capability. Designing for satisfactory performance in such a large system is already a challenge in itself. However, in an age of decreasing budgets and little margin for cost and schedule overruns, it is an even greater challenge to design the FSS for affordability given the multitude of cost and schedule attributes to be considered for every satellite and every constellation. Should one satellite or one constellation experience breaches in cost and schedule budgets, it may impede the overall development of the entire FSS. Performing multi-dimensional tradeoffs among various attributes for the FSS can thus be challenging.

Searching for a technically feasible yet affordable solution during early-phase design may be cognitively demanding on decision makers tasked with operationalizing single monolithic satellites, satellite constellations, or multiple satellite constellations. Decision makers at different levels of FSS development may also be interested in different performance and resource expense attributes, and have different preference levels on each of them depending on the operating context of their entity. Also, the development of new space systems often overlaps with matters related to national security, national budgets and congressional decisions, which are usually considered external to the design process but can impact the selection of preferred designs.

The complexity and scale of this design problem as well as the need to include both internal and external design considerations would benefit from the use of **tradespace-based methods in the design for affordability of the FSS**. The development of the FSS can be considered as the development of a portfolio of various satellite constellation programs. With the application of these methods, the mission needs, stakeholder preferences, design variables and epoch variables for every entity in the FSS can be designed with holistic considerations of performance, cost and schedule. Taking a **bottom-up approach** in this case study can thus ensure that preferred designs for the FSS **remain affordable at the system, program and portfolio level over time**.

4.1 The Federated Satellite Systems Paradigm

The FSS is an advanced instance of distributed satellite systems (DSS), which have been defined as missions whereby multiple satellites collaborate to fulfill a mission. A federated approach to space systems architecting and spacecraft design can potentially revolutionize the design of future spacecraft missions and infrastructure. Spacecraft participating in the FSS can conduct the opportunistic sharing of in-space resources such as computing capacity, communication links, and storage capacity with other spacecraft. The FSS can function based on different exchange mechanisms such as free sharing and market-based trading mechanisms. While the FSS may also appear to be a form of fractionated satellite systems (Brown and Eremenko, 2006) due to the distribution of subsystem capabilities across multiple spacecraft, federated and fractionated concepts are very different in nature. Spacecraft in a FSS are not designed to work as collaborative units from the outset. They share resources on an opportunistic basis, and form temporary networks enabled by integrated hardware or hosted payloads acting as middleware (Golkar, 2013).

If implemented in the near future, it can substantially improve the sustainability of new missions by enabling the use of different satellites for multiple Earth Observation and interplanetary exploration missions that would not be feasible without the support of FSS infrastructure (Golkar, 2013). It will also make space missions more reliable, as the FSS is inherently redundant given the number of participating satellites that all provide the same capability. It will also make space missions more cost effective, as space assets in the form of data storage and processing capacity will be maximally utilized at all times. Lastly, it will provide significant mitigation of demand uncertainty, as it will allow them to rely on federated services to accommodate increased market demand in satellite services (Golkar, 2013). Therefore, the FSS can potentially provide a whole new level of sustainability, reliability, efficiency and confidence to space operations and space systems architecting.

However, the FSS concept has not been thoroughly explored due to the lack of maturity in critical technologies for operation. Much of the FSS potential relies on the ability of satellites in LEO to communicate and eventually exchange data via inter-satellite links (ISL). While much research and development of ISL has been conducted in recent decades, there is still a major gap in knowledge and technology required to realize an in-space “networks of networks” for the opportunistic sharing of resources (Golkar, 2013). Furthermore, the FSS cannot be implemented using existing LEO satellites, as they were designed to perform heterogeneous missions and do not have interfaces required for efficient ISL operation.

Most of these existing satellites contain a significant amount of untapped and unused resources in orbit when they are not performing their missions or not within downlink range of ground stations. (Lluch and Golkar, 2014) define “resources” as any commodity or service that could be potentially traded or shared between spacecraft. However, to avoid confusion with the same term defined specifically in the design for affordability using tradespace-based methods, this in-orbit commodity or service on a participating spacecraft will be herein referred to as “data assets”. The underutilization of data assets is thus a lost opportunity in enhanced performance and reliability of space systems, as well as a lost opportunity for creating in-orbit markets of space commodities (Golkar, 2013). Should the usage of these excess data assets be maximized, this can potentially

yield a profitable market for the exchange of such assets and change the way spacecraft are designed and how space missions are conducted in the future. It is this gap in market opportunity and knowledge that the implementation of the FSS seeks to bridge.

4.2 FSS Architectural Assumptions

The conduct of early-phase design is always necessary in order to determine key concepts that will facilitate the successful development of a complex system. The FSS is one such complex system that has to be designed over many years and managed by many people and institutions in the defense and aerospace industry in order to achieve its envisioned capability. This is a large sociotechnical system that potentially has many mission requirements as well as designed to operate in evolving contexts. The introduction of this chapter described how tradespace-based methods can be applied to explicitly capture various elements of this complexity and allow decision makers to make better informed choices of potential FSS designs based on more holistic considerations of performance, cost and schedule attributes.

As the FSS has not been implemented in the real-world and there are few precedents for reference, a **storyline for the development and implementation of the FSS** has to be proposed in order to create the decision makers, mission concepts, design variables, epoch variables, stakeholder preferences that enable the use of tradespace-based methods in system lifecycle design. In addition, a number of **architectural assumptions** will also have to be made to focus the design process on decisions that directly enable the FSS capability. In this case study, satellites participating in the FSS are not designed on the outset to be solely supporting FSS purposes. The assumptions made will also have significant impacts on how the FSS development time is formulated.

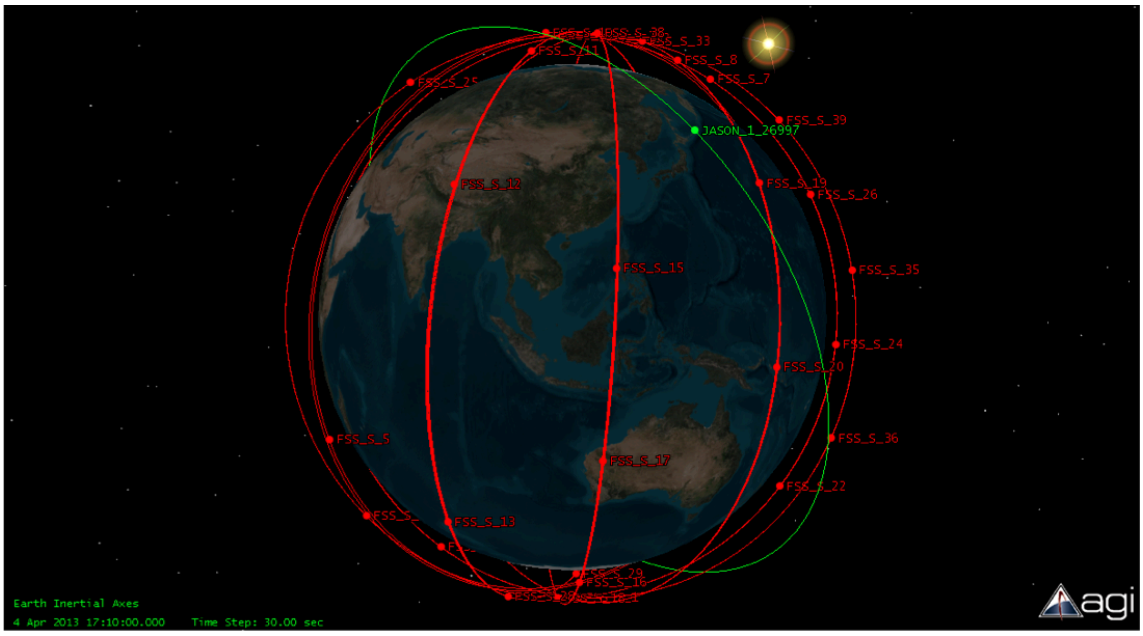


Figure 4-1: FSS Earth Observation Support Infrastructure Example in STK© Visual Representation (Golkar, 2013)

This section now aims to describe the major assumptions made to enable the application of tradespace-based methods for affordability analysis. These assumptions are important as they help focus the design process and narrow down the design variables required for developing the FSS. After making these assumptions, a hypothetical development time for the FSS will be proposed and described in order to facilitate epoch and era construction. Finally, the design methodology and approach taken for affordability analysis will be described in detail.

Assumption 1: New static satellites are needed for the FSS.

The first assumption to be made in this case study is that the construction and launching of new satellites is the only way to achieve the envisioned FSS capability. It was mentioned earlier that existing satellites in orbit do not have the communication interfaces necessary for ISL operation. Given that it is economically unsound and technically challenging to send robots or astronauts to adapt these satellites with FSS-enabling communication devices, the most logical way to establish the FSS will be to launch new satellite constellations. The new satellites will have FSS-enabling communication devices that allow them to communicate with one another via efficient ISL. Assuming that satellites are constructed and launched from Earth also facilitates validation and verification, as there are numerous cost models and design methods available for use, such as those recommended by Wertz and Larson (2011). These satellites are also assumed to be static, meaning that its functionality and attributes are not changed if it is already in operation. Hence, a satellite constructed using present day technology will still maintain present day performance levels, even if a new technology emerges in the future. Only a few satellites constructed in that future time period will be imbued with that future technology and exhibit enhanced performance levels. Therefore, if the FSS initiative is conceptualized in present day, new static satellites will then be designed based on scientific principles derived from present day technology and realistic operating contexts. These satellites will not change over time once they are in operation.

Assumption 2: In this case study, only large satellites are considered and they are always part of a constellation.

In this case study, it is assumed that a participating satellite in the FSS is a large satellite and always part of a greater constellation containing other satellites, such that there is no single satellite with a standalone design and mission agenda that is totally different with respect to other participating satellites. In reality, small satellites can also participate in the FSS, but they are not considered in this case study. By assuming that satellites are large, they can function as major data nodes in the FSS network and this facilitates ease of tradespace analysis as the total number of participating satellites in the FSS becomes manageable. Having a standalone satellite in the FSS is also technically possible. A possible spacecraft is the International Space Station (ISS), which is visited by astronauts from different nations multiple times a year and is continuously resupplied. Relative to other spacecraft that do not enjoy such infrastructural and logistical support, the ISS can potentially be modified to have FSS-enabling communication interfaces and even function as a major node within the FSS network.

The concept of using the ISS for FSS purposes has also been studied by (Golkar, 2013) and it has been shown to be a technically feasible option. The ISS can even be a major node in the FSS network. However, for the purposes of managing the design complexity inherent in this case study, it is assumed that participating satellites in the FSS are large but not on any scale or

versatility comparable to the level of large spacecraft like the ISS. Since the aim of developing the FSS through launching new satellites is to provide an adequate supplier base of data assets, it is more logical to construct multiple units of the same satellite that may be carried by a single launch vehicle or may collaborate together to deliver an emergent capability. Constellations, rather than single standalone units, are thus assumed to be logical base unit for developing the FSS in a quickly, timely and affordable manner.

Assumption 3: The new satellites are homogeneous within their own constellations, but heterogeneous between different constellations.

The designs of these new satellites will be homogeneous within their own constellations, but designs between constellations will be different. For each constellation, it is thus assumed that all satellites are similar in design, and they are developed, launched, operated or retired in the same manner at any point in time. Therefore, a performance attribute for a constellation may simply be the sum of its parts across its constituent homogeneous satellites. If the constellation is designed to be equipped with an emergent capability, then the value of the performance attribute may then be greater than the sum of its parts.

Different satellite constellations will have their unique payload capacity so that they can carry different science payloads. As such, each satellite constellation will have different contributions to scientific value, depending on how it is measured. This assumption is made in order to align the case study to that of the envisioned FSS concept, where heterogeneous satellites participate in a network to provide a supplier base for sharing in-space data assets on an opportunistic basis.

Having different satellite constellations will also result in having different attribute values for each constellation, and this makes program and portfolio level affordability analysis more meaningful than just simply having the sum of similar parts. Since a constellation is the base unit for the FSS, creating multiple constellations with each containing a handful of satellites rather than one large constellation containing many dozen satellites like Iridium or the GPS can deliver a more diverse portfolio of capabilities and scientific value. Having different constellations also implies a disaggregated approach to constructing the FSS, which is believed to be cheaper and more reliable than an aggregated one.

Assumption 4: ISL technology is existent and all FSS satellites are equipped with an additional communication interface for ISL capabilities that is compatible with other subsystems.

All new satellites launched to fulfill the FSS objectives will be equipped with additional communication interfaces designed to facilitate efficient ISL. This in turn is based on the assumption that ISL technology is mature to an extent such that data exchange between satellites can occur within a reasonable slant range, at adequately high levels of network efficiency with few dropped data packets, with low handover time, and adequately high access time for the required data handling tasks. Adding this interface means that there will be an increase in the dry mass of every satellite, as it is an extension to the communications subsystem that is required of every satellite and designed to complement any existing Earth-pointing or omni-directional antennas. Therefore, the new satellites will be able to communicate in the same manner as any existing satellite, but will have the additional capability to establish efficient ISL. This additional interface is also assumed to be radiation-hardened and been tested for compliance and

compatibility with other existing subsystems. Critical subsystems such as power, thermal, structures etc. have been assumed to be compatible with this new interface such that design concepts and information as recommended by (Wertz and Larson 2011) can be used in the case study model.

Assumption 5: Laser Technology is in its nascent stages and has immense potential to develop and mature in the near future.

Apart from radio frequency based communications systems or interfaces, laser technology is assumed to be a feasible means of communication and data transmission. It is assumed that at present day, laser technology exists and but its TRL is low. It is still costly, unreliable, and unable to deliver the high-speed communication channels it promises. However, it is also assumed that research and development in space laser communications will advance rapidly in coming years such that it will become mature within a decade or two. When mature, TRL is high and laser technology will be cheaper than usual, more reliable, and be able to deliver communications capabilities beyond that of any existing medium. Using laser technology can hence produce an unprecedented level of capability for communications and data exchange, which is greatly beneficial to the operation of the FSS.

Assumption 6: Current launch vehicles are available for launching constellations, but Reusable Launch Vehicles may become a realistic option in the near future.

The new satellite constellations require a launch vehicle (LV) to reach the orbit required for operation. Although there are a number of LVs available worldwide for selection, only 4 established LVs were chosen for this case study. They are the Falcon 9, Falcon 9 Heavy, Ariane V, and Atlas V. These LVs have different launch costs and payload capacities, and depending on the size of the constellation, some LVs may be more suitable for certain constellations. The launch costs will hence be directly or indirectly a function of the total mass of the satellite constellation to be launched. While it may still be a distant reality, it is majorly assumed that Reusable Launch Vehicles (RLVs) are a possibility and can be greatly beneficial to the development of the FSS. With RLV, the launch costs for a constellation can potentially be halved and this implies cost savings for FSS development. This is not an impossible scenario in the near future, as the commercial space corporation SpaceX is already in the process of designing such a RLV. The launch costs and the payload capacities for the 4 LVs being considered in this study are shown in Table 4-1.

Table 4-1: Launch cost and payload capacity for selected current launch vehicles.

Launch Vehicle	Launch Cost ($ million)	Payload Capacity (kg)
Falcon 9	56.5	13,500
Falcon 9 Heavy	135	53,000
Ariane V	167.7	16,000
Atlas V	395.6	30,000

Assumption 7: The new satellites are to be launched from the same location to the same orbit inclination angle and altitude, and operated at LEO (Low Earth Orbit). They are all in near circular orbits, have the same constellation configuration, and will have the capability to distribute themselves uniformly across all orbit planes and orbit inclinations in order to maximize Earth coverage as well as accessibility to customer spacecraft.

A promising application of the FSS is to establish opportunistic ISL for data asset sharing between satellites in LEO. Similar forms of ISL have already been implemented in satellite constellations such as Iridium and designed from the outset as predefined communications architecture between involved spacecraft (Golkar, 2013). Since the new satellites forming the FSS will be the first of its kind, operating at LEO reduces the launch cost per satellite and allows more satellites to be launched on a single LV. This implies that constellations can also be formed more quickly, thereby expediting the development of the FSS.

It is assumed that they are all launched from the same location and will end up at the same orbit inclination angle and altitude at LEO. In reality, the choice of launch locations, the final orbit inclination angle, and the final orbit altitude can have considerable impacts on the launch costs, minimum delta-V required and the amount of onboard propellant required. However, this assumption is made to manage the design complexity for the early-phase design process in this case study and eliminates the need to factor in such considerations. Also, by further assuming that they all have the same orbit parameters and Walker constellation configuration immediately after launch, the need to consider subtle impacts to cost, schedule and performance are again avoided in this study. However, these parameters and the constellation configurations can be changed, as it has been assumed that these new satellites have been designed with propulsion capabilities and carry onboard propellant to perform orbit altitude or inclination changes when necessary.

To manage the complexity in this case study, it is assumed that all satellites in the FSS are in near circular orbits with low degrees of eccentricity. In reality, satellites in different constellations can operate in other orbits or at higher degrees of eccentricity, but that introduces the need to consider how ISL can possibly be established at varying distances. By assuming that they are all close to Earth and distributed uniformly across the celestial sphere around it, it can be assumed that the range between each pair of satellites is approximately equal. If there are enough FSS satellites in LEO, it can also be further assumed that customer spacecraft at MEO, GEO or other orbital altitudes can tap on FSS capabilities virtually anywhere in LEO.

Since the satellites are assumed to have the capability to distribute uniformly in LEO, they must have a propulsion capability and must carry some amount of propellant onboard. Depending on the needs of the constellation, propulsion capabilities and propellant mass can vary and they should be key design variables in the satellite system and constellation in order for this assumption to hold. Stronger propulsion capability and higher amount of onboard propellant will enable the satellite to move freely within LEO at all orbital altitudes and inclination angles, and even beyond. However, the tradeoff for such a capability will be higher costs and longer development time.

Assumption 8: Most satellites are designed to have data storage and processing capacity, which can serve as data assets to be utilized in the FSS.

Given that the new satellites are built with the intention that they will eventually be participating in the FSS, they are assumed to have adequate data storage and processing capacity that can serve as the data assets required for opportunistic sharing among customer spacecraft. All satellites are designed with an initial data capacity, and it is assumed that a certain proportion of that capacity will be used to perform the satellite's daily communication and data handling tasks as well as unique science missions. The free capacity on each satellite will be the difference between the initial capacity and used capacity. Depending on the amount of initial capacity or the rate of data capacity usage by the payload, some satellites may have zero free capacity while others may have a high percentage of capacity that remains unused. Satellites with some free capacity will hence be more useful in the FSS. However, they may not providing adequate science value as a result of underutilization and may be considered ineffective.

Assumption 9: All satellites carry a payload to perform unique science missions and the payloads are always in operation. The payloads are the only consumers of data storage and process capacity on the satellite.

It is assumed that all satellites will carry a payload designed to perform a science mission on top of fulfilling objectives for the FSS. The payloads can vary greatly in mass from satellite to satellite and the data usage rates will also vary. Such an assumption is a valid representation of reality. It is also assumed that the payloads are always in operation and will hence be consuming data all the time. This may not be true in reality, as payloads can be switched to idle mode if they are unable to perform their science missions owing to various circumstances. However, if downtimes for the satellites are considered, then the data capacity per satellite can vary very greatly and there is also a need to consider scheduling of data exchange, duration of downtime, and many other factors. By assuming that they are always working, the data usage is constant and the free data capacity on a satellite will likewise be approximately constant. Assuming a constant data usage also means that an upper bound for data usage is always applied. This in turn means that a more conservative estimate will be used to measure a satellite's free data capacity, which can have a greater value in reality as compared to the estimated value in the study. It is also assumed that the payloads are the only consumers of data storage and capacity onboard every satellite. The effects of other subsystems like attitude, determination and control or telemetry and tracking on data capacity are hence negligible.

Assumption 10: A safe and reliable FSS network protocol has been established.

It is assumed that a safe and reliable FSS network protocol has been established, where the physical and application layers of the opportunistic asset sharing platform is built into the communications subsystem and software of every satellite. Therefore, a FSS middleware is installed in all missions, allowing intercommunications between heterogeneous standards acting as a meta-layer similar to what is done in Internet communications (Golkar, 2013). This assumption is made to manage the complexity of the case study, as this eliminates the need to consider different types of routing, packet scheduling and communications protocols that have different degrees of efficiency and compliance standards. The aim of this case study is to

demonstrate the conduct of early-phase design using tradespace-based methods, and that only high-level architectural decisions are of importance.

Assumption 11: The new satellites have capabilities and scientific agendas beyond fulfilling FSS objectives.

The new satellites are not designed like the ones in the Iridium constellation and can provide more capabilities beyond communications. The satellites may be designed for a variety of purposes such as observation, navigation, remote sensing, reconnaissance or military applications. To manage the complexity of the study, these purposes will not be considered explicitly, as the degree to which any of these purposes can be fulfilled can vary greatly and result in a large range of design possibilities for a satellite. Different purposes are factored in this study through varying the payload capacity and data usage rate of the satellite. A greater payload capacity and/or a greater data usage rate will imply a greater degree to which a designed purpose is being satisfied.

Assumption 12: There is a global FSS Directorate to manage the FSS Initiative.

Decision makers are key to the architectural decisions made for the envisioned FSS capability and it is assumed on the outset that a global FSS Directorate (FSSD) has been established to lead the initiative in its research, development, implementation, operation and possibly retirement. The FSSD will comprise key decision makers in the design for affordability process. The FSSD may comprise senior officials from both the commercial and military space sectors from various countries to form the management board. Possible participants in the FSS initiative may include NASA, ESA, JAXA, CSA, CNSA, UKSA and the US Air Force (USAF). In this case study, a participating agency will be referred to as a DM and affixed with a number index.

The FSSD is responsible for funding, scheduling and assessing the development of the FSS. Since the FSSD comprises representatives from participating agencies, the interests of the FSSD are hence aligned to that of all DMs. The main difference is that the FSSD will function as the governing body for the FSS initiative and will be responsible for conducting early-phase design and analysis, as well as providing solutions and development strategies to each DM in order to coordinate development of the FSS.

Each DM is assumed to be interested in launching more of its own monolithic satellites or satellite constellations in the coming years to boost their space capabilities. As participants in the FSSD, the DMs are also interested to see that their satellites can also help contribute to the FSS effort on top of their own science mission agendas. The DMs have also agreed to stagger their satellite development programs over the span of the development timeline to avoid a massive cost commitment in the first few years of the FSS initiative. Also, the DMs also have their own projects and budgets they will like to adhere to, and they are thus not likely to develop satellites all at the same time. The FSSD has also proposed a list of missions that participating satellites and constellations in the FSS should be able to perform, which all DMs have agreed to conform to for their respective satellite projects. The FSSD will be responsible for determining the common design variables that will be used for all satellite projects, as well as the epoch variables that will determine the changes in operating contexts. The FSSD has also elicited preferences of each DM for the satellites, so the preferences for each attribute for the FSS and any of its constituents will be representative of all DMs and the FSSD.

Assumption 13: The FSS is an attractive initiative to all DMs.

In this case study, it is assumed that the FSS is an attractive initiative that is well received by many nations that have satellite capabilities. It has immense potential to become technically feasible and profitable. While no detailed analysis has been performed to assess its technical feasibility and profitability, it has been assumed that the FSS initiative is well received in order to justify the formation of the FSSD and the responsibilities of participating DMs in developing their respective satellite constellations. Following this assumption, private and public corporations as well as academic institutions have recognized the importance of minimizing underutilization of space-based assets and consensually agreed that the opportunistic sharing of these data assets can potentially create a whole new market and revolutionize space systems development. Conducting a high-level design of the FSS can thus help the FSSD determine in future if there exists a potential market where customer spacecraft may choose to leverage FSS capabilities instead of using conventional data handling methods via downlinks to Earth.

By first assuming that the FSS is an attractive, and potentially viable and profitable initiative, the FSSD and its members can focus on performing early-phase design of single satellites, constellations and multiple constellations without having to perform a cost-benefit analysis and comparison of the FSS with conventional data handling methods. Also, it can be further assumed that funding and time will always be available for all satellite constellations that are developed in support of the cause. However, the amount of funding and time available will vary from constellation to constellation since constellations are assumed to be heterogeneous.

Assumption 14: From System to Program to Portfolio using the Bottom-Up Approach

The previous chapter discusses how the design for affordability process can be conducted for a system, program, and portfolio using either the bottom-up or top-down approach. As a logical development timeline for the FSS is the construction of single satellites, then the construction of satellite constellations, and finally the integration of multiple constellations, it is thus most meaningful to take the bottom-up approach in this case study. From herein, a single satellite is defined as a system, a satellite constellation is defined as a program, and multiple constellations or the entire FSS is defined as a portfolio.

4.3 System Definition

Affordability analysis for FSS begins at the system level, which is the design of a monolithic satellite with FSS-enabling capabilities. Based on the earlier assumptions, the FSSD and DMs have been able to narrow the focus of their early-phase design to a number of system performance and resource expense attributes. They are also fully aware of the design for affordability process and are familiar with the need to consider various cost and schedule elements on top of performance.

It has been established that the key decision makers at all levels of the design process, be it system, program or portfolio, are the FSSD and all the DMs.

The mission concept at the system level is then to develop a single satellite that has FSS-enabling capabilities, namely *stakeholder-desired* levels for ISL capability, annual free data

capacity, annual science value, and delta-V, as well as *stakeholder-desired* levels of expenditure for development cost, launch cost, and development time.

The **performance, cost and schedule attributes** are thus derived from the system-level mission statement. These attributes are listed and described in Table 4-2, where system-level performance attributes are labeled SYS-PA and system-level resource expense attributes are labeled SYS-EA. Attributes across performance, cost and time are considered for the system-level analysis in this case study. **It is also assumed that all attributes are independent of each other such that the stakeholder preferences for one attribute have no impact on the preference for another.**

Table 4-2: System Level Performance and Expense Attributes

System Level Attributes	Description
SYS-PA-1 **ISL CAPABILITY**	The **ISL capability** is a proxy measure of the spectral bandwidth and spectral efficiency of the communications channel that can be established by the satellite system. Since the operation of the FSS is highly dependent on the ability of its participating satellites to establish ISL, ISL capability is one of its most important performance attributes. In this case study, the level of ISL capability is a direct function of the platform used for establishing the ISL. If transmission is conducted using **RF technology**, a small RF antenna will give a **low** ISL capability while a big RF antenna will give a **medium** ISL capability. If transmission is conducted using **laser communications** at present day TRL, then the satellite will have a **high** ISL capability, since Laser communications can promise higher transmission rates at better levels of efficiency. If laser communications mature in the future and rated with the highest TRL, then using laser will give an **extremely high** ISL capability. Therefore, both RF and laser communications will be the main choices for communication technology in the FSS.
SYS-PA-2 **ANNUAL FREE DATA CAPACITY**	The **annual free data capacity** is a proxy measure of the in-space data assets that the satellite can offer for opportunistic sharing in the FSS. In this case study, it is simply calculated as the difference between the total data capacity of the satellite and the data capacity used per year by its payload. The data capacity used by its payload is in turn calculated by the product of the payload mass and the annual data usage per kilogram of payload. This is ultimately derived from the assumption that every satellite is constructed for a certain science mission and not just in fulfillment of FSS objectives. Hence, the payload will be the main consumer of data capacity throughout the time of FSS

	development and operation. **The annual free data capacity will be measured using "the number of free data packets available".** Data packets are used as the basic units of measurement for this attribute to **eliminate the need to differentiate between kilo-, mega-, or giga- byte** data storage volumes or rate of data transmission. **Data packets are thus proxy measures of data capacity.** Also, it is also assumed that all FSS satellites obey the same network protocol, hence the size of a data packet can be assumed to be common for all FSS satellites. In this manner, a satellite built with high total data capacity will have a large data packet capacity, and a satellite built with a payload with high data usage per kilogram will have a high overall data packet usage. The annual free data capacity will thus be the difference between the number of data packets available initially and the total number of data packets used by the payload in a year. The annual free data capacity may be increased if the FSSD and DMs decide to buy a **terrestrial capacity real option**, such that they will be able to exercise this option in order to leverage available data capacity on the ground to meet sudden increases in data asset demands.
SYS-PA-3 **ANNUAL SCIENCE VALUE**	The **annual science value** is a proxy measure of the satellite's annual contribution to space science. It has been assumed that the only consumer of data storage and processing capacity is the payload. It is also assumed that the payloads are always in operation. Hence, the annual science value can be calculated by multiplying the payload mass and the annual data packet usage per kilogram of payload. The unit of measurement for annual science value of a satellite is the **number of data packets**.
SYS-PA-4 **DELTA-V**	The **delta-V** value is a direct measure of a satellite's potential ability to change from one trajectory to another by making an orbital maneuver. It can be calculated using the Tsiolkovsky rocket equation $\Delta V = I_{sp} g \ln \frac{m_0}{m_1}$. I_{sp} is the specific impulse expressed for a time period and g is the standard gravitational constant. m_0 is the total mass of the satellite, including its base dry mass, payload mass, and propellant mass. m_1 is simply the mass of the satellite without the propellant, assuming that all the propellant has been used in one sitting for a single orbit maneuver. The delta-V value calculated is thus a measure of its **maximum potential to perform such a maneuver**, and a satellite with high delta-V can easily perform other smaller, less propellant-

	demanding maneuvers. Delta-V is an important consideration, as the satellites are not designed simply for communication purposes, but also for various science missions that may require orbital plane or inclination angle change.
SYS-EA-1 **DEVELOPMENT COST**	The **development cost** of a satellite is the monetary cost required to design, construct, assemble, integrate and test a satellite before it is determined to be ready for operations. This includes the base cost required to develop a typical satellite's subsystems such as thermal, electrical power systems (EPS), structures, tracking telemetry and command (TT&C), attitude determination and control (ADCS) and basic communications. The mass of the communications subsystem may increase if all subsystems are designed for a longer lifetime. The mass of the TT&C unit may increase if the satellite is designed with higher total data capacity. The base mass can also decrease if laser communications is used, as its module is much smaller and lighter than that of antenna dishes used for RF communications. The development cost is typically measured in millions of dollars ($ millions)
SYS-EA-2 **LAUNCH COST**	The **launch cost** of a satellite is the monetary cost required to launch the satellite into its assumed LEO as a monolithic structure. It is a function of the total mass of the satellite, including its onboard propellant, and it is dependent on the choice of launch vehicle. The Ariane V and Delta IV LVs are more expensive options, while the Falcon 9 and Falcon 9 Heavy are relatively cheaper options. However, any of the LVs are used, it must be assumed that the payload capacity of the LV is used entirely, whether if most of the payload comprises the FSS-enabled systems or other unrelated space hardware. Maximizing the utilization of payload capacity on a LV is logical, as the FSSD and DMs will want to reduce the launch cost per kilogram of payload as much as possible. It is also measured in millions of dollars
SYS-EA-3 **DEVELOPMENT TIME**	The **development time** is the time in years required by the FSSD and DMs to design, construct, assemble, integrate and test a satellite before it is determined to be ready for operations. The development time may be dependent on the propulsion technology, the communications technology used for the FSS interface, and the lifetime that the satellite has been designed to operate reliably for.

4.4 System Level Design Variables and Epoch Variables

After establishing the mission concept and attributes, the design variables and epoch variables were then determined from the standpoint of the hypothetical FSSD and DMs. All the assumptions described earlier are also factored into the choices of variables. **Eight design variables** have been chosen for system level design to drive the performance and expense attributes defined in the previous section, and they are listed and described in Table 4-3.

Some of these variables were based on those described by (Underwood 2006) in the design of a distributed satellite communications system. The factor levels for each design variable are also listed. Assuming all combinations of design variables will yield feasible architectures for the satellite system, there are a total of **34,560 possible designs for the satellite system**. **Two context variable**s are considered for system level design and they are based on the assumptions made for laser communications and RLV technology. By assuming their present day low TRLs and their potential to mature in the future with high TRLs, **four possible operating contexts** can exist as a result of combining the low or high TRLs for laser communications and RLV technology. The context variables are the same for system, program and portfolio analysis and they are described in Table 4-4. The design and epoch variables will then be used to establish the system tradespace model, which will eventually calculate the attributes described earlier.

Table 4-3: System Level Design Variables

Design Variables	Description	Factor Levels (units)
SYS-DV-1 **Design Lifetime**	The **design lifetime** of the spacecraft is an important design variable. The longer a satellite is designed to operate for, the longer it can support customer spacecraft. A longer expected design lifetime may also potentially allow FSS users to save cost, as the cost of system spreads out over time. A longer design lifetime also implies that the FSS-enabled satellite has the opportunity to attract more customers in the long run. However, designing spacecraft hardware with a longer lifetime can increase development cost and development time. Operating the satellite for a longer period also means that operations cost will increase greatly over the system lifecycle. These interactions will be accounted for in the system tradespace model. It is also worth noting that designing for a longer lifetime can also impact other factors such as the amount of onboard propellant, hardware reliability, the adoption of new technologies. However, these additional interactions will not be considered in the system	**10** years **15** years

	level tradespace model at this point to manage its computational complexity. The tradespace model will consider the effect of two design lifetimes: **10 and 15 years**.	
SYS-DV-2 **Initial Data Packet Capacity**	The **initial data packet capacity** of a satellite is an important variable, as it is a measure of the maximum number of data packets that the satellite is built to hold. It is in fact a proxy measure of the total amount of onboard memory and processing capability of the satellite electronics, and it also directly affects the free data capacity of a satellite and enables it to participate in FSS operations. It is also equivalent to the annual free data capacity if the satellite payload is not in operation. 5 different data packet capacities are considered in the tradespace model.	**10000** packets **15000** packets **20000** packets **25000** packets **30000** packets
SYS-DV-3 **Annual Data Package Usage Rate** **(per kilogram of Payload)**	The **annual data packet usage per kilogram of payload** of the satellite is a proxy measure of the total amount of onboard memory and processing capability used per kilogram of payload in a year assuming that the satellite payload is always in operation. 3 data packet usage rates are considered in the system tradespace model.	**5** packets **15** packets **25** packets
SYS-DV-4 **FSS Interface Communications Technology**	The **FSS interface communications technology** is the choice of communications technology for establishing efficient ISL between satellites in support of FSS operations. A small RF antenna, a large antenna, and a laser communications module are listed as possible choices for this design variable. A small RF antenna will provide a low ISL capability while laser communications can give a high ISL capability.	Small RF **(1)** Large RF **(2)** Laser **(3)**
SYS-DV-5 **Terrestrial Capacity Real Option**	The **Terrestrial Capacity Real Option** is the design decision of utilizing data packet capacity in the terrestrial system. A satellite system that does not exercise the option to use this capacity must necessarily route all of the packets in their system via satellite. Using terrestrial capacity may help alleviate congestion problems that could occur if the market demand is underestimated. This can	Do not use **(0)** Use **(1)**

	also increase the number of transmission paths available, which should benefit load balancing and reliability. There are more advantages to exercising this option. Having terrestrial capacity can provide a staged deployment platform such that market demand can be built up in certain areas, as the system is being deployed and brought online. Furthermore, as the demand for the service grows, there exists the possibility that a satellite-only system will become overloaded if insufficient excess capacity is designed for. If terrestrial capacity can be utilized upon exercising the option, it can reduce the impact of overloading on the customers and/or to reduce the congestion experienced in high-demand areas. The terrestrial real option is a design decision captured in a binary design variable. If the option to utilize terrestrial capacity is taken, the value of the design variable is 1. Otherwise, it is 0. This variable is based on work by (Underwood, 2009)	
SYS-DV-6 **Propulsion Type**	The **propulsion** type is an important design variable, as these new satellites may have to perform orbital maneuvers for various science missions. This is done in a similar way as the Space Tug example in the earlier chapter. 4 propulsion types are considered in this case study.	Storable Bipropellant **(1)** Cryogenic **(2)** Electric **(3)** Nuclear **(4)**
SYS-DV-7 **Payload Capacity**	The **payload capacity** is a measure of the maximum payload mass that a satellite can carry to perform science missions on top supporting functional requirements for the FSS. It has been assumed that the ability to carry more payloads will mean the ability to carry more instruments, and hence perform more science missions. Hence, the higher the payload capacity, the greater the value to science. Although payload can vary greatly, only 4 discrete payload capacities were chosen. Capacities above 1000kg were not chosen.	**30** kg **100** kg **500** kg **1000** kg

SYS-DV-8 **Propellant Mass**	The **propellant mass** determines the wet mass or total mass of the satellite and its delta-V capability. Similarly, only 3 discrete levels for propellant mass were chosen. Masses above 2000kg were not considered.	**500** kg **1000** kg **2000** kg
SYS-DV-9 **Launch Vehicle**	The **choice of LV** determines the launch cost per kilogram of the satellite, assuming that the entire payload capacity of the LV has been used. Only 4 more recently used LVs are considered.	Falcon 9 **(1)** Falcon 9-Heavy **(2)** Ariane V **(3)** Delta IV **(4)**

Table 4-4: System, Program and Portfolio Level Context Variables

Context Variables	Description
Laser Communications Technology	It is assumed that **laser communications technology** exists at present day. However, its performance and reliability is still questionable given its lack of maturity. As such, it has low TRL at present day. At low TRL, satellites can still have a high ISL capability, but at the expense of higher development cost and longer development time. However, this technology has a potential to make a breakthrough in the future given all the recent development to make free space optical communications work between satellites. Assuming that there is such a breakthrough, laser communications will have a high TRL and satellites can now have an extremely high and reliable ISL capability. Development cost and development time will also be reduced to an extent more favorable than that of RF technology, assuming that laser communications research and manufacturing have already progressed beyond the learning curve. Therefore, this epoch variable has 2 levels: **Low Laser TRL, High Laser TRL**.
Reusable Launch Vehicle Technology	It is assumed that **RLV technology** is still in development and not available for use at present day, hence the launch cost of satellites will remain at the levels expected of currently available LVs like Falcon 9, Falcon 9 Heavy, Ariane V and Atlas. Assuming that that RLV technology makes a breakthrough and eventually matures in the future, launch costs can potentially be halved. Therefore, this epoch variable has 2 levels structured in the same way as that for laser communications: **Low RLV TRL, High RLV TRL**.

The context variables can hence be combined to give 4 different possible contexts that can exist, with Low Laser TRL/Low RLV TRL being representative of present day technology standards, and High Laser TRL/High RLV TRL being representative of advanced future technology. The design variables and epoch variables described in these tables are then used to develop the system level tradespace model.

4.5 Hypothetical FSS Development Missions

After making a number of architectural assumptions about the FSS and determining its design/context variables, hypothetical **development missions** were then proposed for the system, program and portfolio that are to be designed in this case study. The term "development missions" is used because the satellite constellations that are designed, launched and operated in time are not only meant to fulfill their own unique sets of scientific missions, but also to sequentially fulfill the development objectives for the greater entity which is the FSS.

All satellite constellations are designed from the outset that they will eventually be part of the FSS, but some of them will be designed more specifically to meet predicted increases in market demand for FSS assets while some will be designed simply to populate LEO and establish the foundation for the FSS. The types of development missions can vary largely, but 6 plausible ones were chosen for this case study and they are described in Table 4-5. The descriptions for these missions will also help to shape stakeholder preferences for every attribute at the system, program, and portfolio levels.

Table 4-5: Possible development missions for the FSS

Development Mission	Description
0 **Baseline**	A generic mission where the aim is to simply develop, launch and operate the entity reliably for a science mission. Performance, cost and schedule attributes are equally important to the FSSD and DMs, who are already aware that these satellites will be part of the FSS. Having a baseline mission also enables easy comparison between different missions, and allows the FSSD and DMs design other new missions and state preferences for attributes that are clearly different from that of the baseline case.
1 **Space Science**	A realistic present day mission where the aim is to conduct more space science at various orbits. The FSSD and DMs are hence interested in increasing value to science and delta-V of their spacecraft in order to fulfill these science missions, and they are also equally aware that their satellites will eventually be part of the FSS. Reducing launch cost is also critical, as RLV technology has yet to be achieved at present day.

Development Mission	Description
2 **FSS Pilot Mission**	The FSSD and DMs have agreed to develop satellites strictly in support of developing the FSS capability in this pilot mission. Satellites designed for this mission will have less emphasis on science value and delta-V, but will now have increased emphasis on free data capacity and ISL capability. Since a considerable number of satellites will be built for the FSS effort, the FSSD and DMs are also interested in reducing development cost more than anything else.
3 **Initialization and Development**	FSS development is already underway, and the FSSD and the DMs already have knowledge about the expected performance of these satellites and their theoretical first unit costs. There is now even more emphasis on free data capacity and ISL capability of every satellite. Since development cost per satellite has already been controlled for, the emphasis has now shifted to reducing launch cost and development time, since more satellites are required from now on in order to meet the market demand for in-space assets.
4 **Age of Discovery**	Many FSS-enabled satellites are already in LEO and functioning as expected. In the age of discovery, space science has grown exponentially, and this could be attributed to sudden changes in mission requirements such as the discovery of water on moons of nearby planets, or even the discovery of new planetary systems in the Milky way. Existing spacecraft in orbit are conducting numerous science missions, and are constantly demanding for in-space data assets in order to store and process new scientific data. The satellites operating in or designed for this mission must have the scientific capabilities to tap on these new discoveries. As such, ISL capability, free data capacity, and value to science are of the highest priority to the FSSD and DMs. Since more FSS-enabled satellites are still populating LEO in this mission, there is still a need to reduce launch cost and development time in order to launch more satellites quickly to meet market demand.
5 **Orbit Reconfiguration**	In this mission, most of the FSS-enabled satellites have already been launched to LEO and are in operation. As it is assumed that these satellites will be distributed uniformly across the celestial, there is a need to conduct a mission where they actually do so. Adding a new constellation each time with a different number of constituent satellites will change the configuration of this uniform distribution. It was earlier assumed that all new satellites would first end up at the same orbit inclination angle and altitude immediately after launch. However, the FSSD and DMs now know where all the customer spacecraft are and determine that the best way to satisfy market demand for in-space data assets is to distribute all the satellites they have launched in a uniform manner around the Earth. As such, the emphasis on satellites operating in or designed for this mission has high enough delta-V to perform angle and plane changes. They are also interested in lowering development cost and launch cost for every satellite in this mission.

4.6 Epoch Construction and Sequencing for System, Program and Portfolio Level Analysis

Given that there are 4 contexts and 6 sets of mission needs, there are **24 possible epochs** that a satellite system, program or portfolio can be developed or operate in. However, in this case study, not all 24 epochs will be considered. In the potential tradespace model, the attributes of all possible designs will change, especially when the laser communications and/or RLV technology transition to higher TRLs. Since it has been assumed that the satellites are static and are not going to change over time, there will not be many useful designs that exist across all 24 epochs. Therefore, a handful of epochs can be fitted to a possible development timeline for the FSS, where the start and epoch of each epoch corresponds to the start and end of a specific development phase.

A possible development storyline can be as follows:

The FSSD and DMs have finally agreed to the terms and conditions to make the FSS a reality and a space network beneficial to all. However, each of the DMs, which can be a national or multinational space agency, has their own roadmaps for space exploration and they all wish to launch a satellite constellation of their own to perform their own science missions. In the FSS paradigm, each satellite is considered as a system, each satellite constellation is considered as a program, and finally the FSS is considered as a portfolio.

In respect of the terms and conditions on the FSS agreement, the agencies have agreed to equip their satellites with FSS-enabling capabilities. The FSSD, which is responsible for monitoring the performance, cost and schedule for this initiative, is applying the design for affordability process and is considering additional cost and schedule attributes at portfolio inception. However, the FSSD is unaware of the overall time and monetary budget for the FSS portfolio.

Different governments provide different levels of funding for each constellation, and independent funding for FSSD activities like integration and monitoring is also uncertain. However, the FSSD knows the cost of developing a single satellite from its basic materials. Therefore, the FSSD has decided to use the bottom-up approach to designing for affordability. By ensuring that the design of every satellite is affordable, the design of a satellite constellation is also likely to be affordable, and eventually the entire portfolio is also likely to be affordable.

As the FSSD has also recognized the need to keep FSS development and operation affordable at all times, it decides against allowing all DMs to develop their own constellations at the same time. As such, the DMs have agreed to stagger their launches at regular time intervals, say every 5 years as a first-pass analysis. After the first DM has launched the first constellation, each of the remaining DMs will then wait its turn and aim to launch its own constellation at the start of every 5-year interval. Each DM is assumed to begin its development 5 years (or exactly 1 epoch) before its projected year of launch. However, the actual time taken to develop the constellation before eventually launching may be longer than that. As a result, the FSSD may impose cost penalties as a result of the delay in scheduled launch. Each 5-year interval represents a different phase in the FSS development roadmap that has been agreed upon and it has a set of needs that must be fulfilled.

The FSSD has already interviewed every DM about their preferences towards various attributes for different missions in different contexts prior to portfolio inception. Since it has been assumed earlier that the FSSD preferences are also representative of every DM's preferences, there is always a common, unique set of preferences for the satellite system and constellation at every interval. In other words, there is always a new satellite constellation launched by the FSSD and one DM optimistically at the start of every epoch, which will have a unique set of fixed needs and contexts. In this case study, the first epoch has been assumed to last for 10 years, while subsequent epochs are to last for 5 years each. Given that more time is required initially to plan the overall development of the FSS, the first epoch last longer than the others and will transit to the second epoch only after the first FSS-enabled satellite constellation has been launched.

All DMs have also agreed to conduct their constellation development using the common design variables and evaluate them based on the same attributes provided. Assuming that there are 7 DMs agreeing to participate in the FSS initiative, namely ESA (DM1), JAXA (DM2), CSA (DM3), CNSA (DM4), UKSA (DM5), NASA (DM6) and USAF (DM7), each DM will be taking ownership of one satellite constellation and will launch it in the aforementioned order at the beginning of every epoch. Therefore, there will be 7 ***epochs*** *and with the assumed durations of each epoch, the FSSD has optimistically aimed to complete the development of the FSS over a* ***projected period 40 years****.*

The 7 development phases of the FSS can hence be described by 7 selected epochs imposed over the years 2016-2055 at 5-year intervals, apart from the first epoch that last 10 years. Only One constellation will be launched in each phase/epoch. The FSS will thus comprise of 7 different satellite constellations by 2055.

These 7 epochs are chosen because they can be sequenced to form a very plausible development roadmap for the FSS. These epochs correspond to the following 7 phases:

Phase 1, which is conducting **Mission 0 (Baseline) at low laser and RLV TRLs from 2020-2030**, describes the fairly neutral standpoint of DM1, who is the first to launch an FSS-enabled satellite and will not be sure of the attributes they are particularly concerned about.

Five years on, the timeline transits to **Phase 2**, which is conducting **Mission 1 (Space Science) at low laser and RLV TRLs from 2030-2035** describes the more space science-biased standpoint of DM2. DM2 is now more aware of what attributes they are looking out for after observing the development of the baseline constellation for 10 years, and it decides that having larger and more powerful payloads and higher delta-V capability will ensure that its constellation can perform more science missions than ever before. In this time period, concerns about launch cost are valid, as RLV technology is unlikely to mature so quickly and the satellites designed in this epoch will probably have high wet mass.

Five years on, the timeline transits to **Phase 3**, which is conducting **Mission 2 (FSS Pilot Mission) at low laser and RLV TRLs from 2035-2040**, describes the more FSS-focused standpoint of DM3. After recognizing that there are already 2 constellations in LEO, DM3 has been assigned the responsibility of conducting the pilot test for operating the FSS, and it wants to ensure that its own constellation will be capable of fulfilling FSS objectives above anything else. Hence, DM3 is interested in increasing ISL capability and free data capacity. As more satellites

are likely to be built for this pilot mission, DM3 is likely to be interested in lowering its development cost per satellite more than anything else.

Five years on, the timeline transits to **Phase 4**, which is conducting **Mission 3 (Initialization and Development) at low laser TRL but high RLV TRL from 2040-2045**, describes the even more pronounced FSS-centric viewpoint of DM4, who can also enjoy the benefits of mature RLV technology. Since DM4 will be even more driven to support the FSS capability, it is more than interested in increasing ISL capability and free data capacity. Since development cost would have already been controlled for in the previous phase, DM4 will be concerned about reducing development time and launch cost due to the number of satellites required.

The timeline then transits to **Phase 5**, which is conducting **Mission 4 (Age of Discovery) at high laser TRL but low RLV TRL from 2045-50**, describes the standpoint of DM5 who is equally interested in achieving high scientific value and fulfilling FSS objectives. Laser communications technology has matured, enabling an unprecedented level of ISL capability. However, this could be done at the expense of RLV technology, whose TRL could become low again, as the FSSD or DMs may not be entirely confident of both new technologies working together at the same time and prefer to remain conservative by testing one at a time. Science value, free data capacity and ISL capability would all be equally important to DM5. Like the previous phase, reducing launch cost and development would remain important to the DM.

The timeline then transits to **Phase 6**, which is conducting **Mission 5 (Orbit Reconfiguration) at both high laser and RLV TRLs from 2045-2050**, describes the standpoint of DM6 who is now most interested in reconfiguring all FSS-enabled satellite constellations to reconfigure themselves in order to maximize Earth coverage and accessibility to customer spacecraft in other orbits. As such, DM6 is most interested in the delta-V capability of its satellites and would want to make sure that its assigned satellite constellation would be able to perform various orbital maneuvers to meet the objectives of this FSS development phase. Due to the shifted focus on delta-V, DM6 would hence be interested in reducing development cost and launch cost, as the design variables of propulsion technology and propellant mass would become critical.

Finally, the timeline transits to **Phase 7**, which is conducting **Mission 0 (Baseline) at both high laser and RLV TRLs from 2050-2055**, describes the standpoint of DM7 who is taking a neutral approach to developing its constellation since most of the FSS satellites have already been launched. DM7 would just needs to complete the network and have equal preferences for all attributes. All the FSS development missions would be complete after this final launch.

For convenience in notation, a “Low TRL” state for an epoch variable will be assigned to a value of 0 and a “High TRL” will be 1. Hence, a context with both high laser and RLV TRL will be denoted as “Context 1-1”. There are a total of 24 possible epochs, where every 6 epochs correspond to Mission 0 through 5 for contexts 0-0, 0-1, 1-0 and 1-1 respectively. Hence, based on the hypothetical development phase sequence described above, the corresponding epochs and their development missions are summarized in Table 4-6.

Table 4-6: Summary of Hypothetical FSS Development Mission Sequence

Development Phase Sequence	Corresponding Epoch number (out of 24)	Description (Mission, Context)
1	1	Mission 0 in Context 0-0
2	2	Mission 1 in Context 0-0
3	3	Mission 2 in Context 0-0
4	10	Mission 3 in Context 0-1
5	17	Mission 4 in Context 1-0
6	24	Mission 5 in Context 1-1
7	19	Mission 0 in Context 1-1

4.7 System Level Design-to-Value Mapping

After establishing a plausible development timeline for the FSS, it now remains to perform the design to value mapping (DVM) of the design variables to the performance and resource attributes. In the DVM process, the design and epoch variables populate one axis of a matrix while the attributes populate the other. The intersection between a variable and an attribute represents the possibility of an interaction. A value of "0" is assigned to this intersection if there is perceived to be no interaction between the variable and attribute of interest. A value of "1" will signify a perceived weak level of interaction, a value of "3" will then signify a perceived medium level of interaction, and finally a value of "9" will signify a perceived high level of interaction. A Design Structure Matrix (DSM) could also have been used to do this mapping and may even be able to represent the same interactions at a higher degree of fidelity through considering feed-forward and feedback loops simultaneously. However, in view of the high-level approach taken in this first-pass analysis for the bottom-up design of a system to program to portfolio, performing the DVM in this manner will expedite the process of developing the tradespace model. Furthermore, satellite subsystems are neither considered as block modules nor key design variables in this case study. Therefore, the combination of the eight system level design variables and two epoch variables, and their interactions with the five performance attributes and three expense attributes, will provide more than sufficient information to derive a meaningful system level tradespace model that can yield 34,560 possible designs. The DVM table for the satellite system is shown in Table 4-7, where zero values for a matrix element is left as a blank.

Table 4-7: DVM for Satellite System

	ISL Capability	Annual Free Data Capacity	Annual Science Value	Delta-V	Development Cost	Launch Cost	Development Time	Row Total
Spacecraft Lifetime					9			9
Initial Data Packet Capacity		9	3		3		3	18
Annual Data Packet Usage Rate		9	9		3		3	24
FSS Communications Interface Tech.	9				9	3	9	30
Terrestrial Capacity Real Option		9						9
Propulsion Type				9	9	9	9	36
Payload Capacity		9	9	9	3	9	3	42
Propellant Mass				9		9		18
Launch Vehicle						9		9
Laser Comms TRL	9				9		9	27
RLV Tech.						9		9
Column Total	18	36	21	27	45	48	36	

After completing the DVM, more attention was paid to rows and columns with high cumulative values to determine the variables and attributes that have the highest degrees of interaction. Interactions with perceived matrix values of 9 will definitely be accounted for in the system tradespace model, since they are the most critical. Not all interactions with perceived values of 1 or 3 are accounted for. The interactions embedded into the model are described as below, starting from the row with highest total value.

Equations were then formulated for each interaction to establish the computerized model. These system tradespace model equations are not shown in this section and they can be found in the computer code attached in Appendix 1. The scientific principles used were based mainly on the methods for spacecraft subsystem design prescribed by (Wertz and Larson, 2011). Some equations were heuristically derived, as there were no precedent cases for reference. All the equations and their basis for usage are described qualitatively below.

Payload capacity has a major impact on the annual free data capacity, annual science value, delta-V and the launch cost. It has been perceived to have much lower degrees of impact on development cost and development time. This is because the higher the payload capacity, the more payloads the satellite is able to carry. Given that usage of this capacity is also maximize and applying the assumption that the payloads are always in operation, more payloads will consume a higher amount of data capacity, and possibly leaving little annual free data capacity that can be leveraged by FSS operations. If there are more payloads onboard, the amount of science performed onboard will definitely be greater. However, the tradeoff is that total mass will increase as a result, which will in turn increase launch cost. Development cost and development time will also increase, but not to as great an extent. Increasing the capacity on the satellite will require more materials, and development time also increases, as more time is needed to integrate more payloads onboard. However, since the development of payloads from scratch is not considered in this model, its impact on development cost and time is not as significant.

Propulsion type has significant impacts on delta-V, development cost, launch cost and development time. The choice of storable bipropellant, cryogenic, electric or nuclear propulsion technologies give different specific impulses, which directly affect a satellite's dry mass and delta-V potential. Development cost and development time is also greatly affected, as more advanced propulsion technologies, especially electric and nuclear, require more time, more materials and monetary resources to operate effectively.

FSS communications interface technology will have a significant impact on ISL capability, launch cost, and development time, but less impact on the launch cost. Having laser communications greatly enhances a satellite's ISL capability and reduces launch cost since the laser communications module is much smaller than RF modules. However, laser communications is expensive and not as reliable at present day standards, hence it entails higher development cost and development time. Although the laser communications module is much lighter than RF modules, its impact on mass reduction of the satellite is being offset by the payload capacity. Hence, it has a lower impact on launch cost.

Annual data packet usage rate per kilogram of payload has significant impacts on annual free data capacity and annual science, but lower degrees of impact on development cost and development time. A high usage rate, coupled with a high payload mass, can result in a significant decrease in free data capacity that may undermine FSS objectives. However, this coupling has benefits as it increases annual science value. A higher usage rate has also been assumed to be representative of a more enhanced payload that has to use a lot of data capacity to function, which may imply higher development cost and higher development time. However, these interactions are not as significant as the ones discussed earlier.

Propellant mass has significant impacts on delta-V and launch cost. These interactions are obvious, as a satellite with more onboard propellant will have a greater delta-V capability. However, more propellant implies a higher wet mass and this greatly increases launch cost.

Initial data packet capacity has most impact on annual free data capacity, and lower impacts on science value, development cost and development time. The first interaction is straightforward, as the greater data capacity a satellite is designed with, the more likely the satellite has free data capacity regardless of the level of data usage by its payloads. Science value may be impact, as the

amount of science that can be performed is capped by its total capacity. Development cost and development time are slightly affected, as higher data capacities will require more electronics that require more time for integration. However, changes in data storage and processing capabilities have less impact on costs and time given Moore's law.

Spacecraft lifetime is perceived to only have impact on development cost, as more reliable and resistant equipment will require higher cost. Since it can be assumed that none of the subsystems or smaller equipment is designed from scratch, it does not take any longer time to assemble enhanced equipment as compared to regular equipment.

Terrestrial capacity real option is perceived to only have impact on annual free data capacity. All satellites are built with that option since ground stations control and monitor them. If this option is exercised, annual free data capacity increases significantly.

Launch Vehicle is perceived to only have impact on launch cost, since the launch cost per kilogram varies from rocket to rocket.

The epoch variable of **Laser Communications TRL** has significant impacts on ISL capability, development cost and development time. When the TRL increases in the future, ISL capability reaches to an unprecedented new level. Development cost and development time also decreases significantly due to maturity in technology.

The epoch variable of **RLV TRL** is perceived to only have impacts on the launch cost. If the TRL increases in the future, more launches can be conducted every year and this can significantly reduce the launch cost per kilogram of any spacecraft.

4.8 Generation of System Tradespaces and Design Identification

With knowledge of design variables, epoch variables, performance attributes, expense attributes and their interactions, the system tradespace model was developed. Design vectors, performance vectors and expense vectors for the 34,560 designs were obtained for all 24 epochs. Stakeholder preferences, representative of the FSSD and DMs, were hypothetically established based on the different missions conducted in different contexts. MAUT was then applied and a unique SAU curve was obtained for each attribute in each epoch. The SAU curves are not described or shown for brevity. The

MAE and MAU values were then calculated for all the design points and illustrated on a scatter plot for every epoch, yielding 24 tradespaces. The seven tradespaces corresponding to the seven development phases were then selected and their Pareto frontiers identified in red as shown in the subsequent figures. Tradespaces for contexts 0-0, 0-1, 1-0, and 1-1 are colored in blue, green, pink and black respectively. All tradespaces had a 100% yield due to the choice of design and epoch variables. All tradespaces were observed to be different due to changing stakeholder preference levels and operating contexts, as well as changing attribute levels due to the presence of epoch variables. It can also be observed that the tradespaces show more degrees of discretization from Phases 4 to 7, due to the introduction of epoch variables, where an increase in laser or RLV can significantly increase performance attributes and decrease expense attributes.

Phase 1 – Epoch 1: Mission 0 in Context 0-0 (100% Yield)

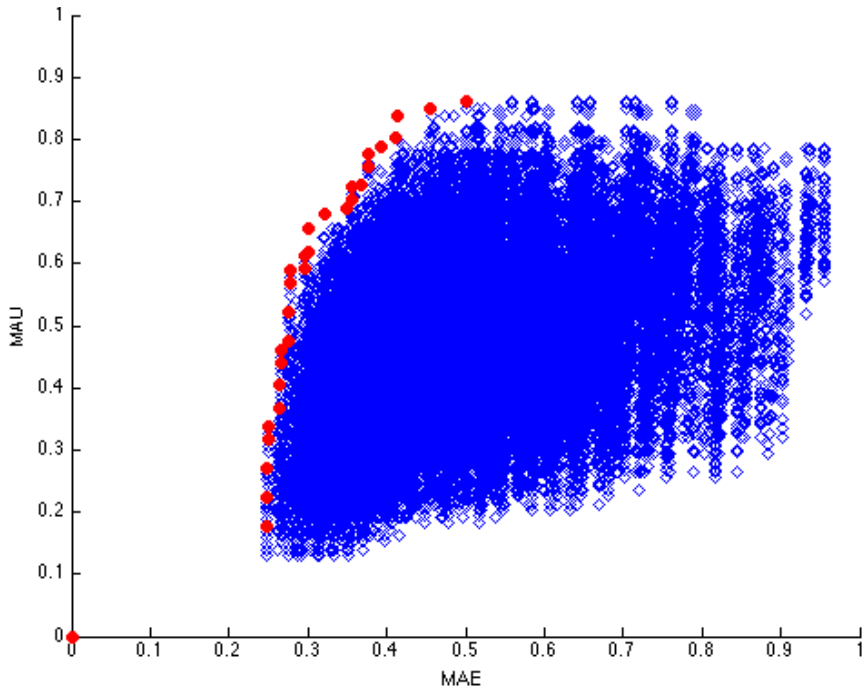


Figure 4-2: System Tradespace for Phase 1 – Epoch 1 – Mission 0 in Context 0-0

Phase 2 – Epoch 2: Mission 1 in Context 0-0 (100% Yield)

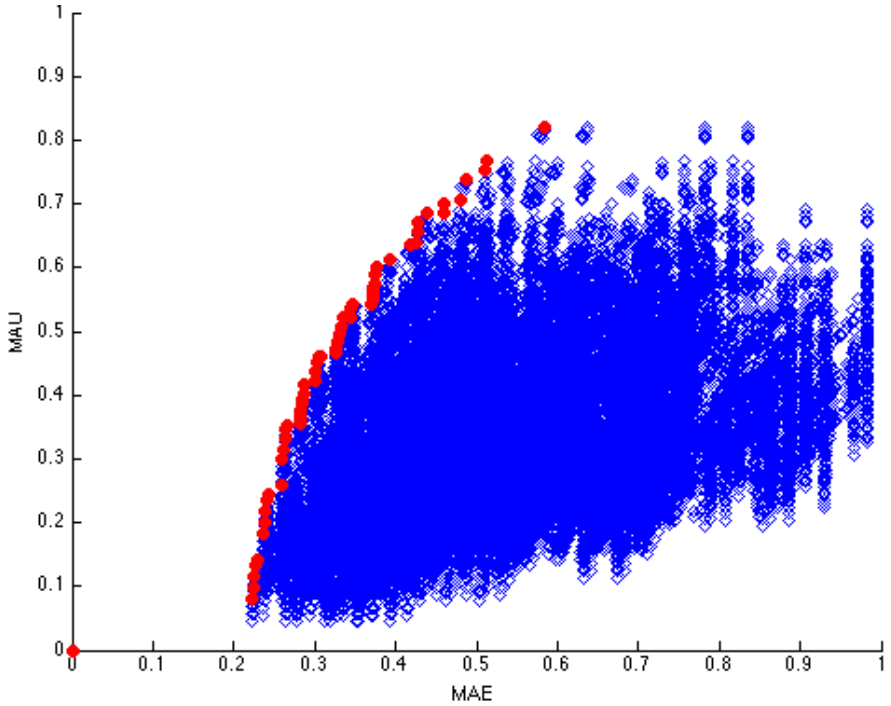


Figure 4-3: System Tradespace for Phase 2 – Epoch 2 – Mission 1 in Context 0-0

Phase 3 – Epoch 3: Mission 2 in Context 0-0 (100% Yield)

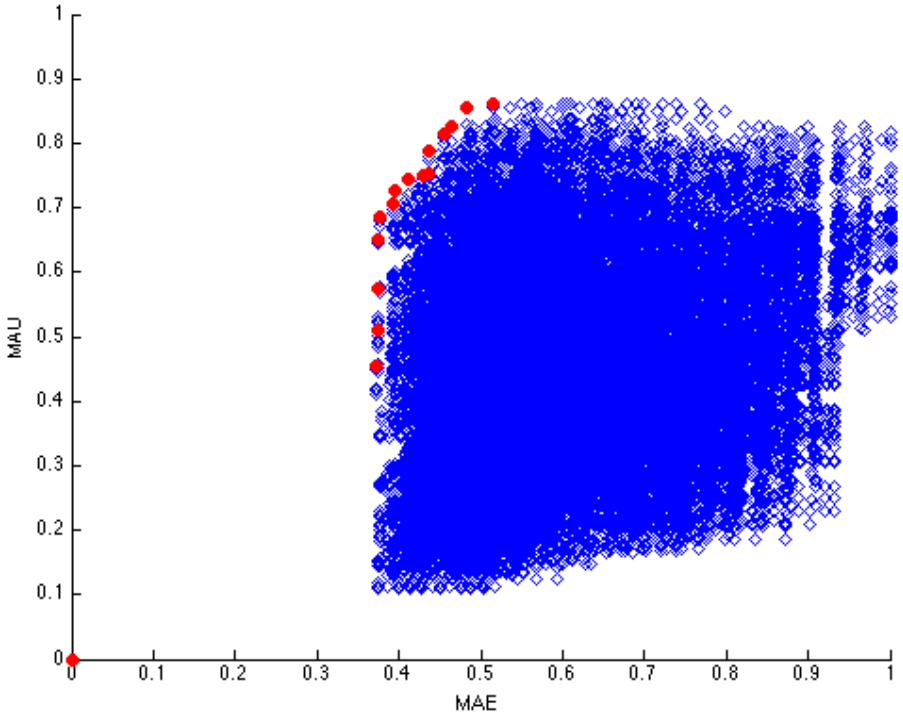


Figure 4-4: System Tradespace for Phase 3 – Epoch 3 – Mission 2 in Context 0-0

Phase 4 – Epoch 10: Mission 3 in Context 0-1 (100% Yield)

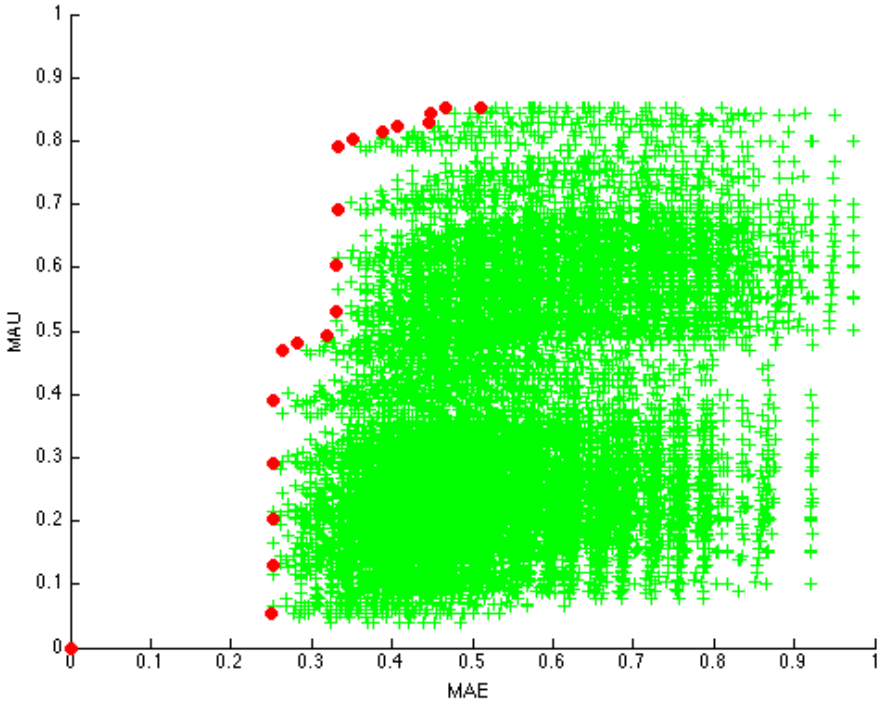


Figure 4-5: System Tradespace for Phase 4 – Epoch 10 – Mission 3 in Context 0-1

Phase 5 – Epoch 17: Mission 4 in Context 1-0 (100% Yield)

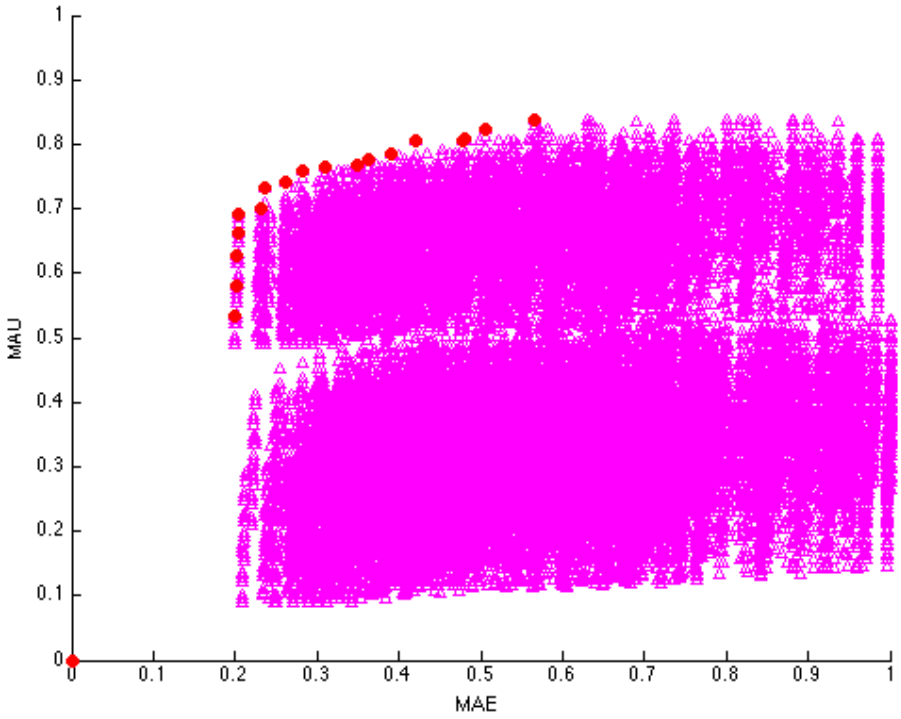


Figure 4-6: System Tradespace for Phase 5 – Epoch 17 – Mission 4 in Context 1-0

Phase 6 – Epoch 24: Mission 5 in Context 1-1 (100% Yield)

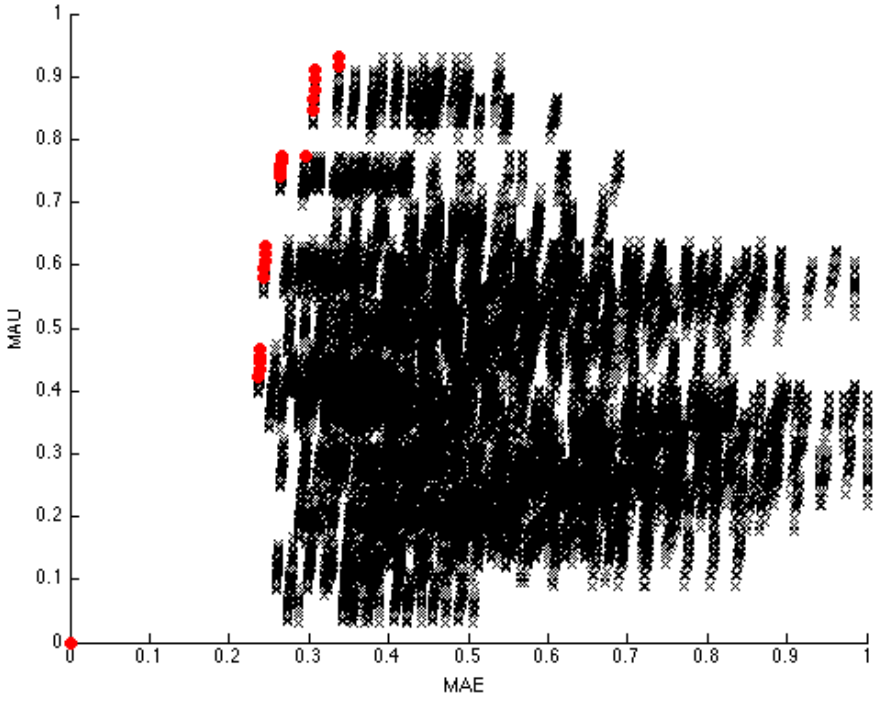


Figure 4-7: System Tradespace for Phase 6 – Epoch 24 – Mission 5 in Context 1-1

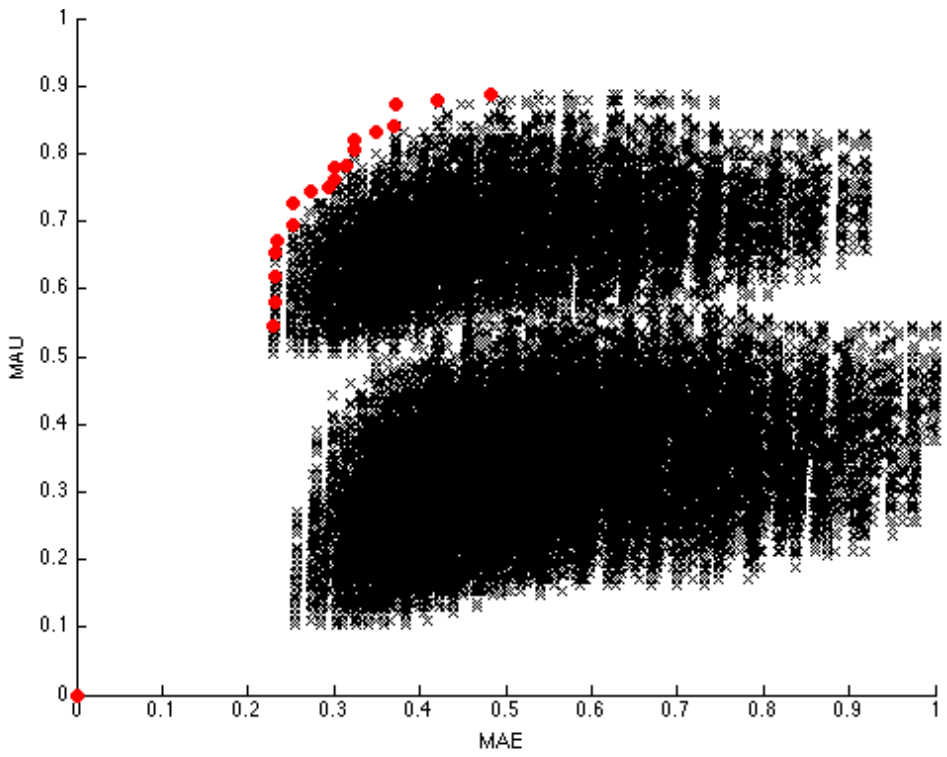


Figure 4-8: System Tradespace for Phase 7 – Epoch 19 – Mission 0 in Context 1-1

4.9 Lognormal Distributions and Confidence Intervals

Dominated designs, which are points not on the Pareto frontier, are filtered out. This leaves only the Pareto frontier set solutions on the tradespace for consideration. In Chapter 2, Kroshl and Pandolfini (2000) prescribed the use of cost interval estimation in order to account for risk and uncertainties in cost and time elements. These can be collectively referred to as **expense budget risks**, which are the probabilities that the actual cost and/or time elements of a system, program or portfolio end up exceeding a given or expected monetary and/or time budget. These expense budget risks can also be accounted for during tradespace exploration, and one of the ways to do so can be the application of **probabilistic distributions to different expense attributes**.

Distributions are applied only to expense attributes because cost and time elements in the system design process experience much greater volatility in the real world. Many probability distribution functions are available for use such as the normal distribution, lognormal distribution, triangular distribution, Weibul distribution etc. that may account for the inherent uncertainties. Performance attributes, on the other hand, are better controlled for as the defense and aerospace industry has many years of experience in maintaining high performance levels. Technically, uncertainty or variance in every variable or attribute of the system may be accounted for using different probabilistic distributions. However, in this case study, only expense risks will be considered. Assuming that a **99% confidence interval** is desired for each expense element, there is then a

need to decide on the choice of probability distribution function that best resembles its actual uncertainty.

With the consideration of the nature of expense budget risks, **risk-adjusted probability distributions** should hence be used, as it gives higher mean values and higher variances than baseline estimates (GAO, 2009). Such distributions will be more skewed to the right. **In this case study, the raw values for the development cost, launch cost and development time for the FSS satellite system can be the mean values of the risk-adjusted estimates for the expense attributes.** Costs to the right of the mean are likely possibilities given that actual costs exceed expected costs in most real-world cases. However, they can also go to the left, giving unexpected cost savings. As such, each cost and time element can have the risk-adjusted probability distribution shown in Figure 4-9. **The mean value is denoted by μ.**

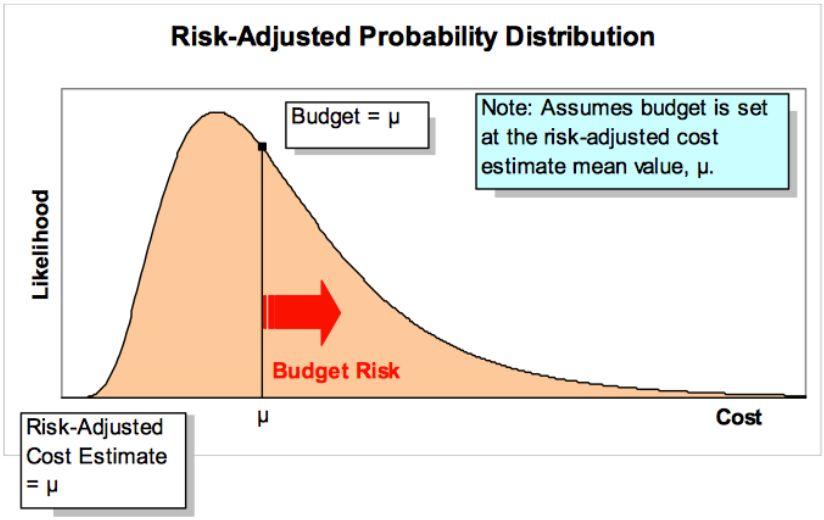


Figure 4-9: Risk-adjusted probability distribution for cost and time elements (NATO OTAN, 2007)

Given the shape of risk-adjusted probability distribution curve, a **lognormal distribution** can be applied to expense attributes since it has the best fit. A set of lognormal distribution curves with different variances is shown in Figure 4-10. A lognormal distribution is more skewed to the right and has a higher variance towards the right, which is reflective of the higher probability that cost and time elements exceed their projected budgets. A higher variance will flatten the curve and increase the skew more towards the right. Expense attributes can have different variances.

For example, development time of a system may have greater volatility than cost since there are more human, technical and time elements involved. Therefore, a lognormal distribution curve with a higher variance can be applied to time- related expenses of a system. The percentage of the area of the distribution to the right of the budget is hence defined as the budget risk. It is usually expressed as a number such a 40%, or 50%, or 60%. To establish a 99% interval for expense attributes, it becomes necessary in this case study to find the x value in Figure 4-10 that yields risk values of 0.5% and 99.5%. Since a mean value of 0 is assumed for the μ parameter in all lognormal distributions, the upper and lower bounds of expense attributes can then be found by multiplying the x value and the mean values of the risk-adjusted estimates for the expense attributes.

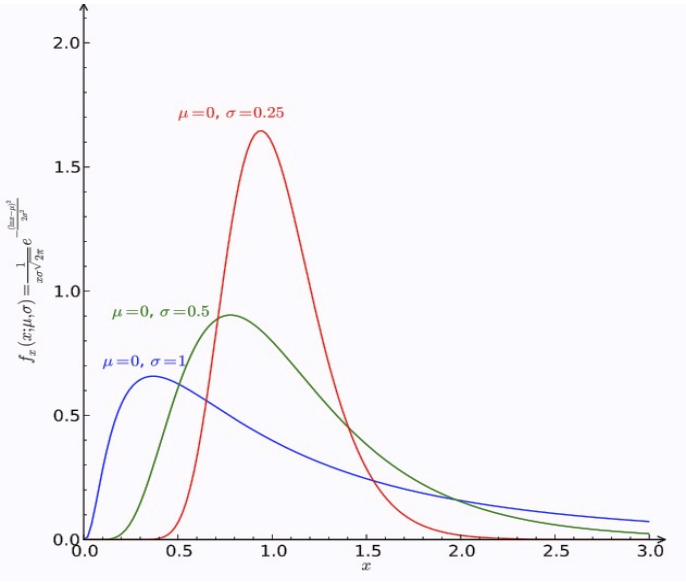


Figure 4-10: Lognormal distributions with different variances (Wikipedia).

Applying MAUT using the same preference sets, the upper and lower bounds for the MAE can be found. Therefore, a 99% confidence interval can be established for the MAE of every design on the Pareto frontier of the tradespaces for the 7 phase-sequenced epochs. A 99% interval is a conservative approach, and selecting a point design that has a 99% MAE confidence interval fully within the affordable solution region on one tradespace allows decision makers to conclude that the design selected can fulfill performance requirements and is affordable for that epoch at a probability of 0.99. With knowledge of the Pareto frontier solutions and the confidence intervals for their MAE values, decision makers can then determine what degree of risk to accept when selecting a particular design.

Different variances are associated with different expense attributes, as described in the earlier example where schedule can vary more than cost in certain cases. In this case study, different σ values are assumed for the development cost, launch cost and development time. These values are summarized in Table 4-8. In Contexts 0-0 and 0-1, development cost and development are assumed to have a variance of 0.1 and 0.15 respectively. Development time has a higher variance, as it has been assumed to have many more uncertainties that impact it during these case studies. The launch cost will have a variance of 0.05 in context 0-0, but that increases to 0.15 in context 0-1. This is because RLV TRL has increased, but there is a greater degree of uncertainty since it is a new technology.

In Contexts 1-0 and 1-1, development cost and development are assumed to have a variance of 0.25 and 0.20 respectively. Development cost has a higher variance, as it has been assumed to have many more uncertainties given the increase in laser communications TRLs. Since it is a new technology, development cost for a satellite with laser communications capability will incur a higher risk in its development cost. The variance in development time also increases slightly as a result. The increases in the variances of the launch cost are similar to those in the previous contexts. Epochs 1, 2, 3 are in Context 0-0; Epoch 10 is in Context 0-1; Epoch 17 is in Context 1-0; and Epoch 24 and 19 are in Context 1-1.

Table 4-8: Assumed σ values for different contexts

	σ value for lognormal distribution		
Context	**Development Cost**	**Launch Cost**	**Development Time**
0-0	0.10	0.05	0.15
0-1	0.10	0.15	0.15
1-0	0.25	0.05	0.20
1-1	0.25	0.15	0.20

A handful of preferred design points were then selected from the Pareto frontier set of each phase-sequenced epoch. They were selected simply by comparing the design vector, performance vector and expense vector. Three points were then picked from each tradespace. A Pareto optimal design point with most of its right-tail confidence interval exceeding the maximum MAE constraint level is hence a **high-risk design**, since there is a high probability of an overrun. A Pareto optimal design point with some of its right-tail confidence interval exceeding the maximum MAE constraint level is a **medium-risk design.**

A Pareto optimal design point with little or none of its right-tail confidence interval exceeding the maximum MAE constraint level is a **low-risk design**. A high, a medium and a low risk were then picked for most epochs and they are colored in black, blue and magenta respectively. For epochs with only two preferred designs, they are labeled as a high and a medium risk design. The Pareto front designs along with their 99% MAE confidence intervals, as well as the preferred designs, are shown in the subsequent figures. The design vector, performance vector, and expense vector corresponding to the preferred designs are also shown in the table below each figure.

Across the Pareto fronts obtained from all seven tradespaces, it can be seen that the right tail of the confidence interval for the MAE of each design point is always longer, implying that there is a greater range to which expense attributes can exceed than fall below their expected values. Another observation is the confidence intervals become wider with each phase, which is reflective of the increasing σ value for the expense attributes. While it is preferable to find a design point whose right tail of its confidence interval lies before the maximum expense constraint level, it also becomes increasingly difficult in later epochs due to the burgeoning budget risks. This increase in confidence intervals is hence representative of the increasing uncertainty and budget risks as time increases.

Typically, such risks accumulate over time as it becomes more difficult to predict levels of expenditure later in the future. It is hence also more difficult to find designs that are always or fully affordable at a later time during the development lifecycle. It was also observed that some epochs are also more restrictive than others in terms of constraint levels, resulting in difference sizes for the affordable solution region. In some epochs, almost all design points are in a wide affordable solution region, hence allowing more choices for consideration, while in other epochs, the affordable solution region is narrower and only a small number of design points are available for down-selection.

Phase 1 – Epoch 1: Mission 0 in Context 0-0

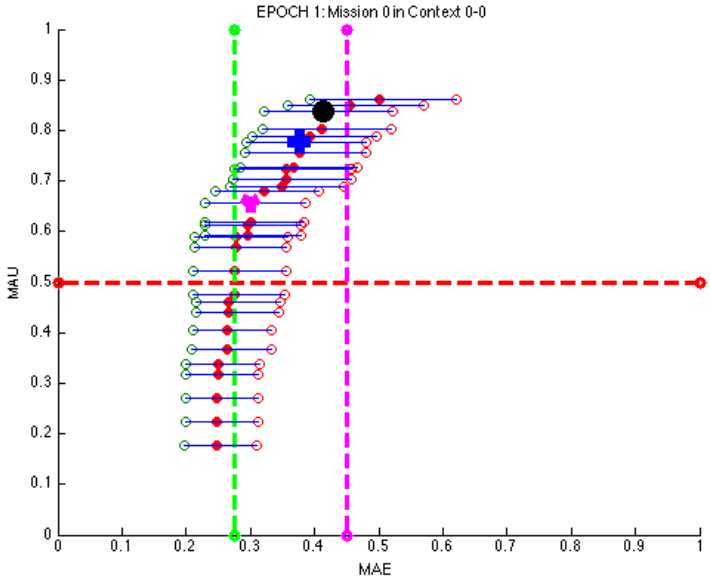


Figure 4-11: Epoch 1 Pareto front design points with 99% MAE confidence intervals

The design vector lists the factor levels for SYS-DV-1 to SYS-DV-9 and it is in the form:

`[spacecraft_lifetime, initial_data_packet_capacity, annual_data_packet_usage_rate, FSS_interface_communications_technology, terrestrial_capacity_real_option, propulsion_type, payload_capacity, propellant_mass, launch_vehicle]`

Performance vector for SYS-PA-1 to SYS-PA-4:

`[ISL_capability, Annual_Free_Data_Capacity, Annual_Science_Value, Delta_V]`

Expense vector for SYS-EA-1 to SYS-EA-3:

`[Development Cost ($ million), LaunchCost ($ million), Development_Time (years)]`

Table 4-9: Preferred designs for Epoch 1 – Mission 0 in Context 0-0

Epoch 1			
Design Number	**Risk**	**Type**	**System Vector**
17202 (2144 kg)	**High (1)**	**Design**	[10,30000,25,3,1,3,100,1000,2]
		Performance	[3,41250,2500,18465]
		Expense	[3.20,5.46,2.5]
17190 (2074 kg)	**Med (2)**	**Design**	[10,30000,25,3,1,3,30,1000,2]
		Performance	[3,43875,750,19345]
		Expense	[3.20,5.28,2]
17102 (1454 kg)	**Low (3)**	**Design**	[10,30000,25,3,1,1,100,500,2]
		Performance	[3,41250,2500,1239]
		Expense	[3.20,3.70,1.25]

Phase 2 – Epoch 2: Mission 1 in Context 0-0

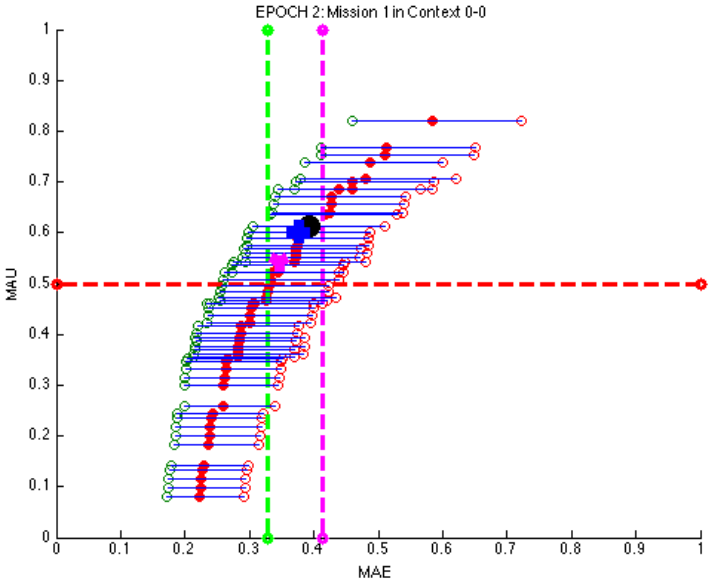


Figure 4-12: Epoch 2 Pareto front design points with 99% MAE confidence intervals

Design vector for SYS-DV-1 to SYS-DV-9:

```
[spacecraft_lifetime, initial_data_packet_capacity, annual_data_packet_usage_rate,
FSS_interface_communications_technology, terrestrial_capacity_real_option,
propulsion_type, payload_capacity, propellant_mass, launch_vehicle]
```

Performance vector for SYS-PA-1 to SYS-PA-4:

```
[ISL_capability, Annual_Free_Data_Capacity, Annual_Science_Value, Delta_V]
```

Expense vector for SYS-EA-1 to SYS-EA-3:

```
[Development Cost ($ million), LaunchCost ($ million), Development_Time (years)]
```

Table 4-10: Preferred designs for Epoch 2 – Mission 1 in Context 0-0

Epoch 2			
Design Number	**Risk**	**Type**	**System Vector**
16830 (2636 kg)	**High (1)**	**Design**	[10,30000,25,2,1,3,500,1000,2]
		Performance	[2,26250,12500,14024]
		Expense	[4.10,6.71,1.50]
17202 (2144 kg)	**Med (2)**	**Design**	[10,30000,25,3,1,3,100,1000,2]
		Performance	[3,41250,2500,18467]
		Expense	[3.20,5.46,2.50]
17190 (2074 kg)	**Low (3)**	**Design**	[10,30000,25,3,1,3,30,1000,2]
		Performance	[3,43875,750,19348]
		Expense	[3.20,5.28,2]

Phase 3 – Epoch 3: Mission 2 in Context 0-0

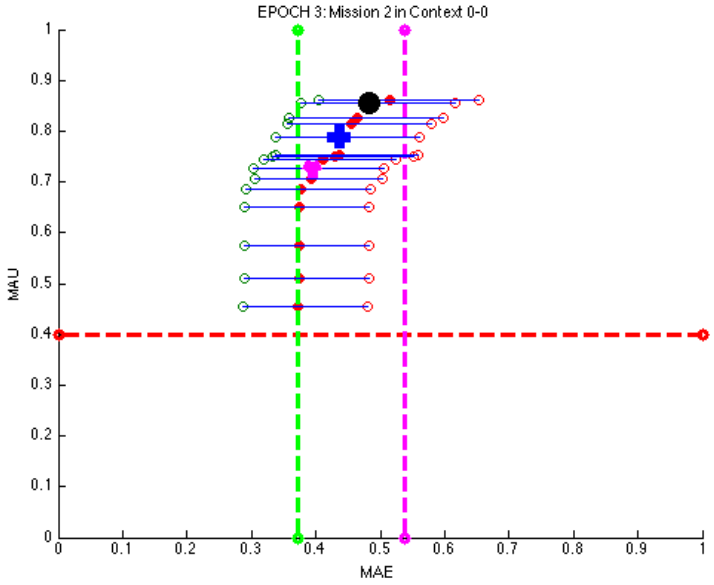


Figure 4-13: Epoch 3 Pareto front design points with 99% MAE confidence intervals

Design vector for SYS-DV-1 to SYS-DV-9:

```
[spacecraft_lifetime, initial_data_packet_capacity, annual_data_packet_usage_rate,
FSS_interface_communications_technology, terrestrial_capacity_real_option,
propulsion_type, payload_capacity, propellant_mass, launch_vehicle]
```

Performance vector for SYS-PA-1 to SYS-PA-4:

```
[ISL_capability, Annual_Free_Data_Capacity, Annual_Science_Value, Delta_V]
```

Expense vector for SYS-EA-1 to SYS-EA-3:

```
[Development Cost ($ million), LaunchCost ($ million), Development_Time (years)]
```

Table 4-11: Preferred designs for Epoch 3 – Mission 2 in Context 0-0

Epoch 3			
Design Number	**Risk**	**Type**	**System Vector**
17202 (2144 kg)	**High (1)**	**Design**	[10,30000,25,3,1,3,100,1000,2]
		Performance	[3,41250,2500,18467]
		Expense	[3.20,5.46,2.50]
17186 (1449 kg)	**Med (2)**	**Design**	[10,30000,25,3,1,3,30,500,2]
		Performance	[3,43875,750,12443]
		Expense	[3.20,3.69,2]
17102 (1454 kg)	**Low (3)**	**Design**	[10,30000,25,3,1,1,100,500,2]
		Performance	[3,41250,2500,1239]
		Expense	[3.20,3.70,1.25]

Phase 4 – Epoch 10: Mission 3 in Context 0-1

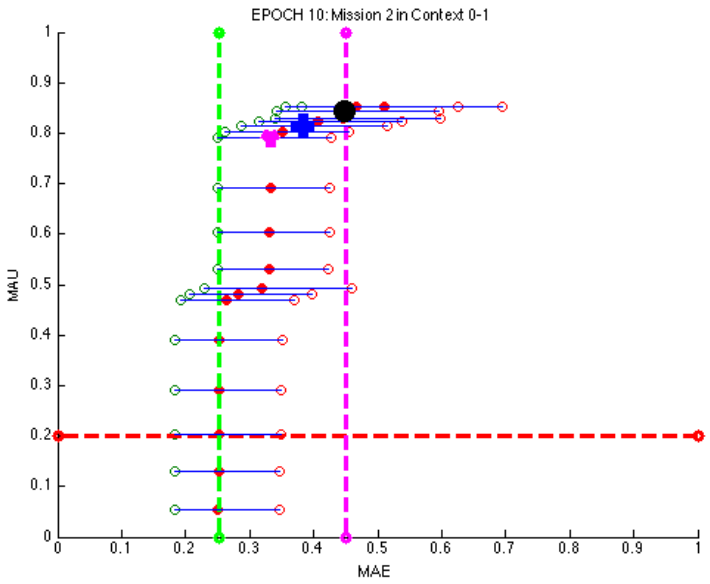


Figure 4-14: Epoch 10 Pareto front design points with 99% MAE confidence intervals

Design vector for SYS-DV-1 to SYS-DV-9:

`[spacecraft_lifetime, initial_data_packet_capacity, annual_data_packet_usage_rate, FSS_interface_communications_technology, terrestrial_capacity_real_option, propulsion_type, payload_capacity, propellant_mass, launch_vehicle]`

Performance vector for SYS-PA-1 to SYS-PA-4:

`[ISL_capability, Annual_Free_Data_Capacity, Annual_Science_Value, Delta_V]`

Expense vector for SYS-EA-1 to SYS-EA-3:

`[Development Cost ($ million), LaunchCost ($ million), Development_Time (years)]`

Table 4-12: Preferred designs for Epoch 10 – Mission 3 in Context 0-1

Epoch 10			
Design Number	**Risk**	**Type**	**System Vector**
17198 (1519 kg)	**High (1)**	**Design**	[10,30000,25,3,1,3,100,500,2]
		Performance	[3,41250,2500,11737]
		Expense	[3.20,1.93,2.50]
17110 (3134 kg)	**Med (2)**	**Design**	[10,30000,25,3,1,1,100,2000,2]
		Performance	[3,41250,2500,2989]
		Expense	[3.20,3.99,1.25]
17102 (1454 kg)	**Low (3)**	**Design**	[10,30000,25,3,1,1,100,500,2]
		Performance	[3,41250,2500,1239]
		Expense	[3.20,1.85,1.25]

Phase 5 – Epoch 17: Mission 4 in Context 1-0

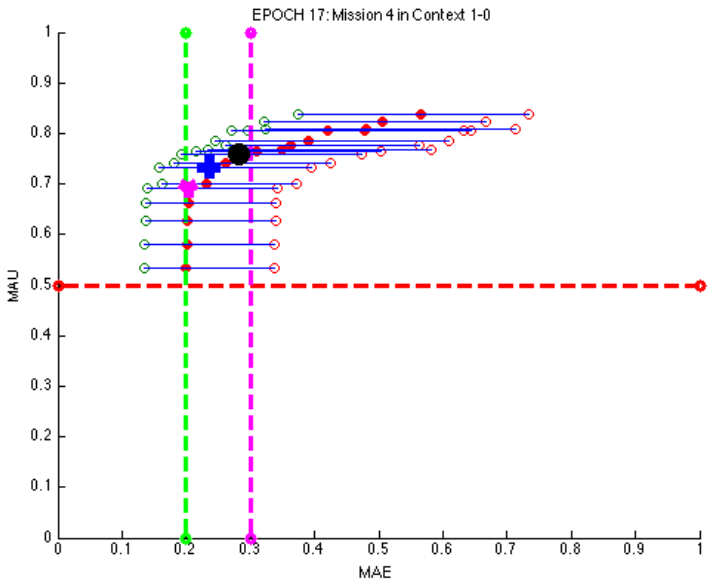


Figure 4-15: Epoch 17 Pareto front design points with 99% MAE confidence intervals

Design vector for SYS-DV-1 to SYS-DV-9:

```
[spacecraft_lifetime, initial_data_packet_capacity, annual_data_packet_usage_rate,
FSS_interface_communications_technology, terrestrial_capacity_real_option,
propulsion_type, payload_capacity, propellant_mass, launch_vehicle]
```

Performance vector for SYS-PA-1 to SYS-PA-4:

```
[ISL_capability, Annual_Free_Data_Capacity, Annual_Science_Value, Delta_V]
```

Expense vector for SYS-EA-1 to SYS-EA-3:

```
[Development Cost ($ million), LaunchCost ($ million), Development_Time (years)]
```

Table 4-13: Preferred designs for Epoch 17 – Mission 4 in Context 1-0

Epoch 17			
Design Number	**Risk**	**Type**	**System Vector**
17114 (1843 kg)	**High (1)**	**Design**	[10,30000,25,3,1,1,500,500,2]
		Performance	[4,26250,12500,931]
		Expense	[2.16,4.69,0.75]
17102 (1443 kg)	**Med (2)**	**Design**	[10,30000,25,3,1,1,100,500,2]
		Performance	[4,41250,2500,1251]
		Expense	[2.16,3.67,0.625]
17090 (1373 kg)	**Low (3)**	**Design**	[10,30000,25,3,1,1,30,500,2]
		Performance	[4,43875,750,1332]
		Expense	[2.16,3.50,0.50]

Phase 6 – Epoch 24: Mission 5 in Context 1-1

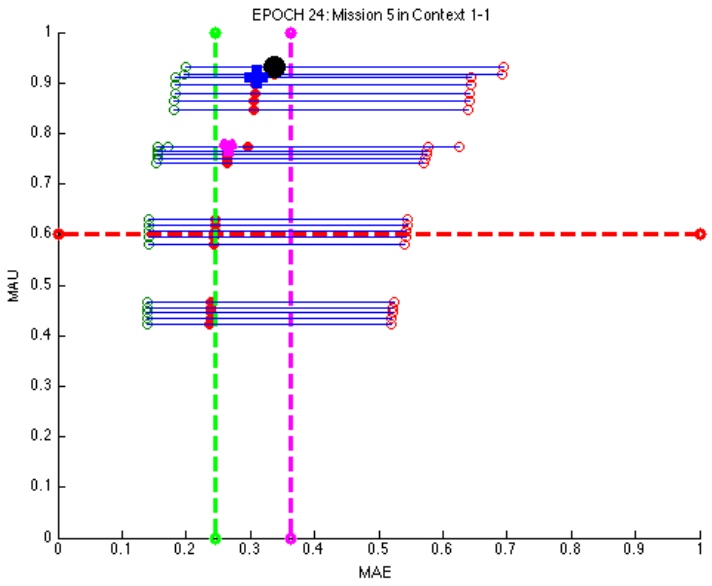


Figure 4-16: Epoch 24 Pareto front design points with 99% MAE confidence intervals

Design vector for SYS-DV-1 to SYS-DV-9:

```
[spacecraft_lifetime, initial_data_packet_capacity, annual_data_packet_usage_rate,
FSS_interface_communications_technology, terrestrial_capacity_real_option,
propulsion_type, payload_capacity, propellant_mass, launch_vehicle]
```

Performance vector for SYS-PA-1 to SYS-PA-4:

```
[ISL_capability, Annual_Free_Data_Capacity, Annual_Science_Value, Delta_V]
```

Expense vector for SYS-EA-1 to SYS-EA-3:

```
[Development Cost ($ million), LaunchCost ($ million), Development_Time (years)]
```

Table 4-14: Preferred designs for Epoch 24 – Mission 5 in Context 1-1

Epoch 24			
Design Number	**Risk**	**Type**	**System Vector**
17206 (3383 kg)	**High (1)**	**Design**	[10,30000,25,3,1,3,100,2000,2]
		Performance	[4,41250,2500,26305]
		Expense	[2.16,4.31,1.25]
17194 (3313 kg)	**Med (2)**	**Design**	[10,30000,25,3,1,3,30,2000,2]
		Performance	[4,43875,750,27218]
		Expense	[2.16,4.22,1]
17190 (2063 kg)	**Low (3)**	**Design**	[10,30000,25,3,1,3,30,1000,2]
		Performance	[4,43875,750,19501]
		Expense	[2.16,2.63,1]

Phase 7 – Epoch 19: Mission 0 in Context 1-1

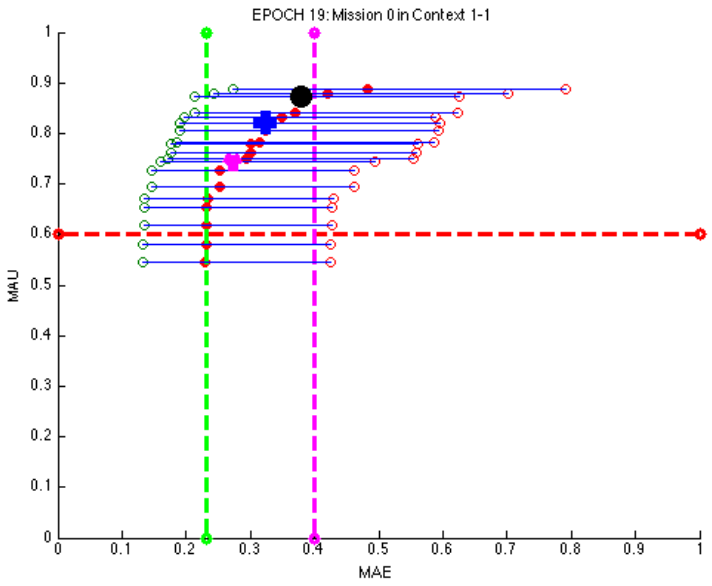


Figure 4-17: Epoch 19 Pareto front design points with 99% MAE confidence intervals

Design vector for SYS-DV-1 to SYS-DV-9:

```
[spacecraft_lifetime, initial_data_packet_capacity, annual_data_packet_usage_rate,
FSS_interface_communications_technology, terrestrial_capacity_real_option,
propulsion_type, payload_capacity, propellant_mass, launch_vehicle]
```

Performance vector for SYS-PA-1 to SYS-PA-4:

```
[ISL_capability, Annual_Free_Data_Capacity, Annual_Science_Value, Delta_V]
```

Expense vector for SYS-EA-1 to SYS-EA-3:

```
[Development Cost ($ million), LaunchCost ($ million), Development_Time (years)]
```

Table 4-15: Preferred designs for Epoch 19 – Mission 0 in Context 1-1

Epoch 24			
Design Number	**Risk**	**Type**	**System Vector**
17202 (2133 kg)	**High (1)**	**Design**	[10,30000,25,3,1,3,100,1000,2]
		Performance	[4,41250,2500,18606]
		Expense	[2.16,2.72,1.25]
17190 (2063 kg)	**Med (2)**	**Design**	[10,30000,25,3,1,3,30,1000,2]
		Performance	[4,43875,750,19501]
		Expense	[2.16,2.63,1]
17106 (2003 kg)	**Low (3)**	**Design**	[10,30000,25,3,1,1,100,1000,2]
		Performance	[4,41250,2500,2034]
		Expense	[2.16,2.55,0.625]

4.10 Discussion of System Level Results

Now, comparing the design vectors of the preferred designs across all seven epochs, a number of commonalities can be noted. Nearly all the designs are in the 17000 series, implying that a certain combination of design variables enable the preferred designs to be Pareto optimal in these epochs. **Common design variables** include a spacecraft lifetime of 10 years, initial data packet capacity of 30000, data packet usage rate per kilogram of payload of 25, exercising the terrestrial capacity real option, and using the Falcon 9 Heavy LV.

Slight variations are noted for the remaining design variables, namely the FSS communications interface technology, propulsion technology, payload capacity and propellant mass. **Higher risk designs** tended to use laser communications modules, more advanced propulsion systems like electric propulsion, higher payload capacities, and carry more onboard propellant. This is expected, as the usage of new technologies and the addition of more capabilities will incur higher degrees of cost and time budget risks. However, the benefit of higher risk is having greater performance attributes.

Higher risk designs generally have at least one attribute that is significantly greater than a conservative lower risk design, such as greater ISL capability, higher annual free data packet capacity, higher science value or higher delta-V. It is also interesting to note that there is an inverse relationship between the annual free data packet capacity and science value. This trend is also expected, as a greater science value implies a greater data usage rate, thereby decreasing the annual free data packet capacity of the satellite.

Comparing the expense vectors, it is worth noting that the development costs of preferred designs are almost always the same in each epoch. This is expected, as nearly all of the preferred designs are in the 17000 series, implying that they have nearly similar design vectors. Development costs for high, medium and low risk designs can remain almost similar due to the tradeoff among propulsion type, payload capacity and propellant mass. Therefore, a high-risk design may have a more advanced propulsion type, but lower payload capacity and propellant mass as compared to the medium and low risk designs.

The launch cost, however, increases with the risk level. This trend can be explained by high-risk designs having more payload capacity and onboard propellant, thereby increasing total mass and launch cost of the satellite. A similar trend is also noted for the development time, where higher-risk designs have longer development times. This trend is also attributed to the same reason.

The total mass of each design was also tracked. The general trend is that higher-risk designs are heavier, as they may be using more advanced propulsion technologies, or have higher payload capacity, or carry more onboard propellant. However, there are instances where a higher risk design is lighter than a lower risk design. This can be attributed to the tradeoff among the aforementioned design variables. The masses of the satellites are between 1373 kg to 3383 kg, thereby confirming the assumption that they are large satellites from the outset.

Therefore, these designs are possibly candidates for constellation development and they will be used as design variables in the program-level tradespace model. Each of these satellites can operate in LEO to support FSS operations, as they have satisfactory performance attributes such as high ISL capability (Levels 3 and 4) and high initial data packet capacity (30000 packets).

Annual science value and delta-V may vary more from satellite to satellite in different epochs, but the satellites are designed accorded to the specified mission needs and contexts. If the epoch demands greater science value, then the satellites designed will have greater science value capacity. If the epoch demands an orbit reconfiguration, then the satellites designed will have greater delta-V.

Therefore, the preferred designs selected for each epoch are determined to be Pareto-optimal, as well as affordable and technically satisfactory within their corresponding epochs.

4.11 Differences in using MAE and Cost

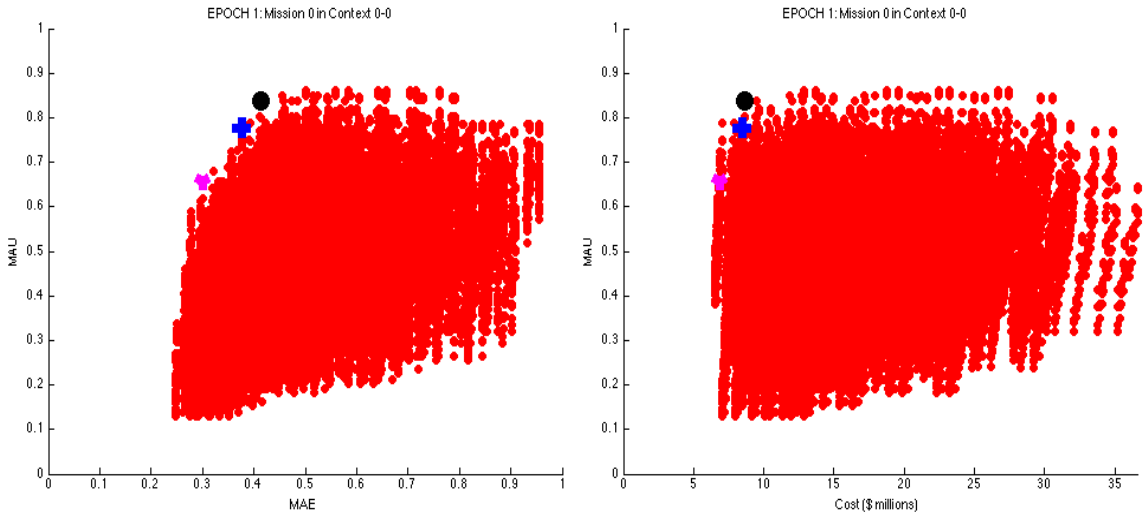


Figure 4-18: Comparing tradespaces for Epoch when MAU is plotted against MAE (left) and when MAU is plotted against total cost (right).

A tradespace parameterized by MAU and cost was also plotted and compared against the tradespace parameterized by MAU and MAE (same as Figure 4-2). The baseline color of the tradespace is red in order to show the high, medium, and low risk designs clearly on both tradespaces. There is clearly a difference in the tradespaces, and having MAE on the x-axis yields a tradespace where the Pareto front is an obvious curve. The three preferred system designs selected in Epoch 1 are shown in both tradespaces. Clearly, parameterizing a tradespace by MAU and cost will guide decision makers towards choosing different preferred designs.

In this epoch, the medium risk design will not have been chosen at all in the MAU-Cost tradespace as it is dominated by designs along its Pareto front. There is thus a strong likelihood that Pareto frontier set designs in the MAU-Cost tradespace will become dominated designs in the MAU-MAE space. While this comparison of tradespaces is not conducted for the remaining epochs, there is sufficient indication to show that considering MAE and considering only cost can yield substantially different results in tradespace analysis, especially when identifying Pareto frontiers and selecting preferred designs.

Therefore, the consideration of additional cost and time elements, as well as stakeholder preferences towards these resource expenses, can provide new information about design concepts that would not have been identified if only total cost was considered. Therefore, parameterizing a tradespace by MAE instead of cost enables the conduct of a more holistic form of tradespace exploration and provides decision makers with new information that can help them design for affordability.

4.12 Summary of System Level Analysis

A tradespace model was formulated for system level analysis of an FSS-enabled satellite. Uncertainty of the different expense elements was also accounted for through the use of lognormal distribution with different variances. The Pareto fronts in selected tradespaces were identified and three preferred system designs were eventually down-selected. Their design variables and attributes were also further investigated to identify commonalities that may reveal key design principles for affordability. The preferred system designs will then be used as design variables for program level analysis.

5 FEDERATED SATELLITE SYSTEMS PROGRAM ANALYSIS

5.1 Program Definition

After selecting 3 preferred satellite system designs in each epoch, the design for affordability process proceeds to program-level analysis. Program-level analysis seeks to determine the most preferred program designs in each epoch, so that the most preferred satellite constellations could be developed and launched. In each epoch, each program or constellation design contains multiple units of one of the 3 preferred satellite system designs that may be developed in different ways or work together to produce an emergent capability. The design variables, performance attributes, and expense attributes will hence be different at the program level. The tradespace formulation and exploration process conducted at the system level will be conducted for the program level in the same manner, but with different design variables, with both new and aggregated performance attributes, and with both new and aggregated expense attributes. Epoch variables, development phases and contexts remain the same in accordance to the hypothetical FSS development roadmap. A new performance attribute is **constellation maneuverability**, and new expense attributes are **labor cost, operations cost, retirement cost and waiting time to launch**. More attributes are expected since program-level analysis is logically more complex.

The key decision makers at all levels of the design process, be it system, program or portfolio, are the FSSD and all the DMs.

The mission concept at the program level is then to develop a single satellite constellation that has FSS-enabling capabilities, with *stakeholder-desired* levels for annual free data capacity, annual science value, and maneuverability, as well as *stakeholder-desired* levels of expenditure for constellation development cost, constellation launch cost, constellation labor cost, constellation operations cost, constellation retirement cost, and waiting time prior to constellation launch.

The **performance, cost and schedule attributes** are thus derived from the program-level mission statement. These attributes are listed and described in Table 5-1, where system-level performance attributes are labeled PRG-PA and system-level resource expense attributes are labeled PRG-EA. Attributes across performance, cost and time are considered for the program-level analysis in this case study. **It is also assumed that all attributes are independent of each other such that the stakeholder preferences for one attribute have no impact on the preference for another.**

Table 5-1: Program Level Performance and Expense Attributes

Program Level Attributes	Description
PRG-PA-1 **Constellation Annual Free Data Capacity**	The **constellation annual free data capacity** is a proxy measure of the in-space data assets that a constellation can offer for opportunistic sharing in the FSS. In this case study, it is simply calculated as the product of the annual free data capacity per satellite and the number of satellites in the constellation. However, this capacity may decrease over time due to data storage. **The annual constellation free data capacity will be measured using "the number of free data packets available".**
PRG-PA-2 **Constellation Annual Science Value**	The **constellation annual science value** is a proxy measure of the constellation annual contribution to space science. It can be calculated by the payload mass and the annual data packet usage per kilogram of payload. It assumes that the constellation is always in constant operation. The unit of measurement for annual science value of a constellation is the **number of data packets**.
PRG-PA-3 **Constellation Maneuverability**	The **constellation maneuverability** is a proxy measure of a constellation's ability to reconfigure all its satellites' positions and orbits in order to achieve maximum effect for FSS operations. Since all satellites in a constellation have the same delta-V, they can alter the constellation's configuration in the same manner. The orbit altitude and inclination angles can be changed to an extent determined by the satellite's delta-V. While the satellites are meant to serve FSS operations mainly in LEO, they can also move out to other orbits to fulfill different science missions if their delta-V capabilities permit.
PRG-EA-1 **Constellation Development Cost**	The **constellation development cost** is the total monetary cost required to design, construct, assemble, integrate and test all satellites before they are ready for operations. It is calculated as the sum of the development cost of all satellites. As satellites are now produced in bulk, there may be economies of scale involved. With more satellites produced, the greater the **discount factor on the development cost per satellite**. It is measured in millions of dollars. The constellation development cost is logically paid in fixed annual installments over the duration of the program contract.

Program Level Attributes	Description
PRG-EA-2 **Constellation Launch Cost**	The **constellation launch cost** is the total monetary cost required to launch the constellation to LEO. It is calculated as the sum of the launch cost of all satellites. It is a function of the total mass of the satellite, including its onboard propellant, and it is dependent on the choice of launch vehicle. However, given that all the satellites will not be able to occupy the entire payload capacity of the Falcon 9 Heavy, they may be a **cost penalty** applied. Occupying the capacity entirely will make the launch most cost effective. If there is still capacity for more payloads, the penalty is applied to account for the inefficiency in payload capacity usage. The lower the percentage payload capacity of the LV or RLV that the constellation occupies, the greater the cost penalty. It is measured in millions of dollars. The constellation launch is logically paid in **fixed annual installments** over the duration of the **program contract**.
PRG-EA-3 **Constellation Labor Cost**	The **constellation labor cost** is the total monetary cost required to pay the work force that is responsible for developing the satellite constellation up till the time of launch. Assuming that there are a fixed number of workers responsible for the development of one satellite at a time and each worker is paid for a fixed salary per year, the constellation labor cost can be found by multiplying the number of workers by the annual fixed salary by the number of satellites being developed at a time. A worker's salary may also increase over time when he stays longer in the development project. Hence, an **interest rate** may be applied to it. The labor cost is measured in millions of dollars. The annual constellation labor cost may hence increase over the years up till the time of launch.
PRG-EA-4 **Constellation Operations Cost**	The **constellation operations cost** is the total monetary cost required to operate all the satellites in the constellation. It is reflective of the labor cost and maintenance cost required to keep the constellation operating in its desired manner. This can be calculated by first assuming a fixed cost for annual operations cost of a satellite, and then multiplying it by the number of satellites in the constellation. The annual constellation operations cost may also increase over time due to inflation or economic fluctuations. Hence, it may increase over the years up till the time of constellation retirement. An interest rate may thus be applied to this cost. If the constellation is operated beyond its projected lifetime, annual operations cost will continue to be incurred at the same interest. It is measured in millions of dollars.

Program Level Attributes	Description
PRG-EA-5 **Constellation Retirement Cost**	The **constellation retirement cost** is the total monetary cost required to retire and de-orbit all the satellites in a single instance. It can be calculated by first assuming a fixed cost to retire a satellite, and then multiplying it by the number of satellites in the constellation. If the constellation is operated only up till its projected lifetime, then the retirement cost is incurred at the end of the lifetime and it is non-recurring. If the constellation is operated beyond its projected lifetime and into legacy, then the retirement cost will not be incurred at all. It is measured in millions of dollars.
PRG-EA-6 **Waiting Time To Launch**	The **waiting time to launch** is the number of years rounded up taken for the entire constellation to be developed and launched. It is a function of the total development time, which is dependent on how long it takes to develop one satellite, how many satellites are being developed at the same time, and the total number of satellites in the constellation. The more satellites being developed concurrently, the lower the total development time, but at the price of higher labor costs. However, given that all the satellites will not be able to occupy the entire payload capacity of the Falcon 9 Heavy, they may be a **time penalty** applied. Occupying the capacity entirely will make the launch most time- effective. If there is capacity for more payloads, the penalty is applied to account for the inefficiency in payload capacity usage. This time penalty simulates the extra time needed to wait on the development of non-FSS space systems that will be loaded onto the same LV. The lower the percentage payload capacity of the LV that the constellation occupies, the greater the time penalty.

5.2 Program Level Design Variables

Six design variables have been chosen for program level design to drive the attributes described above, and they are listed and described in Table 4-17. Assuming all combinations of design variables will yield feasible architectures for a satellite constellation, there are a total of **1080 designs**. The epoch variables considered for program level design are the same as that for system level design and they are based on the assumptions made for laser communications and RLV technology.

Table 5-2: Program Level Design Variables

Design Variables	Description	Factor Levels (units)
PRG-DV-1 **Satellite System Design Choice**	**3 preferred designs** were chosen from the Pareto frontier of the tradespace in each of the 7 epochs. These designs are of **high, medium and low risk** and they are possible candidates for constellation development. A constellation of high-risk designs may yield high performance, but there is a strong likelihood that it will be unaffordable due to compounding risks. On the other hand, a constellation of low-risk designs may always be affordable, but it is not comparable to higher-risk designs in terms of performance. The 3 designs are labeled factor levels 1, 2 and 3. The **delta-V capability** of the choice of satellite will also directly impact the **maneuverability of the entire constellation**. All other attributes and design variables at the program level depends on the system level attributes and design variables of the single satellite system.	High Risk **(1)** Medium Risk **(2)** Low Risk **(3)**
PRG-DV-2 **Number of Satellites**	The **number of satellites** is an obvious design variable for a satellite constellation. Having more satellites in a constellation increases its coverage and accessibility to customer spacecraft that wish to leverage FSS capabilities. More satellites also mean greater value to science since more missions can be conducted. However, it also entails more wet mass, thereby increasing development cost, launch cost, labor cost, operations cost and development time. There is hence a tradeoff between FSS capabilities and science value with costs, time and risk when varying the number of satellites in the constellation. The number of satellites is critical to program level attributes that are based on the aggregation of system level attributes.	**4** satellites **6** satellites **8** satellites **10** satellites **12** satellites

Design Variables	Description	Factor Levels (units)
PRG-DV-3 **Number of Satellites in Concurrent Development**	The **number of satellites in concurrent development** is the number of satellites being developed at the same time. Increasing the number of satellites in parallel development can ensure that the constellation is launched in a much shorter time. This way the satellite constellation will not miss the opportunity to serve more customer spacecraft if satellites are built one at a time over many years. However, having more in parallel development incurs higher labor cost since more people are required to develop, assemble, integrate and test multiple satellites at the same time. More facilities are required and this is factored into the labor cost.	**1** satellite **2** satellites **3** satellites **4** satellites
PRG-DV-4 **Constellation Type**	Satellites in a constellation can be **aggregated, disaggregated or mixed**. Satellites in an aggregated constellation have payloads that are designed to come together to produce an emergent capability that can greatly increase value to science. Satellites can also exist in a mixed orbit, where only some but not all satellites have payloads that come together to produce a smaller emergent capability. Finally, disaggregated satellites have payloads that do not work together for emergent capability.	Aggregated **(1)** Disaggregated **(2)** Mixed **(3)**
PRG-DV-5 **Legacy Operation**	Some satellite constellations can be chosen to **operate for legacy**, meaning that they remain in operation well beyond their projected lifetimes till they malfunction. If not operated for legacy, they can be chosen to retire exactly or sometime after their projected lifetimes.	Not for legacy **(0)** For legacy **(1)**
PRG-DV-6 **Program Contract Length**	**Program contract length** is an important design variable, as it determines how costs are spread out over the coming years. A longer contract length means that the **fixed installments** are smaller and decision makers can use this opportunity to develop concurrent programs under the same budget. A shorter contract length will produce the reverse. Shorter program contract lengths of 4, 7 and 10 years are considered.	**4** years **7** years **10** years

5.3 Program Level Design-to-Value Mapping

DVM of the program design variables to the performance and resource attributes was then performed and the results are shown in Table 5-3. More attention was paid to rows and columns with high cumulative values to determine the variables and attributes that have the highest degrees of interaction. Interactions with perceived matrix values of 9 will definitely be accounted for in the system tradespace model, since they are the most critical. Not all interactions with perceived values of 1 or 3 are accounted for.

Table 5-3: DVM for Satellite Constellation Program

	Annual Free Data Capacity	Annual Science Value	Maneuverability	Total Dev. Cost	Total Launch Cost	Total Labor Cost	Total Operations Cost	Total Retirement Cost	Waiting time to launch	Row Total
Satellite Design Choice	9	9	9	9	9				9	63
Number of Satellites	9	9		9	9	9	9	9	9	72
Number of Satellites in concurrent development						9			9	18
Constellation Type	9	9					9			27
Legacy Operation	3						9			12
Program Contract Length				1	1					2
Laser Comms TRL	9			9					3	21
RLV Tech.					9					9
Column Total	39	27	9	28	28	18	27	9	30	

Equations were then formulated for each interaction to establish the computerized model. These program tradespace model equations are not shown in this section and they can be found in the computer code attached in Appendix 2. The scientific principles used were based mainly on the methods for spacecraft subsystem design prescribed by (Wertz and Larson, 2011). Some equations were heuristically derived, as there were no precedent cases for reference. All the equations and their basis for usage are described qualitatively below.

The **number of satellites** in the constellation is obviously a very important design variable, as it affects many program level attributes that are calculated based on the aggregate sum of system level attributes. Hence, it has significant impacts on constellation annual free data capacity, constellation annual science value, total development cost, total launch cost, total labor cost, total operations cost, total retirement cost, and waiting time to launch. It only does not have impact on maneuverability because the delta-V capabilities are all the same.

The **satellite design choice** for the constellation is another obvious design variable, as it also affects many program level attributes that are calculated based on the aggregate sum of system level attributes. It has significant impacts on all performance attributes, total development cost, total launch cost, and waiting time to launch. It has no impact on labor costs, operations costs and retirement costs because they have been assumed to incur at the same rate in the program tradespace model.

The **number of satellites in concurrent development** affects only the total labor cost and waiting time to launch. Having more satellites in parallel development will require a bigger workforce, thereby incurring greater labor cost. However, the waiting time to launch is reduced because the overall development time for all the satellites is reduced. Even if time penalties are imposed, having a significant number of satellites in concurrent development can potentially offset that penalty. It has no impact on performance attributes and other expense attributes.

The **constellation type** has significant impacts on annual free data capacity, annual science value and operations cost. In an aggregated setup, the satellites can collaborate to produce emergent capabilities that can greatly increase annual science value. However, in order to produce this new capability, a significant amount of data packet capacity has to be consumed and this reduces the annual free data packet capacity that may facilitate FSS operations. In a mixed setup, only some but not all satellites will collaborate and produce the emergent capability. Hence, the increase in annual science and decrease in data packet capacity is of a lesser degree. In a disaggregated setup, all satellites are independently functioning and the annual science value is simply the sum of science values of all the satellites. There is no decrease in free data packet capacity, as no emergent capability is produced. The constellation type also affects operations cost, as having some kind of emergent capability requires more work on the ground for monitoring and assessment, thereby increasing operations cost. Therefore, an aggregated constellation will have the highest operations cost.

The choice of **legacy operation** has a significant impact on operations cost but a smaller impact on annual free data capacity. If operated for legacy, the satellites have to operate well beyond their projected lifetimes and this increases total operations cost for the entire constellation when accumulated over many years. It has a smaller impact on annual free data capacity, as a constellation operated for legacy may experience equipment degradation over many years. This can be represented by apply a discount rate to the annual free data capacity during its years of operation.

The **program contract length** only has little impact on total development cost and launch cost, but it actually has more impact on the *annual development and launch cost*. Since the development and launch cost for the constellation is paid in installments over the agreed duration for the contract, a shorter contract length will imply a higher fixed installment and a longer contract length will imply a lower installment. Despite its low impact, it is still kept as a program design variable, as it will greatly affect the cost commitment profile of a program over its development lifecycle. Its relevance will be seen later in portfolio level analysis.

The epoch variable of **Laser Communications TRL** has significant impacts on annual free data capacity and total development cost, but a smaller impact on waiting time to launch. When the TRL increases in the future, ISL capability of the satellites reaches to an unprecedented new level and more data packet capacity will be consumed in support of FSS operations. Development cost and development time also decreases significantly due to maturity in technology. The epoch variable of **RLV TRL** is perceived to only have impacts on the launch cost. If the TRL increases in the future, more launches can be conducted every year and this can significantly reduce the launch cost per kilogram of any spacecraft.

5.4 Generation of Program Tradespaces and Design Identification

The program tradespace model was then developed. Design vectors, performance vectors and expense vectors for the 1080 designs were obtained for the 7 selected epochs. New program stakeholder preferences were hypothetically established. MAUT was then applied and a unique SAU curve was obtained for each attribute in each epoch. The SAU curves are not described or shown for purposes of brevity. MAE and MAU values were calculated for all designs and illustrated on a scatter plot for each of the 7 epochs. Upper bound constraints on all expense attributes meant to simulate program monetary and time budget were also hypothetically created based on the epoch description to give **maximum expense constraint levels**. These lines are colored in **magenta**. Similarly, lower bound constraints were hypothetically created based on epoch descriptions to give **minimum utility constraint levels**. These lines are colored in **red**. The intersection between the closest design point on the tradespace and the minimum utility constraint level then gives the **derived minimum expected expense level**. These lines are colored in **green**. The constraint levels were applied to every tradespace to determine the affordable solution region. The constraint levels for each epoch are shown in vectors below.

Program Minimum Performance Constraint Levels = [Annual Data Capacity, Annual Science Value, Maneuverability]

Epoch 1: [250000, 25000, 1]

Epoch 2: [250000, 100000, 1]

Epoch 3: [250000, 25000, 1]

Epoch 10: [250000, 20000, 2]

Epoch 17: [250000, 50000, 2]

Epoch 24: [300000, 15000, 3]

Epoch 19: [300000, 25000, 2]

Program Maximum Expense Constraint Levels = [Development Cost ($m), Launch Cost ($m), Labor Cost ($m), Operations Cost ($m), Retirement Cost ($m), Waiting Time to Launch (Years), Program Contract Length (Years)*]

Epoch 1: [25.0, 60, 250, 35, 10, 10, 10]

Epoch 2: [25.0, 55, 225, 30, 10, 10, 7]

Epoch 3: [25.0, 55, 200, 30, 10, 10, 7]

Epoch 10: [22.5, 50, 200, 27.5, 10, 10, 7]

Epoch 17: [22.5, 45, 200, 25, 6, 8, 7]

Epoch 24: [22.5, 45, 200, 25, 6, 8, 7]

Epoch 19: [20.0, 40, 180, 25, 6, 8, 7]

2 or 3 preferred designs for a satellite constellation are then chosen from the affordable solution. Like system level analysis, a **high-risk (black), medium-risk (red), and low-risk (cyan)** program designs were selected. Neither the Pareto front nor the MAE confidence intervals were identified for program level analysis in this case study. Hence, the 'risk' assumed in this analysis is the measure of how close the MAE value of a program design point is to the maximum expense constraint level. The closer it is to the maximum expense line, the more risky it is. However, they can be done in future studies should decision makers be interested in the probabilities of selected program designs remaining affordable. The results of program level tradespace analysis are shown in the subsequent figures below.

CONSTELLATION 1 – Developed in Epoch 1: Mission 0 in Context 0-0 (100% Yield)

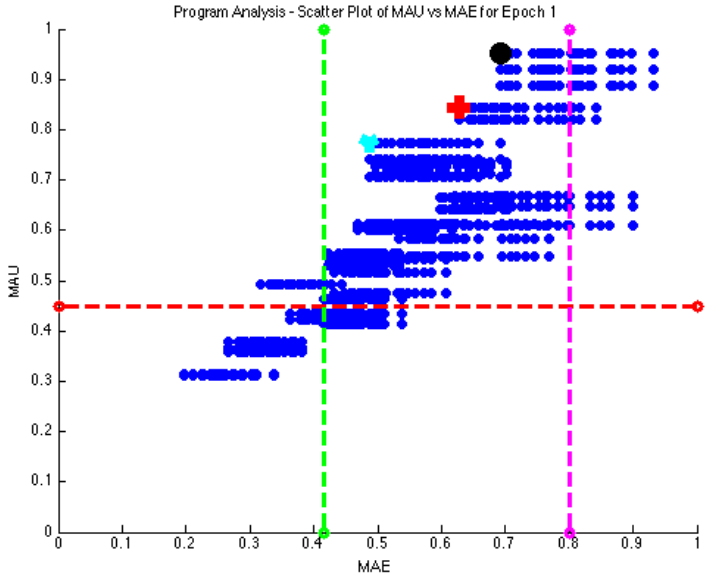


Figure 5-1: System Tradespace for Phase 1 – Epoch 1 – Mission 0 in Context 0-0

The design vector lists the factor levels for PRG-DV-1 to PRG-DV-6:

`[satellite_system_design, number_of_satellites, number_of_satellites_in_concurrent_deelopment, constellation_type, legacy_operation, program_contract_length]`

The performance vector lists the factor levels for PRG-PA-1 to PRG-PA-3:

`[Annual_Free_Data_Capacity, Annual_Science_Value, Maneuverability]`

The expense vector lists the factor levels for PRG-EA-1 to PRG-EA-6:

`[Total_Development_Cost ($ million), Total_Launch_Cost ($ million), Total_Labor_Cost ($ million), Total_Operations_Cost ($ million), Total_Retirement_Cost ($ million), Waiting_time_to_launch (years), Program_Contract_Length (years)*]`

Table 5-4: Preferred designs for Epoch 1 – Mission 0 in Context 0-0

Epoch 1			
Design Number	**Risk**	**Type**	**Program Vector**
349	**High (1)**	**Design**	[1,12,4,2,0,4]
		Performance	[495000,30000,3]
		Expense	[30.76,58.97,251.56,6.03,12,10,4]
283	**Med (2)**	**Design**	[1,10,4,3,0,4]
		Performance	[330000,28125,3]
		Expense	[27.23,49.15,220.53,5.03,10,9,4]
1051	**Low (3)**	**Design**	[3,12,3,2,0,4]
		Performance	[495000,30000,1]
		Expense	[30.76,44.44,143.24,6.03,12,8,4]

CONSTELLATION 2 – Developed in Epoch 2: Mission 1 in Context 0-0 (100% Yield)

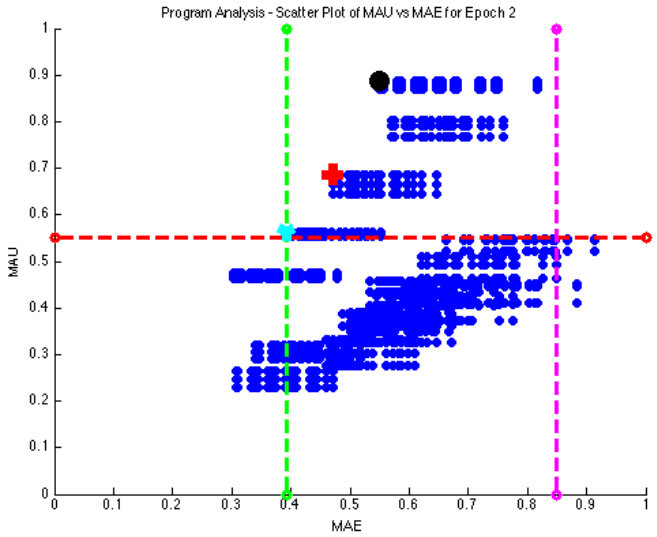


Figure 5-2: System Tradespace for Phase 2 – Epoch 2 – Mission 1 in Context 0-0

The design vector lists the factor levels for PRG-DV-1 to PRG-DV-6:

`[satellite_system_design, number_of_satellites, number_of_satellites_in_concurrent_deelopment, constellation_type, legacy_operation, program_contract_length]`

The performance vector lists the factor levels for PRG-PA-1 to PRG-PA-3:

`[Annual_Free_Data_Capacity, Annual_Science_Value, Maneuverability]`

The expense vector lists the factor levels for PRG-EA-1 to PRG-EA-6:

`[Total_Development_Cost ($ million), Total_Launch_Cost ($ million), Total_Labor_Cost ($ million), Total_Operations_Cost ($ million), Total_Retirement_Cost ($ million), Waiting_time_to_launch (years), Program_Contract_Length (years)*]`

Table 5-5: Preferred designs for Epoch 2 – Mission 1 in Context 0-0

Epoch 2			
Design Number	**Risk**	**Type**	**Program Vector**
331	**High (1)**	**Design**	[1,12,3,2,0,4]
		Performance	[315000,150000,3]
		Expense	[39.40,64.45,122.13,6.03,12,7,4]
181	**Med (2)**	**Design**	[1,8,3,1,0,4]
		Performance	[126000,125000,3]
		Expense	[29.55,48.34,102.03,4.02,8,6,4]
109	**Low (3)**	**Design**	[1,6,3,1,0,4]
		Performance	[94500,93750,3]
		Expense	[23.39,40.28,102.03,3.02,6,6,4]

CONSTELLATION 3 – Developed in Epoch 3: Mission 2 in Context 0-0 (100% Yield)

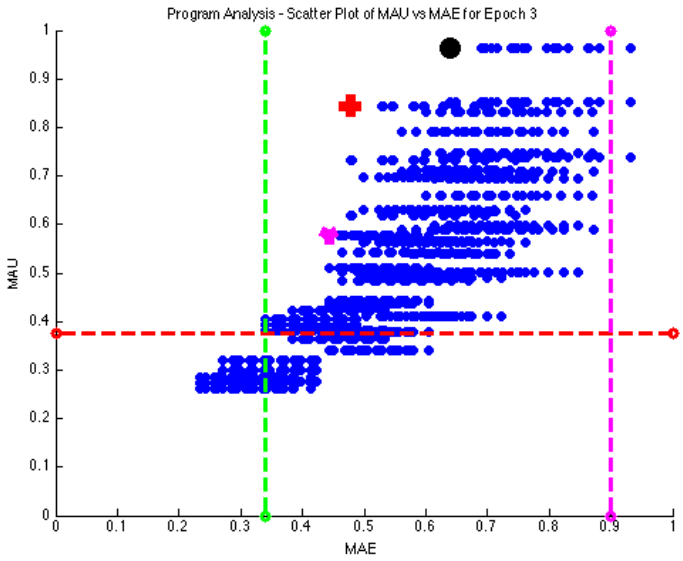


Figure 5-3: System Tradespace for Phase 3 – Epoch 3 – Mission 2 in Context 0-0

The design vector lists the factor levels for PRG-DV-1 to PRG-DV-6:

`[satellite_system_design, number_of_satellites, number_of_satellites_in_concurrent_deelopment, constellation_type, legacy_operation, program_contract_length]`

The performance vector lists the factor levels for PRG-PA-1 to PRG-PA-3:

`[Annual_Free_Data_Capacity, Annual_Science_Value, Maneuverability]`

The expense vector lists the factor levels for PRG-EA-1 to PRG-EA-6:

`[Total_Development_Cost ($ million), Total_Launch_Cost ($ million), Total_Labor_Cost ($ million), Total_Operations_Cost ($ million), Total_Retirement_Cost ($ million), Waiting_time_to_launch (years), Program_Contract_Length (years)*]`

Table 5-6: Preferred designs for Epoch 3 – Mission 2 in Context 0-0

Epoch 3			
Design Number	**Risk**	**Type**	**Program Vector**
349	**High (1)**	**Design**	[1,12,4,2,0,4]
		Performance	[495000,30000,3]
		Expense	[30.76,58.98,251.56,6.03,12,10,4]
1069	**Med (2)**	**Design**	[3,12,4,2,0,4]
		Performance	[495000,30000,1]
		Expense	[30.76,44.44,162.84,6.03,12,7,4]
889	**Low (3)**	**Design**	[3,8,2,2,0,4]
		Performance	[330000,20000,1]
		Expense	[23.07,44.44,110.27,4.02,8,9,4]

CONSTELLATION 4 – Developed in Epoch 10: Mission 3 in Context 0-1 (100% Yield)

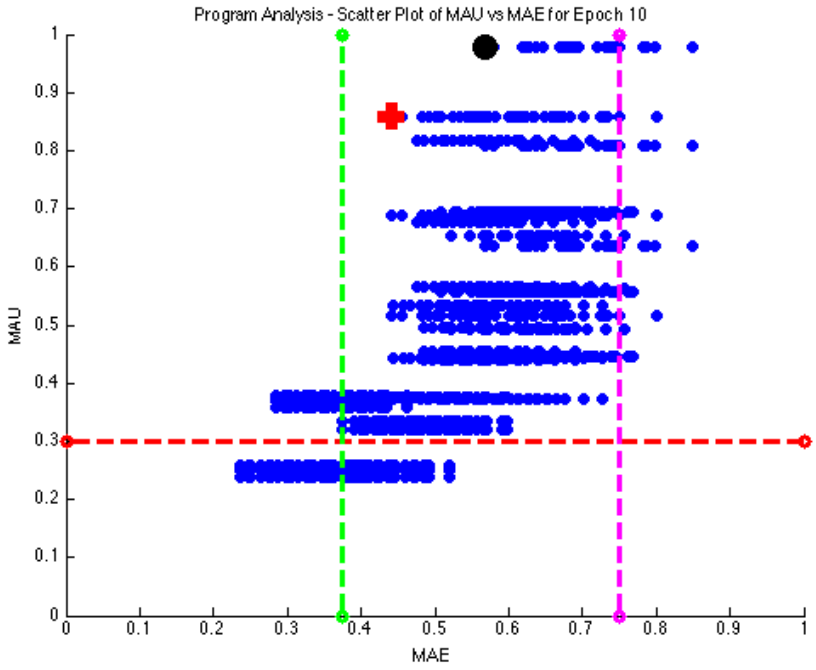


Figure 5-4: System Tradespace for Phase 4 – Epoch 10 – Mission 3 in Context 0-1

The design vector lists the factor levels for PRG-DV-1 to PRG-DV-6:

`[satellite_system_design, number_of_satellites, number_of_satellites_in_concurrent_deelopment, constellation_type, legacy_operation, program_contract_length]`

The performance vector lists the factor levels for PRG-PA-1 to PRG-PA-3:

`[Annual_Free_Data_Capacity, Annual_Science_Value, Maneuverability]`

The expense vector lists the factor levels for PRG-EA-1 to PRG-EA-6:

`[Total_Development_Cost ($ million), Total_Launch_Cost ($ million), Total_Labor_Cost ($ million), Total_Operations_Cost ($ million), Total_Retirement_Cost ($ million), Waiting_time_to_launch (years), Program_Contract_Length (years)*]`

Table 5-7: Preferred designs for Epoch 10 – Mission 3 in Context 0-1

Epoch 10			
Design Number	**Risk**	**Type**	**Program Vector**
349	**High (1)**	**Design**	[1,12,4,2,0,4]
		Performance	[495000,30000,3]
		Expense	[30.76,46.43,284.14,6.03,12,11,4]
1069	**Med (2)**	**Design**	[3,12,4,2,0,4]
		Performance	[495000,30000,1]
		Expense	[30.76,44.44,162.84,6.03,12,7,4]

CONSTELLATION 5 – Developed in Epoch 17: Mission 4 in Context 1-0 (100% Yield)

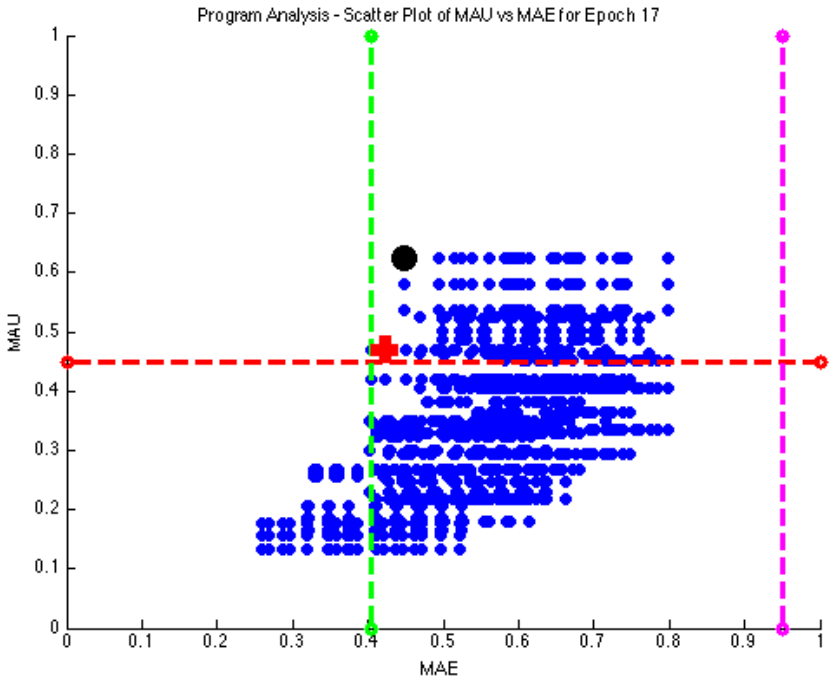


Figure 5-5: System Tradespace for Phase 5 – Epoch 17 – Mission 4 in Context 1-0

The design vector lists the factor levels for PRG-DV-1 to PRG-DV-6:

`[satellite_system_design, number_of_satellites, number_of_satellites_in_concurrent_deelopment, constellation_type, legacy_operation, program_contract_length]`

The performance vector lists the factor levels for PRG-PA-1 to PRG-PA-3:

`[Annual_Free_Data_Capacity, Annual_Science_Value, Maneuverability]`

The expense vector lists the factor levels for PRG-EA-1 to PRG-EA-6:

`[Total_Development_Cost ($ million), Total_Launch_Cost ($ million), Total_Labor_Cost ($ million), Total_Operations_Cost ($ million), Total_Retirement_Cost ($ million), Waiting_time_to_launch (years), Program_Contract_Length (years)*]`

Table 5-8: Preferred designs for Epoch 17 – Mission 4 in Context 1-0

Epoch 17			
Design Number	**Risk**	**Type**	**Program Vector**
331	**High (1)**	**Design**	[1,12,3,2,0,4]
		Performance	[315000,150000,1]
		Expense	[20.77,50.68,82.88,6.03,12,5,4]
1051	**Med (2)**	**Design**	[3,12,3,2,0,4]
		Performance	[526500,9000,1]
		Expense	[20.77,41.95,82.88,6.03,12,5,4]

CONSTELLATION 6 – Developed in Epoch 24: Mission 5 in Context 1-1 (100% Yield)

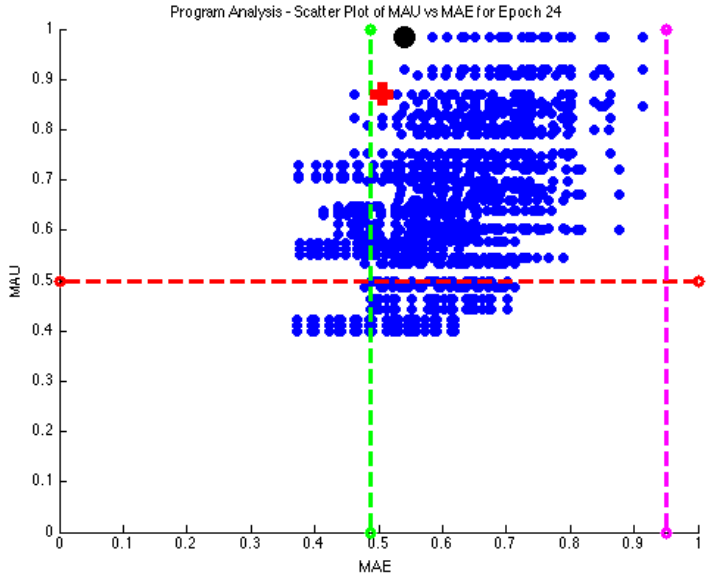


Figure 5-6: System Tradespace for Phase 6 – Epoch 24 – Mission 5 in Context 1-1

The design vector lists the factor levels for PRG-DV-1 to PRG-DV-6:

`[satellite_system_design, number_of_satellites, number_of_satellites_in_concurrent_deelopment, constellation_type, legacy_operation, program_contract_length]`

The performance vector lists the factor levels for PRG-PA-1 to PRG-PA-3:

`[Annual_Free_Data_Capacity, Annual_Science_Value, Maneuverability]`

The expense vector lists the factor levels for PRG-EA-1 to PRG-EA-6:

`[Total_Development_Cost ($ million), Total_Launch_Cost ($ million), Total_Labor_Cost ($ million), Total_Operations_Cost ($ million), Total_Retirement_Cost ($ million), Waiting_time_to_launch (years), Program_Contract_Length (years)*]`

Table 5-9: Preferred designs for Epoch 24 – Mission 5 in Context 1-1

Epoch 24			
Design Number	**Risk**	**Type**	**Program Vector**
349	**High (1)**	**Design**	[1,12,4,2,0,4]
		Performance	[495000,30000,4]
		Expense	[20.78,82.71,110.51,6.03,12,5,4]
691	**Med (2)**	**Design**	[2,12,3,2,0,4]
		Performance	[526500,9000,4]
		Expense	[20.77,80.99,82.88,6.03,12,5,4]

CONSTELLATION 7 – Developed in Epoch 19: Mission 0 in Context 1-1 (100% Yield)

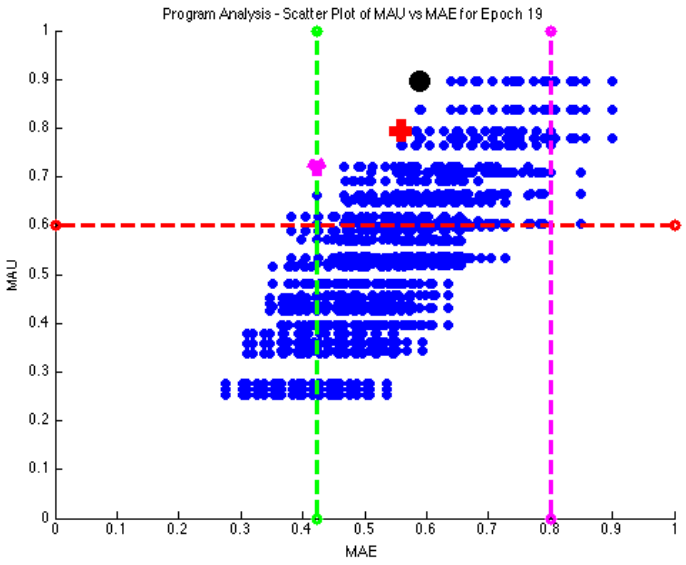


Figure 5-7: System Tradespace for Phase 7 – Epoch 19 – Mission 0 in Context 1-1

The design vector lists the factor levels for PRG-DV-1 to PRG-DV-6:

`[satellite_system_design, number_of_satellites, number_of_satellites_in_concurrent_deelopment, constellation_type, legacy_operation, program_contract_length]`

The performance vector lists the factor levels for PRG-PA-1 to PRG-PA-3:

`[Annual_Free_Data_Capacity, Annual_Science_Value, Maneuverability]`

The expense vector lists the factor levels for PRG-EA-1 to PRG-EA-6:

`[Total_Development_Cost ($ million), Total_Launch_Cost ($ million), Total_Labor_Cost ($ million), Total_Operations_Cost ($ million), Total_Retirement_Cost ($ million), Waiting_time_to_launch (years), Program_Contract_Length (years)*]`

Table 5-10: Preferred designs for Epoch 19 – Mission 0 in Context 1-1

Epoch 19			
Design Number	**Risk**	**Type**	**Program Vector**
349	**High (1)**	**Design**	[1,12,4,2,0,4]
		Performance	[495000,30000,3]
		Expense	[20.77,58.66,136.04,6.03,12,6,4]
277	**Med (2)**	**Design**	[1,10,4,2,0,4]
		Performance	[412500,25000,3]
		Expense	[18.39,48.88,136.04,5.03,10,6,4]
1051	**Low (3)**	**Design**	[3,12,3,3,1,7]
		Performance	[396000,33750,1]
		Expense	[20.77,55.08,82.88,38.05,0,5,7]

An interesting observation was made in both Constellations 1, 4 and 6, where there are points outside the affordable solution region that have expense levels lower than the derived expected minimum expense constraint level and utility levels higher than the minimum utility constraint level. This observation reflects the nature of the minimum utility constraint level, which has been imposed upon externally without knowledge of the program tradespace model and its outputs. Therefore, it is not surprising to find better program designs that can exist and they may be outside the affordable solution region that is bounded by externally imposed constraint levels.

A possible solution to this discrepancy will be to have an *a posteriori derived minimum utility* constraint level. This new minimum utility constraint level will pass through the newly identified design point with better value. This in turn warrants the need to plot an *a posteriori derived expected minimum expense* constraint level that will pass through the same design point. As such, an *a posteriori affordable solution region* can be obtained to facilitate the search for new solutions. This potential transformation is illustrated in Figure 5-8.

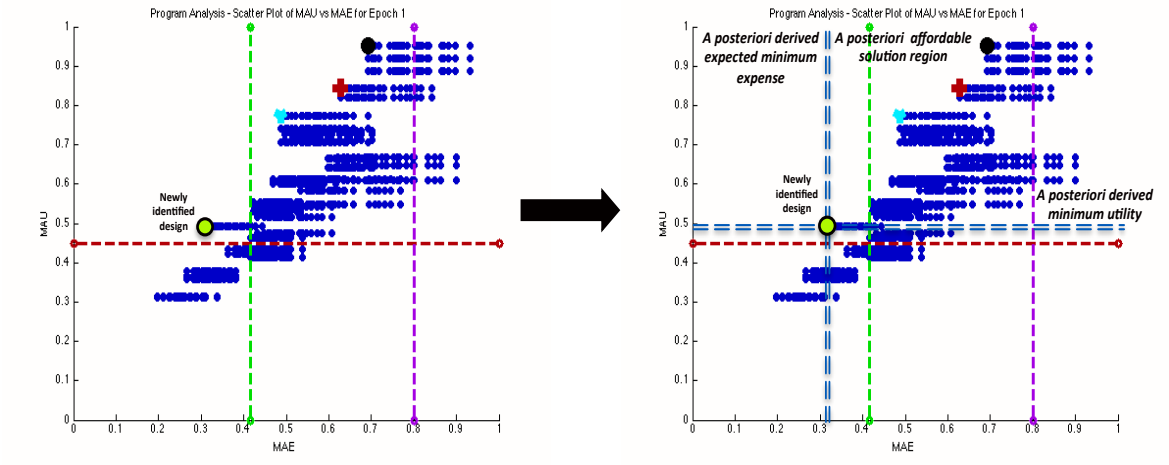


Figure 5-8: Demonstrating transformation to *a posteriori* constraint levels to determine *a posteriori* affordable solution region

This transformation can help reflect new knowledge that has been obtained through tradespace exploration. New information of better value designs with lower expense and higher utility could also be relayed to the authorities responsible for imposing the minimum utility constraint levels. This can help authorities update their mental models about designing satellite constellations in future. Transformations were not performed for the above tradespaces, as it was observed that the preferred designs would still remain in the transformed affordable solution region. Although this new region may contain new low- and medium-risk program designs, a second round of program design selection was not performed since it is beyond the scope of this thesis. Therefore, the preferred program designs selected in the above tradespaces remain valid.

After selecting two or three preferred program designs along the Pareto front of the tradespace for every epoch, there are now $\mathbf{3^4 \times 2^3 = 648}$ **possible portfolio designs** using a full factorial combination. A number of commonalities were observed among all the program designs chosen.

Firstly, majority of the constellation designs were using either the high-risk (Satellite Choice 1) or low-risk (Satellite Choice 3) satellite design. However, the number of designs chosen per epoch is too small to make only conclusive statements as to why only these two satellite design choices constitute all the satellite constellation designs. If more preferred designs were selected per epoch, there is a greater likelihood that a constellation design with medium-risk satellites will be available.

Secondly, it also appears that majority of the constellations, especially the high-risk program designs, will contain 12 satellites. Having more satellites is advantageous, as there is likely to be more free data packet capacity and science value on a whole. Also, there is more redundancy and performance of the constellations will not be as critically impacted should malfunctions occur.

Thirdly, majority of the constellations are to develop by having 3 or 4 satellites in concurrent development. Although high rates of concurrent development are costly, they can significantly reduce the development schedule and ensure that a constellation can be launched in the period it was intended. Therefore, the trade between cost and schedule can be inferred through this observation of all program design vectors.

Fourthly, a majority of the constellations have disaggregated architectures, meaning that they are not designed to perform a new task that is highly dependent on the aggregate and emergent capabilities of constituent satellites. This result ties with recent shifts towards disaggregated space architectures for lower cost and risk (Taverney – Space Review, 2011). This also reflects the tradeoff among performance, cost and risk. Having an aggregated constellation could derive more science value to stakeholders as science experiments of unprecedented scales could be conducted. However, aggregated architectures have lower levels of redundancy and are require more resources to develop. Therefore, there is a tradeoff between aggregation for greater science value and disaggregation for lower cost and shorter development times.

Lastly, almost all the constellations are not to be operated for legacy. This is also aligned to the notion of affordability, as programs operated for legacy incur operations cost that increase annually due to the growing reliance on obsolete technology to serve new purposes. Therefore, retiring a constellation at the end of its expected design lifetime will help save cost and time for the entire program. Also, a shorter contract length of 4 years is also common to all program designs. Although higher installments have to be paid across a shorter contract period, it prevents more cost from being accumulated in later years, where it becomes increasingly difficult to manage budgetary and mission uncertainties. Therefore, shorter contract lengths can prevent the accumulation of cost commitments from across all constellations over the portfolio lifecycle.

Overall, a number of tradeoffs among aggregate performance, cost and schedule could be seen in program level analysis. This allows affordable solutions within each epoch to be chosen in an informed manner. However, the constellations chosen have to operate across multiple epochs, and solutions that are affordable in one epoch may not be unaffordable in another. Therefore, there is a need to determine how the utilities and expenses of each constellation fare across other epochs. If there are solutions that become unaffordable over time, there is going to be **fewer than 648** possible portfolio designs obtained.

5.5 Expense and Utility Trajectories for Constellation Designs

Recalling that the 7 epochs are chosen and arranged in this sequence to simulate the seven development phases required for achieving FSS capability, a new and different satellite constellation is always developed at the start of every epoch. This means that there are a total of seven satellite constellations that would have participated or are still participating in FSS operations after the final launch is conducted. In this case study, a constellation launched in one epoch is expected to operate in its current epoch and the epochs beyond to sustain FSS operations. If chosen to operate for legacy, the constellation is expected to operate through its current epoch, future epochs and beyond since no external or internal disturbances are considered in this tradespace model.

If not operated for legacy, the constellation may be retired at a chosen time before the end of the seventh epoch, depending on mission and budget needs of the FSSD and DMs. As described in the hypothetical mission development timeline, a satellite constellation is developed five years before the start of the epoch and its projected launch. After which, it is expected to operate throughout that epoch and the epochs beyond. Over the course of operation, a satellite constellation is expected to fulfill unique performance requirements and keep within expense budgets when it transits from epoch to epoch. This is known as an affordable satellite constellation that is able to operate across multiple epochs. When described using the terms in MATE and EEA, this is an affordable solution that will remain within different affordable solution regions and above the minimum utility constraint levels across multiple epochs.

A **fully affordable** solution is thus one that does so for all epochs considered, while a **partially affordable** solution is one that only does so for some but not all of the epochs. A partially affordable solution can possibly transit from being affordable in one epoch and to being unaffordable in the next, and vice versa. Therefore, it will be of best interest if at least one fully affordable satellite constellation can be identified for every epoch so that all participating constellations will be affordable across all considered epochs. This eventually increases the probability that the entire portfolio development, launch and operation are also affordable. As demonstrated in the simple Space Tug case study described in Chapter 3, plotting the changes in utility and expense for different designs across multiple epochs can facilitate the identification of affordable designs. Therefore, the same approach is taken here, and the **trajectories for high, medium and low risk designs** selected in each epoch are plotted across multiple epochs. Histograms are also plotted for the high-, medium- and low- risk designs to illustrate the number of epochs in an era during which they remain in the corresponding affordable solution regions.

However, there is a difference in the meanings of the trajectories applied to the FSS case study and the Space Tug example. In the Space Tug example, the constraint levels in each epoch are applied to the same set of Space Tug program designs. In the FSS case study, the constraint levels for one epoch not only applies to the development, launch and operation of a set of constellation designs that is projected for launch in the same epoch, but also to the operation of other sets of constellation designs that were developed, launched and commenced operation in earlier epochs. For example, a constellation projected for launch and operation in Epoch 1 is expected to fulfill constraint level requirements that are meant for the constellation projected for launch and

operation in Epoch 2. This difference is important when comparing utility and expense trajectories for a particular set of constellation designs.

CONSTELLATION 1 – Developed in Epoch 1: Mission 0 in Context 0-0

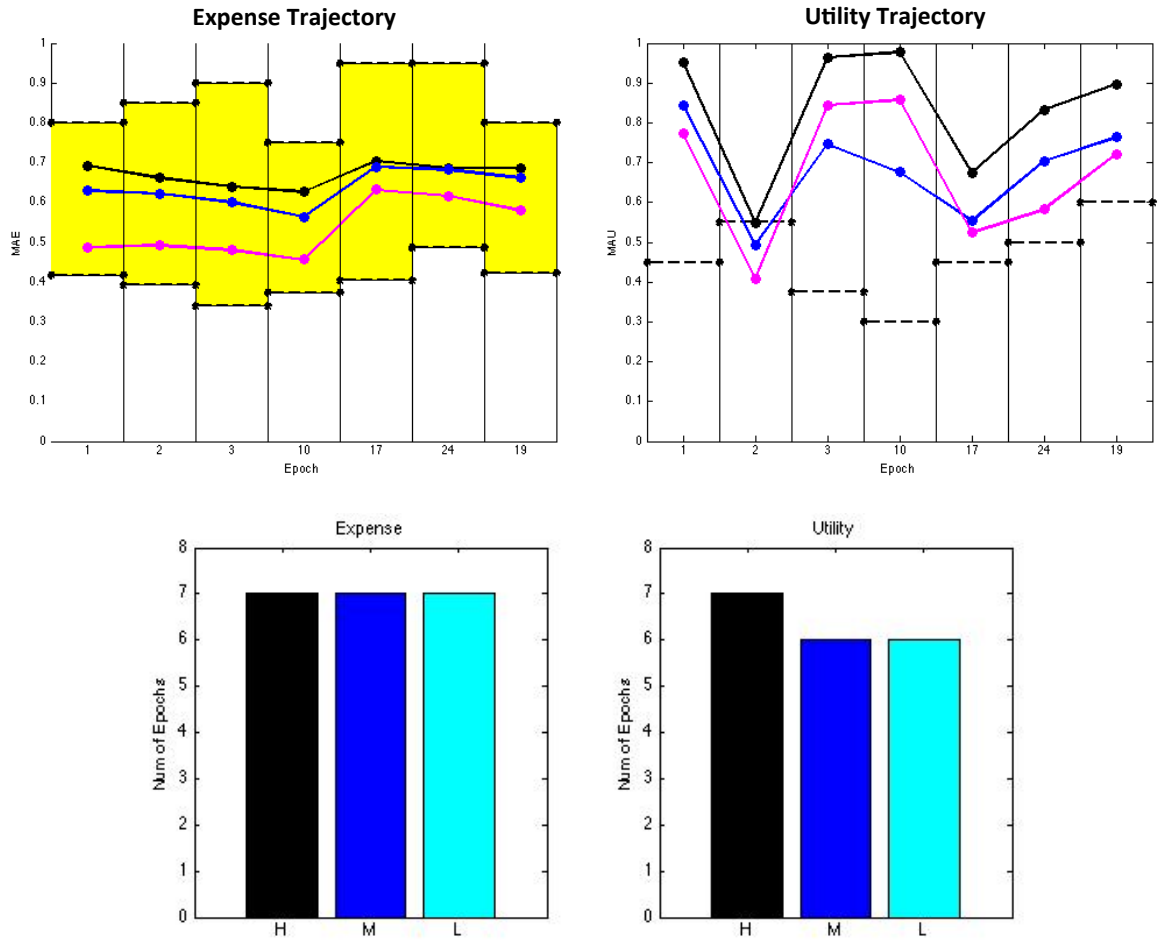


Figure 5-9: Expense trajectory plot, Utility trajectory plot, and Histogram of number of epochs in affordable solution region for each design for Constellation 1 – Epoch 1: Mission 0 in Context 0-0

The expense trajectories for the preferred designs from Epoch 1 show that all three program designs remain affordable across all seven epochs. From a resource-centric perspective, all three designs are possible options for portfolio development. However, the utility trajectories show that the medium and low risk designs will fall below the minimum utility constraint level for Epoch 2. The histograms show that the medium- and low-risk designs confirm that they are above the minimum utility level for only six out of seven epochs. All three designs experience a sharp drop in utility because the constraint level for science value in Epoch 2, which emphasizes the conduct of more space science, is a minimum of 100000 data packets of scientific data. All three designs can only contribute less than 50000 data packets to science value, hence accounting for the drop in utility. Only the high-risk design can generate sufficient science value to have a utility level just barely above the constraint level in Epoch 2. While all three solutions are largely feasible and affordable, the FSSD and DM1 are likely to pick the **high-risk design (Number 349) as the only fully affordable solution that will be applied to portfolio level analysis**.

CONSTELLATION 2 – Developed in Epoch 2: Mission 1 in Context 0-0

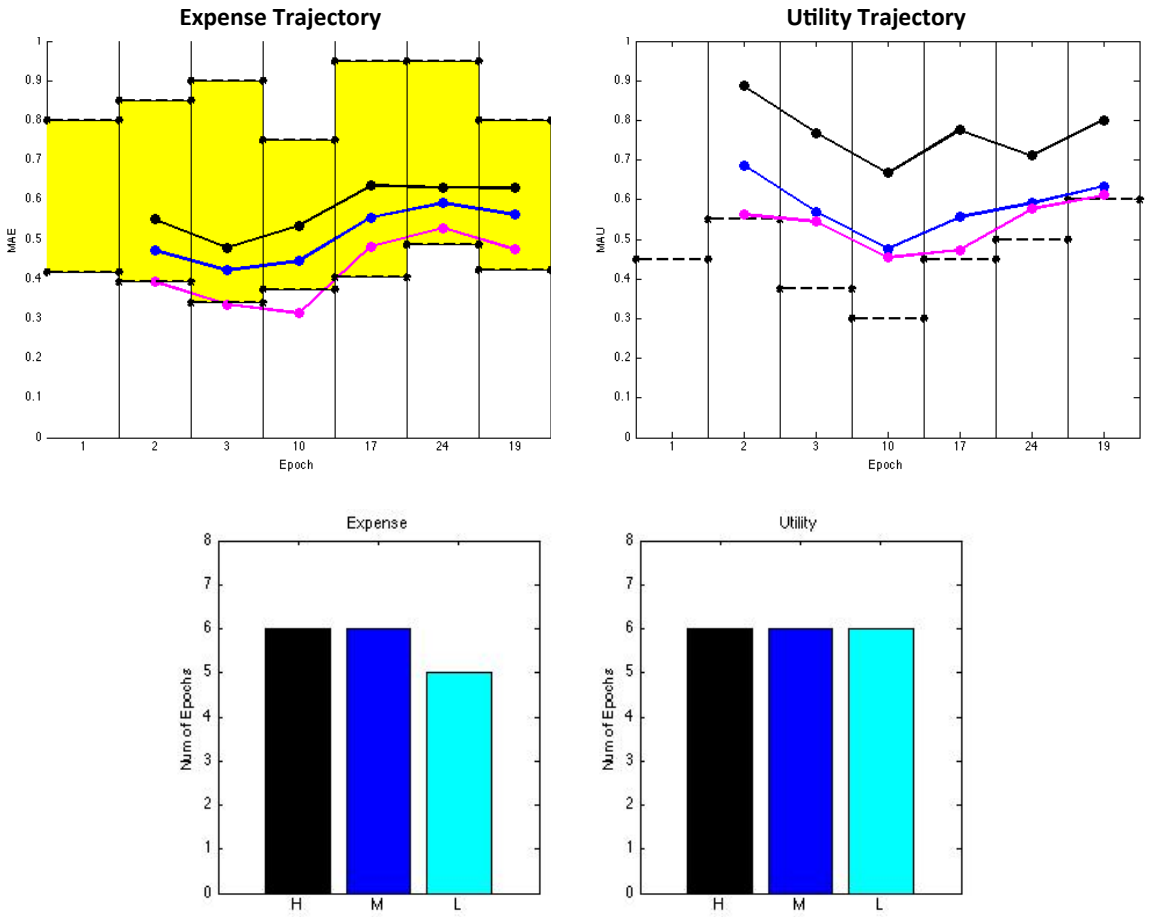


Figure 5-10: Expense trajectory plot, Utility trajectory plot, and Histogram of number of epochs in affordable solution region for each design for Constellation 2 – Epoch 2: Mission 1 in Context 0-0

The expense trajectories for the preferred designs from Epoch 2 show that all three program designs selected from the tradespace in this epoch remain affordable across all six epochs. The low-risk design, however, actually drops below the derived minimum expense level for Epoch 10, thus implying **potential cost savings** when operating the low-risk constellation in that epoch. The corresponding utility trajectories show that all three designs can perform above the minimum performance constraint levels across all six epochs. The utilities of all three designs generally decrease from Epoch 2 to Epoch 10 because of the increasing emphasis on annual free data capacity, but the three preferred constellations for Epoch 2 are designed to perform more science missions than simply to fulfill FSS objectives. Another point worth noting is that the low risk design is marginally above the constraint levels most of the time, but demonstrate a much higher level of utility relative to the utility constraint level in Epoch 10. This is because Epoch 10 only requires 20000 data packets of annual science value, but the low-risk design can potentially generate 93750 packets. However, its utility is at its lowest because it is unable to meet the minimum annual free data capacity constraint level of 250000 data packets in Epoch 10.

The low-risk design in Epoch 2 is a very special design because it actually becomes **unexpectedly affordable** as compared to a constellation projected for Epoch 10 that has to incur a minimum expense in order to meet a minimum utility constraint. It incurs an expense lower than the derived minimum expected expense level for that epoch, and yet still achieve a utility level above the minimum required utility. Therefore, this low-risk design can exhibit cost-savings and 'value-for-money' attributes in Epoch 10. As all three solutions are largely feasible and affordable, the FSSD and DM2 are likely to pick **all three designs as the fully affordable solutions that will be applied to portfolio level analysis**.

CONSTELLATION 3 – Developed in Epoch 3: Mission 2 in Context 0-0

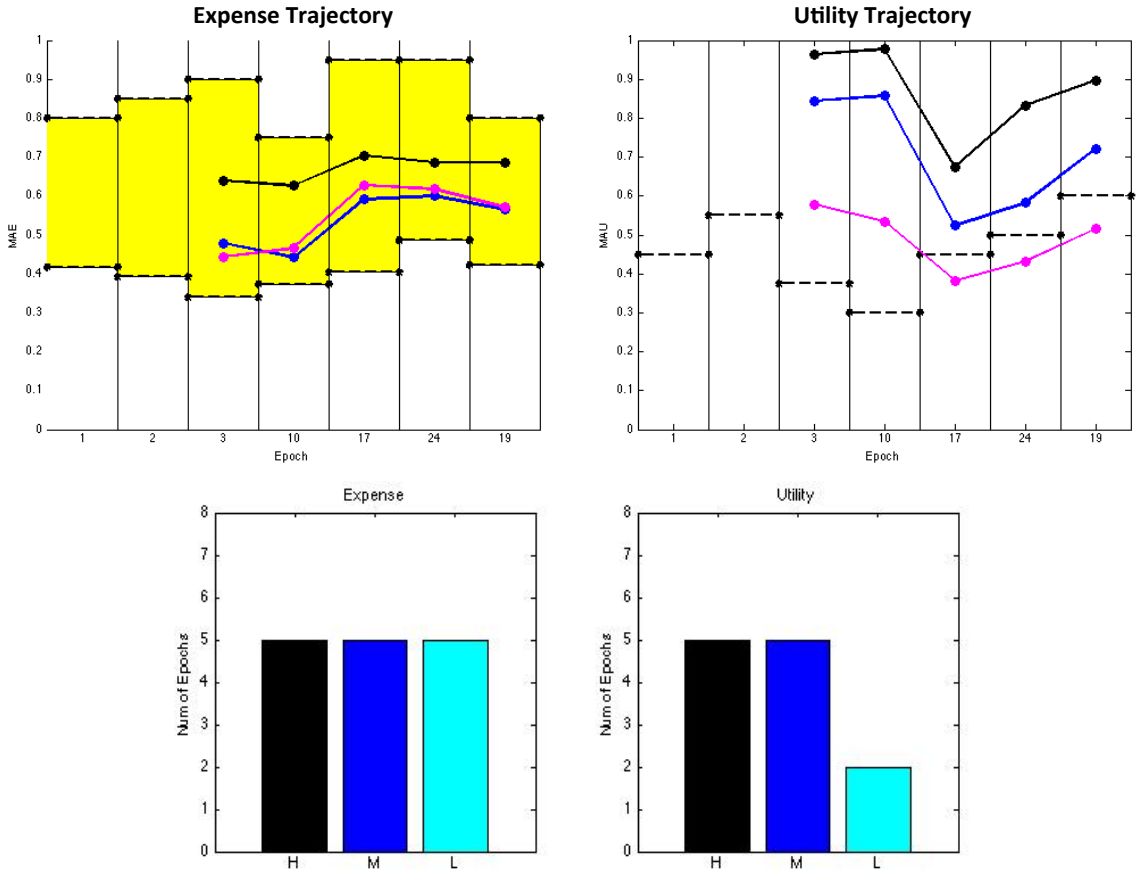


Figure 5-11: Expense trajectory plot, Utility trajectory plot, and Histogram of number of epochs in affordable solution region for each design for Constellation 3 – Epoch 3: Mission 2 in Context 0-0

The expense trajectories for the preferred designs from Epoch 3 show that all three program designs selected from the tradespace in this epoch remain affordable across all five epochs. The low-risk design also appears to incur higher expense than the medium-risk design from Epoch 10 onwards. The utility trajectories for the three designs, however, show that the low-risk design is the only one that is unable to meet the minimum utility constraint level from Epoch 17 onwards, despite incurring a higher degree of monetary and time investment than the medium-risk design. Despite having a high annual free data packet capacity of 330000, the low-risk design is unable to

meet the minimum utility requirement as it only has an overall maneuverability level of 1 and a low annual science value of 20000 packets. The medium and high-risk designs are technically superior with a higher annual free data packet capacity of 495000 and an annual science value of 30000 packets. This means that this constellation does not have a high degree of mobility and may not always have the capability to reconfigure quickly to meet dynamic market demands for FSS data assets. Therefore, the FSSD and DM3 are likely to pick **only the high-risk (Number 349) and medium-risk (1069) designs as the fully affordable solutions that will be applied to portfolio level analysis**.

CONSTELLATION 4 – Developed in Epoch 10: Mission 3 in Context 0-1

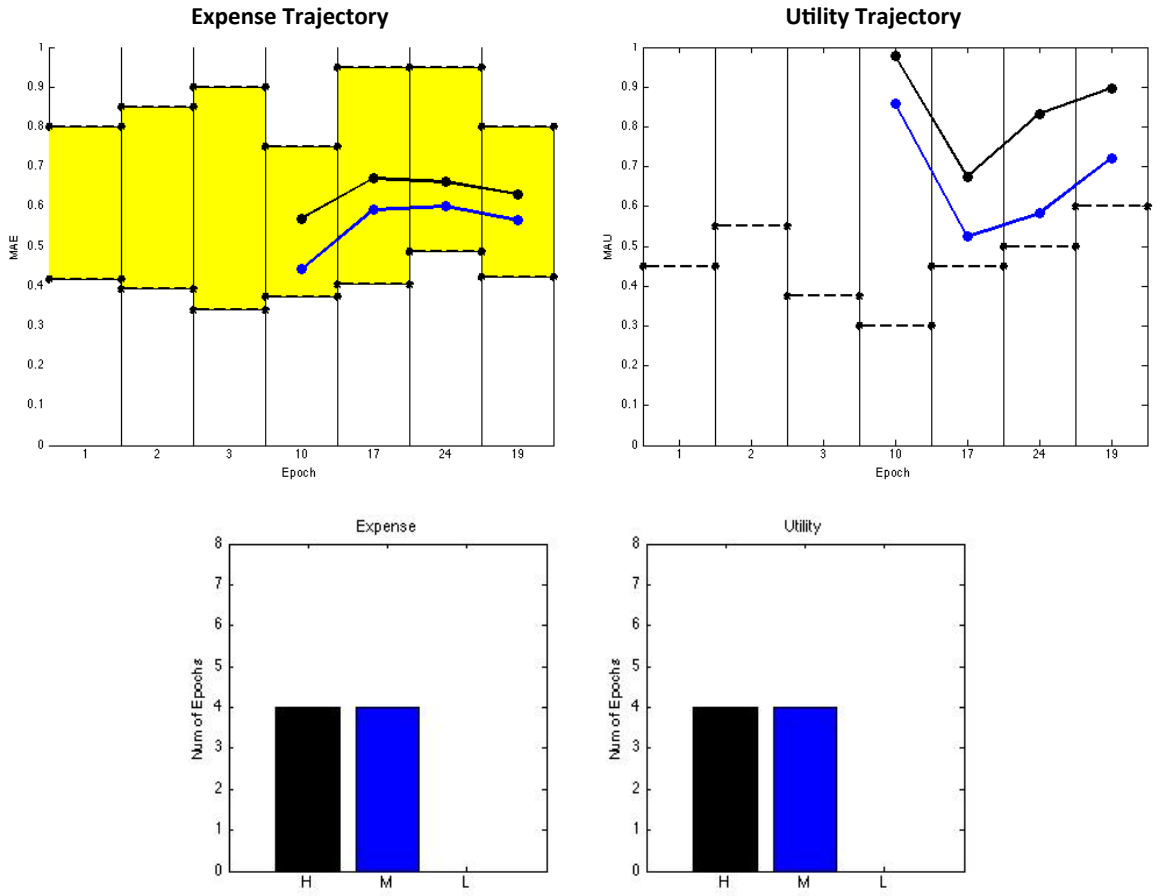


Figure 5-12: Expense trajectory plot, Utility trajectory plot, and Histogram of number of epochs in affordable solution region for each design for Constellation 4 – Epoch 10: Mission 3 in Context 0-1

The expense trajectories for the preferred designs from Epoch 10 show that the two program designs selected from the tradespace in this epoch remain affordable across all four epochs. The utility trajectories also show that both designs are always above the minimum utility level across all four epochs. The sharp drop in utility in Epoch 17 is due to the minimum annual science value being raised from 25000 to 50000 data packets from Epoch 10 to Epoch 17. Both designs are capable of only generating an annual science value of 30000 packets. As both designs are largely feasible and affordable, the FSSD and DM4 are likely to pick **both designs as the fully affordable solutions that will be applied to portfolio level analysis**.

CONSTELLATION 5 – Developed in Epoch 17: Mission 4 in Context 1-0

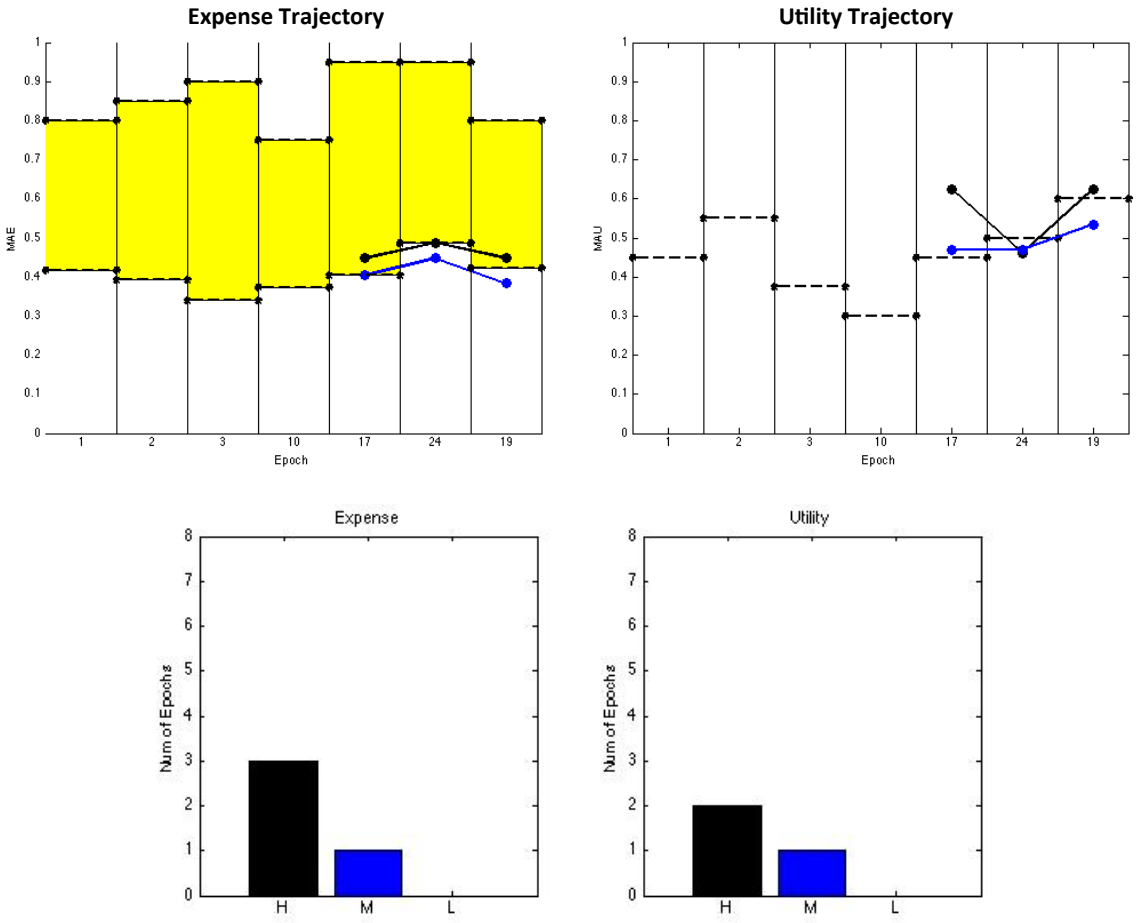


Figure 5-13: Expense trajectory plot, Utility trajectory plot, and Histogram of number of epochs in affordable solution region for each design for Constellation 5 – Epoch 17: Mission 4 in Context 1-0

The expense trajectories for the preferred designs from Epoch 17 show that both program designs selected from the tradespace in this epoch remain affordable across all three epochs. Both designs appear marginally close to the derived minimum expected expense constraint level and there might be doubts about their performance levels. This suspicion is confirmed by the utility trajectories, which are generally low as both designs have a maneuverability level of 1. The medium-risk design does not exhibit a sufficient increase in utility over time to be able to meet the minimum utility constraint levels for Epochs 24 and 19. The high-risk design is just above the minimum utility levels for Epochs 17 and 19, because it has a high annual free data packet capacity of 315000 and a high annual science value of 150000 packets.

However, it dips below the minimum utility level in Epoch 24, which is the reconfiguration mission that requires a high degree of maneuverability and high delta-V for its constituent satellites. As such, the high-risk design is partly but not fully affordable. However, the high-risk design is the closest one to utopia within the affordable solution region and there are no better-value designs in the tradespace (See Figure 4-23). Therefore, the FSSD and DM5 are likely to pick **only the high-risk design (Number 331) as the only partly affordable solutions that will**

be applied to portfolio level analysis. The tradespace and trajectories for this epoch are clear indications that there will be cases where there are no fully affordable solutions across the epochs considered. The solution(s) with the most number of epochs during which it is affordable will be preferred.

CONSTELLATION 6 – Developed in Epoch 24: Mission 5 in Context 1-1

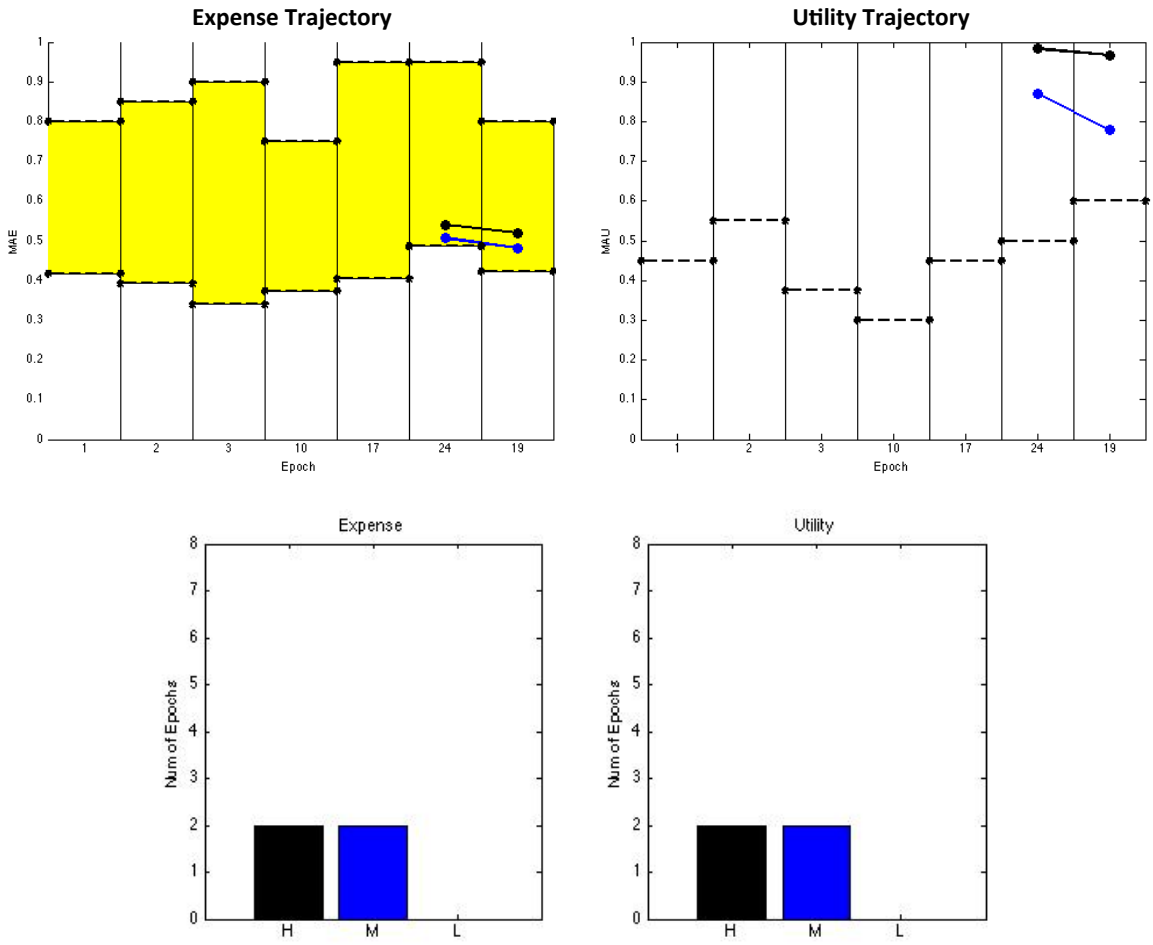


Figure 5-14: Expense trajectory plot, Utility trajectory plot, and Histogram of number of epochs in affordable solution region for each design for Constellation 6 – Epoch 24: Mission 5 in Context 1-1

The expense and utility trajectories for the preferred designs from Epoch 24 show that the two preferred designs selected from the tradespace in this epoch remain affordable across the last two epochs. Both designs have very high annual free data packet capacity of nearly 500000 and have the highest level of maneuverability given that their constituent satellites have high delta-V capabilities. The results for this epoch is straightforward, so the FSSD and DM6 are likely to pick **both designs as the fully affordable solutions that will be applied to portfolio level analysis**.

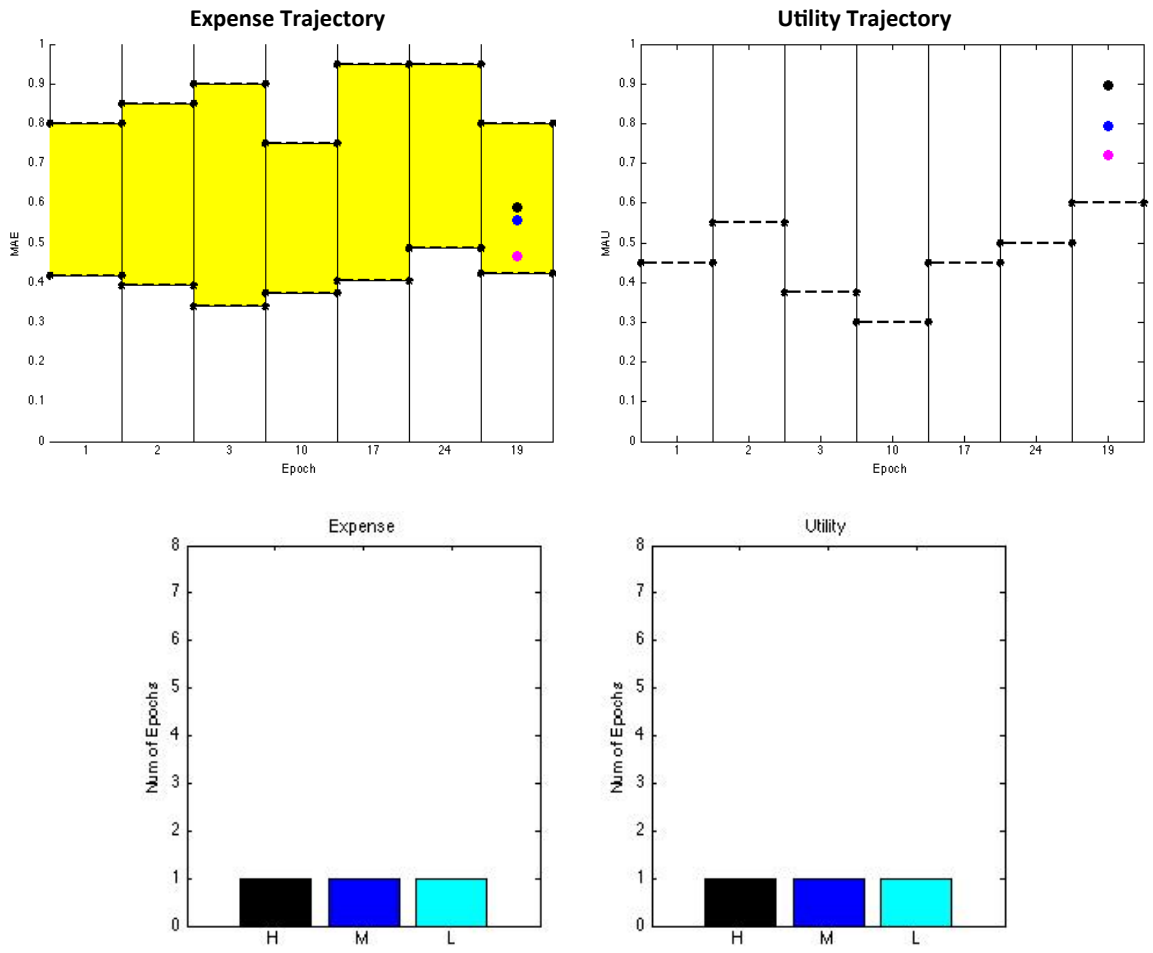


Figure 5-15: Expense trajectory plot, Utility trajectory plot, and Histogram of number of epochs in affordable solution region for each design for Constellation 7 – Epoch 19: Mission 0 in Context 1-1

No expense and utility trajectories are available for the preferred designs in Epoch 19, as they are chosen in the last epoch. This is a Single-Epoch Analysis. The three program designs selected from the tradespace in this epoch are affordable, as they are all above the minimum utility and derived minimum expected expense constraint levels. The results for this epoch is straightforward, so the FSSD and DM6 are likely to pick **all 3 designs as fully affordable solutions that will be applied to portfolio level analysis**.

5.6 Discussion of Program Level Analysis Results

Through the analysis of these trajectories, different programs with varying performance and expense attributes can be identified. As these program designs are selected on the basis that they remain above the constraint levels for most or if not all of the epochs, they are considered to be both cost-effective and time-effective within the definitions of affordability. They also have a satisfactory level of performance, which is perceived by stakeholders and decision makers to be sufficient for fulfilling different mission requirements across multiple epochs. Depending on the

needs and contexts of each epoch, stakeholder preferences and swing weights on program performance and expense attributes are changed accordingly to yield different program tradespaces and different affordable solution regions.

Two or three program designs along the Pareto front constrained by the affordable solution region were then selected. A series of multi-epoch and single-era analyses was then conducted for multiple sets of designs corresponding to different epochs to determine their degrees of affordability. If both the utility and expense trajectories of a design are always above their respective constraint levels, the design is considered fully affordable across those sequenced epochs. However, if either the utility and/or expense trajectory of a design breaches a constraint level and still perceived to be acceptable by stakeholders, then the design is considered partially affordable. Should such designs are deemed unacceptable or if more designs are required for a particular epoch, other points along the Pareto front could also be selected and they could be used in another round of program level analysis.

Among the preferred designs, the point with the highest utility and expense levels is typically regarded as the high-risk design, while a point with the lower utility and expense levels is regarded as a medium or low-risk design. A program design is inherently high-risk if its constituent satellite systems are also at high-risk of exceeding maximum expense constraint levels. A high-risk design will naturally have at least one or more performance attributes that are superior to lower-risk designs, but they are often more expensive in regard to at least one or more expense elements. In this program level analysis, the performance and cost attributes of most designs are of mid-range value, as these designs are found along the Pareto front within the affordable solution region, which has already removed the need to consider high-expense designs.

This effectively means that the mid-range factor levels of design variables define most affordable designs. This is seen in the system and program level tradespaces, as well as the design, performance and expense vectors for preferred designs in their corresponding epochs. As such, basic design principles for designing affordable systems and programs can be established. With respect to the case study, the FSSD and DMs can now know the choice of propulsion system, choice launch vehicle, the satellite payload capacity, communications interface, level of aggregation among satellites etc. to ensure that the individual satellites and the satellite constellations they are designing at the early-phase are more likely to be affordable in reality.

By considering all cost and time related expenditures upfront instead of simply lifecycle cost, tradespace exploration can be conducted more holistically through the inclusion more sociotechnical aspects and enable preferred solutions to be more inherently cost-effective and time-effective. However, designing the FSS requires more than just performing high-level designs of satellites or constellations. It requires an analysis of which combination of satellite constellations is most preferred to ensure that the entire FSS initiative remains affordable across all epochs. A summary of all the affordable programs is shown in Table 5-11.

This necessitates the conduct of **portfolio-level affordability analysis**. Combining all identified affordable solutions in each epoch will then yield possible design options for the portfolio level analysis of the FSS. Considering all preferred program solutions in each of the 7 epochs and combining them all, there are a total of $1\times3\times2\times2\times1\times2\times3$ = **72 designs for the FSS portfolio**.

Table 5-11: Summary table of the characteristics of program designs that have been identified as affordable

Phase (Epoch)	Program Design Number	Risk (Program Option)	Type	Program Vector
1 (1)	349	High (1)	Design	[1,12,4,2,0,4]
			Perf	[495000,30000,3]
			Expense	[30.76,58.97,251.56,6.03,12,10,4]
2 (2)	331	High (1)	Design	[1,12,3,2,0,4]
			Perf	[315000,150000,3]
			Expense	[39.40,64.45,122.13,6.03,12,7,4]
	181	Medium (2)	Design	[1,8,3,1,0,4]
			Perf	[126000,125000,3]
			Expense	[29.55,48.34,102.03,4.02,8,6,4]
	109	Low (3)	Design	[1,6,3,1,0,4]
			Perf	[94500,93750,3]
			Expense	[23.39,40.28,102.03,3.02,6,6,4]
3 (3)	349	High (1)	Design	[1,12,4,2,0,4]
			Perf	[495000,30000,3]
			Expense	[30.76,58.98,251.56,6.03,12,10,4]
	1069	Med (2)	Design	[3,12,4,2,0,4]
			Perf	[495000,30000,1]
			Expense	[30.76,44.44,162.84,6.03,12,7,4]
4 (10)	349	High (1)	Design	[1,12,4,2,0,4]
			Perf	[495000,30000,3]
			Expense	[30.76,46.43,284.14,6.03,12,11,4]
	1069	Med (2)	Design	[3,12,4,2,0,4]
			Perf	[495000,30000,1]
			Expense	[30.76,44.44,162.84,6.03,12,7,4]
5 (17)	331	High (1)	Design	[1,12,3,2,0,4]
			Perf	[315000,150000,1]
			Expense	[20.77,50.68,82.88,6.03,12,5,4]
6 (24)	349	High (1)	Design	[1,12,4,2,0,4]
			Perf	[495000,30000,4]
			Expense	[20.78,82.71,110.51,6.03,12,5,4]
	691	Med (2)	Design	[2,12,3,2,0,4]
			Perf	[526500,9000,4]
			Expense	[20.77,80.99,82.88,6.03,12,5,4]

Phase (Epoch)	Program Design Number	Risk (Program Option)	Type	Program Vector
7 (19)	349	High (1)	Design	[1,12,4,2,0,4]
			Perf	[495000,30000,3]
			Expense	[20.77,58.66,136.04,6.03,12,6,4]
	277	Medium (2)	Design	[1,10,4,2,0,4]
			Perf	[412500,25000,3]
			Expense	[18.39,48.88,136.04,5.03,10,6,4]
	1051	Low (3)	Design	[3,12,3,3,1,7]
			Perf	[396000,33750,1]
			Expense	[20.77,55.08,82.88,38.05,0,5,7]

The design vector lists the factor levels for PRG-DV-1 to PRG-DV-6:

```
[satellite_system_design, number_of_satellites,
number_of_satellites_in_concurrent_deelopment, constellation_type, legacy_operation,
program_contract_length]
```

The performance vector lists the factor levels for PRG-PA-1 to PRG-PA-3:

```
[Annual_Free_Data_Capacity, Annual_Science_Value, Maneuverability]
```

The expense vector lists the factor levels for PRG-EA-1 to PRG-EA-6:

```
[Total_Development_Cost ($ million), Total_Launch_Cost ($ million), Total_Labor_Cost ($
million), Total_Operations_Cost ($ million), Total_Retirement_Cost ($ million),
Waiting_time_to_launch (years), Program_Contract_Length (years)*]
```

6 FEDERATED SATELLITE SYSTEMS PORTFOLIO ANALYSIS

6.1 Portfolio Definition

The FSS is essentially a SoS where multiple satellite constellations collaborate to establish a dynamic "cloud" network where in-space data assets can be shared opportunistically with user spacecraft. The system level and program level analysis have already been conducted for individual satellites and satellite constellations. Therefore, it now remains to conduct portfolio level analysis for the FSS, which seeks to determine the most preferred portfolio designs in the FSS development lifecycle (era) spanned by seven epochs. This facilitates the identification of the most preferred portfolio of satellite constellations to be developed and launched in order to fulfill FSS objectives in a cost-effective and time-effective manner.

Previously in program level analysis, a program or constellation design contains multiple units of one of the three preferred satellite system designs on the system tradespace Pareto fronts that may be developed in different ways or work together to produce an emergent capability. In portfolio level analysis, a portfolio design is a set of heterogeneous satellite constellations developed and launched sequentially to fulfill various science missions, and ultimately fulfill FSS objectives. As the operation of the FSS is dependent on the aggregate characteristics of its constituent constellations, the design variables, performance attributes and expense attributes for the portfolios are straightforward.

The portfolio design variables are simply the choices for satellite constellation for each of the seven epochs, and the performance and expense attributes are simply the aggregate sum of the program level attributes of the constituent constellations. Recall that only one constellation can be developed at the start of every epoch. No new variables or attributes are introduced at this level of analysis for the case study. Tradespace exploration is conducted in a similar manner to determine preferred solutions among all possible portfolio designs. More detailed levels of cost analysis are also performed at this stage to determine the cost profile of individual constellations, and ultimately the entire portfolio of all constellations. In addition, changes in the number of FSS satellites in LEO are also tracked across a period of time in order to explain the changes in annual free data capacity and annual science value.

The key decision makers at all levels of the design process, be it system, program or portfolio are the FSSD and all the DMs.

The mission concept at the portfolio level is to develop the FSS portfolio, which is essentially multiple satellite constellations with FSS-enabling capabilities working together, with *stakeholder-desired* levels for overall annual free data capacity, overall annual science value, and overall maneuverability, as well as *stakeholder-desired* levels of expenditure for overall portfolio development cost, overall portfolio launch cost, overall portfolio labor cost, overall portfolio operations cost, overall portfolio retirement cost, and overall waiting time prior to final constellation launch.

The **performance, cost and schedule attributes** are thus derived from the portfolio-level mission statement. These attributes are listed and described in Table 6-1, where portfolio-level performance attributes are labeled POR-PA and portfolio-level resource expense attributes are labeled POR-EA. Attributes across performance, cost and time are considered for the portfolio-level analysis in this case study. **It is also assumed that all attributes are independent of each other such that the stakeholder preferences for one attribute have no impact on the preference for another.**

Table 6-1: Portfolio Level Performance and Expense Attributes

Portfolio Level Attributes	Description
POR-PA-1 **OVERALL ANNUAL FREE DATA CAPACITY**	The **overall annual free data capacity** is a proxy measure of the in-space data assets that a portfolio of satellite constellations can offer for opportunistic sharing in the FSS. In this case study, it is simply calculated as the sum of the annual free data capacities of all constellations in the portfolio. This sum is calculated on the assumption that all constellations are operating at the same time. This is because the FSSD and DMs currently do not know how it changes during the development lifecycle when constellations emerge or retire. This capacity may also decrease over time due to data storage over the years. The annual constellation free data capacity of the FSS will be measured using "the **number of free data packets** available".
POR-PA-2 **OVERALL ANNUAL SCIENCE VALUE**	The **overall annual science value** is a proxy measure of the portfolio annual contribution to space science. It is calculated as the sum of the annual science value of all constellations in the portfolio. The science value contributed by each constellation is assumed to remain constant so long as it is in operation. This sum is calculated on the assumption that all constellations are operating at the same time for the same reason stated for POR-PA-1. The unit of measurement for the overall annual science value of the FSS is the **number of data packets**.
POR-PA-3 **OVERALL MANEUVERABILITY**	The **overall maneuverability** is a proxy measure of the FSS' ability to reconfigure all its satellites' positions and orbits in order to achieve maximum effect for FSS operations. It is calculated as the sum of maneuverability levels of all constellations in the portfolio. A high overall maneuverability will imply that most constellations and their constituent satellites have the delta-V capability to reconfigure their positions in space to best meet the data asset demands of customer spacecraft.

Portfolio Level Attributes	Description
POR-EA-1 **OVERALL PORTFOLIO DEVELOPMENT COST**	The **overall portfolio development cost** is the total monetary cost required at portfolio inception to design, construct, assemble, integrate and test all constellations and their constituent satellites before they are determined to be ready for operations. It is calculated as the sum of the development cost of all constellations in the portfolio. It is measured in millions of dollars and the development cost for each constellation is paid in fixed annual installments over the duration of the program contract.
POR-EA-2 **OVERALL PORTFOLIO LAUNCH COST**	The **overall portfolio launch cost** is the total monetary cost required to launch all the constellations in the portfolio to LEO. It is calculated as the sum of the launch costs of all constellations. It is measured in millions of dollars and the launch cost for each constellation is paid in **fixed annual installments** over the duration of the program contract.
POR-EA-3 **OVERALL PORTFOLIO LABOR COST**	The **overall portfolio labor cost** is the total monetary cost required to pay all different batches of work forces that are responsible for developing their assigned satellite constellation up till the time of launch. It is calculated as the sum of the labor cost of all constellations in the portfolio. The overall labor cost is measured in millions of dollars and the annual constellation labor cost will increase over the years up till the time of launch due to the presence of interest rates.
POR-EA-4 **OVERALL PORTFOLIO OPERATIONS COST**	The **overall portfolio operations cost** is the total monetary cost required to operate all the constellations in the portfolio. It is calculated as the sum of operations cost of all constellations in the portfolio. The operations cost of a constellation can increase over time due to interest rates up till the time of constellation retirement. If a constellation is operated for legacy, annual operations cost for the constellation will continue to be incurred at the same interest rate, thereby increasing the overall portfolio operations cost. It is measured in millions of dollars.
POR-EA-5 **OVERALL PORTFOLIO RETIREMENT COST**	The **overall portfolio retirement cost** is the total monetary cost required to retire and de-orbit all the constellations at the end of their projected lifetimes. It is calculated as the sum of retirement costs of all constellations in the portfolio. If a constellation is operated only up till its projected lifetime, then the retirement

Portfolio Level Attributes	Description
	cost is incurred at the end of the lifetime and it is non-recurring. If a constellation is operated beyond its projected lifetime and into legacy, then its retirement cost will not be incurred at all, thereby decreasing overall portfolio retirement cost. It is measured in millions of dollars.
POR-EA-6 **OVERALL WAITING TIME TO FINAL CONSTELLATION LAUNCH**	The **waiting time to launch** is the number of years rounded up taken for all the constellations to be developed and launched. It is calculated as the sum of waiting times to launch for all constellations in the portfolio. It assumes that the development of a subsequent constellation commences only after the launch of the previous constellation. However, in a more realistic portfolio development timeline, development schedules for the constellations that are to be launched consecutively can overlap.

6.2 Determining Weights and Preferences for Portfolio Analysis

Since the portfolio design variables are the choices of satellite constellations in each epoch and the portfolio attributes are simply aggregate sums of program attributes, a design to value mapping for portfolio level analysis is redundant as the results would be the same as that of program level analysis. As such, this analysis proceeds straight to tradespace exploration of the FSS portfolio design space.

Equations were then formulated for each interaction to establish the computerized model. These portfolio tradespace model equations are not shown in this section and they can be found in the computer code attached in Appendix 3. The scientific principles used were based mainly on the methods for spacecraft subsystem design prescribed by Wertz and Larson (2011). Some equations were heuristically derived, as there were no precedent cases for reference. All the equations and their basis for usage are described qualitatively below.

However, like other levels of analysis, the weights and stakeholder preference levels for portfolio level analysis have to be determined first and they will be described in detail here. It is important to note that the author of this thesis created all weights and stakeholder preference levels to facilitate research purposes. This case study can be further improved through conducting interviews with actual FSS stakeholders.

Earlier in this case study, seven epochs were chosen and sequenced to simulate a potential development timeline for the FSS. As each epoch describes a fixed set of mission needs and operating contexts, weights and stakeholder preference levels for both system and program level attributes were established for each epoch to facilitate the value-centric design of single satellite systems as well as satellite constellations that would be most suitable and affordable in each epoch. However, the weights and preferences for all seven epochs were not described in more detail for the purposes of brevity, except that utility and expense generally increase for higher

levels of performance attributes and expense attributes respectively. However, the attribute levels that corresponding to minimum and maximum utility or expense varies from epoch to epoch. For example, in an epoch with emphasis on high science value, the highest utility of 1 can only be achieved when a constellation amasses over 1000000 data packets that are considered scientifically valuable, while in another epoch with emphasis on another attribute, a utility of 1 is easily reached at a lower data packet count.

MATE was then conducted for both system and program level analysis, where preferred satellite system designs and satellite constellation designs were then chosen based on their relative positions in the epoch's corresponding affordable solution region and their perceived levels of expense risk. Applying EEA concepts, their expense and utility trajectories were then plotted and compared to determine their extent of affordability. Fully affordable solutions are greatly preferred over partially affordable solutions. The latter were not chosen unless there were really no fully affordable solutions for a particular epoch. Upon the completion of program level analysis, the FSSD and the DMs now know the affordable constellation designs they should use for each epoch in order to develop the FSS sequentially over a period of time.

In portfolio level analysis, the aim is to determine the most affordable portfolio of satellite constellations to develop in order to ensure that the FSS is able to perform when required and remains both cost-effective and time-effective across its projected development lifecycle. As an affordable constellation is to be developed in each epoch, a portfolio will contain seven different satellite constellations developed in seven epochs and the design vector describing a portfolio will comprise the choice of satellite constellation for each epoch. Since the epochs are already sequenced to form an era, all seven epochs need not be considered separately again. The era hence simulates the development lifecycle in years collectively spanned by all seven epochs. Therefore, only one set of weights and one set of preference levels, instead of seven, are required for single era analysis in this case study. The weights for single era analysis describe the general degree of emphasis that the FSSD and DMs will place on performance and expense attributes across the portfolio development lifecycle, regardless of the epoch. Similarly, the preference levels for single era analysis describe the general level of satisfaction the FSSD and DMs will have when they achieve a particular performance level or incur a particular amount of expenditure across the portfolio development cycle, regardless of the epoch. These weights and preference levels are described in more detail for portfolio level analysis.

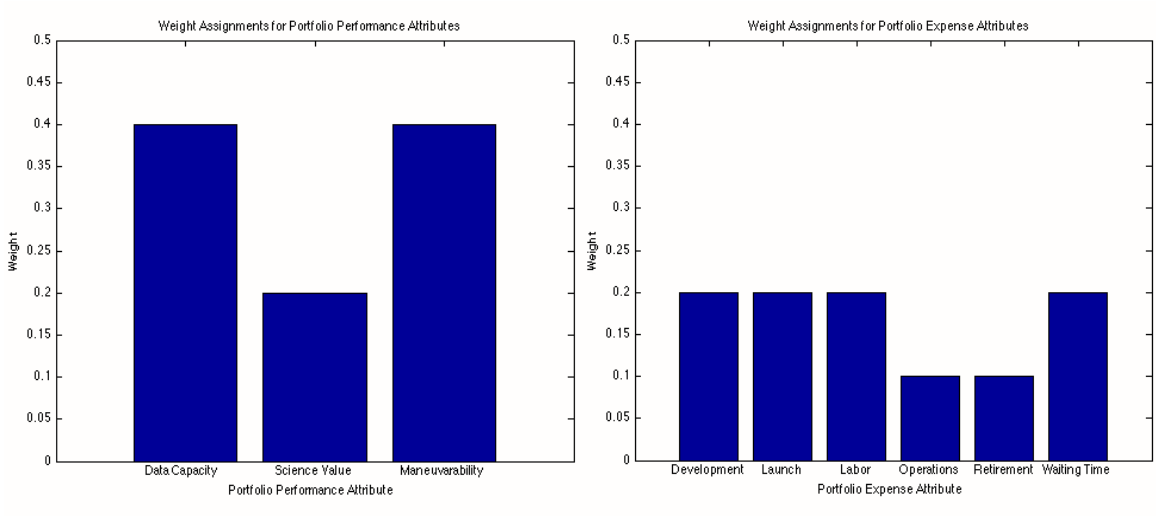


Figure 6-1: Weight assignments to portfolio performance and expense attributes

Figure 6-1 shows the weight assignments to the three portfolio performance attributes and six portfolio expense attributes. In this case study, it was assumed the weights would add up to 1. However, they do not always have to add up to 1 and it is dependent on stakeholder preferences. In this Single Era Analysis, it has been assumed that the FSSD and DMs have placed more emphasis on annual free data packet capacity and maneuverability than on science value since they are more important to FSS operations. As such, higher weights of 0.4 were assigned to annual free data capacity and maneuverability levels. Without sufficient data capacity throughout the whole era, the FSS will be unable to meet market demand for in-space data assets at certain times during the lifecycle spanned by the sequenced seven epochs. Similarly, without high levels of maneuverability, FSS satellites will not be able to respond quickly to changes in market demand and minimize the slant range required to establish efficient ISL.

If the FSS has been assumed to be small satellites at the outset, less emphasis may have been placed on science value delivered. However, since it has been assumed in this case study that large satellites would be the main participants in the FSS, they would have substantial payload capacity. This implies that they have the potential to carry multiple science instruments onboard. Assuming that these satellites are always in operations, larger satellites will inherently have higher science value. While science value is critical to certain constellations, it is not as important to FSS operations. Therefore, a lower weight of 0.2 is still placed on the science value delivered by the FSS to reflect the presence of different scientific interests that the FSSD and DMs may have in addition to fulfilling FSS objectives.

For the weights assigned to the portfolio expense attributes, greater emphasis is placed on the cost elements of development cost, launch cost and labor cost because they are observed to contribute a significant proportion of the total lifecycle cost. The waiting time to launch is the only time element considered in this study and it has been assigned the same weight. A longer wait time for each constellation may mean that the constellation is not operating long enough across the duration during which it was expected to be useful.

Hence, a shorter wait time will allow the development and launch of every constellation to be fairly synchronized, allowing most constellations to provide their highest levels of utility for

longer periods of time. Lesser emphasis is placed on operations cost and retirement cost since they are lower in value. However, operations cost is still important as it can increase over time and becomes significant if the FSSD decides to operate a particular constellation for legacy. Similarly, retirement cost is lower in value but it becomes significant when a constellation is meant to retire at the end of its lifecycle.

The SAU curves for the performance and expense attributes are fairly straightforward as shown in Figure 6-3. For the performance attributes, utility increases almost linearly with individual attribute levels. Utility values of 0 and 1 have been assigned to the lowest and highest existing attribute levels among the 72 portfolio designs. From Figure 6-3, it can be seen that a FSS portfolio design has the potential to provide between 2800000 to 3150000 free data packets per year for opportunistic sharing, 360000 to 450000 data packets worth of science value, and maneuverability levels of 13 to 20.

All 72 FSS designs have already been determined to be technically satisfactory, as they comprise satellite constellation designs that were found in the affordable solution region of each epoch. However, it still remains to determine which of the 72 portfolio designs have the most desired tradeoffs between performance and expense attributes.

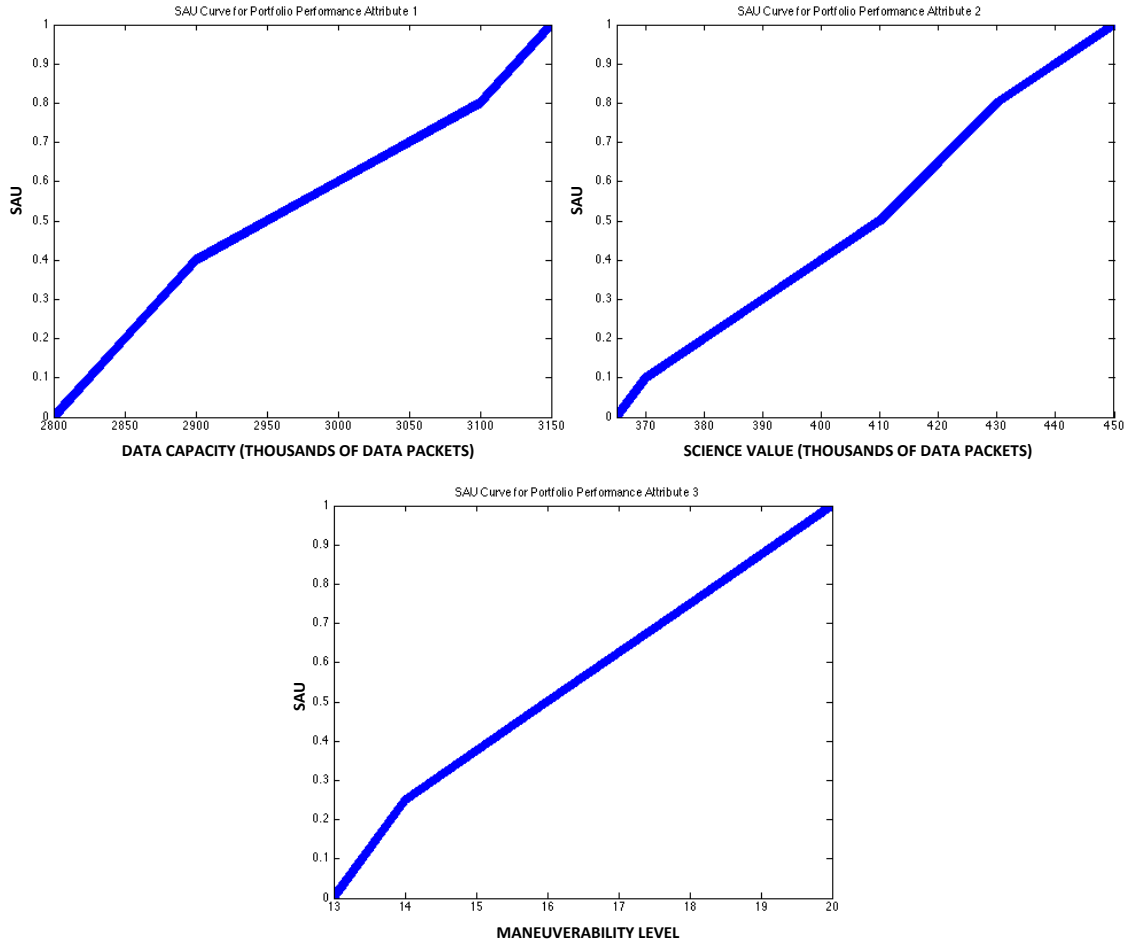


Figure 6-2: SAU curves for portfolio performance attributes in Single Era Analysis

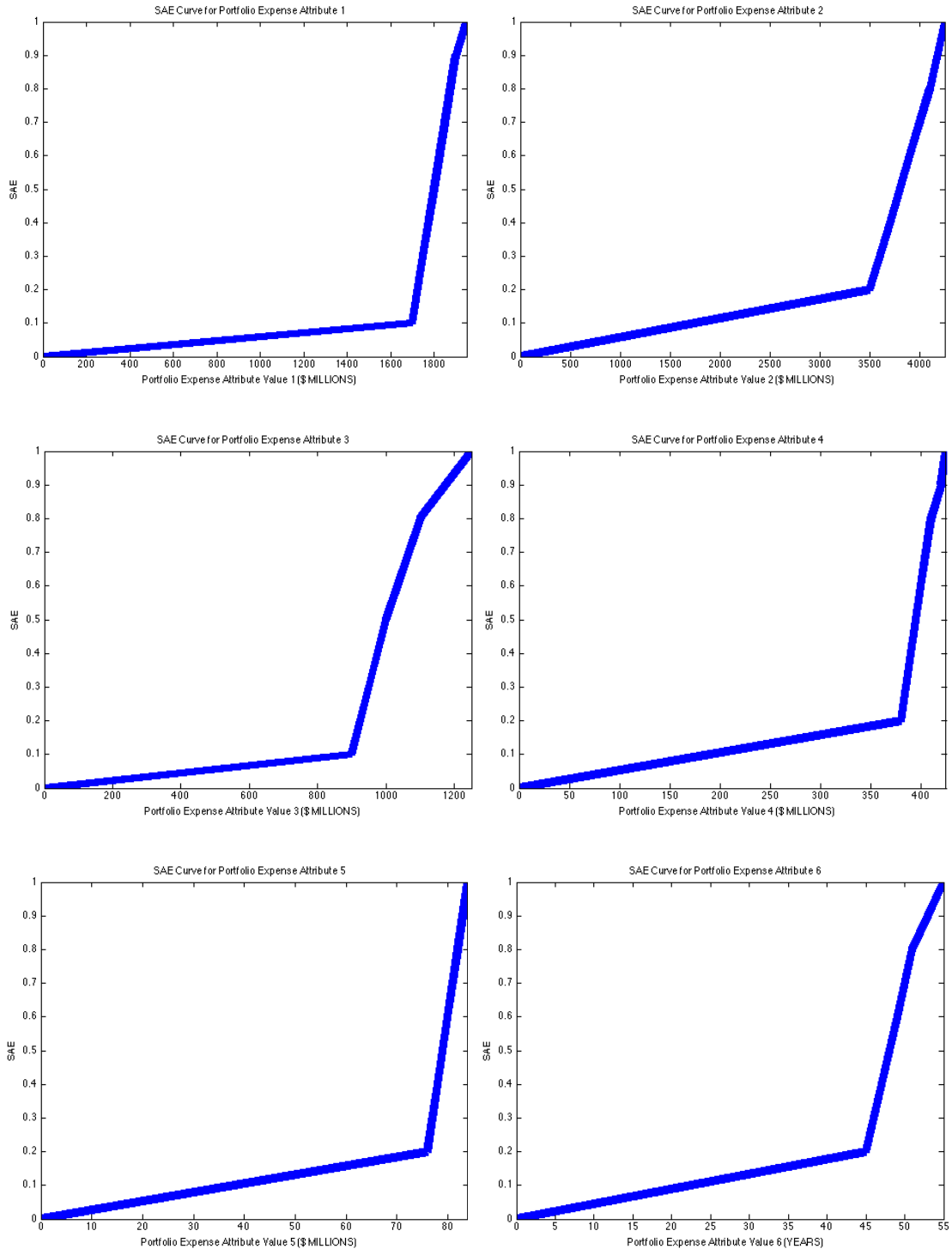


Figure 6-3: SAE curves for portfolio expense attributes in Single Era Analysis

The SAE curves for the expense attributes are shown in Figure 6-3 The lowest level of expense is always at zero monetary cost and zero waiting time because rational stakeholders only derive maximum satisfaction when they can get some return on performance for free and immediately. Hence, expense only increases to a low value of 0.1 or 0.2 at the lowest existing expense attribute

levels among the 72 portfolio designs, before it increases almost linearly to the maximum dissatisfaction level of 1 when the highest existing expense attribute levels are reached. The linear relationships among utility, expense and attribute levels again reflect fairly neutral and predictable views of the FSSD and DMs towards the era. If other eras (other sequences of epochs) are to be considered, then these curves may be skewed to the left or right depending on stakeholder preferences and mission needs across the era.

6.3 Portfolio Cost, Performance, and Satellite Profiles

The MAE and MAU values of the 72 portfolio designs were then plotted against each other to generate the portfolio tradespace. The Pareto frontier was identified and Pareto frontier set solutions are colored in red. 3 portfolio designs were then picked for further analysis: **Portfolio 43** (Mid-Utility/Mid-Expense, Colored in Green), **Portfolio 19** (High-Utility/High-Expense, Colored in Cyan, and **Portfolio 1** (Highest-Utility/Highest Expense). The design vector for a portfolio can be defined as the combination of program choices based on risk level:

`[Constellation 1 Choice, Constellation 2 Choice, Constellation 3 Choice, Constellation 4 Choice, Constellation 5 Choice, Constellation 6 Choice, Constellation 7 Choice]`

The design vectors for the aforementioned 3 designs are:

`[1, 2, 2, 2, 1, 1, 1]`; `[1, 1, 2, 2, 1, 1, 1]` ; `[1, 1, 1, 1, 1, 1, 1]`

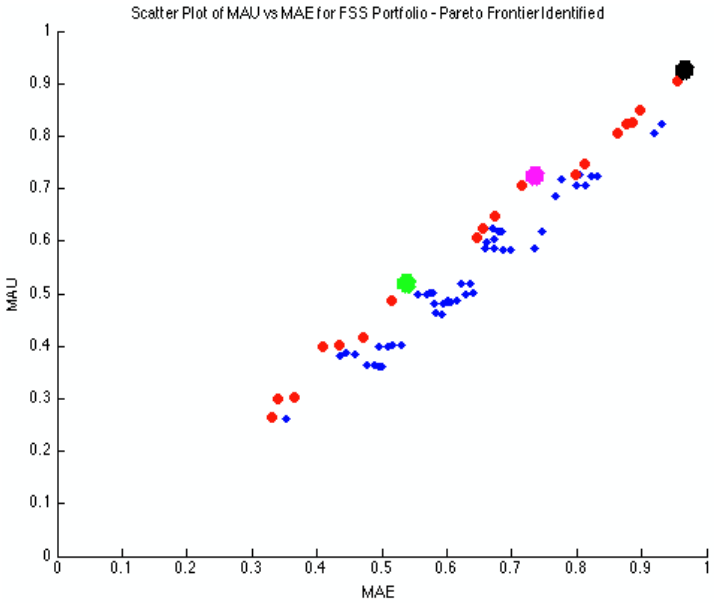


Figure 6-4: Portfolio tradespace with selected designs for further analysis
(Design 43 [1,2,2,2,1,1,1] – Green, Design 19 [1,1,2,2,1,1,1] – Cyan, Design 1 – Black [1,1,1,1,1,1,1])

The **cost and performance profile** of **Portfolio 43** was first analyzed. Figure 6-5 and 6-6 shows the cost and performance profiles of individual satellite constellations. Colored bar charts were used to describe the type and amount of cost commitments each year for every satellite constellation. The total cost per year was also traced. Performance attributes were also tracked and the variation of free data packet capacity and science value delivered by the constellation can be observed. These cost and performance profile plots can facilitate affordability analysis as it

illustrates the order of commitment for different cost elements across the constellation development lifecycle, thereby allowing decision makers to determine which cost commitments they should prioritize or attempt to reduce. They also tell the FSSD and DMs when exactly cost commitments should be made in order to remain affordable and when the constellation should be launched and operated.

Referring to the profile for Constellation 1, which is the high-risk constellation to be designed for Epoch 1 (Phase 1), it can be seen that the development cost and launch cost are paid in fixed installments over the first 4 years. An observation that could possibly worry the FSSD and DMs is that labor cost is high and it is being incurred at an interest rate until the time of launch. Constellation 1 is not launched until Year 11, as it can be seen that only at Year 11 do science value and free data packet capacity become available and labor cost has ceased to recur. It can also be seen that the data packet capacity decreases over time as more data is stored onboard the satellites over time, leaving lesser capacity for FSS purposes. It can also be observed that the annual science value is comparatively low. The FSSD and DMs may then expect the first constellation to deliver only adequate but not groundbreaking science value. It can be inferred that the waiting time to launch is 10 years, which may seem too long and expensive to keep such a program running. As such, the FSSD may wish to re-evaluate their choice of constellation design for Epoch 1, and choose a design that has a greater number of satellites in concurrent development or greater payload capacity in order to reduce its cost penalties. However, they should expect to pay significantly more for development cost and launch cost. Finally, it can be seen at Year 21 that the constellation has been retired due to the incurrence of retirement costs and the absence of data packet capacity and science value. Therefore, the development, cost commitment and performance profiles for each constellation can be summarized and inferred from these plots.

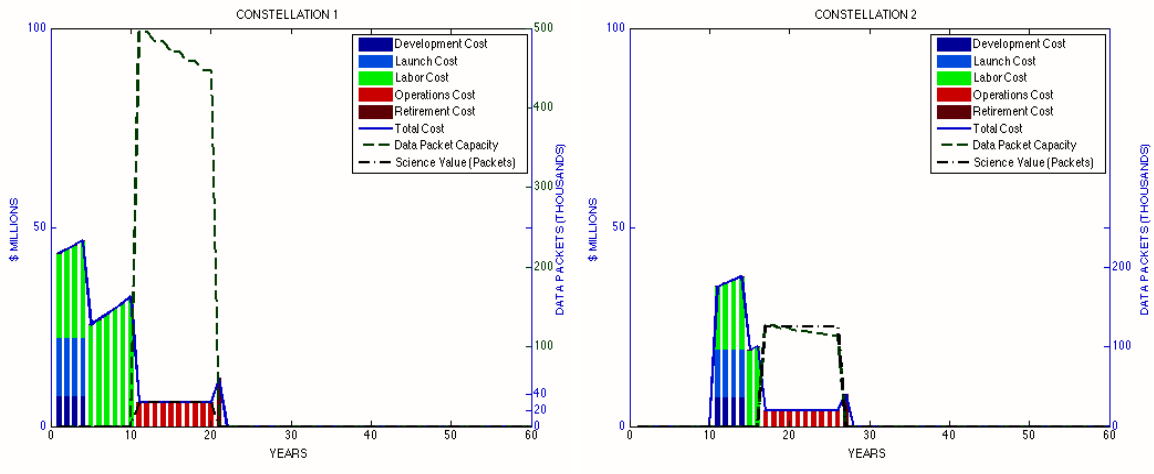


Figure 6-5: Cost and performance profiles of satellite constellations 1-2 of Portfolio 43

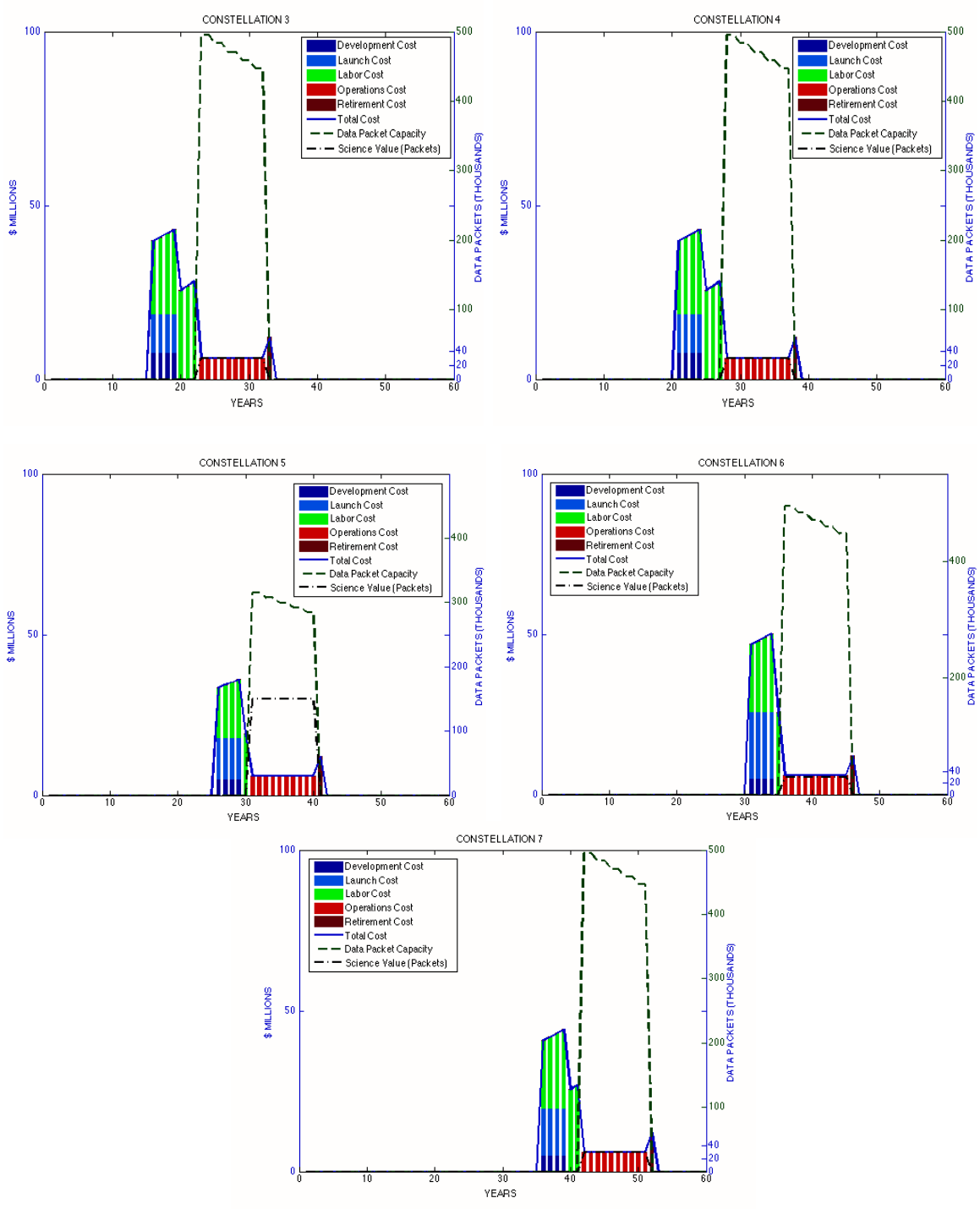


Figure 6-6: Cost and performance profiles of satellite constellations 3-7 of Portfolio 43

Similar interpretations can be made from the plots for the remaining constellations. The plots for Constellations 2,3,4 describe the medium-risk constellation designs for Epochs 2,3,10 (Phases 2,3,4) since the design vector for Portfolio 43 is [1,2,2,2,1,1,1]. It can be seen that Constellation 2 has significantly lower free data packet capacity during its years of operation, but it has a much

higher science value. This observation is expected as Constellation 2 was designed to operate in an epoch that emphasizes the delivery of science value. As more data packets are used to deliver more science value, there are fewer data packets available to fulfill FSS objectives. This constellation is also the cheapest among the 7, as it has the lowest values across all cost elements. Constellations 3 and 4 are rather similar since they can both initially provide 495000 free data packets and are designed specifically to provide FSS capabilities.

Constellation 5, which is a high-risk program design for Epoch 17 (Phase 5), has relatively lower free data packet capacity but much higher science value than ever before. This is because Constellation 5 is designed to operate and conduct Mission 4, which is the "Age of Discovery", and should be capable of delivering science value at unprecedented levels. Constellation 6, which is a high-risk program design for Epoch 24 (Phase 6), is observed to have the highest launch cost among all the constellations. This is due to Constellation 6 being designed to be able to reconfigure itself quickly to meet dynamic market demands for data assets. As such, its constituent satellites have high delta-V potential, thereby implying that they have more massive propulsion equipment and carry more onboard propellant. Finally, Constellation 7 is observed to have a cost and performance profile much like Constellation 1, albeit with lower development cost and launch cost due to high TRLs for laser communications technology and RLV.

A **satellite population profile** can also be plotted for the various constellations. These plots are shown in Figure 6-7 and 6-8, which show the variation in free data packet capacity and science value with the number of satellites in operation during a constellation's projected lifecycle. It can also be observed that all constellations have 12 satellites, except for Constellation 2, which only has 8. Therefore, the capabilities of the satellites in terms of their contribution to free data packet capacity and science value can be inferred. Constellations 1,3,4,7 have the most free annual data packet capacity and would contribute most greatly towards FSS operations. Constellations 5 and 6 have slightly lower capacity, but Constellation 5 provides more science value than any other constellation in the FSS development lifecycle.

Finally, an **overall cost and performance profile** can be established for the FSS by aggregating the plots of all constellations. From this point herein, all costs reported are reported in **then-year dollars**. In Figure 6-9, the annual expenditure for FSS development increases to approximately $82 million in Year 21 before it decreases sharply to approximately $45 million in Year 41. It is observed that labor cost ceased to recur in Year 41, meaning that the last constellation has been launched. This implies that the development and launch of all FSS constellations will take 41 years to complete, giving different levels of performance over the years while taking into consideration different cost and time constraints. The rising edges of the total cost line, coupled with the presence of development and launch costs commitments, represent the development and launch of the 7 constellations. The launches occur in the years when the development and launch costs installments cease to recur. The years in which a constellation is retired is represented by the maroon colored bars that represent the commitment of retirement costs. Assuming that an overall portfolio budget is provided for the FSS initiative over 40 years, it can be seen that the annual cost commitments for Portfolio 43 is always below the budget line. Therefore, it can be determined that Portfolio 43 always remains affordable across the entire era.

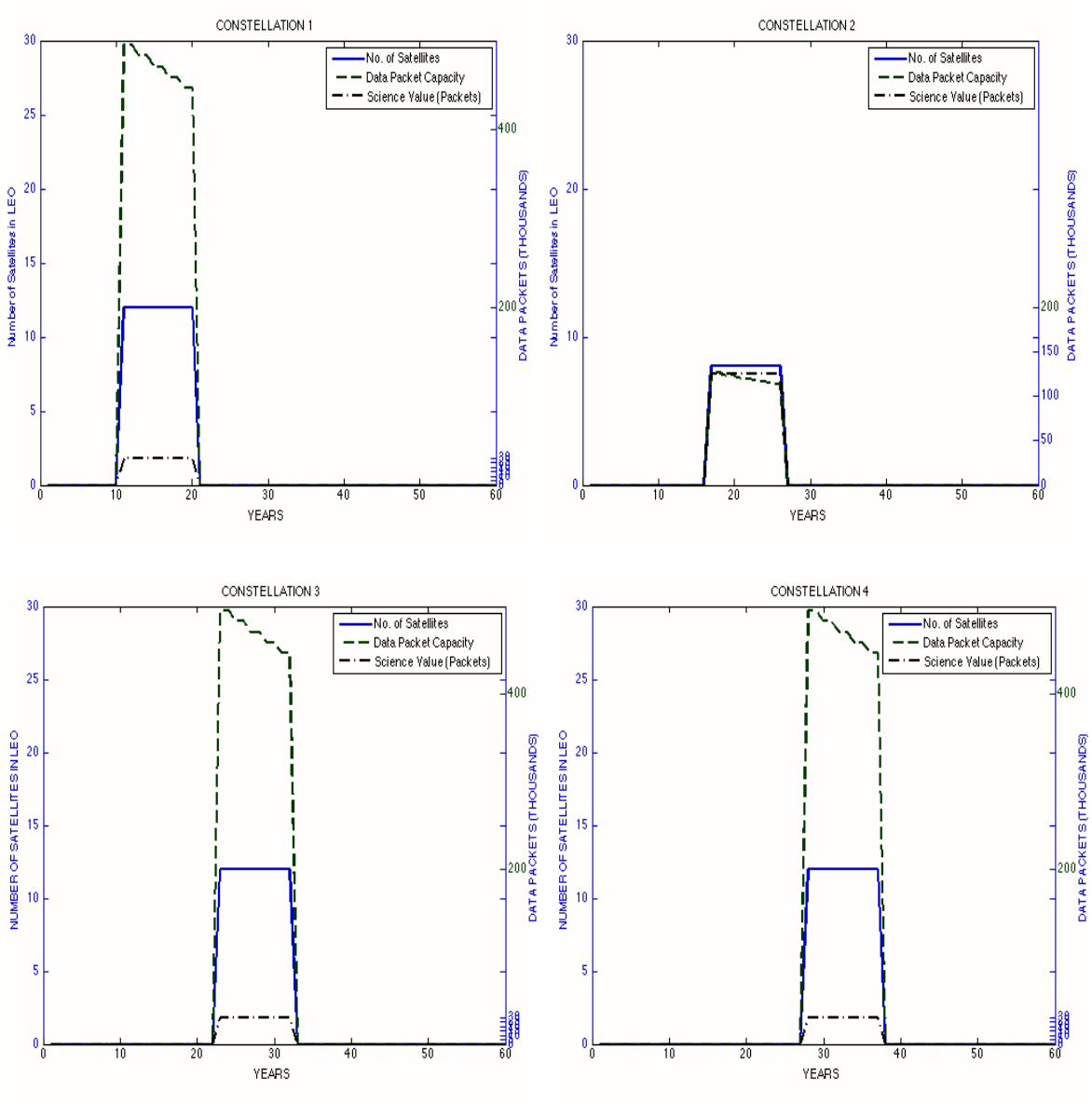


Figure 6-7: Satellite and performance profiles of satellite constellations 1-4 of Portfolio 43

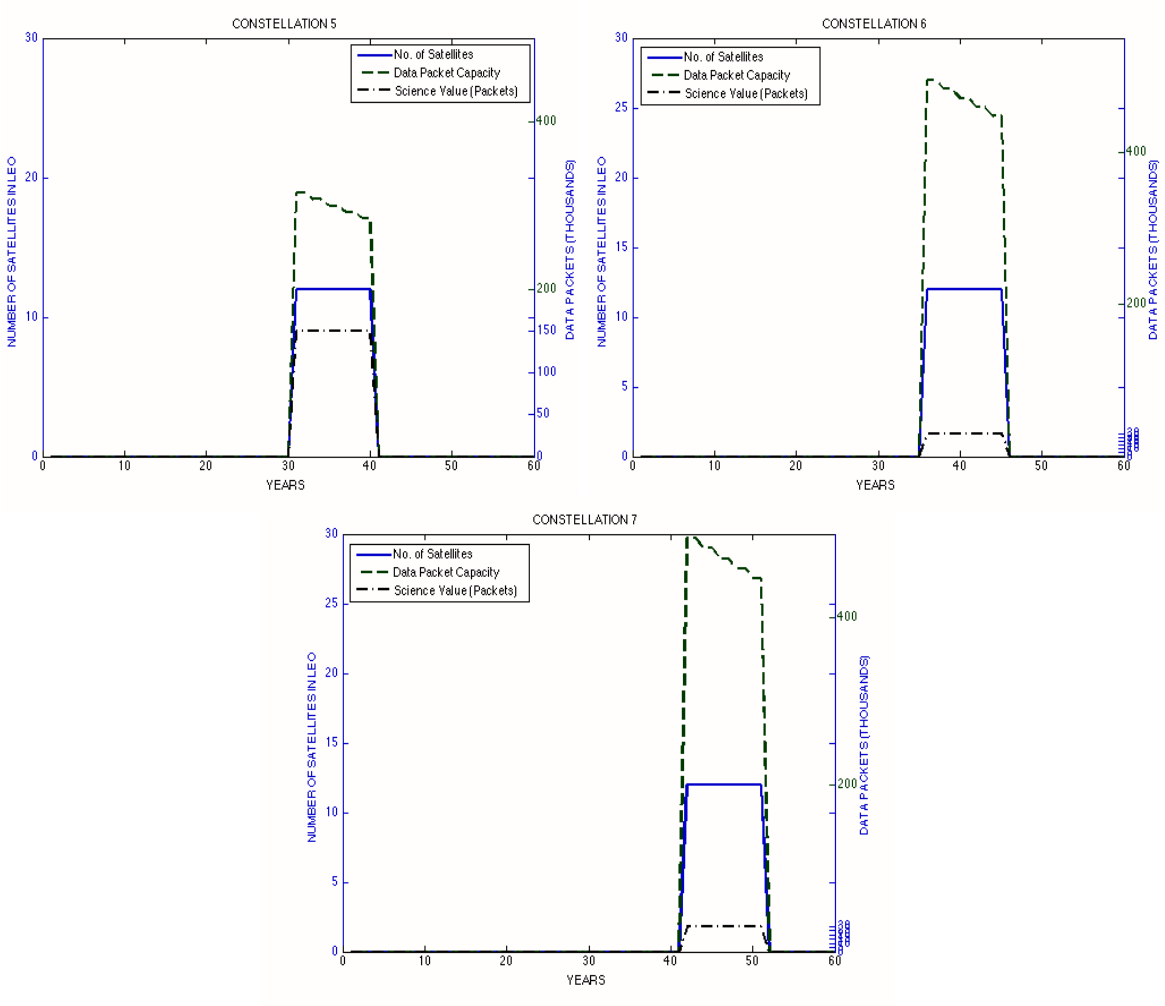


Figure 6-8: Satellite and performance profiles of satellite constellations 5-7 of Portfolio 43

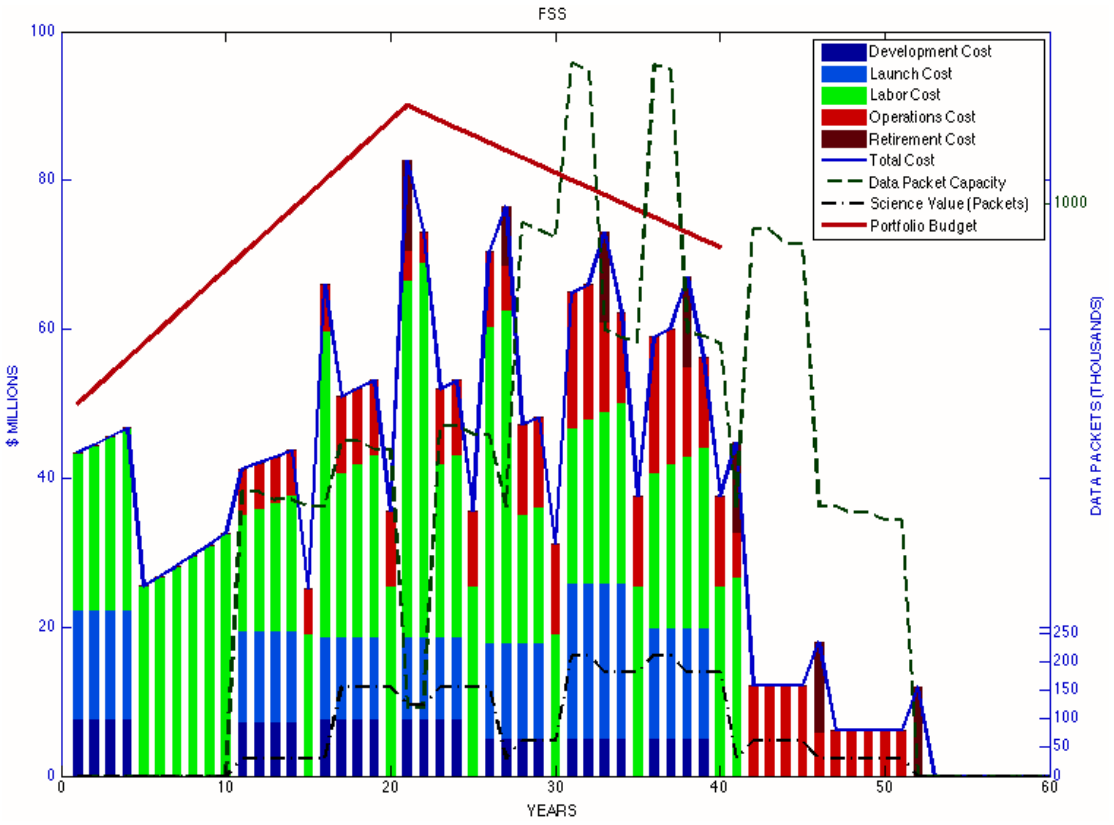


Figure 6-9: Cost and performance profile of FSS Portfolio 43

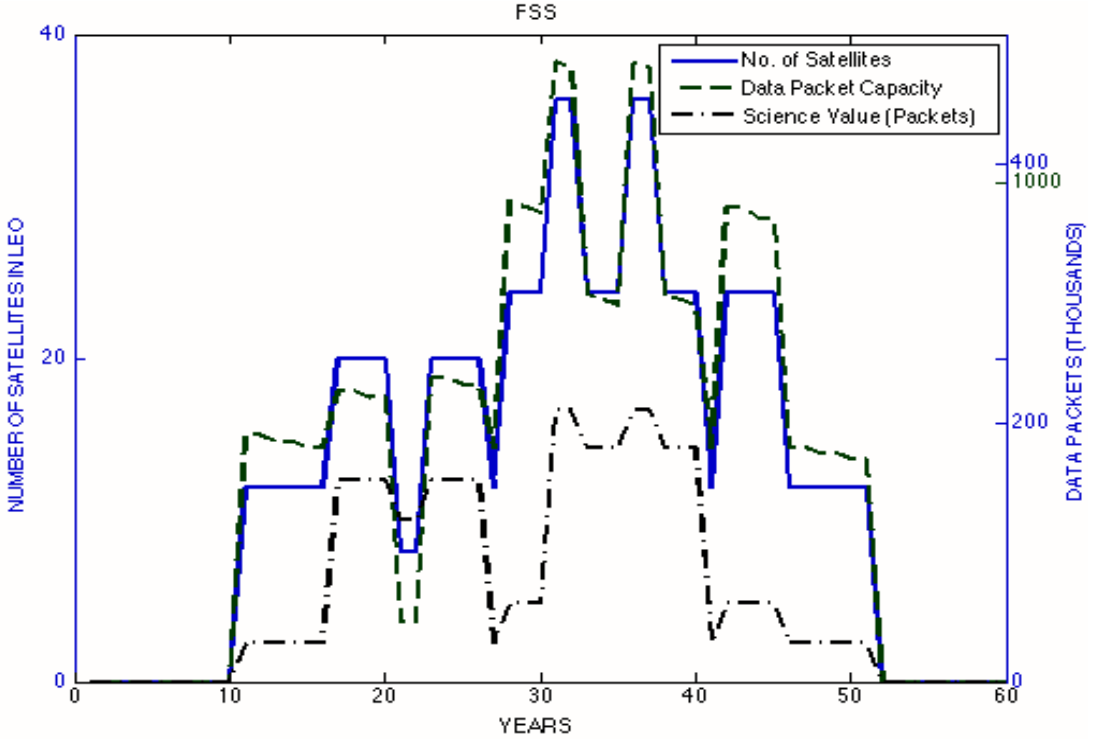


Figure 6-10: Satellite and performance profile of FSS Portfolio 43

In terms of performance, it can be observed that FSS operations can only commence in Year 11 with the launch of the first constellation. Science value for Portfolio 43 is also consistently high, fluctuating between 150000 and 250000 in Years 18 to 40. However, there is a deep trough seen for Years 27-30 owing to the retirement of Constellation 2 that contributed a significant amount of science value. The FSSD and DMS may hence be concerned that there is an opportunity cost incurred as a result of not being able to deliver a required amount of science value in those years.

The free data packet capacity can potentially peak at over 1200000 in Years 31 and 37, and averages over 500000 from Years 24 to 46, which are the peak years for FSS operations in this portfolio. It is also worth noting that operations cost are highest during those years as well due to the number of satellites present in LEO. Sustaining such a large number of spacecraft over time can hence incur high expenses.

However, a more major cause for concern in Portfolio 43 is the deep troughs in data capacity in Years 21-22 and Years 33-35. Data capacity drops drastically to slightly above 150000 and 750000 respectively in these two periods. The first trough is due to the retirement of the first constellation, which was providing about 495000 data packets per year. This sharp decrease also shows that Constellation 2 may not be able to provide sufficient data packet capacity to possibly sustain FSS operations. It can be seen in Years 17-20 that there is only a slight increase in data packet capacity when Constellation 2 joins Constellation 1 in operation.

With the retirement of Constellation 1 in Year 21, only Constellation 2 is available to sustain FSS operations. However, Constellation 2 is designed specifically to provide higher science value rather than to support FSS operations. Should there be sudden spikes in demand for data capacity in Years 21-22, Constellation 2 may not be able to sustain FSS operations efficiently. However, Years 21-22 is in Epoch 3, during which the FSS is still being initialized and there is a lower probability that the FSS may not be able to perform as expected.

However, the second trough requires more attention, as Years 33-35 are in the time period during which FSS operations are expected to be at its peak. While Years 33-35 are in Epoch 6 and there is a lot of ongoing reconfiguration of satellites, the FSSD and DMs should be concerned about the sharp loss in performance as a result of satellite reconfiguration during a time when sustained operations are also expected.

Should there be sudden spikes in demand for data capacity in Years 33-35, the FSS may not be able to operate as desired. There is a much higher probability of such an occurrence. Therefore, there is now a need to possibly address this major concern, either by **modifying one of the constellations** or **choosing an entirely different portfolio design**.

The first alternative is first explored. Instead of retiring Constellation 3, the FSSD and DMs decide to operate it for legacy and extend its period of operation up to 60 years and beyond. The differences in cost and profiles for Constellation 3 when not operated and operated for legacy is shown in Figure 6-11. In terms of cost, operations cost becomes recurring and increases over the years due to interest rates. The interest rates capture the increasing difficulty and economic viability of sustaining operations for a time far in the future using present technology.

Free data packet capacity in Constellation 3 also decreases over the years due to increased storage and equipment degradation over the years. However, the aim of operating a constellation for legacy is in the hope that it will be able to alleviate the second trough observed in the annual free data packet capacity of the FSS. Another advantage is that it provides a baseline data packet capacity and baseline science value even after the FSS initiative has surpassed Year 40. There is now a need to assess the relative increase in cost and benefits when Constellation 3 in Portfolio 43 has been chosen to operate for legacy.

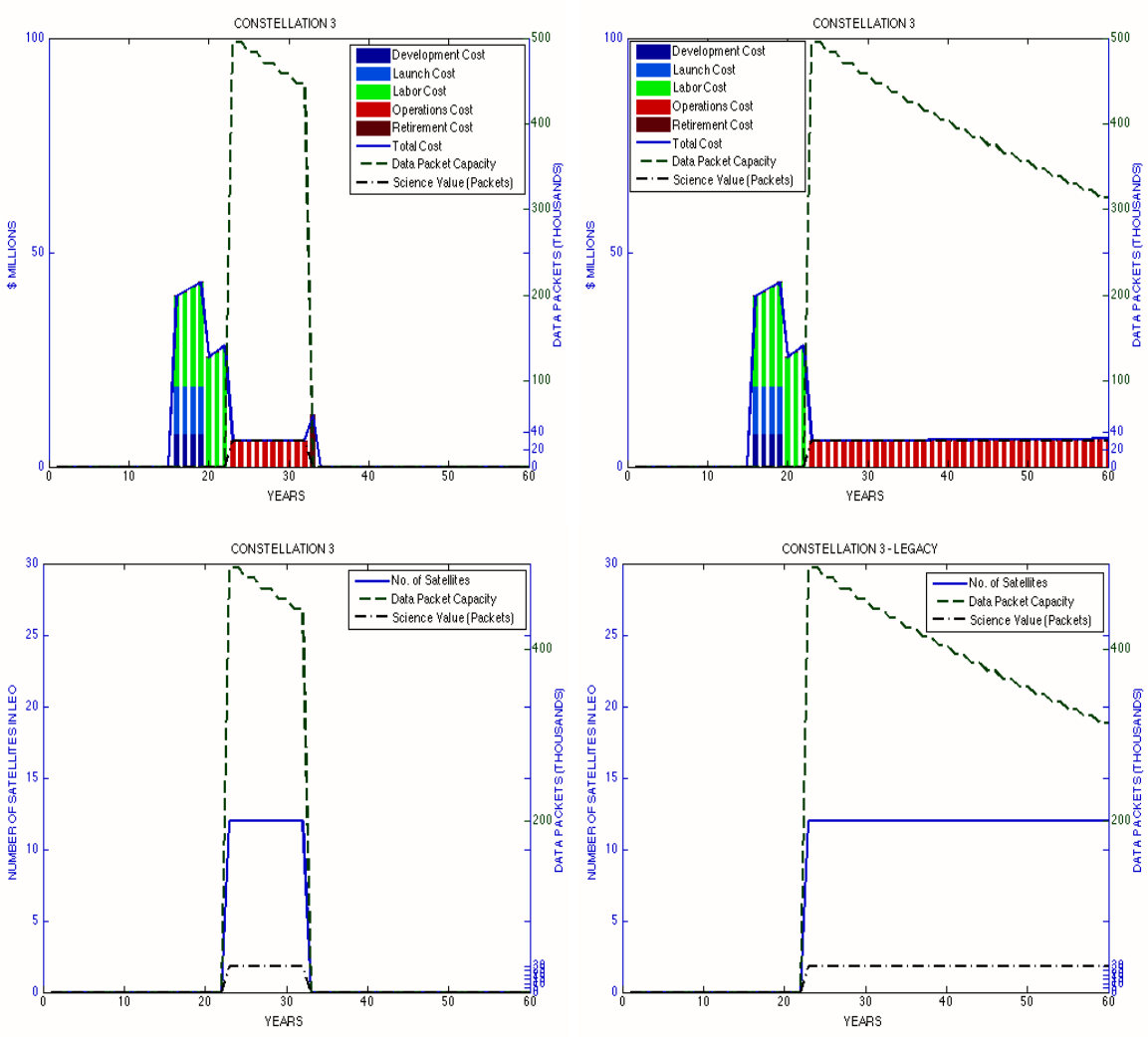


Figure 6-11: Difference in profiles for Constellation 3 when not operated and operated for legacy

Figure 6-12 shows the overall cost and performance profile of Design 43 when Constellation 3 is operated for legacy. The total cost and operations cost from Year 33 and beyond has increased as a result of legacy operation. Operations cost still remain very significant from Years 41 and beyond, which is just after the launch of the last constellation. However, it is observed that the second trough in data packet capacity has been remedied as a result of operating Constellation 3 for legacy. From Years 28-46, the modified version of Portfolio 43 is able to provide over 1000000 data packets to sustain FSS operations and it can even achieve a new peak of over 1500000 data packets in Year 36.

The deep trough in science value in Years 28-30 is also remedied and it now exhibits an increase in science value over those years. Portfolio 43 can even provide its highest level of over 200000 data packets worth of science vale from Years 35-36. From Figure 6-13, it can be observed that there are at least 36 satellites in orbit from Years 31-45 as a result of operating Constellation 3 for legacy. The satellite population even reaches a new peak of 48 in Years 35-36. With extremely high levels of free data packet and science value as well as a large number of satellites for celestial sphere coverage over Years 35-36, Portfolio 43 (modified) can be considered to be operating at its best. Therefore, operating Constellation 3 for legacy in Portfolio 43 appears to a technically superior and viable solution.

However, a caveat of operating for legacy is that there is a breach in the annual portfolio budget in Year 38, as a result of retiring one constellation and sustaining the operations of another. This may render the modified version of Portfolio 43 unaffordable. As a result, the FSSD and DMs may wish to retire that constellation at a later time, but at a risk of incurring even more operations cost in those years with decreasing budgets.

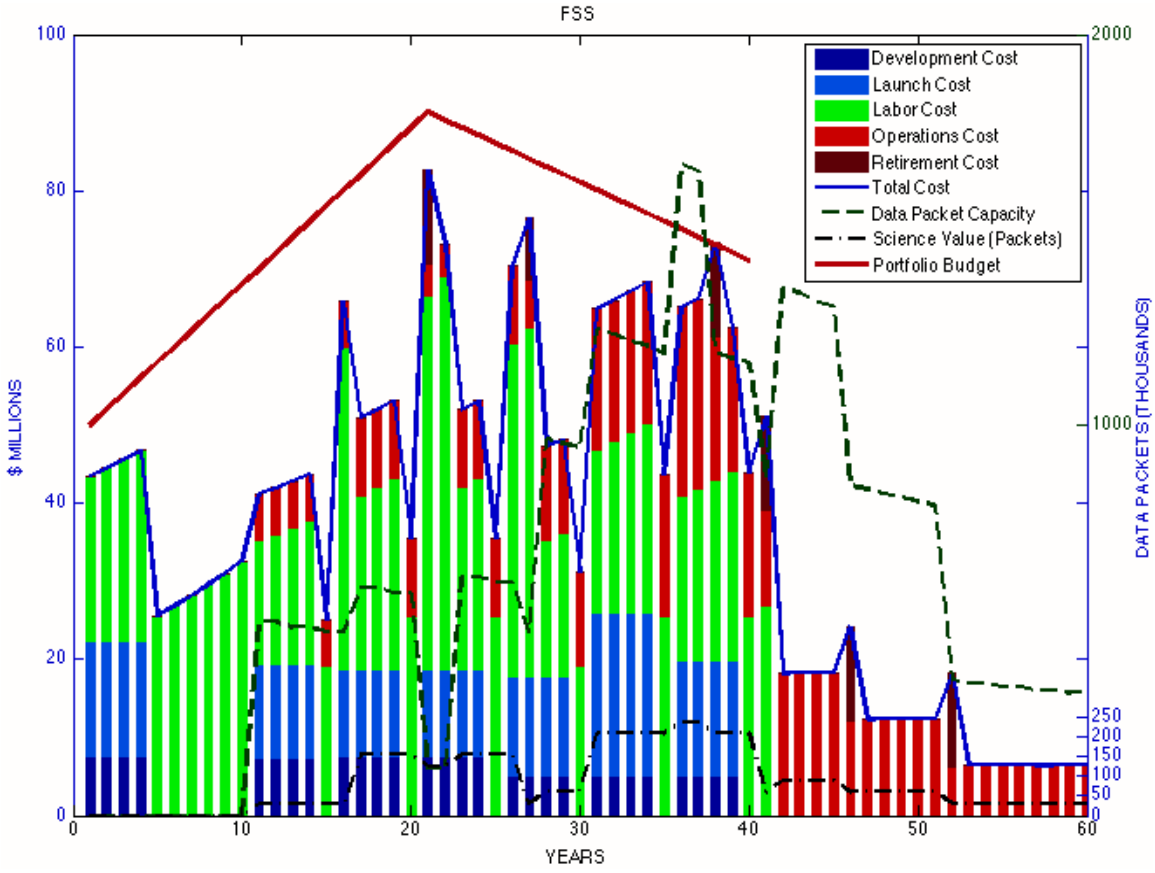


Figure 6-12: Cost and performance profile of Portfolio 43 when Constellation 3 is operated for legacy

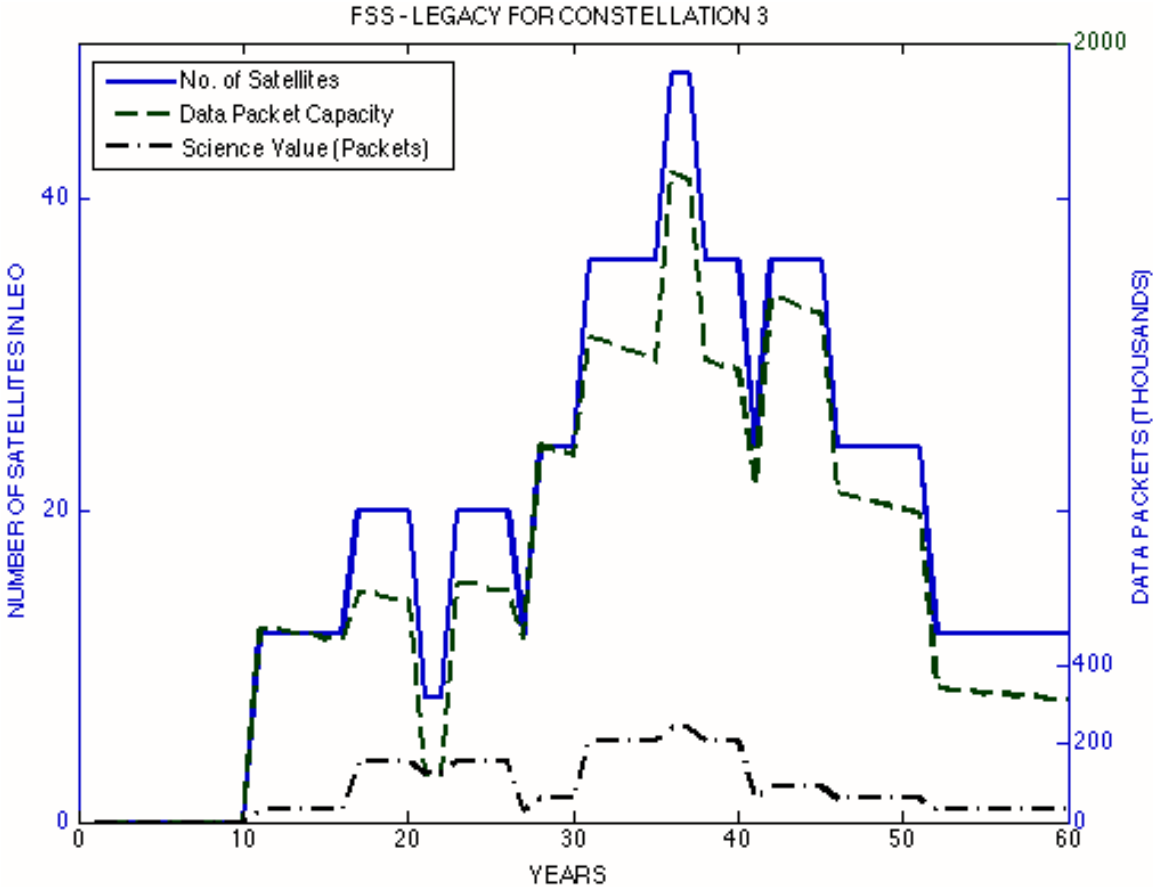


Figure 6-13: Satellite and performance profile of Portfolio 43 when Constellation 3 is operated for legacy

Modifying the projected retirement times of constellations may introduce more work and uncertainty in the analysis, which the FSSD and DMs may not be in favor of. As a result, the second alternative is to pick other portfolio designs. **Portfolio 1**, the highest utility and highest expense design is then chosen for further analysis, as the FSSD and DMs are interested to get the most technically superior yet marginally affordable design. Portfolio 1 comprises the affordable constellations with the highest risk, highest performance attributes, and highest expense attributes. The cost and performance profile, as well as the satellite and performance profile for Portfolio 1 are shown in Figure 6-14 and 6-15 respectively.

Since Portfolio 1 is the most expensive of the 72 possible designs, it is obvious that the total cost per year from Year 11 onwards is much higher than that of Portfolio 43. Years 1-10 are the same across all portfolios, as there is only one affordable constellation design for Epoch 1. The development cost, launch cost, labor cost and operations cost are also much higher than that of Portfolio 43. In terms of performance, Portfolio 1 averages high annual free data packet capacity throughout its lifecycle. Even at its trough in Years 21-25, it can still provide over 250000 data packets and sustain FSS operations better than Portfolio 43, which has a trough of only 150000 data packets in Years 21-22.

Although the trough experienced by Portfolio 1 is sustained for 3 more years, the tradeoff is that it is able to provide consistently higher data packet capacity later in Years 31-40, which is the expected peak period of FSS operations. It is able to provide over 1250000 data packets for nearly 10 years, a performance level that no other portfolio should be able to match. Unlike Portfolio 43, there are no troughs in data capacity during this critical period and the FSS can sustain operations efficiently even when there are sudden changes in market demand for data assets.

In terms of science value, Portfolio 1 is on average more capable than that of Portfolio 43. From Figure 6-14, Portfolio 1 can consistently provide high science value from Years 18-27 and from Years 31-40, during which it is able to deliver over 175000 and over 200000 scientifically valuable data packets. Its performance in these periods even betters that of Portfolio 43 with Constellation 3 operated for legacy. Across Years 31-46, it can be seen that Portfolio 1 has consistently high free data packet capacity and annual science value. These years are hence the "best years" of Portfolio 1, and no other portfolio would be able to sustain such a high level of performance for that period of time. As such, Portfolio 1 would appear to be the most desired solution in terms of performance and if the portfolio budget across its 40 years of development follows its profile.

At the program level, individual constellations that make up Portfolio 1 are affordable with respect to program utility and expense constraint levels, as well as stakeholder preferences for each corresponding constellation. However, at the portfolio level, the aggregation of affordable constellations may produce only a partially affordable portfolio. Assuming that the budget profile for Portfolio 43 is provided for Portfolio 1, there will be 2 cost breaches in Year 28 and Year 31. The second cost breach is of the greater concern, as it exceeds the budget line by over $10 million. 2 cost breaches in a span of 4 years may prove to be a cause of concern for the FSSD and DMs.

This is probably due to the shorter waiting times to launch for all constellations, where the tradeoff is increased cost for concurrent development and more cost elements overlapping every year. Therefore, despite being presented as a possibly affordable and technically superior solution, Portfolio 1 is inherently high-risk and has the most potential to incur cost breaches should there be even if the slightest fluctuations in portfolio budget profile. Unless the given portfolio budget is more abundant, Portfolio 1 may not be a better solution than Portfolio 43 when affordability considerations are of priority.

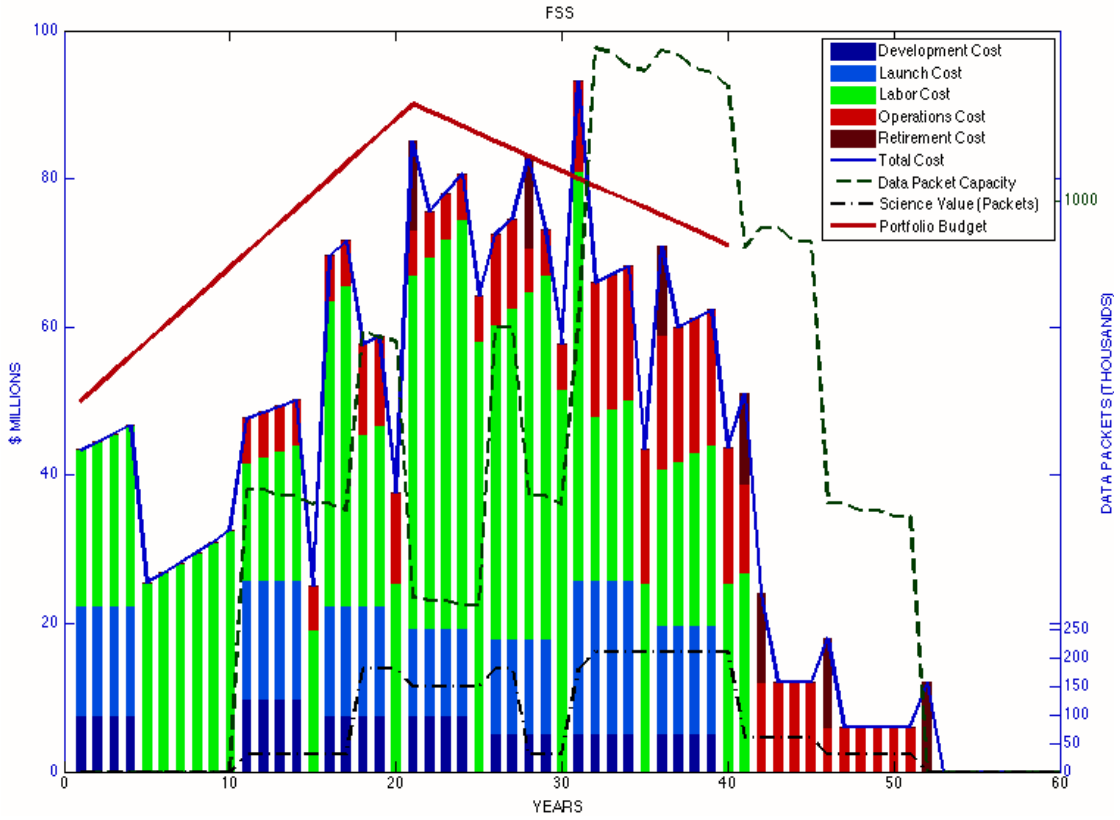


Figure 6-14: Cost and performance profile of Portfolio 1

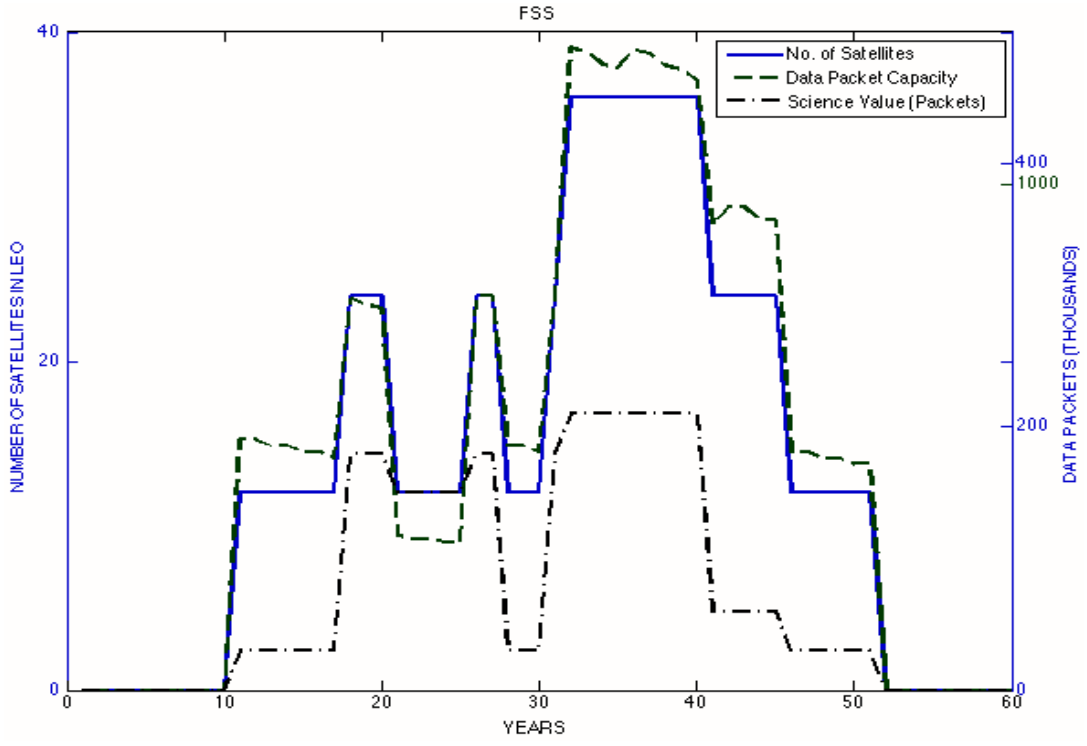


Figure 6-15: Satellite and performance profile of Portfolio 1

Another portfolio from the tradespace is analyzed and this would be **Portfolio 19**. Portfolio 19 has design vector [1,1,2,2,1,1,1], which is almost similar to Portfolio 43 except that Constellation 2 is now a higher-risk and higher-utility program design. Since it was earlier determined that Constellation 2 was a weak link responsible for the performance troughs exhibited by Portfolio 43, replacing it with a better constellation might prove to be a better solution. Therefore, Portfolio 19 is an enhancement of Portfolio 43. The cost and performance profile, as well as the satellite and performance profile for Portfolio 19 are shown in Figure 6-16 and 6-17 respectively.

From Figure 6-16, the total cost per year in Years 11-15 incurred by Portfolio 19 are much higher than that of Portfolio 43, as a result of developing a better and more expensive program design for Constellation 2. The sharp rising edge in years 17-18 show that an enhanced Constellation 2 is now able to provide more free annual data packet capacity. More importantly, the trough experienced by Portfolio 43 is remedied. Although Figure 6-16 also shows a deep trough in Years 21-22, the data packet capacity is still above 250000 data packets despite being at the lowest. Portfolio 43 could only manage to provide 150000 data packets and is at greater risk of not being able to sustain FSS operations.

Also, Constellation 2 now has 12 constituent satellites instead of 8. As a result, there are almost always 24 satellites in operation during Years 17-40, giving consistently high data packet capacity and science value per year. Furthermore, as a result of an enhanced Constellation 2, Years 23-27 now has a significantly higher data packet capacity of approximately 750000. Previously in Portfolio 43, only a slight increase was observed after the launch and operation of the cheaper option for Constellation 2. Years 23-27 is also an unusually long period for a deep trough in data packet capacity for Portfolio 1. In Portfolio 1, despite the launch of Constellation 3, it was only able to recover the data packet capacity it had in Years 18-20 for only a short period of time. While Portfolio 1 also experiences yet another trough at approximately 450000 data packets in Years 28-30, Portfolio 19 actually provides a stepped increase to 1000000 data packets in the same period.

As compared to Portfolios 43 and 1, it can be seen that Portfolio 19 offers a more progressive increase in performance levels over the portfolio development lifecycle. The FSSD and DMs may be more comfortable with such a portfolio if they are more conservative in their engineering design and prefer development phases for the FSS to be conducted in progressive steps. Portfolio 1 may provide attractively high performance from Years 31-40, but the sudden jump in performance prior to the peak period may be too risky. This jump in performance would have to be provided by Constellation 5, and this may place too much pressure and risk on DM5. Should Constellation 5 experience delays in development or launch, the second trough at 450000 data packets may be extended for a longer period of time. However, in Portfolio 19, there is less pressure and risk on DM5. Even if Constellation 5 experiences delays, the FSS would still be able to provide approximately 750000 data packets for a few more years until Constellation 5 recovers.

Compared to Portfolio 1, Portfolio 19 is less risky and offers more progressive performance levels and compared to Portfolio 43, Portfolio 19 offers better performance during the earlier years with rising budgets and will still be able to sustain FSS operations even at its lowest levels. **Therefore, the FSSD and DMs would pick Portfolio 19 as the most preferred solution for the FSS portfolio design in this case study.**

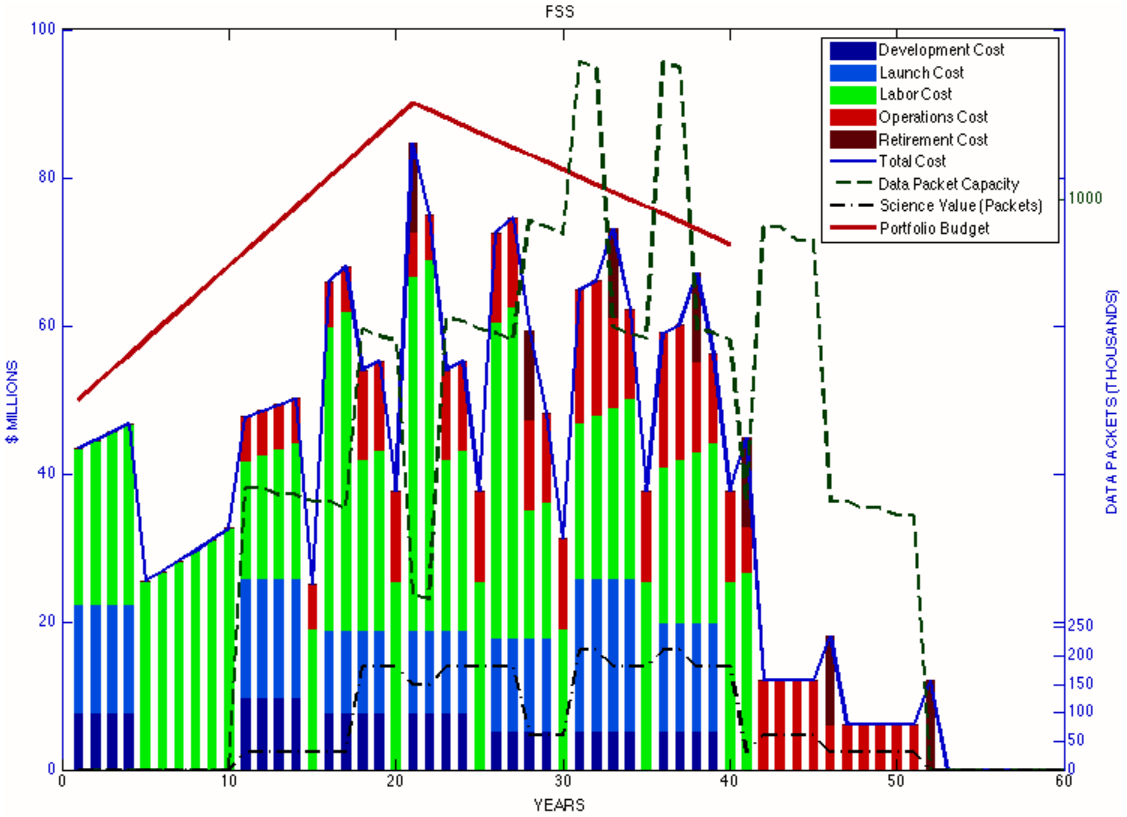


Figure 6-16: Cost and performance profile of Portfolio 19

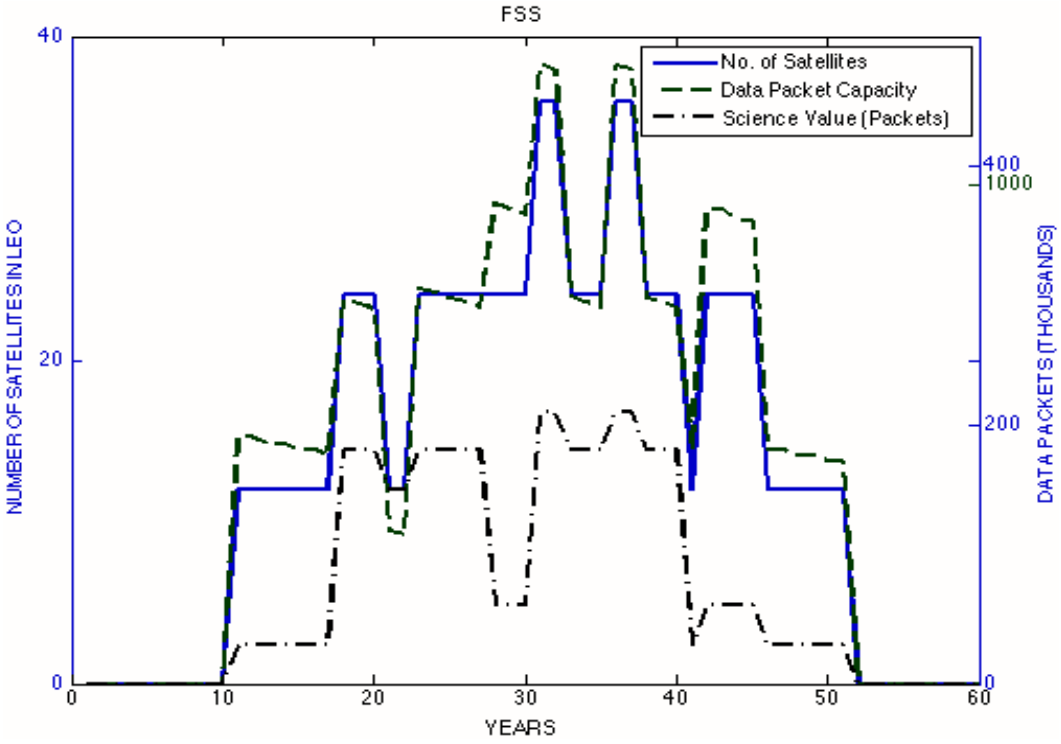


Figure 6-17: Satellite and performance profile of Portfolio 19

6.4 Portfolio Analysis Results

After determining that Portfolio 19 is most preferred and affordable solution for FSS development, it now remains to work backwards from portfolio to program to system level in order to find out what it actually is. Recalling that Portfolio 19 is defined by the vector [1,1,2,2,1,1,1], the FSS can be developed by picking the high-risk designs for Constellations 1,2,5,6,7 and the medium-risk designs for Constellations 3 and 4. The choices of constellations for Portfolio 19 are summarized in Table 6-3.

Table 6-2: Program Constellation Profile of FSS Portfolio 19

Phase (Epoch)	Program Design Number	Risk (Program Option)	Type	Program Vector
1 (1)	349	High (1)	Design	[1,12,4,2,0,4]
			Perf	[495000,30000,3]
			Expense	[30.76,58.97,251.56,6.03,12,10,4]
2 (2)	331	High (1)	Design	[1,12,3,2,0,4]
			Perf	[315000,150000,3]
			Expense	[39.40,64.45,122.13,6.03,12,7,4]
3 (3)	1069	Medium (2)	Design	[3,12,4,2,0,4]
			Perf	[495000,30000,1]
			Expense	[30.76,44.44,162.84,6.03,12,7,4]
4 (10)	1069	Medium (2)	Design	[3,12,4,2,0,4]
			Perf	[495000,30000,1]
			Expense	[30.76,44.44,162.84,6.03,12,7,4]
5 (17)	331	High (1)	Design	[1,12,3,2,0,4]
			Perf	[315000,150000,1]
			Expense	[20.77,50.68,82.88,6.03,12,5,4]
6 (24)	349	High (1)	Design	[1,12,4,2,0,4]
			Perf	[495000,30000,4]
			Expense	[20.78,82.71,110.51,6.03,12,5,4]
7 (19)	349	High (1)	Design	[1,12,4,2,0,4]
			Perf	[495000,30000,3]
			Expense	[20.77,58.66,136.04,6.03,12,6,4]

The design vector lists the factor levels for PRG-DV-1 to PRG-DV-6:

```
[satellite_system_design, number_of_satellites,
number_of_satellites_in_concurrent_deelopment, constellation_type, legacy_operation,
program_contract_length]
```

The performance vector lists the factor levels for PRG-PA-1 to PRG-PA-3:

```
[Annual_Free_Data_Capacity, Annual_Science_Value, Maneuverability]
```

The expense vector lists the factor levels for PRG-EA-1 to PRG-EA-6:

```
[Total_Development_Cost ($ million), Total_Launch_Cost ($ million), Total_Labor_Cost ($
million), Total_Operations_Cost ($ million), Total_Retirement_Cost ($ million),
Waiting_time_to_launch (years), Program_Contract_Length (years)*]
```

In Table 6-3, the first element of every program design vector represents the choice of satellite system design, where "1" denotes the high-risk system and "3" denotes the low-risk system. Similarly, the choices of satellites that make up the constellations are summarized in Table 6-4 with their design, performance attribute and expense attribute vectors. Based on the information provided in these two tables, each DM will have direct information about their most preferred designs at the system, program and portfolio levels. The results obtained from cost, performance and satellite profiles for the portfolio, as well as the **design choices for the satellite system and constellation for each epoch** can be formulated as an **overall development strategy** and a **hypothetical storyline** for the FSS.

Table 6-3: Satellite System Profile of FSS Portfolio 19

Phase (Epoch)	System Design Number	Risk (System Option)	Type	System Vector
1 (1)	17202	High (1)	Design	[10,30000,25,3,1,3,100,1000,2]
			Perf	[3,41250,2500,18465]
			Expense	[3.20,5.46,2.5]
2 (2)	16830	High (1)	Design	[10,30000,25,2,1,3,500,1000,2]
			Perf	[2,26250,12500,14024]
			Expense	[4.10,6.71,1.50]
3 (3)	17102	Low (3)	Design	[10,30000,25,3,1,1,100,500,2]
			Perf	[3,41250,2500,1239]
			Expense	[3.20,3.70,1.25]
4 (10)	17102	Low (3)	Design	[10,30000,25,3,1,1,100,500,2]
			Perf	[3,41250,2500,1239]
			Expense	[3.20,1.85,1.25]
5 (17)	17114	High (1)	Design	[10,30000,25,3,1,1,500,500,2]
			Perf	[4,26250,12500,931]
			Expense	[2.16,4.69,0.75]
6 (24)	17206	High (1)	Design	[10,30000,25,3,1,3,100,2000,2]
			Perf	[4,41250,2500,26305]
			Expense	[2.16,4.31,1.25]
7 (19)	17202	High (1)	Design	[10,30000,25,3,1,3,100,1000,2]
			Perf	[4,41250,2500,18606]
			Expense	[2.16,2.72,1.25]

Design vector for SYS-DV-1 to SYS-DV-9:

```
[spacecraft_lifetime, initial_data_packet_capacity, annual_data_packet_usage_rate,
FSS_interface_communications_technology, terrestrial_capacity_real_option,
propulsion_type, payload_capacity, propellant_mass, launch_vehicle]
```

Performance vector for SYS-PA-1 to SYS-PA-4:

```
[ISL_capability, Annual_Free_Data_Capacity, Annual_Science_Value, Delta_V]
```

Expense vector for SYS-EA-1 to SYS-EA-3:

```
[Development Cost ($ million), LaunchCost ($ million), Development_Time (years)]
```

6.5 FSS Development Strategy

A possible development strategy and the FSS development storyline based on these results are described below. It also serves as a summary of this case study.

Before the commencement of the FSS initiative, a mix of national or multinational space agencies had expressed their desire to launch a new satellite constellation of their own in order to further their various interests in space exploration. Their efforts were initially uncoordinated and many agencies were preparing to launch their own constellations at approximately the same times. However, with the commencement of the FSS initiative, several agencies saw the potential benefits of the FSS to the development of future space systems. As such, several agencies have decided to commit towards its development. These agencies would then be key decision makers (DMs) in the FSS project and the DMs could comprise ESA (DM1), JAXA (DM2), CSA (DM3), CNSA (DM4), UKSA (DM5), NASA (DM6) and USAF (DM7). An overall governing body called the FSS Directorate (FSSD) was formed using key representatives from each of the DMs to oversee the development, design, launch, operation and possibly retirement of the FSS satellites. The interests of the FSSD are hence aligned to and representative of all the DMs. As the DMs have agreed to participate in the FSS initiative, they would equip their own satellite constellations with FSS-enabling capabilities and launch them at strategic times proposed by the FSSD.

In the FSS paradigm, each satellite is considered as a ***system****, each satellite constellation is considered as a* ***program****, and finally the FSS is considered as a* ***portfolio****. The FSSD is hence responsible for monitoring the performance, cost and schedule in the design of the system, program and portfolio. However, the FSSD is unaware of the overall time and monetary budget for the FSS portfolio until they obtain information about the system or program. As such, the FSSD applied the* ***design for affordability process using tradespace-based methods*** *and is considering additional cost and schedule attributes at system, program and portfolio inception. Different cost elements and development schedule would be considered in the design process since designing for affordability warrants the* ***holistic consideration of performance, cost and schedule attributes****. Since the FSSD and DMs are aware of the cost of developing a single satellite from its basic materials, a* ***bottom-up approach*** *was taken for this design process. By ensuring that the design of every satellite is affordable, the design of a satellite constellation is also likely to be affordable, and eventually the entire portfolio is also likely to be affordable across its projected lifecycle. The system, the program and the portfolio would hence be* ***performance-effective, cost-effective and time-effective****.*

At the beginning of this design process, the FSSD established mission statements for the system, program and portfolio. They interviewed every DM about their satellite constellations, gathering information about their design, mission and performance requirements. Through these interviews, the FSSD was able to establish ***design variables, epoch variables, mission types, performance attributes, and resource expense attributes*** *for the system, program and portfolio. All DMs then agreed to conduct their constellation development using the common design variables and evaluate them based on the same attributes.*

The combination of epoch variables aided in establishing different operating contexts in which all mission could possibly be conducted. They then applied ***Multi-Attribute Utility Theory (MAUT)*** *and mapped each DM's preferences to a utility value in order to establish* ***Single Attribute Utility and Single Attribute Expense curves*** *for every attribute.* ***Stakeholder preferences*** *towards various attributes vary for the conduct of different missions in different contexts. The duration during which a specific mission is to be conducted in a specific context is known as an* ***epoch****. Given 6 possible missions and 4 possible contexts, there are a total of* ***24 possible epochs****. Therefore, there is a unique set of preference curves for the satellite system and constellation that could be designed in every epoch.*

The FSSD commenced with ***system level analysis*** *and conducted* ***Multi-Attribute Tradespace Exploration (MATE)*** *for the satellite system. They performed a* ***design to value mapping*** *of the design variables to the performance and resource expense attributes of the system and identified the key interactions that would be fundamental to establishing the system tradespace model. Using the tradespace model, every possible satellite design was generated. Their performance and expense attributes were also calculated. Applying MAUT by explicitly considering stakeholder preferences through attribute weights and utility/expense curves, values for* ***Multi-Attribute Utility (MAU)*** *and* ***Multi-Attribute Expense (MAE)*** *were calculated for each system design. The design points were then plotted on a tradespace parameterized by MAE and MAU, whether the Pareto frontier set solutions were identified. The FSSD was able to generate* ***24 different tradespaces for 24 epochs****.*

However, not all epochs and not all sequences are useful to planning FSS development. Since the FSSD had the highest authority, they established ***7 development phases for the FSS*** *and mapped 7 of the 24 possible epochs to them. 7 phases were chosen because there would be 7 agencies in the FSS initiative and each agency would like to launch its own constellation. A logical approach would hence be to synchronize the development and launch of a new constellation with the start or end of each epoch. As a result, DM1 to DM7 agreed to stagger their constellation developments and commence only at the start of their corresponding development phase (epoch). The 7 phases then followed a logical progression that FSS development could possibly take in reality. The FSSD assumed the* ***first epoch*** *to last for* ***10 years****, while* ***the second up to the final epochs*** *were to last for* ***5 years eac****h. Given that more time is required initially to plan the overall development of the FSS, the first epoch would last longer than the others and would transit to the second epoch only after the first FSS-enabled satellite constellation has been launched. Therefore, there would be* ***7 epochs*** *and with the assumed durations of each epoch, the FSSD has optimistically aimed to complete the development of the FSS over a* ***projected period 40 years****.*

The corresponding 7 tradespaces were then selected. ***Lognormal distributions*** *were applied to the expense attributes to establish confidence intervals for Pareto frontier set solutions.* ***3***

preferred system designs *were chosen from each tradespace. Depending on the extent to which the confidence interval remains in the affordable solution region, the 3 designs were classified as* ***high, medium and low risk****. These 3 designs were then used by the FSSD in* ***program level analysis****. The same steps taken for system level analysis were taken, yielding* ***2 to 3 preferred program designs*** *for each of the 7 epochs.* ***Epoch-Era Analysis (EEA)*** *was then conducted to produce* ***utility and expense trajectories*** *for the preferred designs. Designs whose trajectories would always remain above the constraint levels were considered* ***fully affordable*** *and would be down-selected.* ***Partially affordable*** *designs would not be selected unless there were no better solutions in the epoch. The combination of the down-selected satellite constellation designs would serve as design variables for the FSS portfolio.*

Finally, ***portfolio-level analysis*** *was conducted by the FSSD, who determined that* ***Portfolio 19*** *would be the most preferred design after analyzing the* ***cost, performance and satellite population profiles*** *of several portfolio designs. To initiate the development of Portfolio 19, the FSSD could make the following recommendations to each of the DMs using the design, performance attribute and resource attribute expense vectors obtained for each of its constituent constellation.*

For example, ***ESA (DM1)*** *would be responsible for the development of* ***Constellation 1*** *over a* ***program contract length of 4 years****. The constellation would be* ***disaggregated*** *and contain* ***12 satellites****, which would be* ***retired after 10 years*** *in operation. 4 satellites are to be developed concurrently, and after considering cost and time penalties for launch,* ***the total waiting time to launch would be 10 years****. The constellation would be able to provide free* ***annual data packet capacity of up to 4950000****,* ***annual science value of 30000 data packets****, and a* ***maneuverability level of 3****. Each satellite in the constellation would be designed with a* ***system lifetime of 10 years****, an* ***initial data capacity of 30000 packets****, payloads with* ***data usage rate of 25 packets per kilogram****,* ***laser communications technology****, with a* ***terrestrial capacity real option exercised****,* ***electric propulsion****,* ***payload capacity of 100kg****,* ***propellant mass of 1000kg*** *and to be launched on the* ***Falcon 9 Heavy****.*

Therefore, the FSSD would provide the recommendations for constellation development in the same format to the other DMs. With knowledge of the systems and programs to be developed, the FSSD could now engage with various governments and agencies to determine the overall portfolio budget to be allocated for the development of the FSS over a 40-year period. Assuming that all constellations were developed, launched and operated successfully, the FSS based on Portfolio 19 would be developed in a progressive manner, where its performance in terms of annual free data capacity and science value would increase in step and remain consistently high during critical periods of operation. This would greatly support the development and operation of the FSS.

The cost commitments across the development lifecycle of Portfolio 19 are always below the portfolio budget line with adequate total cost buffer, thereby reducing the probability of cost breaches in any of the years. As a result of affordability considerations at the system, program and portfolio levels, the FSS based on Portfolio 19 would not only be technically superior to most of all possible portfolio designs, it would also be more cost-effective and time-effective. Therefore, the early-phase design for the FSS is complete and the FSS initiative can be implemented based on the development strategy provided by Portfolio 19.

With the consideration of different cost and time elements as well as uncertainties in resource expenses at the outset, the design for affordability process can facilitate the design of a complex SoS from system to program to portfolio levels. The FSS case study demonstrated the applicability of this design method, where the results culminated in producing an overall design and development strategy for all satellite constellations making up the FSS portfolio. The FSS to be designed based on the results of affordability analysis using the bottom-up approach is hence considered to be fully affordable across its development lifecycle. Certainly, there are no bounds on the number of preferred designs that could be picked at the end of system and program level analysis. The design for affordability process could be conducted iteratively with different sets and different numbers of system and program designs until the most desirable portfolios can be obtained. Therefore, the same design process can be applied to any complex SoS whose development roadmap resembles the bottom-up nature of the FSS.

7 INTEGRATING TRADESPACE-BASED METHODS WITH INDUSTRY PRACTICES & POLICIES FOR AFFORDABILITY

7.1 Motivation

The earlier chapters have highlighted affordability as an imperative concept in the system design process, as the current notion in the defense and aerospace industry is to adopt the holistic approach of considering cost and schedule parameters on top of traditional measures of performance. While many methods and frameworks have been introduced in recent years to enable the design for affordability, they have inherent drawbacks such as the lack of time centricity and value centricity. These systems are often large and complex, comprising many subsystems performing different tasks in various operating contexts, involving many people of diverse backgrounds, producing emergent capabilities and even evolving over time. Such is the dynamic nature of systems that it becomes impractical to design a system for single point futures using static quantitative methods and frameworks, and without the consideration of stakeholder preferences.

The design for affordability process is complex, as it requires the holistic consideration of various cost, schedule and performance attributes at the system, program and portfolio levels. The amount of information and uncertainty increases by several orders of magnitude as the design process transits from one level to the next. As the solutions obtained at higher levels become increasingly dependent on the solutions derived at lower levels, the risks of not being able to select and identify preferred designs also increase with each level. Without a systematic framework to manage the transition of the design process between different levels, the design for affordability process can potentially become ill structured and cognitively challenging to system architects due to the overwhelming load of information.

As such, tradespace-based methods are introduced as promising approaches for the design for affordability process. Tradespace-based methods can offer numerous benefits to the early-phase design of complex systems. As demonstrated in the FSS case study, methods such as Multi-Attribute Tradespace Exploration (MATE) and Epoch-Era Analysis (EEA) explicitly consider the notion of time in that systems can change during their lifecycle, the different cost and schedule parameters as key design variables and attributes, the varying stakeholder preferences towards these attributes, and most importantly, the concept of "value" to stakeholders across the system lifetime.

Tradespace-based methods can offer a cross-disciplinary understanding of the goals and expectations for system architects in solving complex design problems. As demonstrated in the FSS case study, tradespace-based methods are pedagogical in nature, and they can guide system architects to iteratively generate and refine design solutions through effective communication and collaboration. With the application of tradespace-based methods in early-phase design, preferred design solutions can potentially be obtained at the system, program, and portfolio levels. Overall design strategies can also be obtained, which will serve as overarching guidelines throughout the

design process. As such, tradespace-based methods can be the answer to the longstanding problems of performance management and reducing cost and schedule overruns in the defense and aerospace industry.

However, there exist three questions pertaining to the role and relevance of tradespace-based methods to engineering design within the greater context of the defense and aerospace industry. These questions were uncovered during the proposal of these methods as key tenets of the design for affordability process, as well as during the application of these methods to the FSS case study. It is necessary to address these questions so that future users of tradespace-based methods will be aware of their pros and cons and apply them fittingly during affordability engineering design and practice.

The *first question is with regards to the implementation issues* that exist in tradespace-based methods. Like other methods currently in existence, tradespace-based methods are not perfect and they have inherent drawbacks that must be recognized and managed by system architects in order to ensure the integrity of the design for affordability process. They do have degrees of uncertainty and subjectivity that may impede the down-selection and identification of preferred solutions at different levels. However, their benefits can outweigh their drawbacks if they are applied appropriately and consistently. Therefore, it is in the interest of system architects to be able to recognize the pros and cons of tradespace-based methods and utilize them in ways that best meets the needs of their design problem at hand.

This then leads to the *second question of when and where tradespace-based methods should be applied* for them to be most effective. Throughout this thesis, they have been introduced and discussed as standalone methods, where they are applied from the start to end of hypothetical case studies to derive preferred design solutions. However, the design process in the real world is fraught with much more complexity, as the early-phase design task is only part of a greater acquisition framework. In the defense and aerospace industry, these acquisition frameworks are key enablers of progress for they facilitate delivery of systems that enhance the capabilities, reputation and influence of many enterprises. It is clear that reforms in acquisition processes are necessary to eliminate the problems currently experienced.

Across these enterprises, there exist various acquisition frameworks that have long taken root in the acquisition community. Despite histories of cost and schedule overruns, experienced acquisition practitioners still use them repeatedly due to inertia in reshaping enterprise practice and engineering policies. Performing a complete upheaval of such acquisition processes and replacing them completely with tradespace-based methods may prove to be a step backwards due to social, economic and political ramifications. Therefore, it may be more useful to determine how tradespace-based methods can be used in ways evolutionary, rather than revolutionary, to enhance existing acquisition frameworks. It is then of interest to determine how tradespace-based methods can complement these acquisition frameworks to reduce incidences of cost and schedule overruns in the defense and aerospace industry.

Finally, the *third question is about the ways the issue of affordability can be resolved beyond the scope of engineering design and analysis*. Design for affordability is not simply an engineering problem and apart from reforming systems engineering practices, many other strategies can be taken to maintain affordable practices throughout industry and enterprise. Through prudent

reformation of engineering and management practices, the principles of affordability may not only be embodied in the physical products delivered by defense and aerospace enterprises, but also may be propagated through wider industry practice and management philosophy. Therefore, the aim of this chapter to address these three questions in order to enable the integration of tradespace-based method with industry practices and policies in order to achieve affordability.

7.2 Benefits and Concerns of Using Tradespace-Based Methods for Affordability Analysis

This section aims to address the first question by discussing implementation issues associated with tradespace-based methods for affordability analysis as demonstrated in the FSS case study. As such, this section will serve as a further discussion of the results derived in Chapters 4-6 and to provide greater insights on the overall effectiveness of these methods. Further examples in systems design will also be used to support the discussions. Over the course of developing the FSS case study, many important lessons were learnt during formulation of the tradespace models and affordability analysis of selected designs. Tradespace-based methods, namely MATE and EEA, can be used concurrently to foster a very systematic and disciplined approach towards system design. By the end of the portfolio analysis, a potential solution with the most preferred tradeoffs among performance, cost and schedule attributes were derived and different program designs were prescribed for each development phase that contributed towards the overall establishment of the FSS. Design vectors were obtained for the entire portfolio, as well as individual constellations and individual satellites, thereby prescribing a prospective design solution to the FSS stakeholders at every design level.

Therefore, the research methods and results presented in the earlier chapters were able to demonstrate the applicability of these methods for designing complex systems over multiple levels. As such, tradespace-based methods can potentially be used in the early-phase design of any complex system. The design procedures prescribed by these methods followed a logical progression and a gradual learning curve can be easily established through formulating tradespaces progressively through different design levels. Although these methods can offer new perspectives towards systems design and engineering, one must be aware of its inherent drawbacks and potential liabilities. Ross (2006) also discussed several implementation issues related to tradespace exploration and analysis. Therefore, implementation issues concerning their benefits, drawbacks and liabilities will be discussed collectively.

7.2.1 Benefit 1: Systematic, Disciplined and Convenient Approach to Design

Affordability analysis can be conducted in a *systematic, disciplined and convenient* manner with holistic considerations of performance, cost and schedule when tradespace-based methods are deployed properly. This approach is *systematic* as it helps stakeholders to conceptualize the design problem in a sequential manner and break it down into separate system, program, and portfolio design problems. Given the need to now consider additional cost and schedule attributes, it is important to manage this growth in information to prevent cognitive overloading of decision makers and system architects. Without this breakdown structure, decision makers will find immense difficulty in drawing the system boundaries, resulting in the over-estimating or under-estimating of the complexity of the design problem. Having the system-program-portfolio

framework helps to structure the design process through a logical progression, and stakeholders can either apply the bottom-up or top-down approach depending on the information made available to them.

This framework, coupled with MATE and EEA, enables the design process to become *disciplined*. At every design level, stakeholders will have to present the mission statement, which will then drive the identification of key design variables, epoch variables, performance attributes and expense attributes that characterizes the current entity. Stakeholders will then generate epochs and can choose to sequence them to form an era in order to simulate an actual portfolio development timeline. They will next perform design to value mapping to determine the key interactions between variables and attributes. With all of this information, stakeholders will formulate the tradespace model, and use the variables as inputs to determine the ranges of values for the outputs.

All possible designs, or some subset, can be enumerated. Stakeholders will then state their preference towards levels of the attributes, which will then be translated to the value metrics of utility and expense. Using the aforementioned model, all sampled design points are evaluated, and applying Multi-Attribute Utility Theory (MAUT), overall utility and expenses are calculated for each of them. A tradespace parameterized by MAU and MAE is then generated and Pareto fronts are identified. Lognormal distributions may then be applied to individual expense elements of Pareto frontier set solutions to realistically account for uncertainty in cost and schedule estimating. External performance and expense constraints are identified and aggregated to produce utility and expense constraint levels that narrow tradespace exploration to the area defined by the affordable solution region. A handful of preferred designs can be investigated and compared, and the most preferred designs may serve as design inputs for the next level of analysis. The same process is then repeated for subsequent levels of analysis until stakeholders can find affordable solutions at every level.

As such, affordability analysis using tradespace-based methods can enforce discipline in the design for affordability process. It proceeds in a logical manner and it is easily repeatable if there is a need to iterate the design process at a particular level to search for more affordable designs. Also, a learning curve is established after stakeholders complete the exploration and analysis at the system level. As experienced in the FSS case study, the formulation of tradespace models for the program and portfolio become increasingly straightforward. Depending on the fidelity of the models and the number of preferred designs, the solution space can become smaller at each level, thereby reducing the design complexity and information that is typically expected at the program and portfolio levels.

Finally, this approach is also *convenient* given the gradual learning curve and widespread availability of computational resources. As all procedures for MATE and EEA have already been sequenced and described, stakeholders simply have to follow the steps prescribed to perform the design process. All information required for further analysis and identifying preferred solutions could be produced using the tradespace models. Unless really necessary, there is no need to formulate different models on different software platforms to arrive at a single solution. Therefore, solutions to early-phase design problems can be derived simply through using tradespace-based methods.

If the tradespace-based methods are applied in a systematic and disciplined manner, they can become computationally less intensive. This is convenient for stakeholders who may have limited computational resources for conducting early-phase design and can only do it progressively. As demonstrated in the FSS case study, tradespace-based methods can potentially reduce the computational complexity at each level of affordability analysis, as the solution space for a program and portfolio becomes increasingly smaller and more manageable. Software tools and platforms for tradespace exploration like MATLAB are also readily available, and stakeholders do not have to procure new equipment or multiple software platforms for the same purposes.

Therefore, the most obvious benefit of using tradespace-based methods for affordability analysis is the *systematic procedure, discipline and convenience* provided. With proper managerial support and knowledge dissemination, the same benefit can be propagated through the enterprise and promote a new system design philosophy.

7.2.2 Benefit 2: Layered and Scalable Methodology for Error and Complexity Reduction

Tradespace-based methods offer a *layered* and *scalable* approach that can potentially reduce complexity and errors typically encountered in the design of large systems. This benefit can be illustrated through the hypothetical design of a new generation of fighter aircraft, where multiple units of different aircraft such as fighter jets and helicopters are to be built under the same defense portfolio, and delivered in batches to the military over a 10-year period to enhance overall air superiority. Potential stakeholders may simply refer to the new fleet of next-generation aircraft as a single "system" or entity to be designed through affordability analysis and tradespace exploration. If only one phase of tradespace exploration was considered, it would result in an attempt to include as many design variables and attributes as possible to characterize this complex system in one instance. As a result, design to value mapping of many design variables to many attributes becomes challenging to stakeholders. The tradespace model also becomes computationally complex due to numerous embedded interactions.

Complex tradespace models also present numerous problems and they are similar to other software designed with several layers of abstraction. Should errors occur in the model, it becomes troublesome to detect and make corrections. Also, complex tradespace models may have many subsystem models interacting with one another through feedback and feed-forward loops that model creation and execution can become computationally intensive. If a high-fidelity model was chosen, the existence of so many subsystems for consideration also implies that a lot of labor is involved, hence driving up the development schedule and labor cost. When models contain many interactions, there exists a high probability that errors are present as well. This renders the validation and verification process both essential and cumbersome, requiring more effort and time spent in developing and correcting, rather than analyzing the model. Although complex models can be of high fidelity and have high degrees of scientific accuracy, there is always a risk that more important variables or attributes are being neglected from the model entirely. The omission of such lurking variables and attributes may undermine the credibility and saliency of the tradespace model, rendering preferred solutions likely to be flawed.

Performing only one phase of tradespace exploration for portfolio design with basic variables is analogous to attempting to design a portfolio of different aircraft using input design variables

meant for a single aircraft. When low-level system design variables are applied to a complex tradespace model to directly derive high-level portfolio attributes, stakeholders may not have visibility of the collective performance and cost attributes at intermediate levels. This intermediate level is likely to be the program level. Without multiple phases of tradespace exploration, stakeholders will face immense difficulty in performing tradeoffs to identify the most preferred designs should they rekindle interest in designing a program, be it a new squadron of fighter jets or helicopters. Therefore, stakeholders would not have the opportunity to identify any potential cost and schedule overruns that may already be present at the intermediate or program level. Even if affordable portfolio solutions were obtained using the single-phase approach, it does not guarantee that affordable program solutions will be inherited. A great risk is incurred with this single-phase approach and should an intermediate program become unaffordable midway through portfolio development, the entire portfolio would become increasingly unaffordable.

Using tradespace-based methods, however, can circumvent the problems of complexity and frequent error occurrence. With the system-program-portfolio framework, stakeholders will be coerced into breaking down the design problem into more manageable parts. In the same hypothetical example, stakeholders will first identify a single fighter jet and a single helicopter as separate systems, then multiple fighter jets and multiple helicopters of the same model as separate programs, and finally multiple squadrons of jets and helicopters as a single portfolio. For every design level, different design variables, performance attributes, and expense attributes can be identified and design to value mapping can be conducted. As a result of the problem breakdown, tradespace models for a system or program or portfolio are not as complex. By breaking down the problem into three distinct blocs, the design process becomes decoupled and layered.

A layered structure in the design process provides more avenues for direct stakeholder control. Each design bloc corresponds to system, program and portfolio level design. The inputs to each bloc will have the design variables as inputs and preferred designs as outputs that will be channeled to the next bloc. The tradespace model in each bloc will have a manageable number of design variables and attributes and with this layered structure in design approach, important variables and attributes are less likely to be neglected. Since the reduced amount of information is cognitively less demanding, key interactions among between design variables are also less likely to be omitted by accident. System architects will only need to focus on a smaller set of design variables when identifying interactions. It is also less computationally intensive to validate and verify separate models for a system, program and portfolio, even if each of them is programmed with high levels of fidelity.

With the layered structure imposed by the framework, there are more checkpoints available through the design process during which stakeholders can assess the saliency and credibility of the models, as well as perform the necessary tradeoffs required for affordability. These checkpoints are the analysis stage for each tradespace exploration phase and they provide opportunities for stakeholders to determine whether the preferred designs for a system or program are indeed satisfactory in terms of performance and expense requirements. Should errors be present in the tradespace model at the system or program level, they can be easily corrected at the end of each bloc and repeated until the model is deemed satisfactory. This can prevent the cascading of modeling errors to the portfolio level.

The breakdown of the design process is not restricted to the system-program-portfolio framework. Depending on the preferences of the stakeholders, they can either be structured into two distinct parts or even more than three parts if the design problem is more complex. If the task is simply to design a new squadron of fighter jets, only the system and program level tradespaces are required. However, should the design problem be "the enhancement of aerial defense of the nation under the same budget portfolio", more intermediate levels can be used during early-phase design. Such levels may include "minor programs" and "major programs". A "minor program" can be the design of a squadron of fighter jets or helicopters, while a "major program" can be the design of different groups of squadrons of fighter jets or helicopters. Hence, the portfolio previously considered in the system-program-portfolio framework can now be regarded as a major program. The portfolio in this case can possibly comprise multiple major programs, such as the design of a new constellation of military satellites and an advanced ballistic missile defense network. Therefore, the number of levels in the design process can be scaled in multiple ways that allows stakeholders to best reduce the complexity and probability of error occurrence throughout the design process.

With the layered structure and scalable properties of tradespace-based methods using the system-program-portfolio framework, stakeholders can perform tradespace exploration and affordability analysis one level at a time and will be better able to identify preferred solutions at each level with better control and more opportunities for error detection.

7.2.3 Benefit 3: Overcoming the Limitations of Existing Methods

An immense benefit of tradespace-based methods is that the common problems currently experienced by existing methods for affordability analysis can be remedied. As tradespace-based methods are an amalgam of many existing methods with each having their own merits, they can address the limitations described earlier in Chapter 2 and potentially enhance the manner in which affordability analysis is conducted in future.

The *first* common limitation that tradespace-based methods can overcome is the ***lack of time-centricity***. Systems can evolve and change, and are subjected to disturbances and perturbations over time. Existing methods focus on designing systems only for single point futures and do not explicitly consider how performance, cost and schedule parameters change over time. This is especially important for systems in the defense and aerospace industry, where the systems are often huge and designed to operate for a number of years and beyond. However, MATE and EEA for affordability analysis easily accounts for time-dependencies throughout the design process. This is because time-related attributes are now considered in the tradespace model to acknowledge that development schedule can be an important resource expense attribute given the potential for schedule overruns in program management.

This addition embodies the principles espoused by the *Time as an Independent Variable (TAIV)* concept. By having preferences and constraints for schedule or other time-related attributes, systems and programs can be designed for better time-effectiveness, affordable solutions can be developed and operated within a period of time most preferred by stakeholders. This was done in the FSS case study, where development schedules for individual satellites and total waiting times to launch for constellations were explicitly considered in the MATE process.

As EEA was used in affordability analysis for the FSS, changes to performance, cost and schedule attributes as a result of changing mission requirements and operating environments were accounted for through the construction of epochs and eras. This demonstrates that the FSS was designed with explicit considerations of time, as its development, launch and operate can be subjected to changes over its lifecycle. A number of epochs were then selected and sequenced to form an era to simulate the development lifecycle of the FSS portfolio. Expense and utility trajectories of preferred designs were plotted across the constructed era, and designs that always maintained their utility and expense levels above minimum constraint levels for corresponding epochs are identified as fully affordable solutions. It was also demonstrated that partially affordable solutions could also be selected if no better solution exists for an epoch.

Simulating a time-period using epochs and eras also embodies the principle of the *multi-period portfolio management approach* described by Davendralingam and DeLaurentis (2013). The FSS development timeline was segmented into different time periods, each characterized by a fixed set of mission needs and operating contexts. As a new constellation was developed at the start of every epoch, the FSS design was based on the sequential acquisition and removal of satellite constellations that describe its evolution over time. It can be seen that design decisions made in each epoch can affect the decision options of future time periods, thus affecting long-term performance and risks of the FSS. Therefore, tradespace-based methods can facilitate the identification and selection of system, program and portfolio designs that can remain affordable throughout their lifecycles.

The *second* limitation that can be circumvented using tradespace-based methods is the ***failure to recognize complexities of scale***. Most methods attempt to design the system and account for its affordability in a single instance or through a single iteration. As such, stakeholders may not initially recognize the complexities of designing huge entities like programs and portfolios, and attempt to do so using methods specifically crafted only for system level analysis. However, few of these methods have broken down the design process into distinct blocs in order to reduce the complexity experienced in analysis. As described earlier, tradespace-based methods offer a systematic, disciplined and convenient approach to system design, and their layered structure and scalable nature can help reduce complexity and probability of modeling errors. Through the system-program-portfolio framework, stakeholders applying the tradespace-based methods can recognize the complexities of scale involved in the design process and attempts to segment the design process into more than one distinct task. The interactions at program and portfolio level can become increasingly complex and they can easily be lost beyond the cognitive limits of stakeholders and system architects if they were all to be designed using a single tradespace model. By conducting tradespace formulation, exploration and analysis at distinct design levels according to scale, affordability analysis can be conducted in a more coherent and controllable manner.

The *third* limitation that tradespace-based methods offer to address is the ***lack of cost breakdown structures***. Many methods use cost, typically lifecycle cost, as the metric to measure the expected expenditure to design a system. Many instances of traditional tradespace exploration have used a single cost metric to make decisions. However, the term "cost" has been described as ambiguous, as there exists "different colors of money" that make up the lifecycle cost. Cost elements such as development costs, launch cost, operations cost, labor cost and other monetary expenses are

incurred in different amounts and at different times. Summing them up into a single cost metric ignores the possibility that having an overrun for a single cost element may result in the entire system becoming unaffordable. Tradespace-based methods for affordability analysis, however, avoid doing so by breaking down "cost" into its distinct elements and apply them to the tradespace model as resource expense elements. This approach is similar to the *affordability method proposed by* Tuttle and Bobinis (2012). Constraint levels are also placed on individual cost elements, reflecting the real world scenario where different cost elements can often be placed under the authority of different program managers. Having constraints on the maximum amount each manager can spend helps impose control on all cost elements and ensures that preferred designs are indeed cost-effective. This mirrors the *Cost as An Independent Variable (CAIV)* concept.

The *fourth* limitation that can be remedied is the ***treatment of affordability as a constraint***. Many methods treat the notion of affordability as an external constraint on monetary cost and time, as most forms of program control are in the form of budget plans that detail how much expenditure can be incurred overall. Traditional tradespace exploration and other methods will typically proceed in generating all possible designs evaluated using some measures of performance and total cost, and then potentially bounding the tradespace by a total cost constraint. While it is true that designing for affordability implies the elimination of designs that are too expensive through the use of constraint levels, it is also about considering the effects of cost and time on the outset of the design process. Therefore, the tradespace-based methods applied to the FSS case study not only used cost and schedule constraints to represent the budgetary needs of different epochs, but also considered different cost and time elements for individual satellites, satellite constellations and the portfolio of constellations when formulating the tradespace models. Affordability is more than just a constraint; it should be a critical part of the design.

The *fifth* and final limitation that tradespace-based methods can address is the ***lack of value-centric perspectives***. The current notion in the defense and aerospace industry is the conduct of Value Engineering and Earned Value Management. However, many methods rely heavily on the use of performance metrics, which facilitates the use of mathematical concepts such as optimization to find the "best" designs. However, what is considered the "best" design by mathematical algorithms may not be the most preferred by stakeholders. Also, arriving at a single "best" solution ignores the investigation of the second or third "best" designs, which may turn out to be more favorable to stakeholders. As such, there is no notion of measuring the level of benefit provided to stakeholders when using a particular design.

By using MATE, the design process automatically becomes value-driven as stakeholder preferences towards various performance, cost and schedule attributes are explicitly considered at the outset and translated to different utility and expense levels. Swing weights are also assigned to different attributes to reflect their relative importance to stakeholders, depending on the mission requirements and operating context. This is typically conducted through formal interviews and surveys, where stakeholder needs for different attributes are carefully elicited and translated to value measures using monotonically increasing single attribute utility and expense curves. Swing weights are then applied to aggregate different utility and expense levels to derive MAU and MAE metrics that can be used to evaluate individual designs. Stakeholders can then search the

tradespace for preferred designs using a variety of methods and not be restrained to selecting only one out of many possible designs.

One way of identifying better value solutions is through the identification of Pareto frontier set solutions that have the most preferred tradeoffs among performance, cost and schedule attributes. The utility and expense metrics were used in MATE for a single satellite, a satellite constellation and a constellation portfolio in the FSS case study. They were also used in the plotting of trajectories during EEA to determine the fully affordable designs for a selected sequence of epochs. As such, stakeholder preferences were always considered at all design levels and solutions are ultimately derived in an exploratory manner. Stakeholders can investigate various designs before collectively deciding on a single solution to develop on. This design process is a learning activity to help stakeholders understand what they truly want to design in a particular system, and it can be iterated as desired in order to search for even better value solutions. Therefore, the application of MATE and EEA enables the design for affordability process to be value-centric throughout, thereby delivering and sustaining value to stakeholders through the design lifecycle.

As they are able to address the limitations observed in many existing methods for affordability analysis, tradespace-based methods in the form of MATE and EEA can offer more advantages and ensure that the systems designed are both cost-effective and time-effective. However, like all methods and models tradespace-based methods have their inherent drawbacks and if used inappropriately, they may present potential liabilities to the design process. Therefore, it is important for practitioners of these methods to be aware of its disadvantages and mitigate their risks as much as possible. Discussion of drawbacks and liabilities shown in the next few sections will be based on insights derived from the FSS case study and the implementation issues discussed in Ross (2006).

7.2.4 Drawback 1: Inherent Subjectivity of Stakeholder Preferences and Multi-Attribute Utility

While the use of stakeholder preferences and multi-attribute utility in the design process can expedite the search for preferred and affordable solutions, it also introduces many degrees of subjectivity that may run counter to the objectivity provided by the scientific principles embedded in tradespace performance and expense models. When formulating tradespace models, the performance and expense attributes are to be calculated using equations and in accordance to scientific principles. For example, in the FSS case study, equations commonly applied in orbital mechanics were used to determine the minimum and maximum factor levels of propellant mass in order to ensure that a satellite has the delta-V potential to perform maneuvers through a range of orbit altitudes and inclination angles. Performance and cost models for individual satellites were also based on the equations and principles prescribed in Wertz and Larsson (2011). The tradespace model then enumerates all possible designs and calculates values for the performance, cost and schedule attributes. Therefore, the tradespace model for a satellite is established upon validated scientific principles and is perceived as an objective representation of a real world system.

However, when MAUT is applied, the ranges of these quantitative measures are translated to subjective measures of benefit in the form of utility and expense. Although this step explicitly

takes into account stakeholder preferences and helps to reduce modeling uncertainties, it also presents several limitations due to its subjective nature. This objective to subjective mapping using utility theory is heavily grounded in concepts from psychology and behavioral economics, and stakeholders often rely on reasonable heuristics to dictate their choice. As a result, they tend to apply what they already know about precedent systems in order to determine their own preferences for a new system to be designed. Such heuristics are not as impeccable as grounded scientific principles and it is often difficult to rely on them for making consistent and informed decisions in situations with high risks and high stakes. Preferences also change with people and the environment and they may not always be established on a rational basis. The minimum and maximum factor levels that correspond to the lowest and highest utility or expense for an attribute, as well as the number of stepped increases in utility or expense and the step size, can all be varied in any way desired by stakeholders. The inclusion of preferences from the start to end of the design process may imbue preferred solutions with high degrees of subjectivity.

Preferences can be captured through the behavior of decision makers, based on statistical analysis of their choices. This implies that the manner in which the system architects conduct the interviews and surveys has a direct impact on measuring the true perceived level of utility or expense of the stakeholders. Conducting an elicitation of needs is not easy, as there are so many design variables and attributes that can be associated with a complex system. The interview process or the survey questionnaire may take too long to be feasible for most real-world situations if all variables and attributes were to be considered for the model. When there are too many questions to think about, it becomes difficult for both system architects and stakeholders to determine which variables and attributes are actually most important to early-phase design. A lot of painstaking effort is also required to map changes in utility or expense to different attribute levels for so many variables and attributes. Furthermore, it is time consuming to develop and administer such lengthy interviews and surveys, as well as selecting individuals who can serve as subject matter experts to answer relevant questions. Therefore, it is important to narrow the design process to a manageable number of variables and attributes.

In the FSS case study at the system level, there are nine design variables, two epoch variables and seven attributes that were considered in the tradespace model. Certain design variables such as data packet capacity and communications interface technology were selected based on the features that distinguished FSS-enabled satellites from existing satellites. Only high-level architectural decisions are of importance. Other variables such as size of solar panels or thermal capacity of materials are also critical to the satellite design and they could have been included in the tradespace model. However, it was decided that they were not considered as key architectural decisions. It is up to the stakeholders to determine which design variables form key architectural decisions depending on their interests, and more importantly, directly impact utility and expense attributes as well drive the desired ilities. Typically, design variables that directly impact the new capability described by the mission statement should be included and those that have lower impacts can serve as intermediate variables or be completely left out of the model. Therefore, it is necessary for both system architects and stakeholders to fully understand the engineering design of the satellite and collaborate at a professional level in order to formulate a cognitively manageable design problem.

Another nuance of using preferences is that stakeholders may not be entirely sure of what they want in a new system until the system itself is actually constructed. They may appear to be sure of the design variables, performance attributes, expense attributes and their preference levels towards each attribute at present, but they may change their opinions upon completion of the models and analysis of the results in the future. Therefore, the tradespace model may not output the attributes that the stakeholders initially expected; or a single attribute level or a range of attribute levels that is mapped to a utility or expense value at present may not actually be the actual utility or expense level experienced by the stakeholders. This raises the dichotomy of expected value and experienced value. While it may be trivial to update preference levels in the tradespace model to fit experienced value, it takes more time and cost to do so when there are multiple changes to be made to the tradespace model. This may include different ways of calculating existing attributes, and the introduction or removal of design variables, epoch variables and attributes. These changes may be an impediment to the aim of designing affordable systems, where different cost and time elements are also of priority.

Also, not all preferences are revealed during the interviews and surveys. Ross (2006) discusses the need to be aware of both articulated and unarticulated value if system architects seek to deliver value through finding preferred solutions in early-phase design. Since articulated values are measures of preferences that have already been captured through the interviews and surveys, it is the unarticulated values that may impede the design process since it represents asymmetric information between the system architects and the stakeholders. Typically, stakeholders have the most knowledge of their preferences, but they may withhold that information since it could negatively affect their utilities. Such scenarios can be especially prevalent within the defense and aerospace industry, where revealing knowledge of future needs, potential new technologies, and new uses for a system may seriously compromise national security or the competitive edge of leading enterprises. The system architects thus experience a dilemma in trying to elicit the vital information needed to deliver value and yet not infringe upon the general interests of the stakeholders. The stakeholders themselves may also experience the same dilemma in trying to convey their preferences as accurately as possible without revealing potentially damaging information on their part. While such asymmetric information may benefit the interests of stakeholders, it also hampers the design process.

If the elicitation and translation of preferences are not done in a transparent manner, the integrity of the results obtained for affordability analysis may be compromised. However, given the nature of the industry at present, both system architects and stakeholders remain likely to encounter this dilemma. It is difficult to justify why highly classified information should be risked if the task is to perform only the early-phase design of the system. Instead, they should collaborate closely with one another in order to reach an agreement on how information can be best exchanged between them to enable tradespace exploration to be meaningful. Only through collaborative efforts and more transparent exchanges can the use of stakeholder preferences in evaluating the tradespace model be prevented from being perceived as a potential liability in the system design process.

Often the design of complex systems involves the participation of multiple stakeholders and it is often difficult to elicit preferences that are agreeable to everyone within a reasonable period of time. MAU models can be used in a group situation, but the presence of multiple sets of

stakeholder preferences may hinder the conduct of MATE. In the FSS case study, it was described that there are hypothetically seven different space agencies participating in the FSS initiative, with each potentially having their own set of preferences towards system and program level designs and their own scientific agendas. Such circumstances may point towards the need to aggregate all preferences in some quantitative manner and even perform multi-stakeholder negotiation. However, this problem is fortunately avoided by introducing the FSS Directorate, which is the hypothetical governing body that coordinates the efforts of all agencies. The FSSD has its preferences aligned to that of every agency due to commence development and launch in each epoch. It also has a separate set of preferences on portfolio design. Therefore, only a single set of preferences has to be considered for each epoch and this renders the design problem more convenient to solve.

However, in reality, there may not always be a similar governing body that may expedite the elicitation and translation of needs. More often there may be situations where multiple preferences sets have to be elicited from stakeholders. This makes the tradespace exploration process become more complex, as all designs have to be evaluated using multiple preference sets and multiple tradespaces will have to be generated to reflect the differing preferences among stakeholders. More cost and time are then incurred when searching for solutions that are common to the preferred solution sets of all stakeholders. Achieving this group consensus for either preferences or preferred solution may be very difficult and time consuming, or even impossible with some groups. The level of detail and specification necessary in the discussion of attributes and their weights can even result in considerable conflict and contention, rather than the move towards consensus.

If preferences reduce the objectivity in tradespace exploration, then the use of MAU and MAE may also pose obstacles for analysis. Both MAU and MAE are measured on a scale of 0 to 1. For utility, a value near 1 represents high satisfaction and a value near 0 represents low satisfaction. Expense is the inverse, where 0 represents high satisfaction and 1 represents low satisfaction. While they can be used to evaluate designs and allow comparison between design points, it is difficult to determine intuitively from the tradespace whether a design point by itself is indeed good or not. More work has to be done to find the design vector, performance vector and expense vector represented by the preferred design point in order to determine whether a design point has the desired attributes. There are multiple attributes aggregated into utility and expense values that make it more difficult to find out what attributes make one design superior to another.

This need to search and compare all the characteristics makes multi-attribute metrics not as intuitive as a traditional performance measure. For example, if a tradespace is solely parameterized by the delta-V and cost of the satellite, it is obvious to stakeholders that designs on the Pareto front have the best tradeoff between the two attributes. Even without the tradespace, it still remains trivial to sort through all the designs and pick a solution with absolute guarantee on the performance level, even if it is not Pareto optimal. This is the intuitive nature offered by traditional performance and cost parameters, as they offer stakeholders direct feedback on the information required for decision-making. With multi-attribute functions, however, a solution is characterized by a single MAU and MAE value, which is indirectly representative of the numerous vectors that actually characterize it.

Describing a point by having a utility of 0.7 and an expense of 0.5 means little unless it is compared to another design point on a tradespace. If a tradespace was never generated, finding preferred solutions becomes even more convoluted. This is especially so if only some but not all attributes in one design are better than another, thereby making it difficult for stakeholders to decide between the two. Having tradespaces and identifying Pareto frontier set solutions then benefits from the use of MAU and MAE since comparisons among different designs and multiple tradeoffs can be conducted on a single platform. Nonetheless, even if interpretation of the design, performance and expense vectors were straightforward for Pareto frontier set solutions, there is still a need to refer to the preference sets and utility curves to determine if the attributes obtained do fulfill the performance, cost and schedule requirements of a particular epoch. MAU and MAE do coerce stakeholders into thinking more holistically across more critical attributes during the design process, but understanding them may also be cognitively challenging to stakeholders unless they have already ascended the learning curve.

Comparing utility and expense values of design points between two epochs also may not come across as intuitive. Two designs in two epochs may both have a utility of 0.7, but their vectors are characterized completely differently. This is due to the changes in preferences and contexts between two epochs, causing two different designs to provide the same measure of benefit. In another situation, a single design can exist in two epochs with the same characteristics, but have different values for utility and expense as a result of the changes in preferences or performance. Therefore, evaluating designs using utility and expense coupled with the inclusion of stakeholder preferences becomes not as intuitive.

As stated during system, program and portfolio level analysis of the FSS, the attributes are also assumed to be perceived independent of one another, thereby facilitating the assignment of swing weights to each attribute and their eventual aggregation through the multi-attribute utility and expense functions. While performance attributes can be chosen and determined to a high degree of certainty that they are completely independent, the same cannot be said of cost attributes. In the Space Tug and FSS demonstrations, development costs, launch cost, operations cost, labor cost, retirement cost and schedule have all been assumed independent of one another. However, this may not be true in reality, as such costs are incurred sequentially and how much monetary resource is incurred for one element in the past can impact the amount that can be incurred for another in the future. Program contracts in the defense and aerospace industry are complex documents that list numerous details describing the agreement on how time and money should be spent to achieve design objectives. Breaching one cost element may risk jeopardizing the entire contract and subsequently other cost elements and the overall development schedule. Furthermore, some performance attributes used for assessment may not be independent of one another. As such, it may not be entirely realistic to make performance, cost and schedule elements fully independent of one another in the design process. However, the benefits of doing so can exceed its drawbacks.

Firstly, the perceived independence assumption helps streamline the elicitation process for determining the SAU and SAE functions of every stakeholder for all attributes. It allows interviews and surveys to be conducted for one of them at a time. Secondly, using multiple cost and time elements allow stakeholders to avoid making decisions based on the ambiguous measure of cost. As explained previously, the term “cost” is typically lifecycle cost, which comprises

many other cost elements that are spent differently and have various degrees of importance to stakeholders. Breaking down cost into its constituent elements help stakeholders gain greater insights into the expenses that are significant in determining the affordability of preferred designs. Therefore, it is still worth assuming the independence of different cost and time elements before aggregating them to calculate expense levels during MATE.

While including stakeholder preferences and using multi-attribute utility and expense functions have their flaws, it still remains a feasible analysis method that can be taken given limited resources and knowledge. Eliciting and including stakeholder preferences from the beginning can help stakeholders design a system to an extent that best meets their needs. Allowing direct human input thus enables the design process to become a learning experience, where better solutions can be identified through further iteration and corrections to the tradespace models.

7.2.5 Drawback 2: Variations in Model Fidelity and Tractability

The fidelity of the tradespace model can vary and it is highly dependent on the preferences of the stakeholder or analyst. Ross (2006) states that choosing the appropriate level of fidelity for model development can mean the difference between discovering key insights, and glossing over critical decision opportunities. The tradeoff determining the fidelity selection for the model is that of accuracy versus effort. If the system architects and stakeholders are very confident about the design principles for the new system, they can aim to create higher fidelity models that can produce more accurate and reliable results. However, it requires more expertise during its formulation as well as more computation power and time to execute. However, the higher the fidelity, the more assumptions are being made. Should one assumption not be realistic, it may undermine the results obtained from the model. Low fidelity models on the other hand have lower degrees of accuracy, but they require less time and effort to formulate. As only a few assumptions are made, it is easier for stakeholders to manage these assumptions and ensure that they can be easily validated. Low fidelity models are used for most early-phase design analysis, while high fidelity models are more often used for analyzing key attribute relationships.

In the FSS case study, data capacity was measured in “data packets” and satellites can establish ISL between one another to transmit data at fixed data rates. This is low fidelity modeling of satellite communications and this may come across as ambiguous to experts, as there is more involved in establishing space communications systems and networks than simply data packets. If stakeholders sought to create a high fidelity model from the start, they would use data capacity values that have been validated against existing spacecraft. This means that initial data capacity can possibly take a large span of values ranging from several megabytes to several gigabytes. Communication between two satellites is possible only if they are within line of sight and the link budget can be closed within a minimum slant range. Other attributes such as mean access time and handover time will also affect the quality of communications and the amount of data actually transferred. Communications capabilities are also dependent on the onboard power of the satellite, as well as the size and type of antenna. Different types of encoding and transmission protocols can also impact bit error rates and data throughput rates. Therefore, there exist many scientific principles in space communications alone that can be incorporated into high fidelity models that derive more meaningful attributes to describe FSS capabilities.

However, pursuing high fidelity models with high accuracy will require the calculation of various parameter values that do not contribute toward answering the question at hand. Closing link budgets and assessing the quality of communications only helps determine the feasibility of sharing in-space data assets among satellites, but does not help determine whether the FSS is a generally feasible and affordable design concept at the system, program and portfolio levels. Trying to create high fidelity models requires high resource expenditures, and results in a false sense of security on the conclusions. Therefore, it is important to perform fidelity matching from the beginning and stakeholders should first focus accuracy and effort in areas with most impact on the important results. Should high fidelity models be eventually required to assess key attributes in more detail, layered software architectures should be employed for MATE studies, with low fidelity models used in the first iteration. As system architects gain insight into the important relations and are able to select preferred solutions first, higher fidelity models can be substituted for their low fidelity predecessors. This helps to achieve a learning process without excessive modeling effort and time spent.

Nonetheless, Ross (2006) describes better models as those that reflect the underlying causal relationship, or "physics" of the design-to-attribute mapping. However, complex systems would often require either the incorporation of too many causal relationships when high fidelity models are needed, or have no relationships at all when there are no precedent cases available for direct validation. This is experienced during implementation of the FSS case study, where it required simulation of shared resources through the transmission of data packets. However, no space-based precedents are available for this new mode of space operations, with the closest being terrestrial applications like cloud computing networks and smart grids. As a result, not all complexities may be describable by causal relationships and only expert opinion or a small sample size dataset may be available. Instead, a data based or even semi-quantitative approach could be used, where factor levels describing low, medium and high attribute values are used. However, this is done at a cost of increased uncertainty (or reduced confidence). In any case, the dynamic MATE framework at least informs the analyst of the necessary information for determining attributes from designs. Regardless of the model type, a dynamic MATE study can still be conducted, though it is important to at least capture the highest-level cost-benefit tradeoffs within the causal structure of the problem even for qualitative analysis. Therefore, stakeholders should decide on the levels of model fidelity and tractability in order to make affordability analysis useful.

7.2.6 Drawback 3: Merits and Drawbacks of Qualitative and Quantitative Measures

Closely related to model fidelity and tractability is the use of qualitative and quantitative measures for attributes and factor levels of design variables. Models with high fidelity and heavily grounded in scientific principles use a lot of quantitative measures and "hard" numbers. If system architects are more inclined towards technical details, having hard numbers in their tradespace studies help convey a sense of concreteness and believability that words or images seem to lack (Ross 2006). However, this reliance can lead to misleading and dangerous results. For example, quantifying ISL capability with parameters such as channel capacity, channel efficiency and bit error rate can make assessing FSS communications capabilities more grounded in principles of space communications engineering. However, given limited knowledge on laser communications technology, attempting to quantify ISL capability provided by laser

communications technology is difficult and even impossible to validate. The next best option will be to use a lower fidelity model, where categorical qualitative measures of "Low", "Medium" and "High" can be used to describe the ISL capability of a satellite. This can then be quantified on a scale from 1-3 and incorporated into the tradespace model. Therefore, if using hard numbers can detract stakeholders' perceptions and undermine the quality of the analysis, qualitative or fuzzy metrics can be used instead.

As described in Ross (2006), the key results from a MATE study are not the numbers per se, but rather the insight into the structure of the design for value problem. The numbers help to make better decisions, but in themselves mean little. A rigorous qualitative MATE study is superior to a shabby quantitative MATE study. MATE has been used to derive affordable solutions at the system, program and portfolio levels of the FSS design case study. If high fidelity models with hard numbers throughout, it becomes harder to search for affordable solutions within a short period of time due to the need to understand and validate all the calculations embedded within the tradespace model. However, it does not mean quantitative methods should not be used at all. The system architects have to weigh the pros and cons of using qualitative versus quantitative approaches based on stakeholder preferences and the availability of accurate data and models.

7.2.7 Drawback 4: Problems with Model Validation

Tradeoffs between model fidelity and tractability as well as the use of various qualitative and quantitative measures have also posed problems for model validation. Validation is one of the key activities in the validation and verification (V&V) process necessary in computer simulation and model development. V&V are the principal means of assessing accuracy and reliability in models. For tradespace-based methods in particular, it is pertaining to the system, program and portfolio tradespace models. *Model verification* is "the substantiation that a computerized model represents a conceptual model within specified limits of accuracy" and *model validation* is "the substantiation that a computerized model within its domain of applicability possesses a satisfactory range of accuracy consistent with the intended application of the model" (SCS).

A *conceptual* model and a *computerized* model are formulated in the design process (Oberkampf and Trucano, 2002). The conceptual model contains all the information and mathematical equations that describe the system of interest. The computerized model is an operationalization of the conceptual model on a computer program. As a result, *model verification* is about assessing the relationship between the conceptual model and the computerized model and *model validation* is about assessing the relationship between the computerized model and reality. Therefore, V&V are necessary for assessing the accuracy and credibility of the conceptual and computerized models before tradespace exploration.

There are many methods available to perform V&V. Verification can be performed more easily through comparison of the conceptual and computerized models. However, model validation is a lot harder and it is only as accurate or as credible as the level of perceived reality of the system. It is often difficult to do so for systems in the defense and aerospace industry, where many systems are revolutionary in nature with unprecedented levels of performance requirements, cost commitments and schedule commitments. These new systems rarely have precedents that are designed or operated in the same manner, thereby making model validation or using any existing systems as a basis for comparison difficult due to potentially large numbers of differences.

Therefore, it is difficult for stakeholders and decision makers to even know what the expected performance, cost and schedule of the new system is going to be, much less the system architects. To say that a model is validated or verified is to say that its truth has been demonstrated, meaning that the model is true but it is impossible to "validate and verify the model" and demonstrate the truth of any proposition in complex open or closed systems (Oreskes et al, 1994).

This is illustrated in the FSS case study, where the opportunistic sharing of in-space data assets is an unprecedented function. Modeling the development strategy of the FSS required numerous assumptions and many mathematical relationships were based on heuristics that may not entirely be reflective of reality. This is especially so for cost estimating relationships, which reflect lower launch costs, lower schedule and lower development costs when laser technology and RLV TRLs become high. As compared to the model, costs and schedule may not adjust in such a discrete fashion in reality. These embedded assumptions thus render the system open and there is also potentially a high degree of aleatory and epistemic uncertainty in the model. Performing V&V with any existing satellite constellations like Iridium and MILSTAR is hence difficult, and at best accurate only for some but not all the subsystems.

Therefore, it is important for practitioners of affordability analysis using tradespace-based methods to understand that models can only corroborate a hypothesis by offering evidence to strengthen what may be already partly established through other means (Oreskes et al, 1994). Ultimately, models are mere representations of reality, and they are useful for highlighting aspects of the system for further study but not susceptible to proof. Models are most useful when they are used to challenge existing formulations, rather than to validate them. However, a model may also confirm biases and support incorrect intuitions.

7.2.8 Drawback 5: Availability of Expertise

Ross (2006) states that effective MATE implementation falls into two categories: process and content, and relevant expertise is required for both of them. Process expertise relates to understanding affordability analysis using tradespace-based methods. The system architects must understand the underlying assumptions and applicability of MATE and EEA in order to conduct the design process appropriately. It has been demonstrated in the FSS case study that affordability analysis for system, program and portfolio can be conducted in a systematic and disciplined manner if the design problem is well formulated. Therefore, it is easy for system architects to ascend the learning curve and understand the procedures of eliciting stakeholder preferences and conducting tradespace exploration. Expertise can thus be developed over time. However, it was noted earlier that the use of preferences and utility may not be intuitive to everyone, and the system architects themselves may have different opinions towards the modeling techniques applied. Therefore, it is important to educate system architects in ways that allow them to understand the process in the same manner so that they can collaborate more effectively in future.

Content expertise relates to the information for the study such as constraints, contexts, domain-specific knowledge, decision maker preferences, design knowledge and modeling knowledge. Availability of content expertise is always helpful but not necessary to ensure that models are constructed using the relevant scientific principles and at the required degree of fidelity. It is not required to have subject matter experts for every subsystem to be involved in the design process. However, the more experts there are, the more reliable and validated are the model and results.

For example, the FSS model requires the consideration of different subsystems such as onboard electronics, data storage, communication technology and propulsion. It would be helpful to have expert opinions on all these subsystems, but there is a trade between the marginal benefits of doing so against spending more time and labor cost. If system architects are confronted with the problem of insufficient expertise, they can rely upon past projects, analogy, and clever assumptions.

7.2.9 Drawback 6: Availability of Decision Makers

The presence of decision makers in the MATE process also help determine if affordability analysis was conducted in the manner desired and whether the preferred designs are useful. Without their input at the beginning and throughout the design process, results from affordability analysis may be invalidated upon their eventual inspection. However, it is difficult and time-consuming to continuously interact with these decision makers, especially in the defense and aerospace industry, where it takes substantial time to engage with senior military and government officials. If it is problematic to get hold of senior decision makers, proxy decision makers may be used, and they may be subordinates or subject matter experts with the true decision maker preferences and thinking frame of reference. However, it is more often that system architects have to assume decision maker preferences as a first-pass analysis in early-phase design, especially when there is no access to decision makers during brainstorming of design variables and attributes. This was demonstrated in the FSS case study, as there is limited time and limited access to relevant experts. Although this greatly expedites the process, it is more useful for different people to play the separate roles of system architect and decision maker in order to better characterize the potentially very different thinking styles. The caveat of taking the former approach is the need to validate the attributes with the decision maker or proxy at a later point in time. The designers and analysts run the risk of having to repeat effort if the attributes are incorrect.

7.2.10 Drawback 7: Computational and Schedule Constraints

Since affordability analysis is based on MATE, and not on other statistical or optimization techniques, it may require more computational effort than a more traditional multi-dimensional optimization exercise. With the inclusion of more design variables, the size of a tradespace increases by many folds. Multiple attributes have to be calculated using the tradespace model and eventually aggregated using multi-attribute utility and expense functions. Tradespace exploration thus requires a lot of computation resources to ensure that it can be conducted within a reasonable period of time. If there are insufficient computation resources, tradespace exploration will then take too much time. Therefore, there are computational and schedule constraints to the conduct of tradespace exploration and analysis.

To avoid overly large tradespaces, system architects may choose to restrict the number of design variables and factor levels applied to the model in order to generate a tradespace with a manageable number of design points. However, there is a risk that potentially better value designs are not enumerated and evaluated. There are many factors that affect cost and schedule attributes and it is important to consider most of their effects during affordability analysis. If tradespaces are large, then it will be necessary to use tradespace sampling techniques such as Latin hypercube sampling in order to scope the tradespace size for consideration to available resources.

Affordability analysis using tradespace-based methods reduces the time needed to search for preferred solutions due to the application of affordable solution regions. These regions help constrain the tradespace to a smaller area, making it easier for system architects and stakeholders to investigate preferred designs further. This allows preferred solutions to be identified and investigated in much quicker time.

Should computation resources pose obstacles to conducting affordability analysis meaningfully, early-phase design should first be done using low fidelity models with a handful of variables and attributes. Although results obtained may not be accurate or reflective of the intended system, they will still enable stakeholders to make some high-level architectural decisions. If more computation resources and time are made available later during the design process, higher fidelity models can then be formulated to provide results that supplement the decisions made earlier. This may help reinforce the choice of preferred designs, or highlight problems that would otherwise not be detected. Using this spiral approach allows system architects to still be able to generate insights under tight schedules.

7.2.11 Summary of Implementation Issues

This section has discussed a number of implementation issues central to the conduct of affordability analysis using tradespace-based methods. These methods offer a systematic, disciplined and convenient approach to the system design and management and its system-program-portfolio framework allows the design process to be layered and scalable. More importantly, they can overcome many limitations found in many existing methods. However, they also have potential drawbacks and liabilities. The usage of stakeholder preferences introduces subjectivity, and the use of multi-attribute utility and expense functions are not as intuitive as traditional measures of performance and cost. It also prevents the usage of established mathematical methods. Furthermore, its results are highly contingent on the levels of model fidelity and tractability, as well as the availability of expertise and decision makers. Tradespace-based methods are also at best close representations of reality and they are to be used to confirm heuristics rather than as proofs. Therefore, it is imperative to apply tradespace-based methods properly in order to obtain best results from affordability analysis.

7.3 Applying Tradespace-based Methods in Defense and Aerospace Acquisition

It was earlier explained how tradespace-based methods could be used to overcome the limitations experienced by existing methods in the design for affordability process. As a result, tradespace-based methods present significant benefits to system design and management in the defense and aerospace industry and they could be applied more often to ensure the delivery of better-valued and more affordable systems in future. However, due to the persistence of cost and schedule overruns for decades, the industry has already constructed various acquisition frameworks to curb them. These frameworks have been applied to numerous design problems by many organizations over the years such that acquisition practitioners have become so well acquainted with them.

As such, it will be an arduous task to demolish them in favor of tradespace-based methods unless there is a major policy upheaval. Therefore, it will be more cost-effective and time-effective to complement the usage of tradespace-based methods with existing acquisition frameworks. This leads to the second question as to where and when should tradespace-based methods be applied for maximal benefit. In answering this question, the primary acquisition frameworks from the US DoD and NASA will first be analyzed to determine areas where tradespace-based methods can be applied, followed by the proposal of an upgraded acquisition framework that can incorporate holistic considerations of performance, cost and schedule throughout design.

7.3.1 US Department of Defense Acquisition Framework

The US DoD was among the first to put into effect policy changes that ensured systems are to be designed with affordability considerations. Following the BBP initiative and the budgetary realities, Dr. Ashton Carter, then Under Secretary of Defense for Acquisition, Technology, and Logistics, issued the memorandum "Better Buying Power: Guidance for Obtaining Greater Efficiency and Productivity in Defense Spending" in 2010 to *target affordability and control cost growth*. Later in 2012, current Under Secretary of Defense Frank Kendall launched the "Better Buying Power 2.0" initiative, an update to the original BBP effort. These memorandums prescribed the following high-level guideline for considering affordability (Carter, 2010a, 2010b):

Mandate affordability as a requirement

- **At Milestone A set affordability target as a Key Performance Parameter**
- **At Milestone B establish engineering trades showing how each key design feature affects the target cost**

To understand the purpose of considering affordability at the milestones specified in this guideline, it is first imperative to understand the five phases within a typical defense program acquisition timeline shown in Figure 7-1. More importantly, it is of interest to determine at which points during the five phases tradespace-based methods can be most effective. These 5 phases are *Material Solution Analysis, Technology Development, Engineering and Manufacturing Development, Production and Deployment*, and finally *Operations and Support* (DAG, 2013). Understanding these five phases well can facilitate the incorporation of affordability considerations at the appropriate points between and during phases. Such an extensive accountability of cost and schedule parameters at every stage of the design lifecycle can help the defense industry mitigate the risk of program overruns and cancellations.

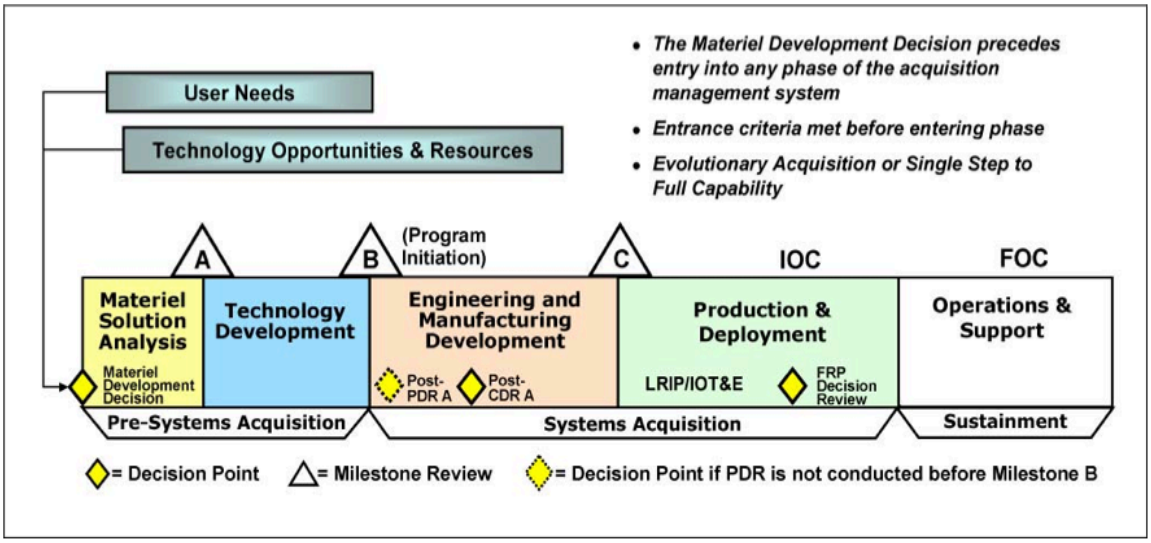


Figure 7-1: Milestones A, B, C and the 5 phases of a typical program acquisition timeline (DAG, 2013).

In the US DoD Defense Acquisition Framework, the **Materiel Solution Analysis Phase** begins with the identification of a military need and the assessment of potential materiel solutions to provide the required capability. In the defense industry, this need is first described in an approved Initial Capabilities Document (ICD), and an **Analysis of Alternatives (AoA)** will be conducted to explore all possible methods of meeting the identified requirement. The AoA will compare solutions based on their effectiveness, cost, schedule, concept of operations, overall risks, application of critical technologies, and their sensitivities to variation in assumptions or input variables.

The AoA in the Materiel Solution Analysis Phase can potentially be replaced with the designing for affordability process using tradespace-based methods, as its procedures are largely similar to the activities conducted in AoA. The military need described in the approved ICD is essentially the "mission statement" that will drive the design process. Practitioners of tradespace-based methods can conduct AoA in a systematic, disciplined and convenient manner by first brainstorming and then deciding upon the key design variables and attributes for formulating the tradespace model. Effectiveness, cost and schedule can all be considered holistically through the MAU and MAE functions that aggregate performance, cost and schedule attributes, while overall risks with regards to exceeding cost and time budgets can be accounted for through lognormal distributions with different variances applied to individual resource expense elements.

The application of critical technologies can then be considered through the use of epoch variables that combine to form different sets of fixed needs in different time periods. Since assumptions were also made in the formulation of the conceptual and computerized tradespace models, sensitivity analysis can also be performed to determine the utility and expense variations of preferred designs when assumptions or input variables are perturbed. The concept of operations can also be formulated in the design model and its consideration in the tradespace models largely depend on the nature of the system as well as the level of the design process. For example, the concept of operations at the FSS system level will be indirectly reflected by the payload size, data capacity and data usage rate, while the concept of operations at the program

level can be aggregated, disaggregated or mixed. Therefore, tradespace-based methods can be applied during AoA as a first-pass analysis and deliver high-level design solutions that are potentially feasible and affordable.

Milestone A is the review and approval stage at the end of this phase, where the materiel solution, cost estimates for the solutions identified in the AoA must be accepted by authorities before a program can be allowed to enter the acquisition system. As realistic cost estimates are required, it is necessary to propose affordable solutions early in the defense acquisition process. This prevents any programs with spiraling costs from entering the development phase and weighing down the entire defense portfolio in the long run.

Milestone A can be the review of results derived from tradespace exploration and analysis during AoA. Preferred solutions for system, program and portfolio levels have been identified and it is necessary for system architects, stakeholders and decision makers to review the design, performance and attribute vectors of preferred designs. This allows them to make critical decisions on the designs and determine whether they are feasible and affordable before being allowed to enter the acquisition system. Should any errors or discrepancies be found, Milestone A also serves as an opportunity to perform corrections and allow decision makers to even return to the AoA phase to reformulate tradespace models.

After passing Milestone A, the program enters the **Technology Development Phase**, during which technologies are developed, matured, and tested under simulated operational environments. Strategies for further capability development will be formulated and industrial contractors will compete to develop prototypes of a required system based on the approved affordable materiel solution.

Affordability considerations have little impact at this point as the preferred designs have already entered the acquisition system. However, tradespace exploration and analysis using different tradespace models can be conducted to determine the most effective and affordable methods of furthering capability development. Similarly, tradespace-based methods may be applied to help decision makers decide which contract to accept. The contract can be assumed to be analogous to a system with attributes such as program contract length, inclusion of options, length of buffer times between development phases etc.

Milestone B is placed at the end of the second phase, and a program is assessed based on its acquisition strategy, acquisition program baseline costs, contracting process and usage of mature technologies. This is done to mitigate any major weaknesses in program management and to include affordability considerations when performing engineering trades to meet target costs. This ensures that the overall architecture and design features of the system remain affordable relative to budget considerations before whole-scale integration and production.

Milestone B is another review and approval stage, where activities such as program or portfolio level analysis can be conducted to determine the acquisition strategy, acquisition program baseline costs, contracting process and usage of mature technologies. The FSS case study demonstrated the conduct of all these activities, where the development and retirement of seven satellite constellations was planned over the portfolio lifecycle. Program development cost, launch cost, operations cost, labor cost and retirement cost for each constellation were also calculated and tracked over their specific lifetimes. Program contract length was also considered

in the tradespace model and they determine the length of time over which installments for development and launch costs are to be paid. The usage of mature technologies is also accounted for through the switch in epochs over the portfolio lifecycle, where the TRLs of laser technology and RLV increase over time. Therefore, Milestone B presents a major opportunity for tradespace-based methods to be applied in order to find the most affordable development strategy for the system, program and portfolio of interest.

After passing Milestone B, the program enters the **Engineering and Manufacturing Development Phase**. This phase consists of two sub-stages: System Integration and System Demonstration. During system integration, full system integration is performed and the various subsystems that provide different capabilities are integrated into a single system. A development model or prototype is also produced. After being subjected to a Post-Preliminary Design Review (PDR) and Post-Critical Design Review (CDR) assessments, the program enters the system demonstration phase. Testing and evaluation will be performed on the model or prototype, which will be refined iteratively to ensure specified performance requirements are met.

After all testing and design corrections have been performed, the program passes **Milestone C** and enters the **Production and Deployment Phase**. The program will proceed to having a low-rate initial production, during which further quality control will be exercised. After completing operations testing and evaluation, the program can finally go into full rate production and enter the **Operations and Support Phase**.

Affordability analysis has little impacts during and between these last two phases as identified affordable systems, programs, and portfolios are already in the development pipeline. However, analysis can still be performed after the final phase to compare actual and expected levels of performance, cost and schedule. If there are few differences between actual and expected values, it can be concluded that affordability analysis has been successful to a large extent. This will increase confidence in the use of tradespace-based methods and other departments within the defense industry can do the same for their programs to sustain affordable acquisition practices. However, if there are major differences, they have to be well documented and the causes for these differences have to be identified. These differences reflect errors in the assumptions and mathematical equations in tradespace models, and corrections to these errors should be proposed. Verification and validation should also be performed again and more experts should also be involved to lend credibility to the further analysis. This allows the establishment of precedent cases upon which affordability analysis can be better performed in future.

After a review of these phases in the defense acquisition process, it can be seen that the Materiel Solution Analysis Phase, AoA, Milestone A and Milestone B are areas in the defense acquisition process where tradespace-based methods can be effective. However, it is also worth noting that once the physical construction of the defense system commences from the Technology Development phase onwards, it becomes increasingly difficult to reduce overhead costs or revert back to solutions that are relatively cheaper. This serves to highlight the need to include most or if not all the affordability considerations during early-phase design, particularly in the Material Solution Analysis Phase and Milestone A review. Conducting affordability analysis at Milestone B and beyond can only provide confirmation that the solutions being developed are within budget and on schedule, but are a lot less effective in remedying a development process that has become unaffordable. However, they can highlight what states the system should exist in to become

affordable and the system architects will have to find the most preferred change strategies to perform that transition.

Through explicit considerations of affordability early in and throughout key phases of the defense acquisition process, tradespace-based methods can facilitate the search for affordable materiel solutions that can provide the full operational capability with minimal cost and schedule overruns.

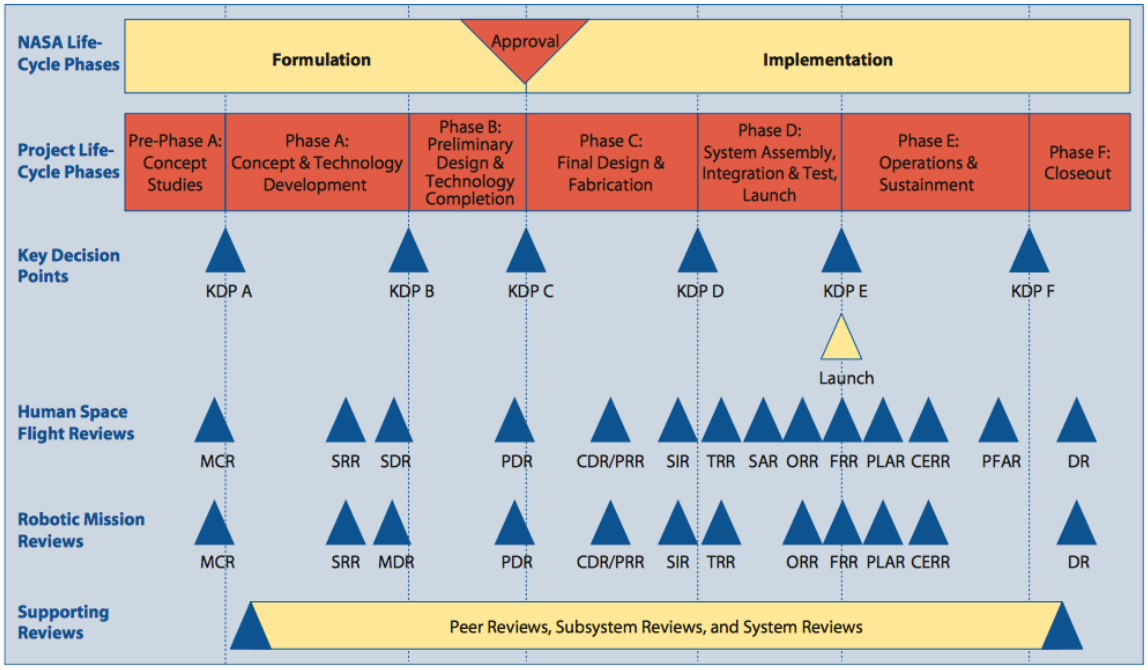
7.3.2 NASA Project Life Cycle and Trade Study Framework

The NASA project life cycle and trade study frameworks are just some of the many frameworks available in the aerospace industry that facilitate effective system design and management. Due to close working relationships with the defense industry, the NASA project life cycle is formulated in a manner similar to the defense acquisition framework as shown in Figure 7-2. In NASA, the project life cycle is decomposed into more manageable phases just like the defense acquisition framework (NASA, 2012). Similar to Milestones A and B, there are key decision points (KDP) between phases that provide managers with incremental visibility into the development progress. The phases of the project life cycle are:

- **Pre-Phase A**: Concept Studies (*Identify feasible alternatives*)
- **Phase A**: Concept and Technology Development (*Define the project and identify and initiate necessary technology*)
- **Phase B**: Preliminary Design and Technology Completion (*Establish a preliminary design and develop necessary technology*)
- **Phase C**: Final Design and Fabrication (*Complete the system design and build/code the components*)
- **Phase D**: System Assembly, Integration and Test, Launch (*Integrate components, and verify the system, prepare for operations, and launch*)
- **Phase E**: Operations and Sustainment (*Operate and maintain the system*)
- **Phase F**: Closeout (*Disposal of systems and analysis of data*)

It is clear that NASA Pre-Phase A is similar to the Materiel Solutions Analysis Phase for defense acquisition, where design alternatives are explored and investigated as in the AoA, while KPD A and KPD B are similar to Milestones A and B. Therefore, tradespace-based methods can also be deployed during these critical phases and decision points in order to ensure that preferred solutions entering the acquisition system are indeed feasible and affordable relative to stakeholder preferences and external constraints.

NASA has also produced many systems engineering guidelines over the years and many methods have been proposed to conduct various design activities such as sensitivity and risk analysis. However, the main process where potential enhancements can be made is the NASA trade study process shown in Figure 7-3 (NASA, 2012). The NASA trade study process begins with defining the system's goals, objectives, and the constraints it must meet. In the early phases of the project life cycle, the goals, objectives, and constraints are usually stated in general operational terms. Functional analysis is then performed, and it involves identifying, describing, and relating the functions a system must perform to fulfill its goals and objectives (NASA, 2012). As many of these steps are similar to conducting MATE for affordability analysis, the MATE procedure can potentially be used to replace the analytical portion of trade studies in the NASA process.



CDR	Critical Design Review
CERR	Critical Events Readiness Review
DR	Decommissioning Review
FRR	Flight Readiness Review
KDP	Key Decision Point
MCR	Mission Concept Review
MDR	Mission Definition Review
ORR	Operational Readiness Review
PDR	Preliminary Design Review
PFAR	Post-Flight Assessment Review
PIR	Program Implementation Review
PLAR	Post-Launch Assessment Review
PRR	Production Readiness Review
P/SDR	Program/System Definition Review
P/SRR	Program/System Requirements Review
PSR	Program Status Review
SAR	System Acceptance Review
SDR	System Definition Review
SIR	System Integration Review
SRR	System Requirements Review
TRR	Test Readiness Review

Figure 7-2: NASA Project Life Cycle (NASA, 2012)

The definition of goals and objectives is equivalent to the definition of the mission statement in MATE, while the definition of plausible alternatives is analogous to the selection of key architectural decisions for the system of interest. Functional analysis and the analytical portion of trade studies is where the remaining MATE activities can be conducted in. EEA should also be conducted to identify the possible futures in which the system can exist. Defining measures and measurement methods for system effectiveness, system performance and system cost is equivalent to the determination of key performance and cost attributes. As part of affordability analysis, schedule attributes should also be included in this step. The collection of data to support evaluation by selected measurement methods is then equivalent to the V&V process that is necessary for the formulation of all trade studies.

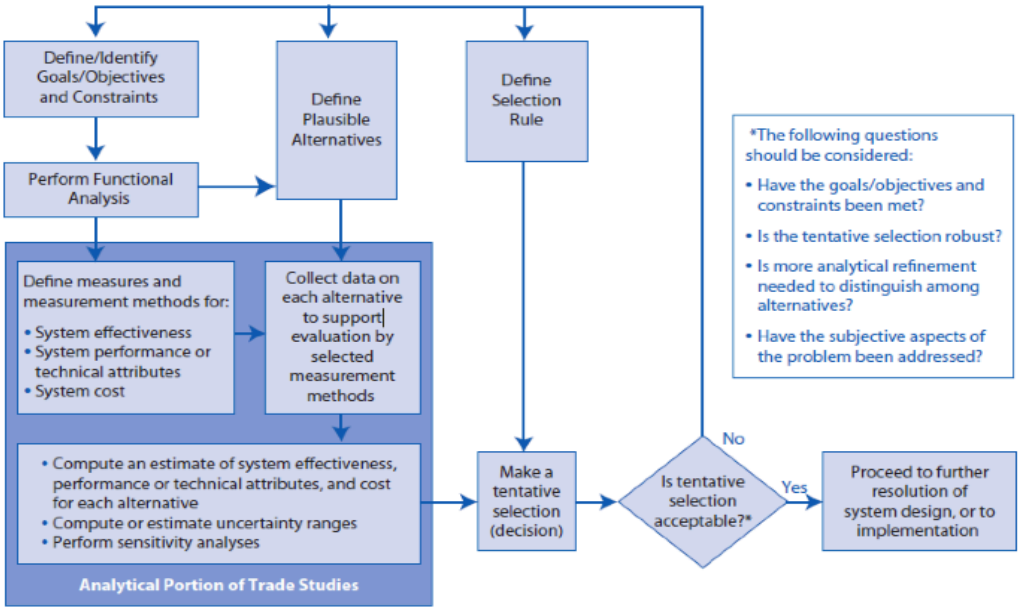


Figure 7-3: NASA Trade Study Process (NASA, 2012)

In computing estimates of system effectiveness, performance attributes, and cost attributes, MAU and MAE functions can be applied to aggregate all considered attributes into utility and expense metrics that parameterize the system tradespace. However, prior to this step, there has to be the elicitation of stakeholder preferences and their translation to single attribute utility and expense levels. The computation and estimation of uncertainty ranges may include the application of various probability distributions to account for uncertainties in different attributes. While lognormal distributions can be applied to cost and schedule elements to yield a confidence interval for aggregate expense, other distributions such as normal or triangular distributions may also be applied to the performance attributes to determine another confidence interval for aggregate utility. The choice of probability distributions is left to the system architects and stakeholders to agree upon. Finally, the making of a tentative selection is equivalent to the identification of the Pareto front and selection of preferred designs. Therefore, MATE is closely mirrored to the NASA trade study process, with the addition of stakeholder preference elicitation. MATE can thus be used in place of the NASA trade study process so as to ensure that selected designs for new space systems are not only technically feasible, but also within budget and likely to be delivered on time.

7.3.3 Affordability Engineering Framework with Tradespace-based Methods

While enhancements can be made to existing frameworks, new frameworks can also be introduced to guarantee that affordability considerations are omnipresent through the design process. One possible option is the **Affordability Engineering Framework (AEF)** proposed by the MITRE Corporation, which has been designed specifically to help the DoD respond to imminent fiscal realities and advance the practice of affordability engineering to improve acquisition program success (MITRE, 2013). Integration of the AEF into the DoD acquisition framework is already in the pipeline and the AEF will be piloted and migrated across selected DoD programs over the next few months for implementation with iterative evaluation and

development. The AEF is not restricted to defense systems, and it can also be applied to aerospace acquisitions across their project lifecycles. An even better approach can be the integration of tradespace-based methods for affordability analysis with the AEF. Tradespace-based methods can facilitate the conduct of a first-pass analysis to deliver high-level solutions for implementation; AEF can be used thereafter to assess the affordability status of preferred program and portfolio designs, as well as allow room taking corrective actions. Therefore, both can be used alongside each other to provide greater insights into design principles with more awareness of the risks posed to performance, cost and schedule attributes.

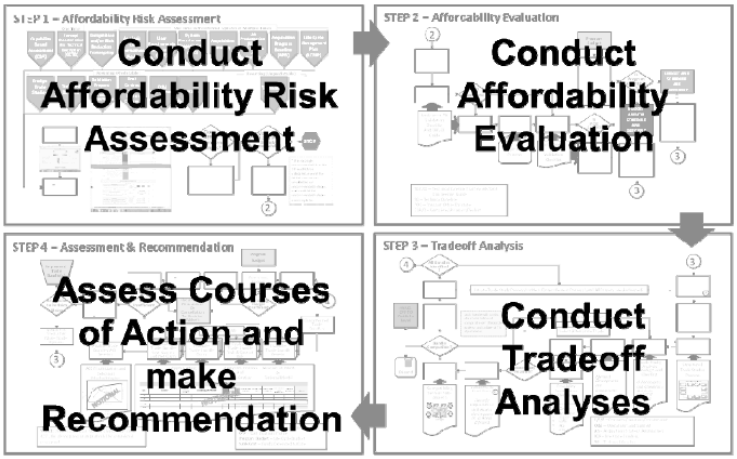


Figure 7-4: Affordability Engineering Framework (MITRE, 2013)

As described by MITRE (2013), the AEF is a structured, actionable approach with tools and techniques to address affordability challenges throughout the life cycle. The AEF uses multi-disciplinary teams to quantitatively evaluate program affordability while identifying integrated cost, schedule, and performance tradespaces. As such, the nature of AEF is very similar to that of affordability analysis using tradespace-based methods, which can potentially become one of the key enablers of the AEF if integrated seamlessly.

The AEF includes four steps:

(1) *Conduct affordability risk assessment*
(2) *Conduct affordability evaluation*
(3) *Conduct tradeoff analyses*
(4) *Assess courses of action and make recommendations*

Step 1 is a qualitative assessment of the program affordability risk and it is accomplished through questionnaire templates on proprietary software tools. Each template contains unique assessment questions that address affordability risk indicators contained in a program's technical baseline. For each question, risk levels such as high, medium, low, unknown, or not applicable, and unique risk-level definitions are provided for each trigger question. Upon selecting a risk level, question-specific recommendations for possible corrective or mitigation actions are provided. An assessment tally is provided upon completion of the assessment and the results can yield evidence of program risks, thereby indicating the state of the program's affordability status. After performing the recommended changes, the technical baseline as well as cost and schedule estimates will be validated (MITRE, 2013).

It is also at this step where tradespace-based methods can first be deployed. The risk levels can pertain to every attribute and this will help in the calculation of upper and lower bounds for utility and expense levels of preferred designs. Additional questions can also be added to the same questionnaire templates to elicit stakeholder preferences on every attribute and variable in addition to their perceived risk levels. However, potential drawbacks of this integration will be the lengthiness of this qualitative assessment since a large number of questions have to be answered. Also, the credibility of these results is highly dependent on the availability of subject matters experts and decision makers, who will be able to make more prudent assessments of various risks levels.

Step 2 is the evaluation of a program's affordability status in a quantitative manner. In the AEF, it is accomplished through an intensive validation process using known parameters from validation checklists and program budgets. Program elements such as risk, cost, schedule, and requirements management and coordination with users and other active stakeholders will be tightly integrated to ensure that tradeoffs among these elements are credible and can potentially result in more affordable solutions. It is also in this step where affordability risk trends can be revealed in more detail. These trends have underlying causes or interaction effects with identified risk factors that might hinder the program in delivering affordable and effective capabilities. Therefore, the identification of all such trends through the AEF can prompt the stakeholders to seek the corrective actions needed as well as the frequency with which they should be applied. The validation of the technical baselines and program estimates can also reveal key performance, cost and schedule drivers of the program. As these elements have a strong influence on the feasibility and affordability of the program, more trades among these design drivers can be performed. The procedures to be taken in Step 2 are shown in Figure 7-5.

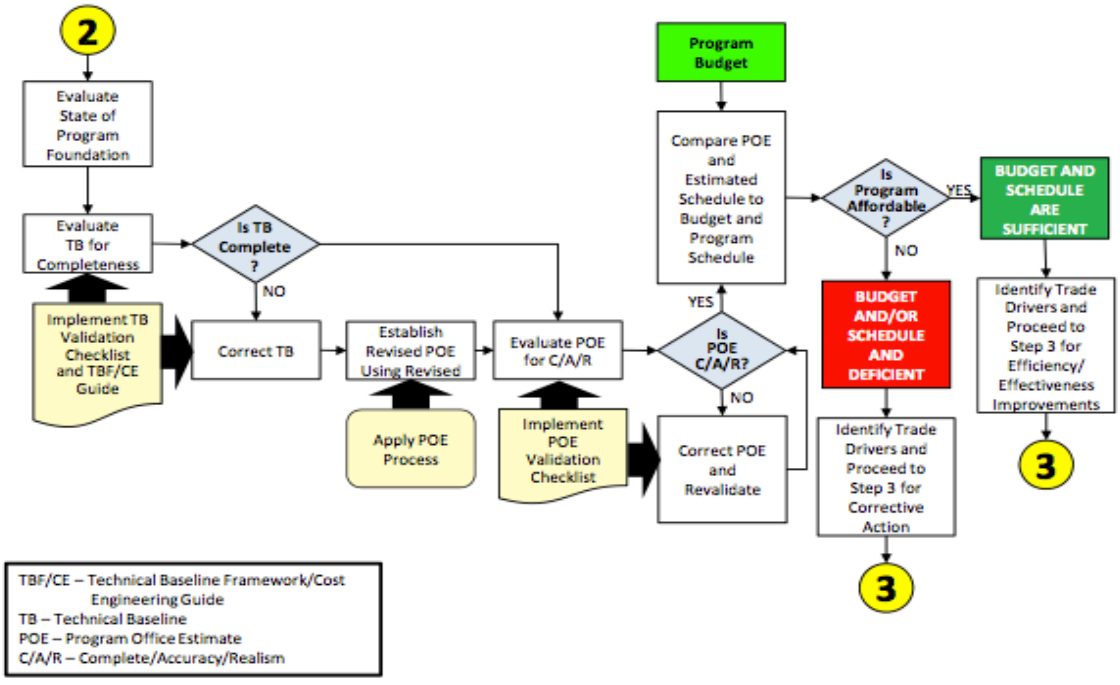


Figure 7-5: AEF Step 2 – Affordability Evaluation (MITRE, 2013)

Tradespace-based methods can be applied in parallel with Step 2. At the beginning, MATE for affordability analysis can be conducted and the tradespace models for the system, program and portfolio should be formulated. Program estimates can be calculated using these models. These models will then have to be validated and this validation procedure can be provided through the AEF, which requires the comparison of program estimates against technical baseline validation checklists, actual program budgets, and program schedules. The tight integration of all program elements can then be achieved through the use of MAU and MAE metrics that allow holistic comparisons to be made among design points on a tradespace.

The subsequent procedure of determining whether a program is affordable or not can be conducted through further tradespace analysis after preferred designs have been identified. As demonstrated in the FSS case study, comparing the cost profiles of individual constellations and the entire portfolio of constellations over their respective operating lifecycles against actual budgets can help determine whether a program is fully or partially affordable. Therefore, similar cost profiles over time for the program of interest can be assessed to determine if its budget and schedule remain sufficient. If determined to be insufficient, further analysis of the program tradespace has to be conducted to identify trade drivers that may allow for corrective action. If determined to sufficient, trade drivers should still be identified so that potential enhancements can be made. The AEF can then proceed to Step 3.

Step 3 is designed to develop and conduct structured tradeoff analyses for either correction action or enhancements to preferred solutions. It aims to evaluate and select among design variables, performance attributes, and expense attributes to achieve the desired capabilities within cost and schedule objectives. Through Step 2, key design drivers were identified and they are used to perform these tradeoffs for better affordability. If the program is determined to be affordable in Step 2, trade opportunities for achieving resource expense savings or performance enhancements can be analyzed and implemented. If the program is determined to be unaffordable, then reduction trades on performance, cost and schedule attributes will be performed to allow the program budget to be sufficient. By the end of Step 3, all preferred trade bundles would have been identified. The procedures to be taken in Step 3 are shown in Figure 7-6.

Step 4 is then the final step where tradeoff bundles that deliver the capabilities that the end user needs within the established budget and time line are selected efficiently. The tradeoff bundles will be assessed according to their benefits, risks, costs, and schedule impacts measured. Finally, recommendations are made to select the tradeoff bundles that help meet affordability goals efficiently for the preferred designs. These trades can be conducted multiple times using multiple methods until stakeholders are satisfied with the final state of the preferred solution.

Similarly, tradespace-based methods can be used in concert with Steps 3 and 4. Such trades can be performed on single points, groups of points, Pareto frontier set solutions, or even the entire tradespace. This can be part of initiating the trade study process in the AEF. Tradeoffs can be performed in multiple ways and multiple times, where a set of consistent evaluation criteria has to be used to assess the marginal benefit or loss as a result of the trades. When stakeholders are satisfied after conducting all possible trades, they will then decide upon the most preferred trades to conduct so as to obtain a refined solution. Application of the AEF will continue through the remaining procedure blocks in Steps 3 and 4 until formal implementation of the preferred design together with the preferred trade bundles commences.

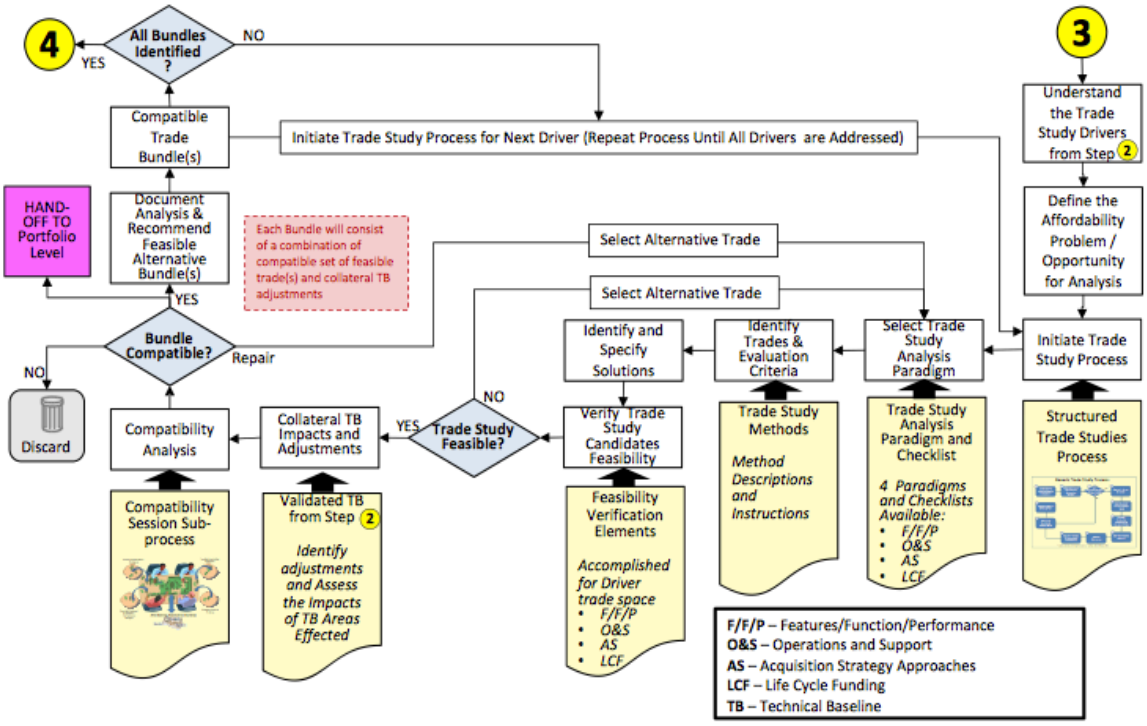


Figure 7-6: AEF Step 3 – Tradeoff Analyses (MITRE, 2013)

MITRE (2013) recommends the AEF process to be conducted throughout the life cycle and initiated via "trigger" point, which may include periods of major program changes, budget preparation and submittal, and existing regulatory and statutory requirements for affordability certification. A typical program profile with trigger points is depicted in Figure 7-7, which is based on the DoD acquisition framework. Numbers along the program lifecycle timeline denote the trigger points and their related activities.

Instead of being limited to a few milestones or FDPs, the AEF can provides a significant increase in the number of affordability analyses using tradespace-based methods relative to current requirements as there would be four or more before Milestone A and seven or more prior to Milestone B (MITRE 2013). The increase in frequency as a result of the coupled usage of AEF and tradespace-based methods can thus provide stronger coherency in results from assessment to assessment and assists in institutionalizing the importance of affordability.

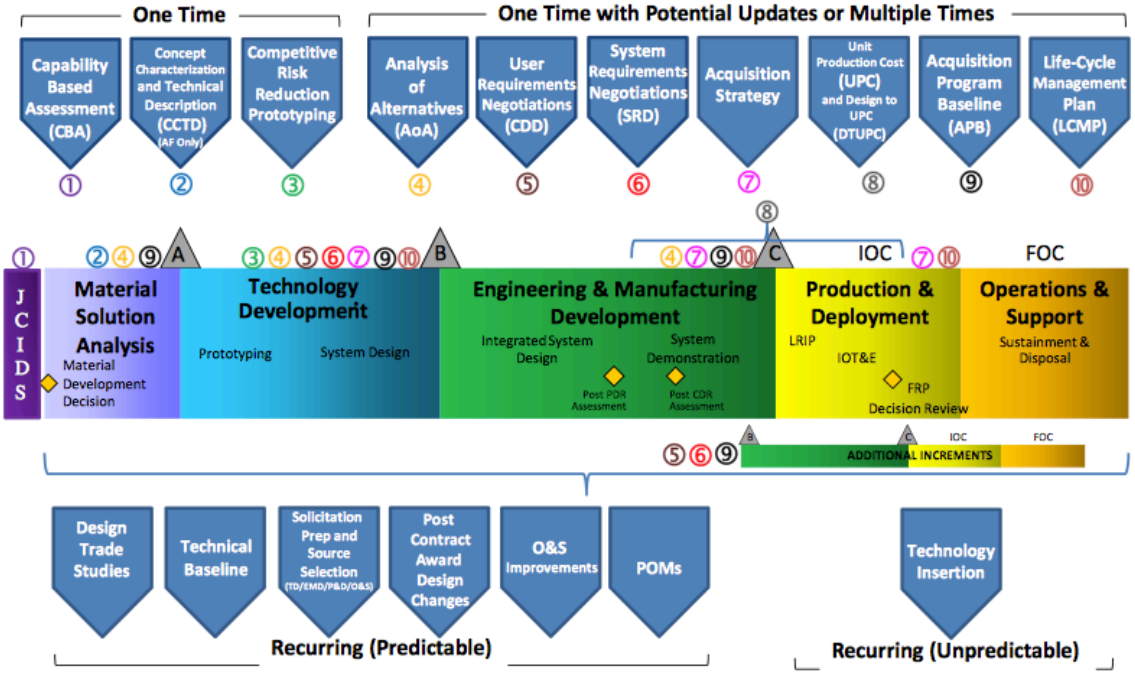


Figure 7-7: AEF Program Trigger Points (MITRE, 2013)

In summary, tradespace-based methods can offer numerous benefits to affordability analysis and they can be applied to current practices in the defense and aerospace industry in a number of ways. A preference towards evolutionary rather revolutionary approaches is taken and tradespace-based methods are to be used in a manner complementary to existing acquisition frameworks. Frameworks from the US DoD and NASA are analyzed to determine areas in which these methods are most applicable and effective. It was determined these methods would be most applicable to the first phase of both acquisition frameworks as well as the first two major decision points. If new frameworks are to be proposed, the AEF by the MITRE Corporation, coupled with tradespace-based methods, can provide a rigorous approach for achieving program affordability.

The AEF with tradespace-based methods is a multi-step process that first qualitatively and quantitatively assesses program affordability risk and stakeholder preferences, then developing and validating tradespace models that enable the generation and identification of preferred solutions, then developing a set of targeted tradeoffs that are bundled for either correction or enhancement purposes, and finally recommending for implementation. Application of the AEF can increase the frequency of affordability analyses and enable the holistic consideration of performance, cost and schedule throughout the design process. This can increase the probability of system, program and portfolio success in the face of declining budgets and mission uncertainties, and provide decision makers with data-driven rationale for program design and trade bundle recommendations.

7.4 Affordability through Effective Policies and Management

This thesis has so far discussed the design of affordable systems, programs and portfolios from an engineering perspective, where tradespace-based methods can be used in the design for affordability process to deliver greater value to stakeholders over time. In answering the third question posed in this chapter, there are other approaches beyond the scope of engineering design and analysis that can be taken to ensure affordability. It is first important to recall that the whole paradigm of affordability emerged through the implementation of the BBP initiative as well as the issuance of defense memorandums and government directives. A single policy guideline of "Mandate Affordability as a Requirement" was sufficient to trigger an avalanche of changes in design methodologies throughout industry and academia . Therefore, it can be seen that a single policy change can potentially impact the uptake of new methods and frameworks for affordability analysis by the entire defense and aerospace community. Therefore, affordability can also be achieved through effective polices and management. Many ways of doing so have been found across a variety of literature sources but the most relevant ones are described in this section.

7.4.1 Implementing New Systems Engineering and Acquisition Policies

It was earlier stated the use of tradespace-based methods and the AEF are meant to be evolutionary rather revolutionary in nature, as the upheaval of current acquisition frameworks would be difficult. This can be attributed to the "dead hand of policy", which results in high inertia for any changes to be made to the current establishment. However, the widespread acceptance of these methods and frameworks are more likely to occur if a government policy or directive is issued regarding the recommendation of their application. If the policy or directive were to make the use of tradespace-based methods or the AEF compulsory in all system design problems in the defense and aerospace industry, the uptake of these new methods would be rapid. As such, only through implementing new systems engineering and acquisition policies can a massive upheaval of acquisition frameworks be possible. Also, this can help establish the use of tradespace-based methods for affordability analysis as the new standard for design. It is also through government policies that all enterprises in the industry can conduct the design for affordability using the same approach and the same definition of affordability. Therefore, new government policies and directives would be the most effective approach to ensuring that affordable systems can be designed in future.

However, implementing a new policy is not a trivial problem. Given that current acquisition frameworks are deeply rooted within the industry, there will be many obstacles encountered in convincing key decision makers within the government about the value of replacing an obsolete or even non-existent policy with a new one. A cost-benefit analysis of the policy change will have to be performed and given the foreseen scale of change throughout the industry, there will be high opportunity costs involved when established enterprises have to spend more time and money training their workforce to be adept in these new methods. Also, the effectiveness of these methods on a large scale has yet to be assessed and this incurs a lot of risks on the stakeholders, who are ultimately the beneficiaries of the new system being designed. If such a policy is to be proposed, it would be highly recommended that changes are made progressively with multiple checkpoints to assess the actual effectiveness and experienced problems in using the new

methods and frameworks. Larger, more influential government agencies and aerospace companies can potentially take the lead in adhering to this new policy shift.

7.4.2 Managing Multiple Stakeholders

Many design problems in the defense and aerospace industry requires the collaboration of multiple stakeholders, such as the government, prime contractors, and suppliers. Some problems such as the design of the FSS portfolio can even require multi-stakeholder collaboration to be conducted across numerous international boundaries. Given the participation of many stakeholders and an impeding clash of design cultures, there is high potential for conflicts of interest to occur (Fitzgerald and Ross, 2014). This may potentially cripple the overall design process, thereby rendering systems unaffordable and unavailable. Therefore, it is important to be able to effectively manage the wide range of interests of multiple stakeholders and devise solutions that are collectively perceived to be acceptable. Multi-stakeholder negotiation has also become an open area of research within the systems engineering community. Having a central management authority is one way of doing so, as an overall governing body can help better organize stakeholder involvements through the design process. It helps to keep participating stakeholders focused on the overall design objective, and not fixated on their personal interests. Therefore, a central management authority like the FSS Directorate will be better positioned to manage different interests and coordinate resulting initiatives.

7.4.3 Better Communication and Work Flow Organization

Owing to the complex and dynamic nature of many systems, there are myriad flows of information and people involved throughout the design process. Even with the aid of new methods and frameworks for implementing affordability, poor communication and work flow organization among participants can seriously inhibit progress in development. Systems and programs can become unaffordable simply due to delays resulting from ineffective communication. To ensure better communication and work flow, program managers can be given greater authority, so that they can take direct control or at least assert a strong influence over tradeoffs among research and development activities, acquisition, operating, and support costs.

Program managers can control development within given cost and time budgets and they should actively seek measures to reduce lifecycle costs of their systems. Program managers also have the responsibility of communicating policy guidelines and best practices to everyone in the design chain so that the entire workforce is working in a disciplined and coherent manner. In terms of work flow organization, they should also be constantly performing updates and checks on system development at specific milestones or decision points to assess the affordability status of the system. Another way of establishing better communications is through the continuation of partnerships among users, developers and other intermediate participants so as to derive better value designs for the available resources. With familiarity and better working relationships, information and people can flow with more ease and clarity.

7.4.4 Integrating Practices from Industry and Academia

Major players in the defense and aerospace industry have devised their own methods of reducing cost and schedule overruns. While there are many efficient practitioners of current acquisition methods, they are not well acquainted with emerging practices originating from academia. These

new practices can potentially facilitate the design of more affordance systems and they should be made known to experienced practitioners quickly to reduce the incidence of cost and schedule overruns. Industry players can directly involve academia in the design process or organize conferences that allow the free exchange of information. In this way, both industry and academia can learn from each other.

7.4.5 Continuously Educate and Train the Acquisition Workforce

Following from the previous point, more investment is needed to continuously educate and train the acquisition workforce, and ensure that acquisition practitioners remain up to date with both current and emerging affordability practices. This ensures that the integration of practices from industry and academia can be completed through the consistent transfer of new knowledge. In this manner, every participant in the design process can embrace the strategic purpose and vision of affordability practices through mastery of affordability analysis techniques and policy guidelines. Revamping the workforce can thus institutionalize cultural changes that ultimately guide the defense and aerospace industry towards having the same conceptions towards affordability.

7.5 Summary

This chapter sought to answer three key questions pertaining to the role and relevance of tradespace-based methods to engineering design within the context of the defense and aerospace industry.

The first question is with regards to the implementation issues that exist with tradespace-based methods. Like other methods currently in existence, tradespace-based methods are not perfect and they have inherent drawbacks that must be recognized and managed by system architects in order to ensure the integrity of the design for affordability process. Generally, tradespace-based methods offer a systematic, disciplined and convenient approach to design. The system-program-portfolio framework allows the design process to be layered and scalable. More importantly, they can overcome many limitations found in many existing methods. However, they also have potential drawbacks and liabilities such as the subjectivity introduced through the usage of stakeholder preferences and multi-attribute utility. Utility and expense metrics are also not as intuitive as traditional measures of performance and cost. Ultimately, tradespace-based methods are at best close representations of reality and they are better used to confirm heuristics rather than proofs. Therefore, it is imperative to apply tradespace-based methods properly in order to obtain best results from affordability analysis.

This then leads to the second question of when and where tradespace-based methods should be applied for them to be most effective. Tradespace-based methods can be used in a manner complementary to existing acquisition frameworks such as those from the US DoD and NASA. If new frameworks can be proposed, the AEF by the MITRE Corporation, coupled with tradespace-based methods, can provide a rigorous approach for achieving program affordability. Application of the AEF can potentially increase the frequency of affordability analyses and enable the holistic consideration of performance, cost and schedule throughout the design process.

Finally, the third question is about the approaches beyond engineering design and analysis that can be taken to resolve the affordability issue. Some approaches include making policy changes, managing multiple stakeholders, better communication and work flow organization, better integration between industry and academia, and the continued education and training of the workforce. Application of these approaches in the future can produce positive impacts on the future of design, as this ensures that affordability principles can propagate throughout wider industry practice and management philosophy. In doing so, every system, program or portfolio that is being designed can become and remain affordable over time.

8 CONCLUSION & FUTURE WORK

8.1 General Conclusion

Program management failures have plagued the defense and aerospace industry for decades, as unanticipated cost and schedule overruns have rendered the development of systems ineffective in terms of both time and cost considerations. This raises the need to holistically include *performance, cost and schedule* considerations during the *early-phase design* of systems in order to perform useful tradeoffs to derive more feasible and affordable solutions. This paradigm is the *design for affordability* and it is the principle that motivates this research. Specifically, this design conundrum is targeted at *defense and aerospace* systems, which have complex mission requirements and stakeholder involvement that are susceptible to changes and perturbations over time. In this research, *tradespace-based methods* are introduced to incorporate these holistic considerations into the design process and facilitate the progressive and disciplined search for affordable solutions to the system, program and portfolio of interest. A potential method for affordability analysis has been identified and its feasibility has been demonstrated through application to the Space Tug and FSS design case studies. Its benefits and limitations were also assessed. This method can also be integrated with current acquisition frameworks and other affordability methods to reduce the occurrence of cost and schedule overruns in future. Strategies for policy change and management were also outlined so that they can complement the tradespace-based methods to better design for affordability.

8.2 Research Contributions

This section aims to summarize the research contributions by matching sets of research work that have answered the four key research questions.

1. *What is affordability in the context of defense and aerospace systems?*

Affordability is defined as the property of becoming or remaining feasible relative to resource needs and resource constraints over time. **Affordability can be treated as an ility that drives the design of more affordable yet technically sound architectures.**

Affordability can be applied to any entity, be it a system or a program or portfolio, with the bigger entities warranting a higher degree of complexity and more attributes and resources for consideration. A resource may be defined as the aggregation of cost, schedule and other non-monetary factors necessary for architecting, development and operation. Resource needs are the set of resource requirements elicited from stakeholders, and resource constraints are the statements of restrictions on these requirements that limit the range of feasible solutions.

In defining affordability, a comprehensive review of current affordability research and practices in academia, industry and government was first conducted. A repertoire of journal articles,

conference papers, industry reports, government documents, theses and books were then reviewed to obtain a holistic understanding of the approaches taken to integrate the affordability into existing system engineering frameworks and acquisition practices.

A lexicographic analysis of the term *affordability* was performed, and its increasing relevance and usage in daily applications to the engineering of complex systems were also tracked. The scope of affordability studies was narrowed down to the defense and aerospace industry, where the construction of complex systems, programs and portfolios is immensely challenging and they necessitate the added consideration of cost and time elements in the design process.

Current definitions for affordability within the industry were collated and overlapping themes were identified. Apart from cost, schedule and performance, other major themes that were identified include value, systems engineering, attributes, constraints, mission needs, risks, lifecycle, budget and strategy. Existing frameworks for understanding affordability and conducting affordability analysis were also described and reviewed. Their disadvantages were identified and they include the lack of time centricity, failure to recognize complexities of scale, lack of cost breakdown structures, treating affordability as a constraint rather than a requirement, and a lack of a value-centric perspective. After aggregating the state of current practice and analyzing the drawbacks of existing methods, it was inferred that systems are currently not designed with explicit affordability considerations. It was also concluded that better methods for affordability analysis must build upon the strengths of existing frameworks and without their weaknesses. Therefore, new methods introduced should cover the major themes in affordability analysis and address its limitations.

2. *How can affordability be incorporated in early-phase engineering design?*

Affordability considerations can be incorporated in early-phase design using tradespace-based methods such as Multi-Attribute Tradespace Exploration and Epoch-Era Analysis. The notion of total lifecycle cost can also be replaced with the aggregate measure of Multi-Attribute Expense. These methods can be applied to the progressive design of systems, programs and portfolios using the bottom-up or top-down approach. The feasibility of these methods was demonstrated in the Space Tug and FSS design case studies.

To conduct affordability analysis and perform affordability tradeoffs in an informed manner, systems engineering methods have to demonstrate changes in resource expenses as major decision parameters and times to completion are varied. The minimization of resource expenses, while maintaining or increasing performance specifications across changing contexts over time, motivates the construction of tradespaces with considerations of temporality. Affordability analysis can thus be conducted through *tradespace exploration*.

Tradespace exploration allows a holistic consideration of capabilities and mission utility during early-phase design, instead of being locked too early into requirements and key performance parameters. As tradespace exploration entails the enumeration and evaluation of a large number of potential designs, this method is most relevant to the design of complex systems with multiple dimensions of benefits and expenses. The use of tradespaces instead of simple tradeoffs of several point designs can lead to better lifecycle results.

As tradespace exploration enables the promulgation of affordability as an ility, tradespace-based methods are introduced for designing for affordability. *Multi-Attribute Tradespace Exploration (MATE)* can be used in the value-driven search for affordable designs by aggregating multiple dimensions of benefits into a single utility metric. Tradespaces have been traditionally viewed as two-dimensional plots bounded by the parameters of utility and costs. Since the design process considers more than just cost, cost can be replaced with an aggregate measure for resource expenses to enable affordability analysis. The measure is *Multi-Attribute Expense (MAE)* and it aggregates cost, schedule and other non-monetary factors into a single expense metric. Design points on a tradespace can then be compared using their unique MAE and MAU metrics, which are representative of all performance and resource expense attributes, as well as their corresponding stakeholder preferences. Therefore, MATE can be modified for the purposes of affordability as shown in Figure 8-1.

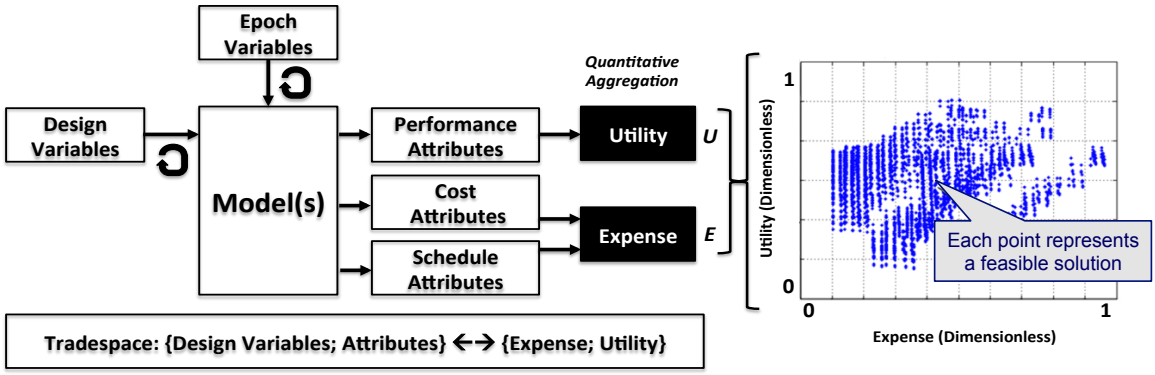


Figure 8-1: Data flow for the Tradespace Exploration process in Affordability Analysis (adapted from Ross and Hastings, 2005).

Constraint levels for each attribute are also determined and aggregated to form MAE and MAU constraint levels for the tradespace in each epoch. The area bounded by the maximum MAE, minimum MAU, and derived expected minimum MAE is then the *affordable solution region*, in which the most affordable solutions are most likely to be located. By searching in the narrowed affordable solution regions in every tradespace for every epoch, affordable solutions for systems, programs and portfolios can be found. This concept was demonstrated in Figure 8-2.

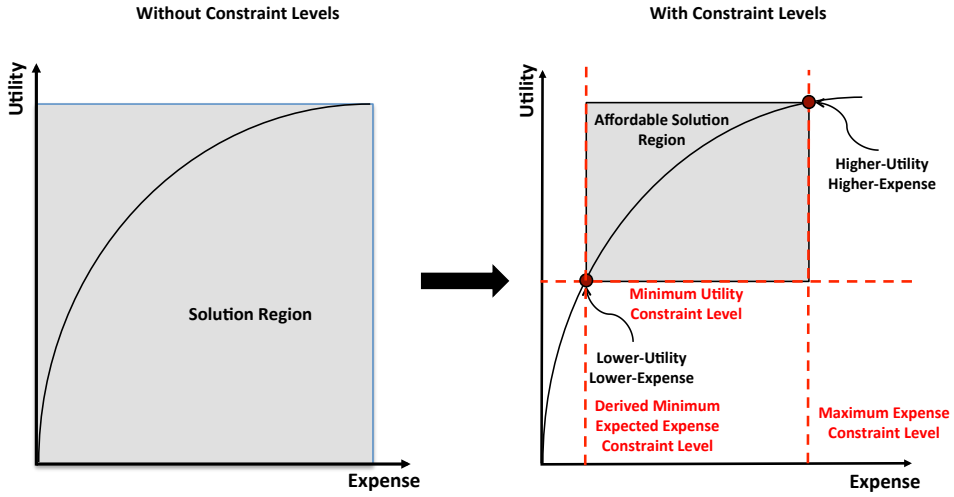


Figure 8-2: Narrowing the affordable solution space using external constraint levels for a fixed context.

Lognormal distributions with different variances can then be applied to individual expense elements to account for uncertainty that grows over time. In the FSS case study, a 99% confidence interval was established on the expense values of Pareto frontier set solutions in every system tradespace. This yields a longer upper-tail and a shorter lower-tail that collectively account for expense variability. Depending on the extent to which the upper-tail is within the affordable solution, three preferred Pareto optimal solutions can be labeled as "High", "Medium", or "Low" risk. High-risk designs have better performance attributes, but are at the highest risk of exceeding time and budgetary constraints. A sample of this application is shown in Figure 8-3, where high-, medium, and low-risk designs are colored in black, blue and cyan respectively. As such, this is one of the ways in which time and budgetary uncertainties can be accounted for during system level analysis and Single-Epoch Analysis.

FSS Case Study: Phase 1 – Epoch 1: Mission 0 in Context 0-0

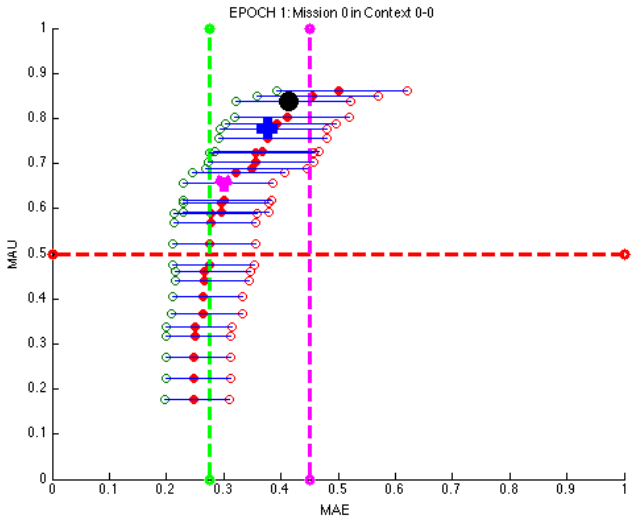


Figure 8-3: FSS Case Study: Epoch 1 Pareto front design points with 99% MAE confidence intervals

Epoch-Era Analysis (EEA) can then be used to account for how the performance, cost and schedule attributes of a system or program evolve over time across dynamic operating environments. EEA is a design approach used to clarify the impacts of time and context on the value of the system or a program, and can be modified and applied to enable affordability analysis over multiple *epochs* (periods of fixed contexts) and multiple *eras* (ordered sequences of epochs). Trajectories that track the changes in utility and expense of a system, program or portfolio can be plotted to determine which solutions are fully affordable or partially affordable across a period of time. It was evident during the case study that a program level analysis is more complex than that of a system due to more design variables, performance attributes and expense attributes. This complexity can only increase when the design for affordability process cascades down to the portfolio level.

CONSTELLATION 1 – Developed in Epoch 1: Mission 0 in Context 0-0

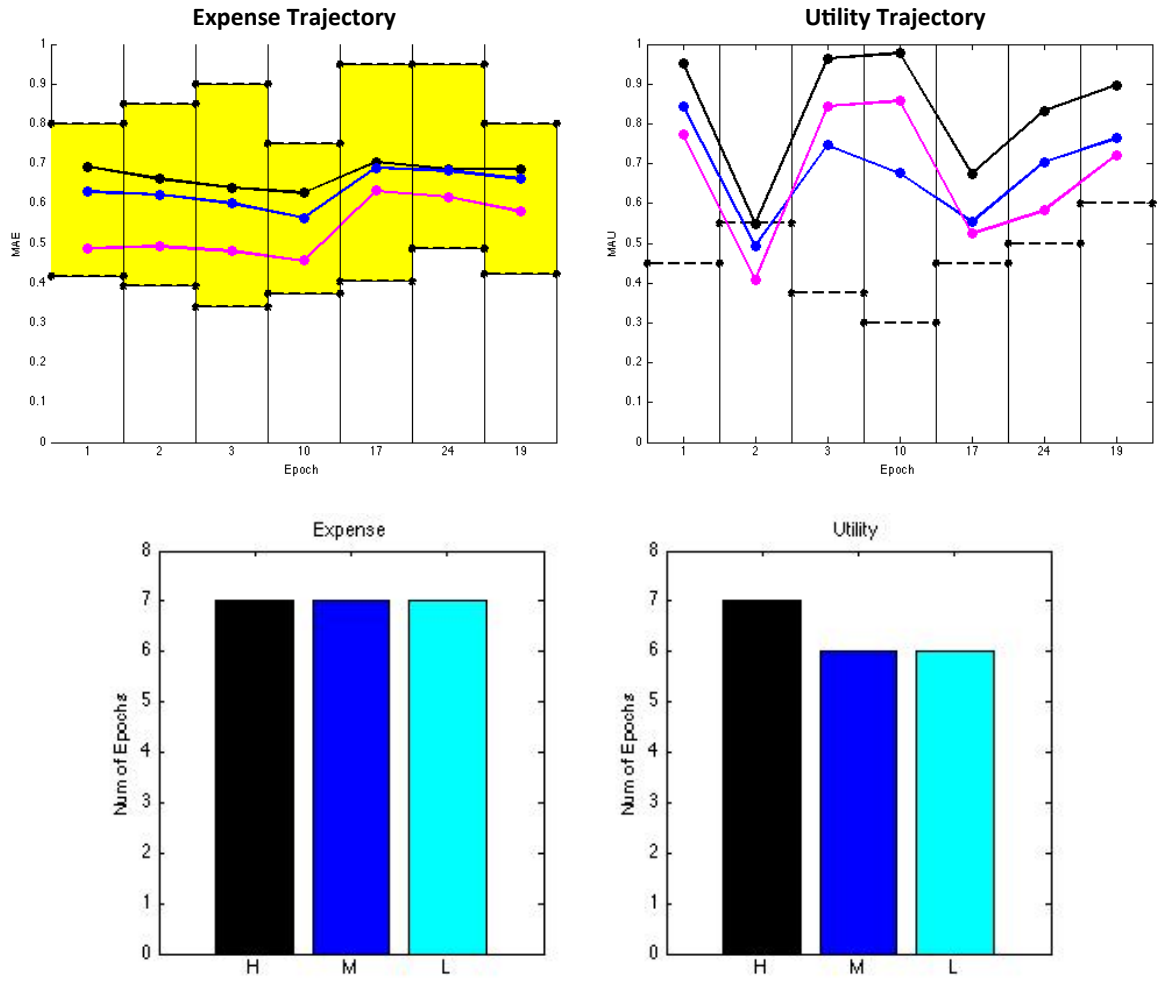


Figure 8-4: FSS case study: Expense trajectory plot, Utility trajectory plot, and Histogram of number of epochs in affordable solution region for each design for Constellation 1 – Epoch 1: Mission 0 in Context 0-0

In the FSS case study, the three preferred system solutions derived in system level analysis were used as design variables for program level analysis, during which two or three preferred program solutions from each tradespace were selected using the same methods as before. However, at program level analysis, it has become critical to find program solutions that remain largely affordable and feasible across multiple epochs or an era. This necessitates the conduct of Multi-Epoch and Single-Era Analysis. This was performed through the plotting of utility and expense trajectories of preferred program solutions across the epochs or era of interest, as well as plotting histograms that indicate the number of epochs in which a particular program remains affordable or feasible. A sample of this program level application is shown in Figure 8-4.

Once affordable program solutions have been found, they can be used as design variables for portfolio level analysis to generate the portfolio tradespace. A number of portfolios can then be picked to further investigate changes in their performance, cost and schedule profiles over their designed lifetimes. In the FSS case study, the cost and performance profiles, as well as the satellite population and performance profiles, were obtained for individual constellations and entire portfolios. Cost profiles for entire portfolios can reveal any breaches in portfolio budgets at any time during the portfolio lifecycle. Should variations in annual total cost commitments remain below the portfolio budget and performance levels are at acceptable levels, an affordable portfolio solution has been found. At the end of the FSS case study, Portfolio 19 was found to be a desirable and affordable solution. A portfolio development strategy can then be potentially formulated based on its constituent program and system level design vectors. The profiles for Portfolio 19 are shown in Figures 8-4 and 8-5.

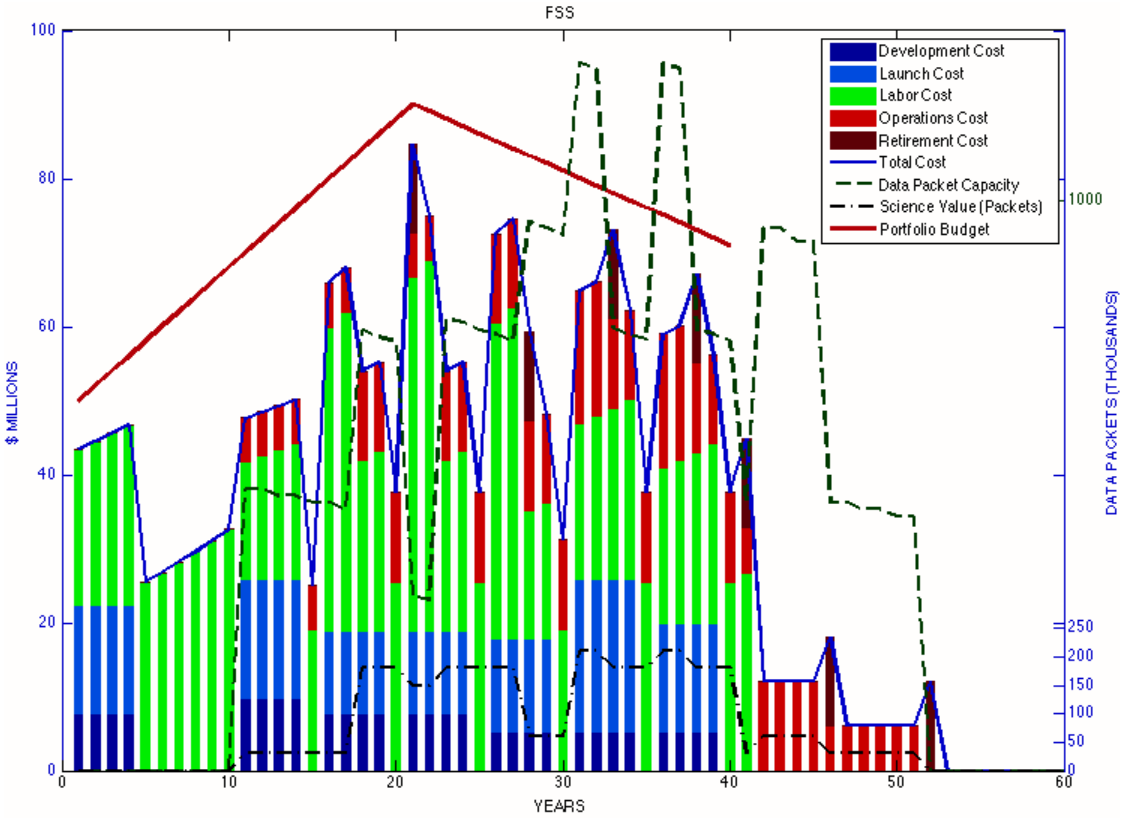


Figure 8-5: Cost and performance profile of Portfolio 19

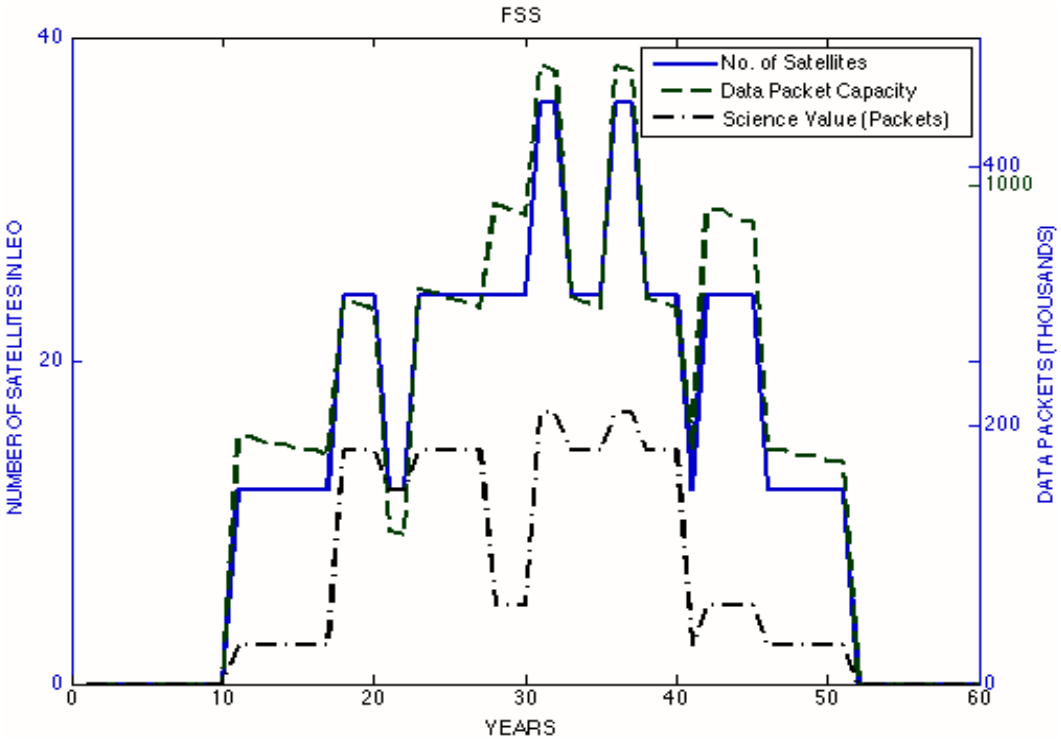


Figure 8-6: Satellite and performance profile of Portfolio 19

Generally, systems, programs and portfolios in the real world are much more complex, and there are many considerations to be taken into account during the design process. As such, a more convenient way to proceed forward from system to portfolio will be to down-select a handful of preferred designs at the end of each level, which will become design inputs at the next level of analysis. This has been demonstrated in the FSS case study. This narrows down the number of possible designs for systems, programs and portfolios, enabling a more focused and easy path for tradespace exploration. In the design of the FSS, a bottom-up approach was taken and separate affordability analyses for the system, program and portfolio were conducted sequentially. Results from this case study culminated in producing an overall design and development strategy for all satellite constellations making up the FSS portfolio. The FSS to be designed based on the results of affordability analysis using the bottom-up approach is hence considered to be fully affordable and feasible across its development lifecycle.

In summary, the design for affordability process can be conducted in a layered, scalable approach based on MATE as shown in Figure 8-1 and complemented with EEA for time-centric design considerations.

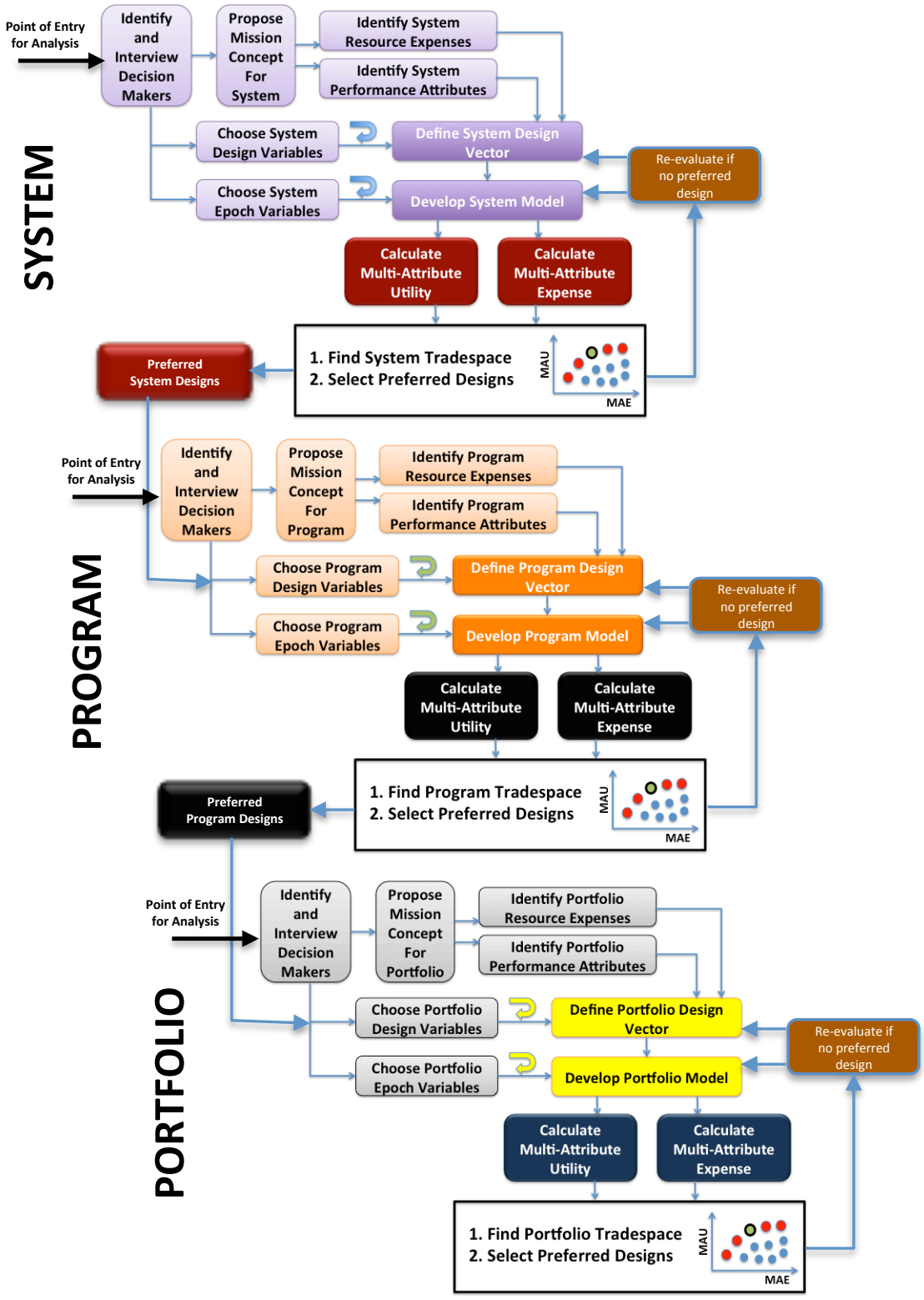


Figure 8-7: Flow chart for conducting the tradespace exploration process from system to portfolio

.

3. *What are the issues associated with considering affordability in early-phase design?*

Designing for affordability using tradespace-based methods offers several advantages that can overcome limitations experienced by other methods currently used for affordability analysis, but they also have their limitations and liabilities. They must be used appropriately to achieve maximum satisfaction.

Tradespace-based methods for affordability analysis offer a systematic, disciplined and convenient approach to system design and management and its system-program-portfolio framework allows the design process to be layered and scalable. Most importantly, they can overcome many limitations found in many existing methods. However, they also have potential drawbacks and liabilities. The usage of stakeholder preferences introduces subjectivity, and the use of multi-attribute utility and expense functions are not as intuitive as traditional measures of performance and cost. Furthermore, its results are highly contingent on the levels of model fidelity and tractability, as well as the availability of expertise and decision makers. Tradespace-based methods are at best close representations of reality and they are to be used to confirm heuristics rather than as proofs. Therefore, it is imperative to apply tradespace-based methods properly in order to obtain best results from affordability analysis.

4. *How can affordability concepts be propagated throughout the defense and aerospace industry?*

Affordability can be propagated through the defense and aerospace industry if methods are used in evolutionary ways that complement existing acquisition frameworks. Affordability can also be approached through perspectives beyond engineering design. New policies and refined management approaches can be used alongside tradespace-based methods to ensure the delivery of affordable systems.

Tradespace-based methods can offer numerous benefits to affordability analysis and they can be applied to current practices in the defense and aerospace industry in a number of ways. Evolutionary approaches are preferred and tradespace-based methods can be used to complement existing acquisition frameworks. Frameworks from the US DoD and NASA were analyzed to determine areas in which these methods were most applicable and effective. It was determined that these methods would be most applicable to the first phase of both acquisition frameworks as well as the first two major decision points.

If new frameworks can be proposed, the Affordability Engineering Framework (AEF) by the MITRE Corporation, coupled with tradespace-based methods, can provide a rigorous approach for achieving program affordability. The AEF with tradespace-based methods becomes a multi-step process that first qualitatively and quantitatively assesses program affordability risk and

stakeholder preferences, then developing and validating tradespace models that enable the generation and identification of preferred solutions, then developing a set of targeted tradeoffs that are bundled for either correction or enhancement purposes, and finally recommending for implementation. Application of the AEF can increase the frequency of affordability analyses and enable the holistic consideration of performance, cost and schedule throughout design.

Affordability can also be achieved through approaches outside systems engineering. Some approaches include making policy changes, managing multiple stakeholders, better communication and work flow organization, better integration between industry and academia, and the continued education and training of the workforce. Application of these approaches can produce positive impacts on the future of design. This ensures that affordability principles can propagate through industry practices and management philosophies. In doing so, every system, program or portfolio being designed can become and remain affordable over time.

8.3 Future Work

The research conducted in this thesis has established the foundation for affordability studies using MIT SEAri methods and constructs. With understanding of key affordability concepts and the design methodology described through case study applications, acquisition practitioners can apply the tradespace-based methods to derive affordable solutions for their unique system design problems. However, the design for affordability process can always be refined and enhanced in order to increase its credibility. Therefore, there remains considerable research to be conducted in the future to advance the field of affordability studies. The future work discussed in this section will be based on the work completed for this thesis as well as further research considerations beyond the scope of the thesis.

8.3.1 Enhancement to the FSS case study

The enhancement that can most immediately be completed in the near future will be increasing the fidelity of the tradespace models used for the FSS case study. The tradespace models used in this thesis were of lower fidelity but they were more than sufficient and useful in deriving key concepts that facilitate the basic design of the FSS. Since the FSS comprises multiple satellite constellations, a considerable amount of effort was already channeled towards modeling it from the system to program to portfolio levels. While models can never be complete and perfect, they can always be made better.

Currently, the exchange of information among satellites in the FSS is rudimentarily modeled using basic units of “data packets”. While the assumption of using data packets as the common means of information flow in the space communications network is a key enabler of this early-phase design process, it is also a major drawback in terms of scientific credibility. The establishment of inter-satellite links cannot be realistically modeled by regular inflows and outflows of data packets, as there are other complex processes such as channel modulation, phase-shift keying and encoding going on in a communications system.

Also, ISL is only possible if the link budgets between two or more satellites are closed when they are within line of sight of one another. However, none of these were taken into account for the system or program tradespace models in the FSS case study due to time and manpower

constraints on the research. Therefore, the first phase of future work should include enhancement of the FSS tradespace models to ensure that the space communications aspects are scientifically validated. Communications in a dynamic space network is the backbone of the FSS concept of operations. Apart from communications, other areas for improvement include cost estimating relationships and profiles of cost commitments. The lifetime of a spacecraft may not always be readily fall into 10-year and 15-year categories and the commitment of different cost elements may not always be distributed evenly over a number of years. More validation and verification with cost, schedule and performance profiles of existing satellites can be performed.

Interviews and surveys can actually be conducted with FSS experts to elicit the preferences of actual stakeholders. These stakeholders are available within the MIT community and the Skolkovo Institute of Science and Technology. Currently, the preferences sets used in the FSS tradespace models are hypothetical and they were created by the author of this thesis based on a personal assessment of the mission and operating context that characterizes an epoch of interest. However, actual stakeholders may have different and more updated perspectives towards the missions and operating contexts that FSS satellites are expected to operate in. Furthermore, the development of the FSS is likely to be a multi-stakeholder effort and its early-phase design can become more credible if a number of actual stakeholders were interviewed and later involved in the design process.

Given that FSS research is still in its nascent stages, any enhancement on the conceptual and computerized models would lend more credibility to the results gathered from affordability analysis and contribute to the advancement of the FSS concept for future space operations.

8.3.3 Applying the top-down approach

Earlier in the thesis, it was discussed that the top-down approach to designing for affordability can be taken in order to avoid the risk of selecting sub-optimal portfolios. This risk is present in the bottom-up approach if system or program designs were chosen too conservatively. As a lot of uncertainty exists at the portfolio level, more research has to be done in future to be able to characterize uncertainty and ensure that the selection of affordable portfolios can subsequently facilitate the selection affordable systems or programs that are less likely to be sub-optimal.

8.3.4 Enhance the formulation and aggregation of expenses

It was discussed earlier that one of the drawbacks of affordability analysis using tradespace-based methods is the use of the Multi-Attribute Expense (MAE) function to aggregate all cost and time elements. Validity of the function is based on the assumption that these elements are all perceived independent. However, these elements are not independent in reality. Cost and time elements are usually tied closely to one another during system development as a result of bundled requirements in program contracts. Hence, there are likely to be spillover effects if one of them exceeds initial estimates. For the purposes of affordability analysis, the assumption of perceived independence still holds. Its usage in the analyses performed in this thesis is justified, as there can be unexpected synergies that result as cross-terms in the multiplicative MAU and MAE functions. Currently, there are no obvious candidate functions that allow the better aggregation of interdependent expense elements together with stakeholder preferences. Therefore, future

research work can be conducted to find out how to enhance the formulation and aggregation of multiple resource expenses and overcome their interdependent nature.

8.3.5 Trading Affordability with other Ilities

Affordability has been defined as ility in this thesis and analyses conducted using tradespace-based methods were driven towards the constituent properties of cost-effectiveness and time-effectiveness. In order to achieve affordability, performance was traded for better cost and schedule margins. As a result, this may compromise other ilities that were driving the design of a system. Such ilities may include robustness, survivability, evolvability and changeability and they generally drive preferred design solutions to have properties that enable them to resist or respond positively to changes. These solutions may have higher performance margins or built with design options that allow them to do so. With the consideration of affordability, a significant number of design points in the tradespace can be immediately regarded as unaffordable and will not be explored further. As a result, affordable design solutions are not likely to the same as solutions that were derived using any of the other ilities. Therefore, it is of interest to find out how affordability can be best traded with other ilities, so that preferred designs can be affordable and changeable, or affordable and survivable. Considerable research has already been conducted for the change-related ilities like changeability, evolvability and survivability. Hence, a possible area for future work can be the trading of affordability with other ilities during the design process.

8.3.6 Multi-Era Analysis

In this thesis, single-epoch, multi-epoch and single-era analysis was conducted in the Space Tug and FSS case studies. However, multi-era analysis was not conducted as there was insufficient knowledge regarding its concepts and procedures at the point of inception for this thesis. Now, the basics of multi-era analysis have been established and it can possibly be applied to any of the case studies to see what designs can become or remain affordable across many possible futures (Schaffner, 2014).

8.3.7 The Concept of Product-Service Systems

Sometimes, referring to an entity being designed as a 'system' is not sufficient and more than often, it is hard to delineate the variables that characterize a system, a program, and a portfolio. A possible improvement can be the introduction of the Product-Service System (PSS) concept (Ray et al 2006). A PSS is defined as 'a system of products, services, network partners and supporting infrastructure that is economically feasible, competitive and satisfies customer needs. It offers dematerialized solutions that minimize the environmental impact of consumption'. Therefore, a PSS consists of products and services, which have tangible, and intangible elements combined together to deliver value to the customer throughout its life cycle while ensuring economic profitability for the manufacturer (Bankole, 2011). As such, the notion of PSS can potentially resonate with the defense and aerospace industry. Its holistic consideration of so many factors affecting product development and delivery can facilitate the promulgation of more affordable contractual agreements.

8.3.7 *Affordability, Profitability and Sustainability*

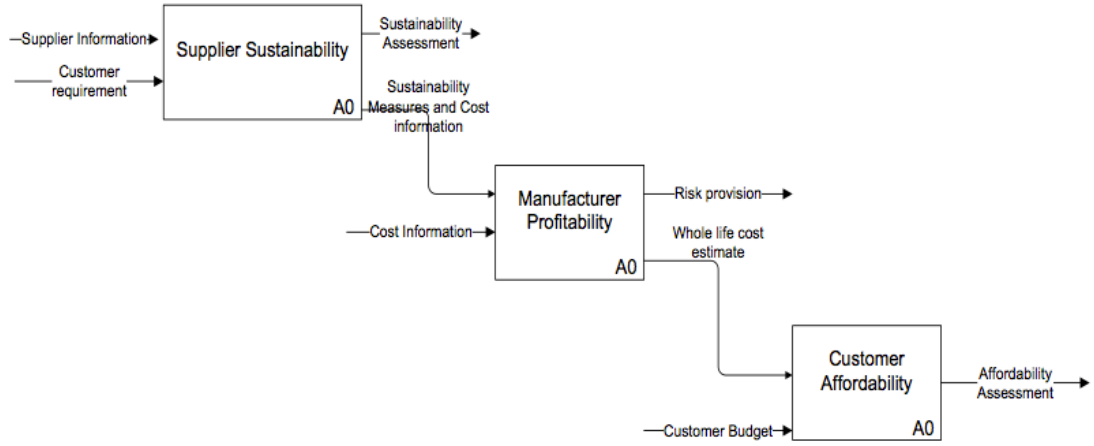


Figure 8-8: Links between supplier sustainability, manufacturer profitability and customer affordability (Bankole, 2011)

There exists a longer and more complex value chain in the delivery of defense and aerospace systems. Although affordability was the main driver of design in this thesis and assessed from the perspectives of the customer, there are also other important players and properties that exist. As shown in Figure 8-8, manufacturer profitability and supplier sustainability can impose significant downstream effects on customer affordability (Bankole, 2011). Therefore, it may be of interest in the future to define manufacturer profitability and supplier sustainability using MIT SEAri constructs and concepts, and assess how these three interacting "ilities" can be best achieved across a value chain.

8.3.8 *Extending the Scope of Affordability*

Performance, cost and schedule were used as the three main elements for characterizing affordability. However, there are more factors that influence affordability in the defense and aerospace industry as shown in Figure 8-9. Additional factors such as legislation, world economic climate, global competition, supply chain issues, environmental factors and political situations were not explicitly considered. It is obvious that a change in acquisition laws and policies, or a shift in government priorities can greatly impact the design process and eventually determine whether a system is affordable or not. Therefore, further research from both systems engineering and policy perspectives can be conducted in future to determine the impacts of legal, political and environmental disturbances on affordability.

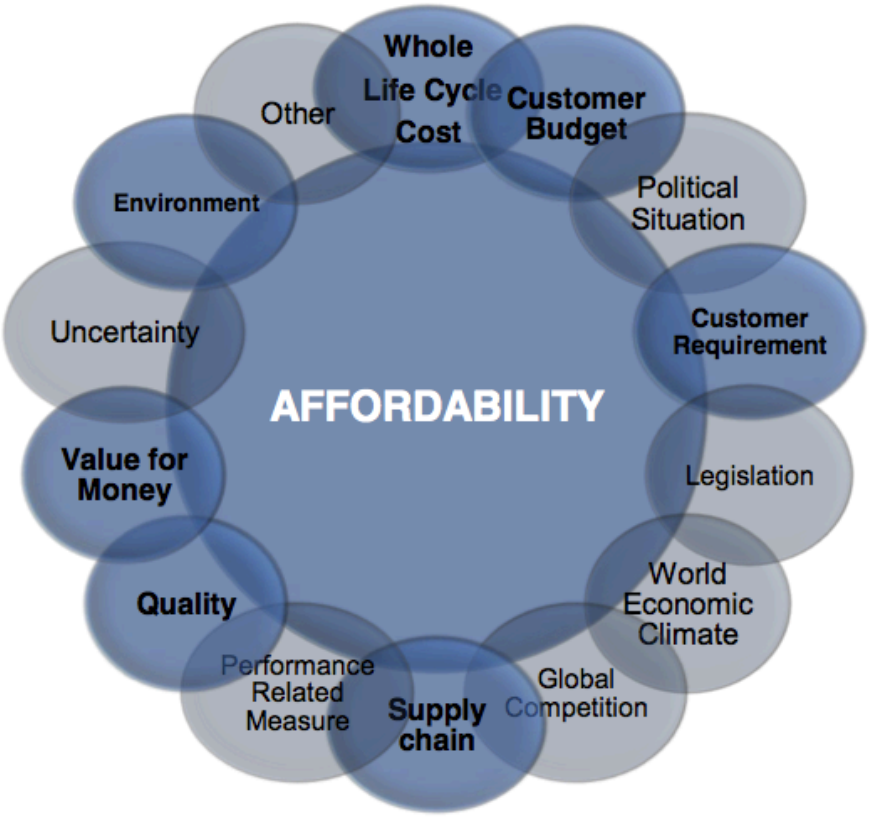


Figure 8-9: Factors influencing Affordability (Bankole, 2011)

In summary, a number of research thrusts can possibly be taken in future to advance the field of affordability systems. Through the generation and exchange of more knowledge, best concepts and practices can be propagated through the defense and aerospace industry, thereby motivating the continued design and delivery of more affordable systems, programs and portfolios in the future.

8.4 Epilogue

Designing for affordability is a major challenge that has to be taken seriously by the defense and aerospace industry if future cost and schedule overruns are to be avoided. Balancing affordability and risk in a cost-constrained acquisition environment may be difficult, but industry leaders are beginning to take initiatives that can shape the future of affordable system design and acquisition.

At the 6th annual US Space Mission Assurance Summit held on 5-6 February 2014 at the National Reconnaissance Office (NRO) in Chantilly, VA, the theme was "Creating a More Affordable Enterprise: Best Practices for Life Cycle Mission Success". Top leaders in the government and industry space community shared the best practices and lessons learned on maintaining affordability while achieving mission success (Aerospace 2014). Dr. Wanda Austin, President and CEO of The Aerospace Corporation, said the following words to the attendees:

"Our space systems continue to provide extraordinary value. Our efforts to be more cost effective are providing great results while our systems continue to provide essential and reliable products to support national security,"

"As we become more innovative in the development and acquisition of our space systems, we continue to apply lessons learned and leverage our best practices to deliver 100 percent mission success."

These comments show that major industry players like The Aerospace Corporation have been taking affordability into consideration and taking more cost-effective measures into their design.

Betty Sapp, Director of the NRO, also discussed past launch vehicle and space vehicle failures and successes in her keynote address to the summit participants. During her talk, she described a cycle where sustained good performance leads to a pressure to divert resources to other areas, especially during times of budget constraint, which can then result in mission failure or degradation (Aerospace 2014). The failure is then followed by a "back to basics" approach and added resources, which returns successful performance. Ms. Sapp said:

"The challenge for us is to figure out how we deal with a resource-constrained environment and maintain success and how we measure the risk associated with dialing up or down mission assurance."

Many others have also contributed their opinions towards the creation of affordability through their enterprises. It is clear that the concept of affordability is not being overlooked. Cost-effectiveness and time-effectiveness are increasingly recognized as key attributes in the design process. As it is impossible for affordability measures to be perfect and to accurately predict the future, cost overruns, schedule delays and program management failures are still likely to occur. However, it is the responsibility of those involved to do their best and commit their knowledge towards reducing their likelihood. As Dr. Wanda Austin said at the 2014 Conference on Systems Engineering Research (CSER),

"Affordability is ultimately in the eyes of the beholder"

To define what is affordable, design what is affordable and sustain what is affordable,

It is our responsibility to make it happen.

REFERENCES

Allard, Wayne, United States Senator. (2005). *Address to the National Defense Industrial Association, 2005 Space Policy and Architecture Symposium*. 23 September 2005. Retrieved from: http://www.globalsecurity.org/space/library/news/2005/space-050923- ndia-allard.htm.

Alleman, G.B. (2010). *Performance-Based Project Management – An Integrated Approach to Principles, Practices, and Processes to increase Probability of Project Success*. Retrieved from: http://herdingcats.typepad.com/my_weblog/2010/11/domain-dependent-concepts.html

AT Kearney (2011). *Should cost review to improve affordability*. Retrieved from: http://www.atkearney.com/aerospace-defense/featured-article/-/asset_publisher/S5UkO0zy0vnu/content/should-cost-review-to-improve-affordability/10192

Bankole, O.O. (2011). *Development of an Affordability Assessment Framework for Defense Contracts at the Bidding Stage*. Cranfield University.

BBC News (2013). *ESA Selects Ariane 6 Basic Design*. Retrieved from: http://www.bbc.com/news/science-environment-23241158

Beesemyer, J.C. (2012). *Empirically Characterizing Evolvability and Changeability in Engineering Systems*. Master of Science Thesis, Aeronautics and Astronautics, MIT, June 2012.

Bever, Bob, and James S. Collofello. (2002). "*An investigation of techniques for addressing software affordability*." *Aerospace Conference Proceedings, 2002. IEEE*. Vol. 5. IEEE, 2002.

Blanchard B. and W. Fabrycky. (2006). *Systems Engineering and Analysis* (4th ed.). Upper Saddle River, NJ: Prentice Hall.

Bobinis, J., Haimowitz, J., Tuttle, P., Garrison, C. (2012). *Affordability Considerations: Cost Effective Capability. INCOSE Affordability Working Group*. Retrieved from: http://www.mors.org/UserFiles/file/2012%20-%20Meeting%20Affordabilty%20Analysis/Read%20Aheads/Affordability%20Considerations_Cost%20Effective%20Capability.pdf

Booton, R., and Ramo, S. (2012). "*The Development of Systems Engineering*," IEEE Aerospace and Electronic Systems, vol. AES-20, no.4, pp. 306-310, July 1984.

Booz Allen Hamilton (2011). *Why it's needed and VIRGINIA Case Study*. Washington DC, May 17 2011.

Borky, J., W. Bauhaus Jr, and A. Brown. (1998) "A Space Roadmap for the 21st Century Aerospace Force." *Air Force Scientific Advisory Board Presentation.*

Broniatowski, D.A., and Weigel A.L. (2008). "*Political Sustainability in Space Exploration*," *Space Policy*, Vol. 24, No. 3, pp. 148-157, August 2008.

Brown, O. and Eremenko, P. (2006). *The Value Proposition for Fractionated Space Architectures*, in: AIAA (Ed.) AIAA Space 2006, AIAA, San Jose, CA, 2006.

Cancian, M.F. (2010). *Cost Growth: Perception and Reality. A Publication of the Defense Acquisition University*. Retrieved from http://www.dau.mil/pubscats/PubsCats/AR%20Journal/arj55/Cancian_55.pdf

Carter, A. B. (2010a). *Better buying power: Guidance for obtaining greater efficiency and productivity in defense spending* [Memorandum]. Retrieved from http://www.acq.osd.mil /docs/USD_ATL_Guidance_Memo_September_14_2010_FINAL.PDF.

Carter, A. B. (2010b). *Better buying power: Mandate for restoring affordability and productivity in defense spending* [Memorandum]. Retrieved from http://mil-oss.org/resources/us-dod_policy-memo_carter-memo-on-defense-spending-28-jun 2010.pdf

Carter, A. B. (2011). *Should-cost and Affordability* [Memorandum]. Retrieved from http://bbp.dau.mil/docs/Should-cost%20and%20Affordability.pdf

Carter, A. B. and Hale, R.F. (2011). *Joint Memorandum on Savings Related to "Should Cost*. Retrieved from http://bbp.dau.mil/docs/Joint%20Memorandum%20on%20Savings%20 Realted%20to%20Should%20Cost.pdf

Cordesman, A.H. and Frederiksen, P.S. (2006). *America's Uncertain Approach to Strategy and Force Planning*. Working Draft Center submitted to Center for Strategic and International Studies.

Dahmann, J., Rebovich, G., Lane, J., Lowry, R., & Baldwin, K. (2011). *An implementers' view of systems engineering for system of systems.* IEEE Systems Conference, Montreal, Quebec.

Defense Acquisition University (DAU) (2010). *Introduction to Defense Acquisition Management.* Retrieved from: http://www.dau.mil/publications/publicationsDocs/Intro%20to%20Def%20Acq% 20Mgmt%2010%20ed.pdf

Defense Acquisition University (DAU) (2013). *Improving the Professionalism of the Defense Acquisition Workforce.* Retrieved from: http://www.dau.mil/aboutDAU/AboutDocs/StrategicPlan.pdf

Defense Acquisition Guidebook (DAG) (2013). Retrieved from https://acc.dau.mil/Community Browser.aspx?id=654219, ID 654219, Created: May 9, 2013, Modified: July 15, 2013.

de Weck, O.L., Ross, A.M., and Rhodes, D.H. (2012). *Investigating Relationships and Semantic Sets amongst System Lifecycle Properties (Ilities)*. 3rd International Conference on Engineering Systems, TU Delft, the Netherlands, June 2012.

de Weck, O.L., de Neufville, R. and Chaize, M. (2004). *Staged Deployment of Communications Satellite Constellations in Low Earth Orbit*. Journal of Aerospace Computing, Information, and Communication, Vol. 1, March 2004

Davendralingam, N and DeLaurentis, D. (2013). *Acquisition Management for System of Systems: Affordability Through Effective Portfolio Management*. 10th Annual Acquisition Research Symposium Systems of Systems Management, Naval Postgraduate School.

Dean, E.B. (2012). Affordability. Presented at the 2012 ISPA/SCEA Joint International Conference & Training Workshop 14--16 May 2012 Brussels, Belgium. Retrieved from http://www.mors.org/UserFiles/file/2012%20-%20Meeting%20Affordabilty%20Analysis/ Read%20Aheads/Affordability%20Paper%20120409%20Final.pdf

Derleth, J. E. (2003). *Multi-Attribute Tradespace Exploration and its Application to Evolutionary Acquisition*. SM thesis in Aeronautics and Astronautics. Massachusetts Institute of Technology. Cambridge, MA: 144.

Diller, N. P. (2002). Utilizing Multiple Attribute Tradespace Exploration with Concurrent Design for Creating Aerospace Systems Requirements Engineering. S.M. Thesis. Massachusetts Institute of Technology. Cambridge, MA.

Emmons, D.L., Lobbia, M., Radcliffe, T., and Bitten, R.E. (2010). *Affordability assessments to support strategic planning and decisions at NASA*. Aerospace Conference, 2010 IEEE, vol., no., pp.1,13, 6-13 March 2010. doi: 10.1109/AERO.2010.5446893

Fitzgerald, M.E., and Ross, A.M., "Controlling for Framing Effects in Multi-Stakeholder Tradespace Exploration," 12th Conference on Systems Engineering Research, Redondo Beach, CA, March 2014

Golkar, A. (2013) "*Federated satellite systems: an innovation in space systems design*," in 9th IAA Symposium on Small Satellites for Earth Observation. Berlin, Germany: International Academy of Astronautics, April 2013.

Golkar, A. (2013) "*Federated satellite systems: A case study on sustainability enhancement of space exploration systems architectures*," in 64th International Astronautical Congress, no. IAC-13-D3.4. Beijing, China: International Astronautical Federation, September 2013.

Gourley, Scott R. (2004). "*Space Warriors and Wizards*," Military Aerospace Technology, 3.3 (15 November 2004). Retrived from: http://www.military-aerospace- technology.com/article.cfm?DocID=683.

Herald, Tom. (2011) "Affordable Architectures -- How do I recognize one?" Lockheed Martin, INCOSE ESEP.

Higgins, G. (1997). *CAIV – An Important Principle of Acquisition Reform*. PM: Special Issue. January – February 1997.

Hubka, V., Andreasen, M., Eder, W. and Hills, P. (advisory editor) (1988), *Practical Studies in Systematic Design,* Butterworths, London.

International Council on Systems Engineering (INCOSE) Affordability Working Group (2011). Retrieved from http://www.incose.org/practice/techactivities/wg/details.aspx?id=affwg

International Council on Systems Engineering (INCOSE) Affordability Working Group (2011). "*Affordability Analysis: How do we do it?" 2011.* Retrieved from: http://www.mors.org/UserFiles/file/2013%20-%20Meeting%20Affordabilty%20Analysis/Outbriefs/06%20_%20INCOSE%20Plenary%20Panel%20Brief%202012-08-01.pdf

Institute of Defense Analyses (IDA) (2009). *The Major Causes of Cost Growth in Defense Acquisition. Volume I: Executive Summary*. IDA Paper P-4531 Log: H 09-001671, December 2009.

Keeney, R. L. and Raiffa, H. (1993). *Decisions with Multiple Objectives-Preferences and Value Tradeoffs*. Cambridge, UK, Cambridge University Press.

KLR – Leaders in Project Management. (2008). *Project/program/portfolio management: What does it really mean?* Retrieved from http://www.klr.com/articles/Articles_PM_project_program_portfolio_mgmt.pdf

Kroshl, W.M. and Pandolfini, P.P. (2000). *Affordability Analysis for DARPA Programs*. John Hopkins APL Digest, Volume 21, Number 3 (2000).

Taverney, T.D. (2011). *Resilient, disaggregated, and mixed constellations*. The Space Review. Retrieved from: http://www.thespacereview.com/article/1918/1

Lee, Douglas E. (2004). "*Space Reform*," Air and Space Power Journal, 18.2: 103-112 (Summer 2004). Retrieved from: http://www.airpower.maxwell.af.mil/airchronicles/apj/apj04/sum04/ lee.html.

Lluch, I. and Golkar, A. (2014). *Satellite-to-Satellite Coverage Optimization Approach for Opportunistic Inter-Satellite Links*. (Working Paper)

Maier, M. W. (1999). *Architecting principles for systems-of-systems*. Systems Engineering, 1:pp. 267–284.

Mallory, Craig J. (2011). *International Programs Contribute to Affordability*. Defense Acquisition Univerisity.

Mavris, Dimitri N., and Daniel A. DeLaurentis. (1998) "A stochastic design approach for aircraft affordability." Aerospace System Design Laboratory (ASDL) School of Aerospace Engineering, Georgia Institute of Technology.

Merriam-Webster Online Dictionary (2014). Retrieved from: http://www.merriam-webster.com/info/copyright.htm

Mikaelian, T., Nightingale, D., Rhodes, D., and Hastings, D. (2011). "*Real Options in Enterprise Architecture: A Holistic Mapping of Mechanisms and Types for Uncertainty Management*," *IEEE Transactions on Engineering Management*, Vol. 58, No. 3, August 2011.

MITRE Corporation: Affordability, Efficiency and Effectiveness. (2012). Retrieved from: http://www.mitre.org/work/systems_engineering/guide/se_lifecycle_building_blocks/other_life_cycle_building_blocks_articles/afford_efficiency_effect.html

MITRE Corporation - Anderson, S., Wydler, V., and Duquette, J. (2012). *Affordability Engineering Framework Overview*. 9th Annual Acquisition Research Symposium, Naval Postgraduate School.

NASA Systems Engineering Handbook. (2012)

National Research Council. (2010). *Controlling Cost Growth of NASA Earth and Space Science Missions*. The National Academies Press.

NDIA Affordability Working Group. (2011). *Status Report, 14th Annual Systems Engineering Conference*. Retrieved from: http://www.psmsc.com/Downloads/2012WinterTelecoms/Affordability%20Working%20Group%20-%20SE%20Division%20Strategy%20Meeting%20111207.pdf

Nickel, J. (2010). *Trading Political Feasibility and Technical Optimality: Multi-Attribute Tradespace Exploration as Communication Tool in Transportation Planning*. Industry Studies Conference 2010. Chicago, IL.

Oreskes, N., Shrader-Frechette, K., and Belitz, K. (1994). *Verification, Validation, and Confirmation of Numerical Models in the Earth Sciences*. Science, New Series, Vol. 263, No. 5147, pp. 641-646

Oberkampf, W. L. and T. G. Trucano (2002), "Verification and Validation in Computational Fluid Dynamics," *Progress in Aerospace Sciences,* vol. 38, No. 3, pp. 209-272.

Otto, K. and Wood, K. (2000), Product Design: Techniques in Reverse Engineering and New Product Development, Prentice Hall, Upper Saddle River, NJ.

Pahl, G. and W. Beitz (1984), *Engineering Design: A Systematic Approach,* Design Council, London.

Parabolic Arc. (2013). *SpaceX Competition Has Arianespace Looking at New Pricing*. Retrieved from: http://www.parabolicarc.com/2013/11/26/spacex-competition-arianespace-pricing/#sthash.6KceCvN1.dpuf

Patterson, J.D. (2013). *Time as an Independent Variable: A Tool to Drive Cost Out of and Efficiency into Major Acquisition Programs*. 10th Annual Acquisition Research Symposium Acquisition Management. Published April 1, 2013.

Plummer, B. (2013). *America's staggering defense budgets, in charts*. The Washington Post, January 7, 2013. Retrieved from http://www.washingtonpost.com/blogs/wonkblog /wp/2013/01/07/everything-chuck-hagel-needs-to-know-about-the-defense-budget-in-charts/

Ray, A, Baguley, P. and Roy, R. (2006). "Developing aFramework for Affordability Engineering" The 4th International Conference on Manufacturing Research (ICMR 2006), Liverpool John Moores University, 5-7 September 2006:11-16.

Redman, Q.; Stratton, G. (2001). "Why affordability is a systems engineering metric," *Digital Avionics Systems, 2001. DASC. 20th Conference* , vol.1, no., pp.4B3/1-4B3/12 vol.1, 14-18 Oct 2001 doi: 10.1109/DASC.2001.963371

Rhodes, D.H., Ross, A.M., and Nightingale, D.J. (2009). "*Architecting the System of Systems Enterprise: Enabling Constructs and Methods from the Field of Engineering Systems*," 3rd Annual IEEE Systems Conference, Vancouver, Canada, March 2009.

Richards, M.G., Viscito, L., Ross, A.M., and Hastings, D.E. (2008) "*Distinguishing Attributes for the Operationally Responsive Space Paradigm*," Responsive Space 6 Conference, Los Angeles, CA, April 2008.

Richards, M.G., Ross, A.M., Hastings, D.E., and Rhodes, D.H. (2009). Multi-Attribute Tradespace Exploration for Survivability. *7th Conference on Systems Engineering Research*. Loughborough University, UK.

Roberts, C. J. (2003). *Architecting Strategies Using Spiral Development for Space Based Radar*. SM thesis in Technology and Policy Program. Massachusetts Institute of Technology. Cambridge, MA: 86.

Ross, A.M. (2003). *Multi-Attribute Tradespace Exploration with Concurrent Design as a Value-centric Framework for Space System Architecture and Design*. Dual Master of Science Thesis, Aeronautics and Astronautics and Technology and Policy Program, MIT, June 2003.

Ross, A.M., Hastings, D.E., and Diller, N.P. (2003). *Multi-Attribute Tradespace Exploration with Concurrent Design for Space System Conceptual Design*. AIAA Paper 2003-1328, 41st Aerospace Sciences Meeting, Reno, NV, January 2003.

Ross, A.M., Hastings, D.E., Warmkessel, J.M., and Diller, N.P. (2004). *Multi-Attribute Tradespace Exploration as a Front-End for Effective Space System Design*. AIAA Journal of Spacecraft and Rockets, pp. 20-28, Jan/Feb 2004

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

Ross, A.M. (2006). *Managing Unarticulated Value: Changeability in Multi-Attribute Tradespace Exploration*. Doctor of Philosophy Dissertation, Engineering Systems Division, MIT, June 2006.

Ross, A.M., and Rhodes, D.H (2008a), "*Architecting Systems for Value Robustness: Research Motivations and Progress*," 2nd Annual IEEE Systems Conference, Montreal, Canada, April 2008a.

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

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

Ross, A.M., McManus, H.L., Long, A., Richards, M.G., Rhodes, D.H., and Hastings, D.E. (2008). *Responsive Systems Comparison Method: Case Study in Assessing Future Designs in the Presence of Change*. AIAA Space 2008. San Diego, CA.

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

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

Ross, A.M., McManus, H.L., Rhodes, D.H., and Hastings, D.E. (2010a), "*Revisiting the Tradespace Exploration Paradigm: Structuring the Exploration Process*," AIAA Space 2010, Anaheim, CA.

Ross, A.M., O'Neill, M.G., Hastings, D.E., and Rhodes, D.H. (2010b) "Aligning Perspectives and Methods for Value-Driven Design," AIAA Space 2010, Anaheim, CA, September 2010.

Saleh, J. H., Mark, G. T. and Jordan, N. C. (2009). *Flexibility: a multi-disciplinary literature review and research agenda for designing flexible engineering systems*. Journal of Engineering Design, Vol. 20, No. 3, pp. 307-323.

Schaffner, M.A. (2014). *Designing Systems for Many Possible Futures: The RSC-based Method for Affordable Concept Selection (RMACS), with Multi-Era Analysis*. Master of Science Thesis, Aeronautics and Astronautics, MIT, June 2014.

Schwartz, M. (2010). *The Nunn-McCurdy Act: Background, Analysis, and Issues for Congress*. Washington, DC: Congressional Research Service. Retrieved from http://www.fas.org/sgp/crs/misc/R41293.pdf

Schwartz, M. (2013). *Defense acquisitions: How DoD acquires weapon systems and recent efforts to reform the process*. Washington, DC: Congressional Research Service. Retrieved from http://www.defenseinnovationmarketplace.mil/resources/RL340261.pdf

Semple, J. (2007). *Review into Affordability Housing Final Report, Department for Social Development in Northern Ireland*. Retrieved from: http://www .dsdni.gov.uk/consultation-review_affordable_housing.pdf (accessed 21st October 2009).

Smyth, P. (2005). *"Water and wastewater Finance and Pricing"*, Taylor and Francis Group LCC., p. 309-323

Stagney, David B. (2003). *The Integrated Concurrent Enterprise*. SM thesis in Aeronautics and Astronautics. Massachusetts Institute of Technology. Cambridge, MA: 180.

Szajnfarber, Z., and Weigel, A.L. (2009). "*What Drives Spacecraft Innovation? A Quantitative Analysis of Communication Satellite Innovation History (1964-2006)*," Technology Management and Policy Consortium meeting, Vancouver, BC, June 2009.

The Aerospace Corporation (2014). Space Industry Leaders Address Affordability of Space Systems. Retrieved from: https://www.aerospace.org/2014/02/11/space-industry-leaders-address-affordability-of-space-systems/

Tuttle, P. and Bobinis, J. (2012). *Affordability Specification*. Retrieved from http://www.mors.org/UserFiles/file/2012%20%20Meeting%20Affordabilty%20Analysis/Read%20Aheads/Affordability%20Specification%20MORS.pdf

Ulrich, K. T., and Eppinger, S. (1995), *Product Design and Development,* McGraw-Hill, NY.

Underwood, J. (2003). Distributed Satellite Communication System Design: First-Order Interactions between System and Network Architectures. Master of Science Thesis, Aeronautics and Astronautics, MIT, June 2005.

U.S. Department of Defense (DoD). (2008). *Operation of the defense acquisition system* (DoDI 5000.02). Washington, DC: Office of the Under Secretary of Defense (Acquisition, Technology, & Logistics).

U.S. Department of Defense (DoD). (2012). *Better Buying Power 2.0*. Washington, DC: Office of the Under Secretary of Defense, Acquisition, Technology and Logistics, 2012.

U.S. Department of Defense (DoD) (2013). *Performance of the Defense Acquisition System, 2013 Annual Report*. Washington, DC: Office of the Under Secretary of Defense, Acquisition, Technology and Logistics, 2013.

U.S. Department of Defense (DoD), Office of the Under Secretary of Defense (Comptroller) (2013). *National Defense Budget Estimates for FY 2014*.

U.S. Government Accountability Office (2009). *GAO Cost Estimating and Assessment Guide (Report No. GAO-09-3SP)*. Washington, DC.

U.S. Government Accountability Office (2011). *High-Risk series: An update (Report No. GAO-11-394T)*. Washington, DC.

U.S. Government Accountability Office (2012). *Weapons Acquisition Reform: Reform Act Is Helping DOD Acquisition Programs Reduce Risk, but Implementation Challenges Remain (Report No. GAO-13-103)*. Washington, DC.

U.S. Government Accountability Office (2012). *James Webb Space Telescope: Actions Needed to Improve Cost Estimate and Oversight of Test and Integration (Report No. GAO-13-4)*. Washington, DC.

U.S. Government Accountability Office (2013). *Defense acquisitions: Assessments of selected weapon programs (Report No. GAO-13-294SP)*. Washington, DC.

U.S. Government Accountability Office (2013). *High-Risk Series: An Update (Report No. GAO-13-283)*. Washington, DC.

Viscito, L. (2009). *Quantifying Flexibility in the Operationally Responsive Space Paradigm*, Master of Science Thesis, Aeronautics and Astronautics, MIT, June 2009.

Viscito, L., and Ross, A.M. (2009). *Quantifying Flexibility in Tradespace Exploration: Value-Weighted Filtered Outdegree*. AIAA Space 20*09*. Pasadena, CA.

Weigel, A. L. (2002). *Bringing Policy into Space Systems Conceptual Design: Quantitative and Qualitative Methods*. PhD thesis in Technology, Management, and Policy. Massachusetts Institute of Technology. Cambridge, MA: 168.

Wertz, J.R. and Larson, W.J. (2011). *Space Mission Analysis and Design (SMAD)*.

Wiggins, S. (2009). The Federal Budgeting Process and The Color of Money. Retrieved from http://resources.ncmahq.org/chapters/Beach_Cities/Shared%20Documents/Presentations/Color%20of%20Money%2011-10-09.pdf

Wu, M.S., Ross, A.M., and Rhodes, D.H. (2014). "*Design for Affordability in Complex Systems and Programs Using Tradespace-Based Affordability Analysis*," 12th Conference on Systems Engineering Research, Redondo Beach, CA, March 2014.

APPENDIX 1:
SYSTEM TRADESPACE MODEL

The MATLAB code for the entire FSS tradespace model will not be included in this appendix. Only important sections are shown. Contact MIT SEAri or the author of this thesis for access to the full code.

This section contains the MATLAB code written to **generate all possible system designs** for an FSS-enabled satellite.

```
function [satellite_performance_attributes,satellite_expense_attributes,satellite_intermediate,
system_DV] = Generate_Satellite_Design_For_Epoch(Laser,ReusableLV)
%% This code is written by Marcus Shihong Wu.
%% Copyright (c) 2014, SEARI MIT.

%% Performance Attributes
% 1 - ISL Data Rate
% 2 - Data Available per Satellite
% 3 - Science Value
% 4 - Delta V

%% Expense Attributes
% 1 - Development Cost
% 2 - Launch Cost
% 3 - Development Time

%% Intermediate Variables
% 1 - Comms Weight
% 2 - TCC Weight
% 3 - Data Packet Available (can be positive or negative)
% 4 - Base Mass
% 5 - Dry Mass
% 6 - Wet Mass

%% Satellite Variables
spacecraft_lifetime = [10 15];
datapacket_capacity = [10000 15000 20000 25000 30000]; % total number of packets
datapacket_usage = [5 15 25]; % packets per kg of payload
transmitter_type = [1 2 3]; %1 - RF Small Antenna, RF Big Antenna, 2- Laser
terrestrial_capacity_real_option = [0 1]; % 0: dont buy, 1: Buy
propulsion_type = [1 2 3 4]; % storable bi, cryogenic, electric, nuclear
payload_cap = [30 100 500 1000]; % kg
propellant_mass = [500 1000 2000]; % kg % DeltaV =4km/s for 30deg change in inclination at LEO, 11km/s
for 90deg
% Changing orbital altitude within LEO 400-1000km = 0.3km/s
% Refer to https://www.amacad.org/publications/Section_6.pdf
% LEO orbit velocity 6.9-7.8km/s - Use 7.8km/s. Use 6.9, 7.2, 7.5, 7.8
% Orbital Inclination Change: Delta_V = 2vsin(Delta_Angle/2)
% Max Delta_V of 2849.6 allows for a Delta_angle of 21.1-23.8deg
% Min Delta_V of 292 allows for a Delta_angle of 2.14-2.42 deg
% Launch Variables
launcher = [1 2 3 4];
% 1 - Falcon 9 ($56.M 13500kg $4185/kg
% 2 - Falcon 9 Heavy ($135M 53000kg $2547/kg)
% 3 - Ariane V ($167.7M 16000kg $5072/kg)
% 4 - Atlas V ($395.6M 30000kg 6182/kg)
% See: http://space.stackexchange.com/questions/1989/what-is-the-current-cost-per-pound-to-send-
something-into-leo

%% Calculate Small Spacecraft Weight
% From (Wertz and Larson, 2011) SMAD Pg 297 Table 11-7
weight_structure = 205; %kg for structures
weight_thermal = 40; % kg for thermal control
weight_ADCS = 65; % kg for attitude determination and control systems
weight_EPS = 251; % kg for electrical power system
weight_propulsion = 75; % kg for propulsion system
weight_TTC = 45; % kg for telemetry tracking and control
weight_CDH = 46; % kg for command and data handling


%% Generate Satellite Designs
system_DV = zeros(34560,9);
satellite_performance_attributes = zeros(34560,4);
```

```
satellite_expense_attributes = zeros(34560,3);
satellite_intermediate = zeros(34560,6);

count=1;

for a=1:numel(spacecraft_lifetime)
    for b=1:numel(datapacket_capacity)
        for c=1:numel(datapacket_usage)
            for d=1:numel(transmitter_type)
                for e=1:numel(terrestrial_capacity_real_option)
                    for f=1:numel(propulsion_type)
                        for g=1:numel(payload_cap)
                            for h=1:numel(propellant_mass)
                                for i=1:numel(launcher)
                                    %% Storing design variables
                                    system_DV(count,1) = spacecraft_lifetime(a);
                                    system_DV(count,2) = datapacket_capacity(b);
                                    system_DV(count,3) = datapacket_usage(c);
                                    system_DV(count,4) = transmitter_type(d);
                                    system_DV(count,5) = terrestrial_capacity_real_option(e);
                                    system_DV(count,6) = propulsion_type(f);
                                    system_DV(count,7) = payload_cap(g);
                                    system_DV(count,8) = propellant_mass(h);
                                    system_DV(count,9) = launcher(i);

                                    %% Propulsion Type
                                    if propulsion_type(f)==1 % storable bi
                                        propulsion_weight = weight_propulsion; % Remains at 75kg
                                        Isp = 300;
                                        mass_frac = 0.12; % of propellant mass
                                        development_time = 0.5; % years (3 months)
                                    elseif propulsion_type(f)==2 % cryogenic
                                        propulsion_weight = weight_propulsion; % Remains at 75kg
                                        Isp = 550;
                                        mass_frac = 0.13; % of propellant mass
                                        development_time = 0.75; % years (4 months)
                                    elseif propulsion_type(f)==3 % electric
                                        propulsion_weight = weight_propulsion; % Remains at 75kg
                                        Isp = 3000;
                                        mass_frac = 0.25; % of propellant mass
                                        development_time = 1.0; % years (6 months)
                                    elseif propulsion_type(f)==4 % nuclear
                                        propulsion_weight = 500; % assume 500kg for nuclear propulsion
                                        Isp = 1500;
                                        mass_frac = 0.2; % of propellant mass
                                        development_time = 1.5; % years (12 months)
                                    end

                                    %% Transmitter Type - ATTRIBUTE 1: ISL Data Rate
                                    if Laser == 0
                                        if transmitter_type(d)==1 % RF - Small Antenna
                                            comms_weight = 1.5*weight_CDH; % with small antenna, total
weight CD&H increases
                                            ISL_data_rate = 1; % LOW
                                        elseif transmitter_type(d)==2 % RF - Large Antenna
                                            comms_weight = 2.5*weight_CDH; % with large antenna, total
weight CD&H increases more
                                            ISL_data_rate = 2; % MEDIUM
                                        elseif transmitter_type(d)==3 % Laser - Affects weight and
development time
                                            comms_weight = 0.5*weight_CDH; % with laser, total weight
CD&H decreases by a lot
                                            ISL_data_rate = 3; % HIGH
                                            development_time = 2.0*development_time; % Assume to take
2.0 times longer to perfect Laser Comms
                                        end

                                        development_cost_factor_to_BaseMass = 1;

                                    elseif Laser ==1

                                        if transmitter_type(d)==1 % RF - Small Antenna
                                            comms_weight = 1.5*weight_CDH; % with small antenna, total
weight CD&H increases
                                            ISL_data_rate = 1; % LOW
                                        elseif transmitter_type(d)==2 % RF - Large Antenna
                                            comms_weight = 2.5*weight_CDH; % with large antenna, total
weight CD&H increases more
                                            ISL_data_rate = 2; % MEDIUM
                                        elseif transmitter_type(d)==3 % Laser - Affects weight and
development time
```

```
                                        comms_weight = 0.25*weight_CDH; % with laser maturity,
total weight CD&H decreases by even more
                                        ISL_data_rate = 4; % VERY HIGH
                                        %development_time = development_time; % Laser technology
has matured. Take same amount of time to develop
                                    end

                                    development_cost_factor_to_BaseMass = 0.7;

                                end

                                % STORE INTERMEDIATE - 1 COMMS WEIGHT
                                satellite_intermediate(count,1) = comms_weight;

                                % STORE ATTRIBUTE 1
                                satellite_performance_attributes(count,1) = ISL_data_rate;

                                %% ATTRIBUTE 2: Datapacket Capacity and Datapacket Usage

                                % Adjust weight of TTC equipment for
                                % different data packet size
                                if datapacket_capacity(b)==10000
                                    TTC_weight = 1.0*weight_TTC;
                                elseif datapacket_capacity(b)==15000
                                    TTC_weight = 1.5*weight_TTC;
                                elseif datapacket_capacity(b)==20000
                                    TTC_weight = 2.0*weight_TTC;
                                elseif datapacket_capacity(b)==25000
                                    TTC_weight = 2.5*weight_TTC;
                                elseif datapacket_capacity(b)==30000
                                    TTC_weight = 3.0*weight_TTC;
                                end

                                % STORE INTERMEDIATE - 2 TTC WEIGHT
                                satellite_intermediate(count,2) = TTC_weight;

                                datapacket_available = datapacket_capacity(b) -
datapacket_usage(c)*payload_cap(g); % Calculate amount of data packet space available

                                % Terrrestrial Capacity
                                % Data Availability increases by 1.5x when you
                                % buy terrestrial option, but cost increases
                                if (terrestrial_capacity_real_option(e)==1)
                                    new_datapacket_available = 1.5*datapacket_available;
                                else
                                    new_datapacket_available = datapacket_available;
                                end

                                % STORE INTERMEDIATE - 3 Datapacket available
                                satellite_intermediate(count,3) = new_datapacket_available;

                                if new_datapacket_available <=0
                                    data_available_per_satellite = 0;
                                else
                                    data_available_per_satellite = new_datapacket_available;
                                end

                                % STORE ATTRIBUTE 2
                                satellite_performance_attributes(count,2) =
data_available_per_satellite;

                                %% ATTRIBUTE 3: Value to Science

                                % Value to Science measures from 1-12
                                % Depends on payload_cap and
                                % datapacket_usage
                                % Development time factors of 1, 1.25,
                                % 1.5 and 1.75 depend on payload_cap

                                if payload_cap(g)==30 && datapacket_usage(c)==5 % 30 x 5 = 150
                                    science_value = payload_cap(g)*datapacket_usage(c);
                                    development_time = 1*development_time; % does not take much
time to develop small payload
                                elseif payload_cap(g)==100 && datapacket_usage(c)==5 % 100 x 5 =
500
                                    science_value = payload_cap(g)*datapacket_usage(c);
                                    development_time = 1.25*development_time; % takes slightly
longer
                                elseif payload_cap(g)==500 && datapacket_usage(c)==5 % 500 x 5 =
```

```
2500
                                    science_value = payload_cap(g)*datapacket_usage(c);
                                    development_time = 1.5*development_time; % takes even longer
                                elseif payload_cap(g)==1000 && datapacket_usage(c)==5 % 1000 x 5 =
5000
                                    science_value = payload_cap(g)*datapacket_usage(c);
                                    development_time = 1.75*development_time; % takes longest
                                elseif payload_cap(g)==30 && datapacket_usage(c)==15 % 30 x 15 =
450
                                    science_value = payload_cap(g)*datapacket_usage(c);
                                    development_time = 1*development_time; % does not take much
time to develop small payload
                                elseif payload_cap(g)==100 && datapacket_usage(c)==15 % 100 x 15 =
1500
                                    science_value = payload_cap(g)*datapacket_usage(c);
                                    development_time = 1.25*development_time; % takes slightly
longer
                                elseif payload_cap(g)==500 && datapacket_usage(c)==15 % 500 x 15 =
7500
                                    science_value = payload_cap(g)*datapacket_usage(c);
                                    development_time = 1.5*development_time; % takes even longer
                                elseif payload_cap(g)==1000 && datapacket_usage(c)==15 % 1000 x 15
= 15000
                                    science_value = payload_cap(g)*datapacket_usage(c);
                                    development_time = 1.75*development_time; % takes longest
                                elseif payload_cap(g)==30 && datapacket_usage(c)==25 % 30 x 25 =
750
                                    science_value = payload_cap(g)*datapacket_usage(c);
                                    development_time = 1*development_time; % does not take much
time to develop small payload
                                elseif payload_cap(g)==100 && datapacket_usage(c)==25 % 100 x 25 =
2500
                                    science_value = payload_cap(g)*datapacket_usage(c);
                                    development_time = 1.25*development_time; % takes slightly
longer
                                elseif payload_cap(g)==500 && datapacket_usage(c)==25 % 500 x 25 =
12500
                                    science_value = payload_cap(g)*datapacket_usage(c);
                                    development_time = 1.5*development_time; % takes even longer
                                elseif payload_cap(g)==1000 && datapacket_usage(c)==25 % 1000 x 25
= 25000
                                    science_value = payload_cap(g)*datapacket_usage(c);
                                    development_time = 1.75*development_time; % takes longest
                                end

                                % STORE ATTRIBUTE 3
                                satellite_performance_attributes(count,3) = science_value;

                                %% INTERMEDIATE: Calculate MASS
                                % Sum of weights of all basic components
                                BaseMass = weight_structure + weight_thermal + weight_ADCS +
weight_EPS +...
                                    propulsion_weight + TTC_weight + comms_weight; % only
propulsion, TTC and comms weight can be changed for base mass

                                % STORE INTERMEDIATE - 4 BaseMass
                                satellite_intermediate(count,4) = BaseMass;

                                % Sum of total dry mass
                                DryMass = payload_cap(g) + BaseMass +
mass_frac*propellant_mass(h); % affects launch cost
                                % multiplied by drymass_factor to
                                % account for different propulsion
                                % types

                                % STORE INTERMEDIATE - 5 DryMass
                                satellite_intermediate(count,5) = DryMass;

                                % Sum of Total Mass including propellant
                                WetMass = DryMass + propellant_mass(h);

                                % STORE INTERMEDIATE - 6 WetMass
                                satellite_intermediate(count,6) = WetMass;

                                %% ATTRIBUTE 4: Delta V
                                gravity = 9.8;
                                delta_V = gravity*Isp*log(WetMass/DryMass);
                                % Satellites in LEO travel at 6900 -
                                % 7800m/s
                                % Need to find the lower bound of angle
                                % change possible, so use highest
```

```
                                    % orbital velocity possible at LEO

                                    % STORE ATTRIBUTE 4
                                    satellite_performance_attributes(count,4) = delta_V;

                                    %% EXPENSE 1: Calculate DEVELOPMENT COST

                                    % Refer to SMAD Pg 297 Table 11-7
                                    development_cost_BaseMass =
development_cost_factor_to_BaseMass*(22.8*weight_structure + 22.8*weight_thermal + 165.9*weight_ADCS +
25.5*weight_EPS +...
                                        44.1*propulsion_weight + 82.5*weight_TTC + 97.8*comms_weight);

                                    if spacecraft_lifetime(a)==10
                                        development_cost_per_satellite =
1.0*development_cost_BaseMass;
                                        development_time = 1.0*development_time;
                                    elseif spacecraft_lifetime(a)==15
                                        development_cost_per_satellite =
1.25*development_cost_BaseMass;
                                        development_time = 1.2*development_time;
                                    end

                                    % STORE EXPENSE 1
                                    satellite_expense_attributes(count,1) =
development_cost_per_satellite;

                                    %% EXPENSE 2: Calculate LAUNCH COST

                                    % Falcon 9 v 1.1- $4,109
                                    % Falcon 9 Heavy - $2, 547
                                    % Delta IV- $13,072
                                    % Atlas V- $13,182

                                    if launcher(i)==1 % Falcon 9
                                        launch_cost_per_satellite = 4109*WetMass/1000000;
                                    elseif launcher(i)==2 % Falcon 9 Heavy
                                        launch_cost_per_satellite = 2547*WetMass/1000000;
                                    elseif launcher(i)==3
                                        launch_cost_per_satellite = 5072*WetMass/1000000;
                                    elseif launcher(i)==4
                                        launch_cost_per_satellite = 6182*WetMass/1000000;
                                    end

                                    if ReusableLV==0

                                        % STORE EXPENSE 2
                                        satellite_expense_attributes(count,2) =
launch_cost_per_satellite;

                                    elseif ReusableLV==1

                                        satellite_expense_attributes(count,2) =
0.5*launch_cost_per_satellite; %half cost
                                    end

                                    %% EXPENSE 3: Calculate SCHEDULE
                                    % STORE EXPENSE 3
                                    satellite_expense_attributes(count,3) = development_time;
                                    % Store Extra Expense - Total Cost
                                    satellite_expense_attributes(count,4) =
development_cost_per_satellite + launch_cost_per_satellite;
                                    %% COUNTER
                                    count = count + 1;
                                end
                            end
                        end
                    end
                end
            end
        end
    end
end

save satellite_performance_attributes
save satellite_expense_attributes
save satellite_intermediate
save system_DV
end
```

This section contains the MATLAB code written to **generate the weights and preferences sets** for an FSS-enabled satellite in 24 different epochs. Brief descriptions of the epochs are also provided.

```
%% This code is written by Marcus Shihong Wu.
%% Copyright (c) 2014, SEARI MIT.
%% Generate Weights and Preferences for all 24 Epochs for Single Satelliteexp

% Create weights for 6 missions for satellite system

% Performance Attribute 1 - ISL Data Rate
% Performance Attribute 2 - Data Packet Available per Satellite
% Performance Attribute 3 - Science Value
% Performance Attribute 4 - Delta-V

% Expense Attribute 1 - Development Cost
% Expense Attribute 2 - Launch Cost
% Expense Attribute 3 - Development Time

% Mission 0 - Baseline, All equally important
% Mission 1 - Current: Emphasis on Science Value and Delta V, Emphasis on Launch Cost
% Mission 2 - FSS Datellite Development: Emphasis on Data Packet Available per
% Satellite and ISL data rate, Emphasis on LOWER DEVELOPMENT COST since you
% need to drive down cost to develop new technology in order to realize the
% aforementioned attributes
% Mission 3 - FSS Initialization: Even more emphasis most on ISL Data Rate and Data packet available,
Emphasis on
% lowering launch cost and development time since we need to launch as many
% satellites as possible to meet market demand for FSS services
% Mission 4 - FSS Discovery: Equal Emphasis on ISL Data Rate, Data Packet Available and
% Science Value only, Equal emphasis on launch cost and development time
% Mission 5 - FSS Reconfiguration: Emphasis most on Delta_V, Emphasis on
% lowering Development cost and Launch cost

% 2 context variables - Availability of Laser, Availability of Reusable
% 00 - No Laser, No RLV
% 01 - No Laser, Got RLV
% 10 - Got Laser, No RLV
% 11 - Got Laser, Got RLV

% Launch Vehicle
% Gives rise to 4 different contexts
% 6 missions x 4 operating scenarios =  24 Epochs

%% Set up Weights for System Performance Attributes

% E1 to E12 = RLV does not affect any performance attributes, only cost

% E1 - 0.25 0.25 0.25 0.25 - All even in 00
% E2 - 0.1 0.1 0.4 0.4 - Emphasis on Science Value and Delta in 00
% E3 - 0.3 0.3 0.2 0.2 - Emphasis on ISL and Data Available in 00
% E4 - 0.4 0.4 0.1 0.1 - More Emphasis on ISL and Data Available in 00
% E5 - 0.3 0.3 0.3 0.1 - Equal Emphasis on ISL,Data,Science Value in 00
% E6 - 0.1 0.1 0.1 0.7 - Most emphasis on Delta V

% E7 - 0.25 0.25 0.25 0.25 - All even in 01
% E8 - 0.1 0.1 0.4 0.4 - Emphasis on Science Value and Delta in 01
% E9 - 0.3 0.3 0.2 0.2 - Emphasis on ISL and Data Available in 01
% E10 - 0.4 0.4 0.1 0.1 - More Emphasis on ISL and Data Available in 01
% E11 - 0.3 0.3 0.3 0.1 - Equal Emphasis on ISL,Data,Science Value in 01
% E12 - 0.1 0.1 0.1 0.7 - Most emphasis on Delta V

% E13 to E24 - Laser changes some performance attributes

% E13 - 0.4 0.2 0.2 0.2 - Got Laser in 10 - Focus on ISL
% E14 - 0.2 0.1 0.35 0.35 - Emphasis on Science Value and Delta V in 10
% E15 - 0.4 0.4 0.1 0.1 - Emphasis on ISL and Data Available in 10
% E16 - 0.45 0.45 0.05 0.05 - More Emphasis on ISL and Data Available in 10
% E17 - 0.4 0.25 0.25 0.1 - Equal Emphasis on ISL,Data,Science Value in 10
% E18 - 0.3 0.1 0.1 0.5 - Most emphasis on Delta V

% E19 - 0.4 0.2 0.2 0.2 - All even in 00
% E20 - 0.2 0.1 0.35 0.35 - Emphasis on Science Value and Delta in 11
% E21 - 0.4 0.4 0.1 0.1 - Emphasis on ISL and Data Available in 11
% E22 - 0.45 0.45 0.05 0.05 - More Emphasis on ISL and Data Available in 11
% E23 - 0.4 0.25 0.25 0.1 - Equal Emphasis on ISL,Data,Science Value in 11
% E24 - 0.3 0.1 0.1 0.5 - Most emphasis on Delta V
```

```
system00_performance_weights = [0.25 0.25 0.25 0.25;
    0.1 0.1 0.4 0.4;
    0.3 0.3 0.2 0.2;
    0.4 0.4 0.1 0.1;
    0.3 0.3 0.3 0.1;
    0.1 0.1 0.1 0.7];

system01_performance_weights =  [0.25 0.25 0.25 0.25;
    0.1 0.1 0.4 0.4;
    0.3 0.3 0.2 0.2;
    0.4 0.4 0.1 0.1;
    0.3 0.3 0.3 0.1;
    0.1 0.1 0.1 0.7];

system10_performance_weights =  [0.4 0.2 0.2 0.2;
    0.2 0.1 0.35 0.35;
    0.4 0.4 0.1 0.1;
    0.45 0.45 0.05 0.05;
    0.4 0.25 0.25 0.1;
    0.3 0.1 0.1 0.5];

system11_performance_weights =  [0.4 0.2 0.2 0.2;
    0.2 0.1 0.35 0.35;
    0.4 0.4 0.1 0.1;
    0.45 0.45 0.05 0.05;
    0.4 0.25 0.25 0.1;
    0.3 0.1 0.1 0.5];

save system00_performance_weights;
save system01_performance_weights;
save system10_performance_weights;
save system11_performance_weights;

%% Set up Weights for System Expense Attributes

% E1 - 0.34 0. 0.33 0.33 - All even in 00
% E2 - 0.25 0.5 0.25  - Emphasis on launch Cost in 00
% E3 - 0.5 0.25 0.25  - Emphasis on development cost in 00
% E4 - 0.1 0.45 0.45  - Emphasis on launch cost and development time in 00
% E5 - 0.2 0.4 0.4  - Emphasis on launch cost and development time in 00
% E6 - 0.4 0.4 0.2  - Emphasis on development cost and launch cost in 00
% E7 - 0.4 0.2 0.4 - Got RLV in 01 - If got RLV, launch cost is lower!
% E8 - 0.3 0.4 0.3  - Emphasis on launch Cost in 01 - Less emphasis now
% E9 - 0.6 0.2 0.2  - Emphasis on development cost in 01
% E10 - 0.2 0.35 0.45  - Emphasis on launch cost and development time in 01
% E11 - 0.3 0.3 0.4  - Emphasis on launch cost and development time in 01
% E12 - 0.5 0.3 0.2  - Emphasis on development cost and launch cost in 01
% E13 - 0.2 0.4 0.4 - Got Laser in 10
% E14 - 0.15 0.6 0.25  - Emphasis on launch Cost in 10
% E15 - 0.4 0.3 0.3  - Emphasis on development cost in 10
% E16 - 0.05 0.5 0.45  - Emphasis on launch cost and development time in 10
% E17 - 0.1 0.45 0.45  - Emphasis on launch cost and development time in 10
% E18 - 0.3 0.5 0.2  - Emphasis on development cost and launch cost in 10
% E19 - 0.34 0.33 0.33 - Got Laser and RLV in 11
% E20 - 0.3 0.3 0.4  - Emphasis on launch Cost in 11
% E21 - 0.4 0.3 0.3  - Emphasis on development cost in 11
% E22 - 0.1 0.4 0.5  - Emphasis on launch cost and development time in 11
% E23 - 0.1 0.3 0.6  - Emphasis on launch cost and development time in 11
% E24 - 0.35 0.35 0.3  - Emphasis on development cost and launch cost in 11

system00_expense_weights = [0.34 0.33 0.33;
    0.25 0.5 0.25;
    0.5 0.25 0.25;
    0.1 0.45 0.45;
    0.2 0.4 0.4;
    0.4 0.4 0.2];

system01_expense_weights =  [0.4 0.2 0.4;
    0.3 0.4 0.3;
    0.6 0.2 0.2;
    0.2 0.35 0.45;
    0.3 0.3 0.4;
    0.5 0.3 0.2];

system10_expense_weights =  [0.2 0.4 0.4;
    0.15 0.6 0.25;
    0.4 0.3 0.3;
    0.05 0.5 0.45;
    0.1 0.45 0.45;
    0.3 0.5 0.2];
```

```
system11_expense_weights =   [0.34 0.33 0.33;
    0.3 0.3 0.4;
    0.4 0.3 0.3;
    0.1 0.4 0.5;
    0.1 0.3 0.6;
    0.35 0.35 0.3];

save system00_expense_weights;
save system01_expense_weights;
save system10_expense_weights;
save system11_expense_weights;

%% Set up System Performance Preferences
system00_performance_preferences = {};
system01_performance_preferences = {};
system10_performance_preferences = {};
system11_performance_preferences = {};

% SAU for ISL Data Rate
system00_performance_preferences{1,1}=[1,0;2,0.5;3,1]; %00
system00_performance_preferences{2,1}=[1,0;2,0.5;3,1];
system00_performance_preferences{3,1}=[1,0;2,0.4;3,1];
system00_performance_preferences{4,1}=[1,0;2,0.2;3,1];
system00_performance_preferences{5,1}=[1,0;2,0.4;3,1];
system00_performance_preferences{6,1}=[1,0;2,0.5;3,1];

system01_performance_preferences{1,1}=[1,0;2,0.5;3,1];%01 Got RLV
system01_performance_preferences{2,1}=[1,0;2,0.5;3,1];
system01_performance_preferences{3,1}=[1,0;2,0.4;3,1];
system01_performance_preferences{4,1}=[1,0;2,0.2;3,1];
system01_performance_preferences{5,1}=[1,0;2,0.4;3,1];
system01_performance_preferences{6,1}=[1,0;2,0.5;3,1];

system10_performance_preferences{1,1}=[1,0;2,0.3;3,0.6;4,1];%10 Got Laser
system10_performance_preferences{2,1}=[1,0;2,0.4;3,0.7;4,1];
system10_performance_preferences{3,1}=[1,0;2,0.25;3,0.5;4,1];
system10_performance_preferences{4,1}=[1,0;2,0.15;3,0.35;4,1];
system10_performance_preferences{5,1}=[1,0;2,0.3;3,0.5;4,1];
system10_performance_preferences{6,1}=[1,0;2,0.3;3,0.6;4,1];

system11_performance_preferences{1,1}=[1,0;2,0.3;3,0.6;4,1];%11
system11_performance_preferences{2,1}=[1,0;2,0.4;3,0.7;4,1];
system11_performance_preferences{3,1}=[1,0;2,0.25;3,0.5;4,1];
system11_performance_preferences{4,1}=[1,0;2,0.15;3,0.35;4,1];
system11_performance_preferences{5,1}=[1,0;2,0.3;3,0.5;4,1];
system11_performance_preferences{6,1}=[1,0;2,0.3;3,0.6;4,1];

% SAU for Data Packet Available per Satellite
system00_performance_preferences{1,2}=[0,0;10000,0.25;20000,0.5;30000,0.75;40000,1];%00
system00_performance_preferences{2,2}=[0,0;10000,0.25;20000,0.5;30000,0.75;40000,1];
system00_performance_preferences{3,2}=[5000,0;15000,0.25;25000,0.5;40000,1];
system00_performance_preferences{4,2}=[10000,0;20000,0.25;30000,0.5;45000,1];
system00_performance_preferences{5,2}=[5000,0;10000,0.25;20000,0.5;30000,0.75;45000,1];
system00_performance_preferences{6,2}=[5000,0;15000,0.4;30000,0.7;45000,1];

system01_performance_preferences{1,2}=[0,0;10000,0.25;20000,0.5;30000,0.75;40000,1];%01
system01_performance_preferences{2,2}=[0,0;10000,0.25;20000,0.5;30000,0.75;40000,1];
system01_performance_preferences{3,2}=[5000,0;15000,0.25;25000,0.5;40000,1];
system01_performance_preferences{4,2}=[10000,0;20000,0.25;30000,0.5;45000,1];
system01_performance_preferences{5,2}=[5000,0;10000,0.25;20000,0.5;30000,0.75;45000,1];
system01_performance_preferences{6,2}=[5000,0;15000,0.4;30000,0.7;45000,1];

system10_performance_preferences{1,2}=[0,0;10000,0.25;20000,0.5;30000,0.75;40000,1];%10
system10_performance_preferences{2,2}=[0,0;10000,0.25;20000,0.5;30000,0.75;40000,1];
system10_performance_preferences{3,2}=[5000,0;15000,0.25;25000,0.5;40000,1];
system10_performance_preferences{4,2}=[10000,0;20000,0.25;30000,0.5;45000,1];
system10_performance_preferences{5,2}=[5000,0;10000,0.25;20000,0.5;30000,0.75;45000,1];
system10_performance_preferences{6,2}=[5000,0;15000,0.4;30000,0.7;45000,1];

system11_performance_preferences{1,2}=[0,0;10000,0.25;20000,0.5;30000,0.75;40000,1];%11
system11_performance_preferences{2,2}=[0,0;10000,0.25;20000,0.5;30000,0.75;40000,1];
system11_performance_preferences{3,2}=[5000,0;15000,0.25;25000,0.5;40000,1];
system11_performance_preferences{4,2}=[10000,0;20000,0.25;30000,0.5;45000,1];
system11_performance_preferences{5,2}=[5000,0;10000,0.25;20000,0.5;30000,0.75;45000,1];
system11_performance_preferences{6,2}=[5000,0;15000,0.4;30000,0.7;45000,1];

% SAU for Science Value
system00_performance_preferences{1,3}=[0,0;2500,0.4;10000,0.7;20000,1];%00
system00_performance_preferences{2,3}=[0,0;2500,0.25;10000,0.5;20000,1];
```

```
system00_performance_preferences{3,3}=[0,0;2500,0.3;10000,0.6;20000,1];
system00_performance_preferences{4,3}=[2000,0;10000,0.4;20000,1];
system00_performance_preferences{5,3}=[0,0;2500,0.3;10000,0.6;20000,1];
system00_performance_preferences{6,3}=[0,0;2500,0.4;10000,0.7;20000,1];

system01_performance_preferences{1,3}=[0,0;2500,0.4;10000,0.7;20000,1];%01
system01_performance_preferences{2,3}=[0,0;2500,0.25;10000,0.5;20000,1];
system01_performance_preferences{3,3}=[0,0;2500,0.3;10000,0.6;20000,1];
system01_performance_preferences{4,3}=[2000,0;10000,0.4;20000,1];
system01_performance_preferences{5,3}=[0,0;2500,0.3;10000,0.6;20000,1];
system01_performance_preferences{6,3}=[0,0;2500,0.4;10000,0.7;20000,1];

system10_performance_preferences{1,3}=[0,0;2500,0.4;10000,0.7;20000,1];%10
system10_performance_preferences{2,3}=[0,0;2500,0.25;10000,0.5;20000,1];
system10_performance_preferences{3,3}=[0,0;2500,0.3;10000,0.6;20000,1];
system10_performance_preferences{4,3}=[2000,0;10000,0.4;20000,1];
system10_performance_preferences{5,3}=[0,0;2500,0.3;10000,0.6;20000,1];
system10_performance_preferences{6,3}=[0,0;2500,0.4;10000,0.7;20000,1];

system11_performance_preferences{1,3}=[0,0;2500,0.4;10000,0.7;20000,1];%11
system11_performance_preferences{2,3}=[0,0;2500,0.25;10000,0.5;20000,1];
system11_performance_preferences{3,3}=[0,0;2500,0.3;10000,0.6;20000,1];
system11_performance_preferences{4,3}=[2000,0;10000,0.4;20000,1];
system11_performance_preferences{5,3}=[0,0;2500,0.3;10000,0.6;20000,1];
system11_performance_preferences{6,3}=[0,0;2500,0.4;10000,0.7;20000,1];

% SAU for Delta V
system00_performance_preferences{1,4}=[0,0;1000,0.2;3500,0.5;10000,0.7;20000,1];%00
system00_performance_preferences{2,4}=[1000,0;3000,0.2;10000,0.5;20000,0.8;25000,1];
system00_performance_preferences{3,4}=[0,0;1000,0.3;3500,0.6;10000,0.8;20000,1];
system00_performance_preferences{4,4}=[0,0;1000,0.35;3500,0.7;10000,0.9;20000,1];
system00_performance_preferences{5,4}=[0,0;1000,0.2;3500,0.5;10000,0.7;20000,1];
system00_performance_preferences{6,4}=[3000,0;5000,0.25;15000,0.5;25000,1];

system01_performance_preferences{1,4}=[0,0;1000,0.2;3500,0.5;10000,0.7;20000,1];%01
system01_performance_preferences{2,4}=[1000,0;3000,0.2;10000,0.5;20000,0.8;25000,1];
system01_performance_preferences{3,4}=[0,0;1000,0.3;3500,0.6;10000,0.8;20000,1];
system01_performance_preferences{4,4}=[0,0;1000,0.35;3500,0.7;10000,0.9;20000,1];
system01_performance_preferences{5,4}=[0,0;1000,0.2;3500,0.5;10000,0.7;20000,1];
system01_performance_preferences{6,4}=[3000,0;5000,0.25;15000,0.5;25000,1];

system10_performance_preferences{1,4}=[0,0;1000,0.2;3500,0.5;10000,0.7;20000,1];%10
system10_performance_preferences{2,4}=[1000,0;3000,0.2;10000,0.5;20000,0.8;25000,1];
system10_performance_preferences{3,4}=[0,0;1000,0.3;3500,0.6;10000,0.8;20000,1];
system10_performance_preferences{4,4}=[0,0;1000,0.35;3500,0.7;10000,0.9;20000,1];
system10_performance_preferences{5,4}=[0,0;1000,0.2;3500,0.5;10000,0.7;20000,1];
system10_performance_preferences{6,4}=[3000,0;5000,0.25;15000,0.5;25000,1];

system11_performance_preferences{1,4}=[0,0;1000,0.2;3500,0.5;10000,0.7;20000,1];%11
system11_performance_preferences{2,4}=[1000,0;3000,0.2;10000,0.5;20000,0.8;25000,1];
system11_performance_preferences{3,4}=[0,0;1000,0.3;3500,0.6;10000,0.8;20000,1];
system11_performance_preferences{4,4}=[0,0;1000,0.35;3500,0.7;10000,0.9;20000,1];
system11_performance_preferences{5,4}=[0,0;1000,0.2;3500,0.5;10000,0.7;20000,1];
system11_performance_preferences{6,4}=[3000,0;5000,0.25;15000,0.5;25000,1];

save system00_performance_preferences;
save system01_performance_preferences;
save system10_performance_preferences;
save system11_performance_preferences;

%% Set up System Expense Preferences
system00_expense_preferences ={};
system01_expense_preferences ={};
system10_expense_preferences ={};
system11_expense_preferences ={};

% Create SAU for Development Cost
system00_expense_preferences{1,1}=[0,0;30000,0.4;50000,0.6;60000,0.8;70000,1];%00
system00_expense_preferences{2,1}=[0,0;30000,0.3;50000,0.6;60000,0.9;70000,1];
system00_expense_preferences{3,1}=[0,0;30000,0.5;50000,0.8;60000,1];
system00_expense_preferences{4,1}=[0,0;30000,0.4;50000,0.6;60000,0.8;70000,1];
system00_expense_preferences{5,1}=[0,0;30000,0.3;50000,0.6;60000,0.9;70000,1];
system00_expense_preferences{6,1}=[0,0;30000,0.4;50000,0.8;65000,1];

system01_expense_preferences{1,1}=[0,0;30000,0.4;50000,0.6;60000,0.8;70000,1];%01- No laser
system01_expense_preferences{2,1}=[0,0;30000,0.3;50000,0.6;60000,0.9;70000,1];
system01_expense_preferences{3,1}=[0,0;30000,0.5;50000,0.8;60000,1];
system01_expense_preferences{4,1}=[0,0;30000,0.4;50000,0.6;60000,0.8;70000,1];
system01_expense_preferences{5,1}=[0,0;30000,0.3;50000,0.6;60000,0.9;70000,1];
system01_expense_preferences{6,1}=[0,0;30000,0.4;50000,0.8;65000,1];
```

```
system10_expense_preferences{1,1}=[0,0;25000,0.4;35000,0.6;45000,0.8;50000,1];%10 - Got laser
system10_expense_preferences{2,1}=[0,0;25000,0.3;35000,0.6;45000,0.9;50000,1];
system10_expense_preferences{3,1}=[0,0;27500,0.5;37500,0.8;45000,1];
system10_expense_preferences{4,1}=[0,0;25000,0.4;35000,0.6;45000,0.8;50000,1];
system10_expense_preferences{5,1}=[0,0;25000,0.3;35000,0.6;45000,0.9;50000,1];
system10_expense_preferences{6,1}=[0,0;27500,0.4;37500,0.8;45000,1];

system11_expense_preferences{1,1}=[0,0;25000,0.4;35000,0.6;45000,0.8;50000,1];%11 - Got laser
system11_expense_preferences{2,1}=[0,0;25000,0.3;35000,0.6;45000,0.9;50000,1];
system11_expense_preferences{3,1}=[0,0;27500,0.5;37500,0.8;45000,1];
system11_expense_preferences{4,1}=[0,0;25000,0.4;35000,0.6;45000,0.8;50000,1];
system11_expense_preferences{5,1}=[0,0;25000,0.3;35000,0.6;45000,0.9;50000,1];
system11_expense_preferences{6,1}=[0,0;27500,0.4;37500,0.8;45000,1];

% Create SAU for Launch Cost
system00_expense_preferences{1,2}=[0,0;5,0.2;10,0.4;15,0.6;18,0.8;22,1];%00
system00_expense_preferences{2,2}=[0,0;9.5,0.5;13,0.75;15,1];
system00_expense_preferences{3,2}=[0,0;5,0.25;10,0.45;15,0.65;18,0.8;25,1];
system00_expense_preferences{4,2}=[0,0;9.5,0.5;13,0.75;15,1];
system00_expense_preferences{5,2}=[0,0;12,0.5;15,0.75;20,1];
system00_expense_preferences{6,2}=[0,0;12,0.5;15,0.75;20,1];

system01_expense_preferences{1,2}=[0,0;3.5,0.3;6.5,0.6;10,0.9;13,1];%01 - Got RLV - Launch cost pref
should change!
system01_expense_preferences{2,2}=[0,0;3.5,0.25;6.5,0.5;10,0.7;13.5,1]; % Relax prefs because of lower
launch cost
system01_expense_preferences{3,2}=[0,0;3.5,0.3;6.5,0.6;10,0.9;13,1];
system01_expense_preferences{4,2}=[0,0;3.5,0.25;6.5,0.5;10,0.7;13.5,1];
system01_expense_preferences{5,2}=[0,0;3.5,0.25;6.5,0.5;10,0.7;13.5,1];
system01_expense_preferences{6,2}=[0,0;3.5,0.25;6.5,0.5;10,0.7;13.5,1];

system10_expense_preferences{1,2}=[0,0;5,0.2;10,0.4;15,0.6;18,0.8;22,1];%10 - No RLV, same as initial
system10_expense_preferences{2,2}=[0,0;9.5,0.5;13,0.75;15,1];
system10_expense_preferences{3,2}=[0,0;5,0.25;10,0.45;15,0.65;18,0.8;25,1];
system10_expense_preferences{4,2}=[0,0;9.5,0.5;13,0.75;15,1];
system10_expense_preferences{5,2}=[0,0;12,0.5;15,0.75;20,1];
system10_expense_preferences{6,2}=[0,0;12,0.5;15,0.75;20,1];

system11_expense_preferences{1,2}=[0,0;3.5,0.3;6.5,0.6;10,0.9;13,1];%11 Got RLV - Launch cost pref
should change!
system11_expense_preferences{2,2}=[0,0;3.5,0.25;6.5,0.5;10,0.7;13.5,1];
system11_expense_preferences{3,2}=[0,0;3.5,0.3;6.5,0.6;10,0.9;13,1];
system11_expense_preferences{4,2}=[0,0;3.5,0.25;6.5,0.5;10,0.7;13.5,1];
system11_expense_preferences{5,2}=[0,0;3.5,0.25;6.5,0.5;10,0.7;13.5,1];
system11_expense_preferences{6,2}=[0,0;3.5,0.25;6.5,0.5;10,0.7;13.5,1];

% Create SAU for Development Time
system00_expense_preferences{1,3}=[0,0;1.5,0.4;3,0.7;5,1];%00
system00_expense_preferences{2,3}=[0,0;1.5,0.4;3.5,0.8;5,1];
system00_expense_preferences{3,3}=[0,0;1.5,0.4;3.5,0.8;5,1];
system00_expense_preferences{4,3}=[0,0;1,0.4;2,0.6;3,0.8;4,1];
system00_expense_preferences{5,3}=[0,0;1,0.4;2,0.6;3,0.8;4,1];
system00_expense_preferences{6,3}=[0,0;1.5,0.4;3.5,0.8;5,1];

system01_expense_preferences{1,3}=[0,0;1.5,0.4;3,0.7;5,1];%01 - No laser yet, pref should not change
system01_expense_preferences{2,3}=[0,0;1.5,0.4;3.5,0.8;5,1];
system01_expense_preferences{3,3}=[0,0;1.5,0.4;3.5,0.8;5,1];
system01_expense_preferences{4,3}=[0,0;1,0.4;2,0.6;3,0.8;4,1];
system01_expense_preferences{5,3}=[0,0;1,0.4;2,0.6;3,0.8;4,1];
system01_expense_preferences{6,3}=[0,0;1.5,0.4;3.5,0.8;5,1];

system10_expense_preferences{1,3}=[0,0;1,0.4;1.55,0.7;2.1,0.8;2.5,1];%10 - Got laser, pref changes!
system10_expense_preferences{2,3}=[0,0;1,0.3;1.55,0.5;2.1,0.7;2.5,1];
system10_expense_preferences{3,3}=[0,0;1,0.3;1.55,0.5;2.1,0.7;2.5,1];
system10_expense_preferences{4,3}=[0,0;1,0.5;1.5,0.8;2,1];
system10_expense_preferences{5,3}=[0,0;1,0.5;1.5,0.8;2,1];
system10_expense_preferences{6,3}=[0,0;1,0.3;1.55,0.5;2.1,0.7;2.5,1];

system11_expense_preferences{1,3}=[0,0;1,0.4;1.55,0.7;2.1,0.8;2.5,1];%11
system11_expense_preferences{2,3}=[0,0;1,0.3;1.55,0.5;2.1,0.7;2.5,1];
system11_expense_preferences{3,3}=[0,0;1,0.3;1.55,0.5;2.1,0.7;2.5,1];
system11_expense_preferences{4,3}=[0,0;1,0.5;1.5,0.8;2,1];
system11_expense_preferences{5,3}=[0,0;1,0.5;1.5,0.8;2,1];
system11_expense_preferences{6,3}=[0,0;1,0.3;1.55,0.5;2.1,0.7;2.5,1];

save system00_expense_preferences;
save system01_expense_preferences;
save system10_expense_preferences;
save system11_expense_preferences;
```

This section contains the MATLAB function written to **generate the upper and lower bounds for a 99% confidence interval using lognormal distributions.** This was applied to only Pareto frontier set solutions.

```
%% Get upper and lower interval values for expense attributes
%% This code is written by Marcus Shihong Wu.
%% Copyright (c) 2014, SEARI MIT.

function [Pareto_Expense_Attributes_LOWER, Pareto_Expense_Attributes_UPPER] =
Get_Pareto_ExpenseAttribute_Intervals(Pareto_Expense_Attributes,sigma_devcost,sigma_launchcost,sigma_s
chedule)

% sigma_devcost = 0.1;
% sigma_launchcost = 0.05;
% sigma_schedule = 0.15;

Pareto_Expense_Attributes_LOWER = cell([1,size(Pareto_Expense_Attributes,2)]);
Pareto_Expense_Attributes_UPPER = cell([1,size(Pareto_Expense_Attributes,2)]);

for i = 1:size(Pareto_Expense_Attributes,2)

    count = 1;

    for j = 1:size(Pareto_Expense_Attributes{i},1)

        Pareto_Expense_Attributes_LOWER{i}(count,1) =
logninv(0.005,0,sigma_devcost)*Pareto_Expense_Attributes{i}(count,1);
        Pareto_Expense_Attributes_LOWER{i}(count,2) =
logninv(0.005,0,sigma_launchcost)*Pareto_Expense_Attributes{i}(count,2);
        Pareto_Expense_Attributes_LOWER{i}(count,3) =
logninv(0.005,0,sigma_schedule)*Pareto_Expense_Attributes{i}(count,3);

        Pareto_Expense_Attributes_UPPER{i}(count,1) =
logninv(0.995,0,sigma_devcost)*Pareto_Expense_Attributes{i}(count,1);
        Pareto_Expense_Attributes_UPPER{i}(count,2) =
logninv(0.995,0,sigma_launchcost)*Pareto_Expense_Attributes{i}(count,2);
        Pareto_Expense_Attributes_UPPER{i}(count,3) =
logninv(0.995,0,sigma_schedule)*Pareto_Expense_Attributes{i}(count,3);

        count = count + 1;
    end

end

end
```

This section contains the MATLAB mode written to **plot the 99% confidence interval using lognormal distributions for Pareto frontier set solutions of 7 epochs.**

```
%% This code is written by Marcus Shihong Wu.
%% Copyright (c) 2014, SEARI MIT.
%% Get upper and lower intervals for expense attributes

[Sys_00_Expense_Lower, Sys_00_Expense_Upper] =
Get_Pareto_ExpenseAttribute_Intervals(Sys_00_Pareto_Expense_Attributes,0.1,0.05,0.15);

[Sys_01_Expense_Lower, Sys_01_Expense_Upper] =
Get_Pareto_ExpenseAttribute_Intervals(Sys_01_Pareto_Expense_Attributes,0.1,0.15,0.15);

[Sys_10_Expense_Lower, Sys_10_Expense_Upper] =
Get_Pareto_ExpenseAttribute_Intervals(Sys_10_Pareto_Expense_Attributes,0.25,0.05,0.2);

[Sys_11_Expense_Lower, Sys_11_Expense_Upper] =
Get_Pareto_ExpenseAttribute_Intervals(Sys_11_Pareto_Expense_Attributes,0.25,0.15,0.2);

%% Create cells to store lower MAE thresholds, upper MAE thresholds, and Pareto MAU

MAE_Lower_System_00 = cell(1,size(Sys_00_Expense_Lower,2));
MAE_Upper_System_00 = cell(1,size(Sys_00_Expense_Upper,2));
MAU_Pareto_Only_System_00 = cell(1,size(Sys_00_Expense_Upper,2));

MAE_Lower_System_01 = cell(1,size(Sys_01_Expense_Lower,2));
MAE_Upper_System_01 = cell(1,size(Sys_01_Expense_Upper,2));
MAU_Pareto_Only_System_00 = cell(1,size(Sys_00_Expense_Upper,2));

MAE_Lower_System_10 = cell(1,size(Sys_10_Expense_Lower,2));
MAE_Upper_System_10 = cell(1,size(Sys_10_Expense_Upper,2));
MAU_Pareto_Only_System_00 = cell(1,size(Sys_00_Expense_Upper,2));

MAE_Lower_System_11 = cell(1,size(Sys_11_Expense_Lower,2));
MAE_Upper_System_11 = cell(1,size(Sys_11_Expense_Upper,2));
MAU_Pareto_Only_System_00 = cell(1,size(Sys_00_Expense_Upper,2));

%% Calculate for 00

% requires upper and lower thresholds for MAE, prefs, weights
for i = 1:size(Sys_00_Expense_Lower,2)

    [MAE_lower00] =
Calculate_MAE(Sys_00_Expense_Lower{i},system00_expense_preferences(i,:),system00_expense_weights(i,:))
;
    [MAE_upper00] =
Calculate_MAE(Sys_00_Expense_Upper{i},system00_expense_preferences(i,:),system00_expense_weights(i,:))
;
    [MAU_00] =
Calculate_MAU(Sys_00_Pareto_Performance_Attributes{i},system00_performance_preferences(i,:),system01_p
erformance_weights(i,:));

    MAE_Lower_System_00{i} = MAE_lower00;
    MAE_Upper_System_00{i} = MAE_upper00;
    MAU_Pareto_Only_System_00{i} = MAU_00;

end

%Plot_Pareto_System_Only(clean_pareto_set_MAE_00(:,1), clean_pareto_set_MAU_00(:,1));
for j=1:size(Sys_00_Pareto_MAE,2)
    figure()
    scatter(Sys_00_Pareto_MAE{j},Sys_00_Pareto_MAU{j},'r','fill');
    title(['Pareto Set Only with 99% Lognormal CI - Context 00 - Scatter Plot of MAU vs MAE for
Mission ',num2str(j)]);
    axis([0 1 0 1]);
    xlabel('MAE');
    ylabel('MAU');
    hold on
    scatter(MAE_Lower_System_00{j},MAU_Pareto_Only_System_00{j});
    hold on
    scatter(MAE_Upper_System_00{j},MAU_Pareto_Only_System_00{j});
    hold on

    for i = 1:size(MAE_Lower_System_00{j},1)
        plot([MAE_Lower_System_00{j}(i) MAE_Upper_System_00{j}(i)],[MAU_Pareto_Only_System_00{j}(i)
MAU_Pareto_Only_System_00{j}(i)]);
```

```
        hold on
    end
end

%% Calculate for 01

% requires upper and lower thresholds for MAE, prefs, weights
for i = 1:size(Sys_01_Expense_Lower,2)

    [MAE_lower01] =
Calculate_MAE(Sys_01_Expense_Lower{i},system01_expense_preferences(i,:),system01_expense_weights(i,:))
;
    [MAE_upper01] =
Calculate_MAE(Sys_01_Expense_Upper{i},system01_expense_preferences(i,:),system01_expense_weights(i,:))
;
    [MAU_01] =
Calculate_MAU(Sys_01_Pareto_Performance_Attributes{i},system01_performance_preferences(i,:),system01_p
erformance_weights(i,:));

    MAE_Lower_System_01{i} = MAE_lower01;
    MAE_Upper_System_01{i} = MAE_upper01;
    MAU_Pareto_Only_System_01{i} = MAU_01;

end

%Plot_Pareto_System_Only(clean_pareto_set_MAE_00(:,1), clean_pareto_set_MAU_00(:,1));
for j=1:size(Sys_01_Pareto_MAE,2)
    figure()
    scatter(Sys_01_Pareto_MAE{j},Sys_01_Pareto_MAU{j},'r','fill');
    title(['Pareto Set Only with 99% Lognormal CI - Context 01- Scatter Plot of MAU vs MAE for Mission
',num2str(j)]);
    axis([0 1 0 1]);
    xlabel('MAE');
    ylabel('MAU');
    hold on
    scatter(MAE_Lower_System_01{j},MAU_Pareto_Only_System_01{j});
    hold on
    scatter(MAE_Upper_System_01{j},MAU_Pareto_Only_System_01{j});
    hold on

    for i = 1:size(MAE_Lower_System_01{j},1)
        plot([MAE_Lower_System_01{j}(i) MAE_Upper_System_01{j}(i)],[MAU_Pareto_Only_System_01{j}(i)
MAU_Pareto_Only_System_01{j}(i)]);
        hold on
    end
end

%% Calculate for 10

% requires upper and lower thresholds for MAE, prefs, weights
for i = 1:size(Sys_10_Expense_Lower,2)

    [MAE_lower10] =
Calculate_MAE(Sys_10_Expense_Lower{i},system10_expense_preferences(i,:),system10_expense_weights(i,:))
;
    [MAE_upper10] =
Calculate_MAE(Sys_10_Expense_Upper{i},system10_expense_preferences(i,:),system10_expense_weights(i,:))
;
    [MAU_10] =
Calculate_MAU(Sys_10_Pareto_Performance_Attributes{i},system10_performance_preferences(i,:),system10_p
erformance_weights(i,:));

    MAE_Lower_System_10{i} = MAE_lower10;
    MAE_Upper_System_10{i} = MAE_upper10;
    MAU_Pareto_Only_System_10{i} = MAU_10;

end

%Plot_Pareto_System_Only(clean_pareto_set_MAE_00(:,1), clean_pareto_set_MAU_00(:,1));
for j=1:size(Sys_10_Pareto_MAE,2)
    figure()
    scatter(Sys_10_Pareto_MAE{j},Sys_10_Pareto_MAU{j},'r','fill');
    title(['Pareto Set Only with 99% Lognormal CI - Context 10 - Scatter Plot of MAU vs MAE for
Mission ',num2str(j)]);
    axis([0 1 0 1]);
    xlabel('MAE');
    ylabel('MAU');
    hold on
    scatter(MAE_Lower_System_10{j},MAU_Pareto_Only_System_10{j});
    hold on
```

```
    scatter(MAE_Upper_System_10{j},MAU_Pareto_Only_System_10{j});
    hold on

    for i = 1:size(MAE_Lower_System_10{j},1)
        plot([MAE_Lower_System_10{j}(i) MAE_Upper_System_10{j}(i)],[MAU_Pareto_Only_System_10{j}(i)
MAU_Pareto_Only_System_10{j}(i)]);
        hold on
    end
end

%% Calculate for 11

% requires upper and lower thresholds for MAE, prefs, weights
for i = 1:size(Sys_11_Expense_Lower,2)

    [MAE_lower11] =
Calculate_MAE(Sys_11_Expense_Lower{i},system11_expense_preferences(i,:),system11_expense_weights(i,:))
;
    [MAE_upper11] =
Calculate_MAE(Sys_11_Expense_Upper{i},system11_expense_preferences(i,:),system11_expense_weights(i,:))
;
    [MAU_11] =
Calculate_MAU(Sys_11_Pareto_Performance_Attributes{i},system11_performance_preferences(i,:),system11_p
erformance_weights(i,:));

    MAE_Lower_System_11{i} = MAE_lower11;
    MAE_Upper_System_11{i} = MAE_upper11;
    MAU_Pareto_Only_System_11{i} = MAU_11;

end

%Plot_Pareto_System_Only(clean_pareto_set_MAE_00(:,1), clean_pareto_set_MAU_00(:,1));
for j=1:size(Sys_11_Pareto_MAE,2)
    figure()
    scatter(Sys_11_Pareto_MAE{j},Sys_11_Pareto_MAU{j},'r','fill');
    title(['Pareto Set Only with 99% Lognormal CI - Context 11 - Scatter Plot of MAU vs MAE for
Mission ',num2str(j)]);
    axis([0 1 0 1]);
    xlabel('MAE');
    ylabel('MAU');
    hold on
    scatter(MAE_Lower_System_11{j},MAU_Pareto_Only_System_11{j});
    hold on
    scatter(MAE_Upper_System_11{j},MAU_Pareto_Only_System_11{j});
    hold on

    for i = 1:size(MAE_Lower_System_11{j},1)
        plot([MAE_Lower_System_11{j}(i) MAE_Upper_System_11{j}(i)],[MAU_Pareto_Only_System_11{j}(i)
MAU_Pareto_Only_System_11{j}(i)]);
        hold on
    end
```

This section contains the MATLAB mode written to **construct the constraint levels on performance and expense attributes.**

```
%% Construct constraints on performance attributes
% [ISL-Data-Rate Data-Packet-Capacity Science-Value Delta-V]
Mission_0_Context_00_Performance_Attribute_Constraints = [2 35000 1000 5000];
Mission_1_Context_00_Performance_Attribute_Constraints = [2 40000 10000 10000];
Mission_2_Context_00_Performance_Attribute_Constraints = [2 40000 5000 5000];
Mission_3_Context_01_Performance_Attribute_Constraints = [2 40000 2500 5000];
Mission_4_Context_10_Performance_Attribute_Constraints = [3 35000 10000 15000];
Mission_5_Context_11_Performance_Attribute_Constraints = [3 35000 10000 15000];
Mission_0_Context_11_Performance_Attribute_Constraints = [3 40000 2000 10000];

All_Performance_Attribute_Constraints=[Mission_0_Context_00_Performance_Attribute_Constraints;
    Mission_1_Context_00_Performance_Attribute_Constraints;
    Mission_2_Context_00_Performance_Attribute_Constraints;
    Mission_3_Context_01_Performance_Attribute_Constraints;
    Mission_4_Context_10_Performance_Attribute_Constraints;
    Mission_5_Context_11_Performance_Attribute_Constraints;
    Mission_0_Context_11_Performance_Attribute_Constraints];

%% Construct constraints on expense attributes

Mission_0_Context_00_Expense_Attribute_Constraints = [35000 4 2];
Mission_1_Context_00_Expense_Attribute_Constraints = [37500 3.5 2];
Mission_2_Context_00_Expense_Attribute_Constraints = [32500 4 2];
Mission_3_Context_01_Expense_Attribute_Constraints = [35000 2 1.5];
Mission_4_Context_10_Expense_Attribute_Constraints = [25000 4 1];
Mission_5_Context_11_Expense_Attribute_Constraints = [25000 2 1];
Mission_0_Context_11_Expense_Attribute_Constraints = [25000 2 1];


All_Expense_Attribute_Constraints=[Mission_0_Context_00_Expense_Attribute_Constraints;
    Mission_1_Context_00_Expense_Attribute_Constraints;
    Mission_2_Context_00_Expense_Attribute_Constraints;
    Mission_3_Context_01_Expense_Attribute_Constraints;
    Mission_4_Context_10_Expense_Attribute_Constraints;
    Mission_5_Context_11_Expense_Attribute_Constraints;
    Mission_0_Context_11_Expense_Attribute_Constraints];
%% Construct constraint levels for PERFORMANCE - preferences for epochs 1,2,3,10,16,23,19

Performance_Preferences_Constraints = {};

% for ISL
Performance_Preferences_Constraints{1,1}=[1,0;2,0.5;3,1]; %1
Performance_Preferences_Constraints{2,1}=[1,0;2,0.5;3,1]; %2
Performance_Preferences_Constraints{3,1}=[1,0;2,0.4;3,1]; %3
Performance_Preferences_Constraints{4,1}=[1,0;2,0.2;3,1]; %10
Performance_Preferences_Constraints{5,1}=[1,0;2,0.3;3,0.5;4,1];%17
Performance_Preferences_Constraints{6,1}=[1,0;2,0.3;3,0.6;4,1];%24
Performance_Preferences_Constraints{7,1}=[1,0;2,0.3;3,0.6;4,1];%19

% for Data Packet Available
Performance_Preferences_Constraints{1,2}=[0,0;10000,0.25;20000,0.5;30000,0.75;40000,1]; %1
Performance_Preferences_Constraints{2,2}=[0,0;10000,0.25;20000,0.5;30000,0.75;40000,1]; %2
Performance_Preferences_Constraints{3,2}=[5000,0;15000,0.25;25000,0.5;40000,1]; %3
Performance_Preferences_Constraints{4,2}=[10000,0;20000,0.25;30000,0.5;45000,1]; %10
Performance_Preferences_Constraints{5,2}=[5000,0;10000,0.25;20000,0.5;30000,0.75;45000,1];%17
Performance_Preferences_Constraints{6,2}=[5000,0;15000,0.4;30000,0.7;45000,1];%24
Performance_Preferences_Constraints{7,2}=[0,0;10000,0.25;20000,0.5;30000,0.75;40000,1];%19

% for Science Value
Performance_Preferences_Constraints{1,3}=[0,0;2500,0.4;10000,0.7;20000,1]; %1
Performance_Preferences_Constraints{2,3}=[0,0;2500,0.25;10000,0.5;20000,1]; %2
Performance_Preferences_Constraints{3,3}=[0,0;2500,0.3;10000,0.6;20000,1]; %3
Performance_Preferences_Constraints{4,3}=[2000,0;10000,0.4;20000,1]; %10
Performance_Preferences_Constraints{5,3}=[0,0;2500,0.3;10000,0.6;20000,1];%17
Performance_Preferences_Constraints{6,3}=[0,0;2500,0.4;10000,0.7;20000,1];%24
Performance_Preferences_Constraints{7,3}=[0,0;10000,0.25;20000,0.5;30000,0.75;40000,1];%19

% for Delta V
Performance_Preferences_Constraints{1,4}=[0,0;10000,0.25;20000,0.5;30000,0.75;40000,1]; %1
Performance_Preferences_Constraints{2,4}=[0,0;10000,0.25;20000,0.5;30000,0.75;40000,1]; %2
Performance_Preferences_Constraints{3,4}=[5000,0;15000,0.25;25000,0.5;40000,1]; %3
Performance_Preferences_Constraints{4,4}=[10000,0;20000,0.25;30000,0.5;45000,1]; %10
Performance_Preferences_Constraints{5,4}=[0,0;1000,0.2;3500,0.5;10000,0.7;20000,1];%17
Performance_Preferences_Constraints{6,4}=[3000,0;5000,0.25;15000,0.5;25000,1];%24
Performance_Preferences_Constraints{7,4}=[0,0;2500,0.4;10000,0.7;20000,1];%19
```

```
%% Construct constraint levels for EXPENSES - preferences for epochs 1,2,3,10,16,23,19

Expense_Preferences_Constraints = {};

% for Development Cost
Expense_Preferences_Constraints{1,1} = [0,0;30000,0.4;50000,0.6;60000,0.8;70000,1];
Expense_Preferences_Constraints{2,1} = [0,0;30000,0.3;50000,0.6;60000,0.9;70000,1];
Expense_Preferences_Constraints{3,1} = [0,0;30000,0.5;50000,0.8;60000,1];
Expense_Preferences_Constraints{4,1} = [0,0;30000,0.4;50000,0.6;60000,0.8;70000,1];
Expense_Preferences_Constraints{5,1} = [0,0;25000,0.3;35000,0.6;45000,0.9;50000,1];
Expense_Preferences_Constraints{6,1} = [0,0;27500,0.4;37500,0.8;45000,1];
Expense_Preferences_Constraints{7,1} = [0,0;25000,0.4;35000,0.6;45000,0.8;50000,1];

% for Launch Cost
Expense_Preferences_Constraints{1,2} = [0,0;5,0.2;10,0.4;15,0.6;18,0.8;22,1];
Expense_Preferences_Constraints{2,2} = [0,0;9.5,0.5;13,0.75;15,1];
Expense_Preferences_Constraints{3,2} = [0,0;5,0.25;10,0.45;15,0.65;18,0.8;25,1];
Expense_Preferences_Constraints{4,2} = [0,0;3.5,0.25;6.5,0.5;10,0.7;13.5,1];
Expense_Preferences_Constraints{5,2} = [0,0;12,0.5;15,0.75;20,1];
Expense_Preferences_Constraints{6,2} = [0,0;3.5,0.25;6.5,0.5;10,0.7;13.5,1];
Expense_Preferences_Constraints{7,2} = [0,0;3.5,0.3;6.5,0.6;10,0.9;13,1];

% for Schedule
Expense_Preferences_Constraints{1,3} = [0,0;1.5,0.4;3,0.7;5,1];
Expense_Preferences_Constraints{2,3} = [0,0;1.5,0.4;3.5,0.8;5,1];
Expense_Preferences_Constraints{3,3} = [0,0;1.5,0.4;3.5,0.8;5,1];
Expense_Preferences_Constraints{4,3} = [0,0;1,0.4;2,0.6;3,0.8;4,1];
Expense_Preferences_Constraints{5,3} = [0,0;1,0.5;1.5,0.8;2,1];
Expense_Preferences_Constraints{6,3} = [0,0;1,0.3;1.55,0.5;2.1,0.7;2.5,1];
Expense_Preferences_Constraints{7,3} = [0,0;1,0.4;1.55,0.7;2.1,0.8;2.5,1];
%

%% Construct constraint levels for PERFORMANCE - weights for epochs 1,2,3,10,17,24,19

Performance_Weights_Constraints = [0.25 0.25 0.25 0.25;
    0.1 0.1 0.4 0.4;
    0.3 0.3 0.2 0.2;
    0.4 0.4 0.1 0.1;
    0.4 0.25 0.25 0.1;
    0.3 0.1 0.1 0.5;
    0.4 0.2 0.2 0.2];

%% Construct constraint levels for EXPENSE - weights for epochs 1,2,3,10,17,24,19

Expense_Weights_Constraints = [0.34 0.33 0.33;
    0.25 0.5 0.25;
    0.5 0.25 0.25;
    0.2 0.35 0.45;
    0.1 0.45 0.45;
    0.35 0.35 0.3;
    0.34 0.33 0.33];

%% Calculate MAE and MAU constraint levels

MAE_Constraint_Levels = zeros(size(All_Expense_Attribute_Constraints,1),1);
MAU_Constraint_Levels = zeros(size(All_Performance_Attribute_Constraints,1),1);

for e = 1:size(Performance_Preferences_Constraints,1)

    [MAU_Constraint] =
Calculate_MAU(All_Performance_Attribute_Constraints(e),Performance_Preferences_Constraints(e,:),Perfor
mance_Weights_Constraints(e,:));
    [MAE_Constraint] =
Calculate_MAE(All_Expense_Attribute_Constraints(e),Expense_Preferences_Constraints(e,:),Expense_Weight
s_Constraints(e,:));

    MAE_Constraint_Levels(e) = MAE_Constraint;
    MAU_Constraint_Levels(e) = MAU_Constraint;

end
```

APPENDIX 2:
PROGRAM TRADESPACE MODEL

The MATLAB code for the entire FSS tradespace model will not be included in this appendix. Only important sections are shown. Contact MIT SEAri or the author of this thesis for access to the full code.

This section contains the MATLAB code written to **generate all possible program designs** for an FSS-enabled satellite constellation.

```
%% This code is written by Marcus Shihong Wu.
%% Copyright (c) 2014, SEARI MIT.

function [program_performance_attributes,program_expense_attributes,program_intermediate,
program_DV]...
    =
Generate_Constellation_Design_For_Epoch(Epoch_Expense_Attributes,Epoch_Performance_Attributes,Epoch_In
termediate, Epoch_System_DV)

program_satellite_design_choice = [1 2 3]; % 1 - Low Risk, 2 - Medium Risk, 3 - High Risk
program_number_of_satellites = [4 6 8 10 12]; % number of satellites per constellation
program_num_satellites_concurrent_development = [1 2 3 4]; % number of satellites developed
concurrently per satellite development time
program_constellation_type = [1 2 3]; % 1 - aggregated, 2 - disaggregated, 3 - mixed
program_legacy_operation = [0 1]; % 0 - retire at end of lifetime, 1 - operate as a legacy SoS
program_contract_length = [4 7 10]; % in years
% Total number of design points  = 3 x 5 x 4 x 3 x 2 = 360

program_DV = zeros(1080,6);
program_performance_attributes = zeros(1080,3); % 3 Performance Attributes
program_expense_attributes = zeros(1080,7);% 7 Expense Attributes
program_intermediate = cell(1080,15);

%annual_constellation_labor_cost = cell(1080);
%annual_constellation_operations_cost = cell(1080);

count  = 1;

for a = 1:numel(program_satellite_design_choice)
    for b = 1: numel(program_number_of_satellites)
        for c = 1:numel(program_num_satellites_concurrent_development)
            for d = 1:numel(program_constellation_type)
                for e = 1:numel(program_legacy_operation)
                    for f = 1:numel(program_contract_length)

                        %% store all DV
                        program_DV(count,1) = program_satellite_design_choice(a);
                        program_DV(count,2) = program_number_of_satellites(b);
                        program_DV(count,3) = program_num_satellites_concurrent_development(c);
                        program_DV(count,4) = program_constellation_type(d);
                        program_DV(count,5) = program_legacy_operation(e);
                        program_DV(count,6) = program_contract_length(f);

                        %% get data for each satellite design choice

                        if program_satellite_design_choice(a)==1
                            system_expense = Epoch_Expense_Attributes(1,1:3);
                            system_performance = Epoch_Performance_Attributes(1,:);
                            system_intermediate = Epoch_Intermediate(1,:);
                            system_DV = Epoch_System_DV(1,:);
                        elseif program_satellite_design_choice(a)==2
                            system_expense = Epoch_Expense_Attributes(2,1:3);
                            system_performance = Epoch_Performance_Attributes(2,:);
                            system_intermediate = Epoch_Intermediate(2,:);
                            system_DV = Epoch_System_DV(2,:);
                        elseif program_satellite_design_choice(a)==3
                            system_expense = Epoch_Expense_Attributes(3,1:3);
                            system_performance = Epoch_Performance_Attributes(3,:);
                            system_intermediate = Epoch_Intermediate(3,:);
                            system_DV = Epoch_System_DV(2,:);
                        end
```

```
                        %% PERFORMANCE ATTRIBUTE 3: MANEUVERABILITY
                        % function of individual satellite Delta V
                        % Split into 3 different levels: Low, Medium,
                        % High,Extreme

                        if system_performance(4)<=5000
                            constellation_maneuverability = 1; % LOW
                        elseif (system_performance(4)>5000)&&(system_performance(4)<=10000)
                            constellation_maneuverability = 2; % MEDIUM
                        elseif (system_performance(4)>10000)&&(system_performance(4)<=20000)
                            constellation_maneuverability = 3; % HIGH
                        elseif (system_performance(4)>20000)
                            constellation_maneuverability = 4; % HIGH
                        end

                        %% EXPENSE ATTRIBUTE 1: Total Constellation Development Cost

                        % basic constellation development cost directly proportional to number of
                        % satellites in a constellation
                        % system_expense(1) = DEVELOPMENT COST OF 1 SATELLITE

                        constellation_development_cost =
program_number_of_satellites(b)*system_expense(1);

                        % include discount factor for bulk development
                        if program_number_of_satellites(b)==4
                            total_constellation_development_cost = 1.0*constellation_development_cost;
                        elseif program_number_of_satellites(b)==6
                            total_constellation_development_cost =
0.95*constellation_development_cost; % 5% discount per satellite
                        elseif program_number_of_satellites(b)==8
                            total_constellation_development_cost =
0.90*constellation_development_cost; % 10% discount per satellite
                        elseif program_number_of_satellites(b)==10
                            total_constellation_development_cost =
0.85*constellation_development_cost; % 15% discount per satellite
                        elseif program_number_of_satellites(b)==12
                            total_constellation_development_cost =
0.80*constellation_development_cost; % 20% discount per satellite
                        end
                        % STORE EXPENSE ATTRIBUTE 1 - Dev Cost

                        % basic constellation launch cost (single launch) directly proportional to
number of
                        % satellites in a constellation
                        % system_expense(1) = DEVELOPMENT COST OF 1 SATELLITE

                        % It cost $2547/kg to launch on Falcon 9 Heavy
                        % Payload capacity is 53000kg

                        launch_cost_per_kg = 2547; % for Falcon 9 Heavy

                        constellation_wetmass =
system_intermediate(6)*program_number_of_satellites(b);
                        % STORE INTERMEDIATE 1 - CONSTELLATION WET MASS

                        if constellation_wetmass < 10000
                            cost_penalty = 2.0; % x2 price for not making efficient use of capacity
                            time_penalty = 5; % Add 5 years because you need to wait for other non-FSS
systems to be developed and loaded onto launcher
                        elseif (constellation_wetmass >= 10000)&&(constellation_wetmass < 15000)
                            cost_penalty = 1.5; % 1.5 price for not making efficient use of capacity
                            time_penalty = 4; % Add 4 years
                        elseif (constellation_wetmass >= 15000)&&(constellation_wetmass < 20000)
                            cost_penalty = 1.0; % no penalty for using at least 25% of capacity
                            time_penalty = 3; % Add 3 years
                        elseif (constellation_wetmass >= 20000)&&(constellation_wetmass < 25000)
                            cost_penalty = 0.9; % 10% discount for using slightly below half the
capacity
                            time_penalty = 2; % Add 2 years
                        elseif (constellation_wetmass >= 30000)
                            cost_penalty = 0.8; % 20% bulk discount for using more than half the
capacity
                            time_penalty = 1; % Add 1
                        end
                        % get cost penalty as intermediate
                        % STORE INTERMEDIATE 2 - COST PENALTY
                        % STORE INTERMEDIATE 3 - TIME PENALTY
```

```
                        %% EXPENSE ATTRIBUTE 2: Total Constellation Launch Cost
                        total_constellation_launch_cost =
constellation_wetmass*launch_cost_per_kg*cost_penalty;
                        % STORE EXPENSE ATTRIBUTE 2 - Launch Cost

                        % Constellation development is equal to the number of
                        % satellites multiplied by the development time for
                        % each satellite, then divided by the number of
                        % satellites being developed concurrently, rounded up
                        % to the nearest integer value

                        % ***IMPORTANT!***
                        % If total payload mass on launcher is much lesser
                        % than total launcher payload capacity, apply cost
                        % and schedule penalty as you are not making use of
                        % capacity efficiently. Also, you are likely to
                        % share the use of the launcher with other systems
                        % outside the scope of the FSS. By not making full
                        % use of all the space, there is a also penalty to
                        % development schedule time or launch time. We need
                        % to include this penalty because the system design
                        % points chosen were based on the precedent that
                        % Falcon 9 Heavy offered the lowest launch cost per
                        % kg of mass. However, it is logical that penalties
                        % will be put in place if the constellation does
                        % not occupy an economically sufficient amount of
                        % payload capacity in order to make the launch
                        % worth it.

                        %% EXPENSE ATTRIBUTE 3: Total Constellation Development Time
                        constellation_development_time =
ceil(program_number_of_satellites(b)*system_expense(3)/program_num_satellites_concurrent_development(c
));
                        % STORE EXPENSE ATTRIBUTE 6

                        % Calculate labor cost
                        % Assume a satellite system requires a team of 50 workers
                        % Assume a worker is paid $100,000 a year
                        % Assume worker salary increases by 20% with each year
                        % If more than 1 satellite is developed concurrently,
                        % then multiply number of workers by the number of
                        % satellites
                        num_workers = 50;
                        worker_salary = 0.1; % %100,000 in millions
                        salary_interest = 1.05; % increase by 5% each year

                        % Calculate labor cost using S_n = a*(1-r^n)/(1-r)
                        % This is formula for sum of geometric series
                        % principal labor cost = num workers * per worker
                        % salary * num of satellites developed concurrently

                        %% EXPENSE ATTRIBUTE 5 - WAITING TIME TILL LAUNCH IN YEARS
                        total_constellation_launch_time = constellation_development_time +
time_penalty;

                        %% EXPENSE ATTRIBUTE 4 - TOTAL DEVELOPMENT AND WAITING LABOR COST
                        total_num_workers =
num_workers*program_num_satellites_concurrent_development(c);
                        total_constellation_labor_cost = worker_salary*total_num_workers...
                            *((1-salary_interest^total_constellation_launch_time)/(1-
salary_interest));
                        % STORE EXPENSE ATTRIBUTE 3

                        % intermediate - labor cost per year of the
                        % constellation development schedule

                        % STORE INTERMEDIATE 4 - ANNUAL LABOR COST
                        annual_constellation_labor_cost = zeros(total_constellation_launch_time,1);

                        for i=1:total_constellation_launch_time
                            annual_constellation_labor_cost(i,1) =
num_workers*worker_salary*program_num_satellites_concurrent_development(c)*salary_interest^(i);
                        end

                      % STORE EXPENSE ATTRIBUTE 7
                      % Calculate Program Data Capacity
                        if program_constellation_type(d)==1 % if satellites are working together to
achieve some bigger purpose
                            data_capacity_penalty = 0.6;
                            science_value_extra = 1.25; % 25% more annual science value because of
```

```
emergent capability
                            ops_cost_extra = 1.2; % 20% more to operate and achieve emergent
capability
                        elseif program_constellation_type(d)==2 % disaggregated
                            data_capacity_penalty = 1.0;
                            ops_cost_extra = 1.0; % no extra operations cost
                            science_value_extra = 1.0; % no extra science value
                        elseif program_constellation_type(d)==3 % mixed
                            data_capacity_penalty = 0.8;
                            ops_cost_extra = 1.1;% 10% more to operate and achieve smaller emergent
capability
                            science_value_extra = 1.125; % 12.5% more annual science value because of
smaller emergent capability
                        end

                        % penalty*data capacity per satellite* num of
                        % satellites

                        %% EXPENSE ATTRIBUTE 7 - TOTAL RETIREMENT COST
                        if program_legacy_operation(e)==0

                            program_duration = system_DV(1); % system lifetime DV1 10/15 years
                            capacity_decrease_period = 2; %2 years
                            retirement_cost_per_satellite = 1; % Assume 1 million per satellite
                            total_constellation_retirement_cost =
retirement_cost_per_satellite*program_number_of_satellites(b);

                        elseif program_legacy_operation(e)==1 % legacy operation

                            program_duration = 60; % work for 60 years
                            total_constellation_retirement_cost = 0;
                            % Comment: legacy systems have zero retirement
                            % cost
                        end
                        % STORE EXPENSE ATTRIBUTE 5

                        %% PERFORMANCE ATTRIBUTE 1 -  Immediate Annual Total Data Capacity Available
in Constellation
                        % immediately upon launch
                        constellation_data_capacity =
data_capacity_penalty*system_performance(2)*program_number_of_satellites(b);
                        % STORE PERFORMANCE ATTRIBUTE 1  - Data Cap
                        biannual_capacity_decrease = 0.025; % decrease by 2.5% every 2 years

                        annual_constellation_data_capacity = zeros(program_duration,1);
                        % intermediate - annual data capacity for program
                        % duration
                        for i=1:program_duration
                            annual_constellation_data_capacity(i) = constellation_data_capacity*(1-
biannual_capacity_decrease)^(ceil(i/capacity_decrease_period)-1);
                        end

                        %% PERFORMANCE ATTRIBUTE 2 - Immediate Annual Science Value from Constellation

                        constellation_science_value =
science_value_extra*system_performance(3)*program_number_of_satellites(b);
                        % STORE PERFORMANCE ATTRIBUTE 2 - Science Value
                        annual_constellation_science_value = zeros(program_duration,1);

                        for i=1:program_duration
                            annual_constellation_science_value(i) = constellation_science_value;
                        end
                        % science value should not be discounted assuming
                        % that equipment does not degrade over time or
                        % experience malfunction

                        % Assume operations cost per satellite to be
                        % $50,000
                        annual_ops_cost_per_satellite = 0.05; % $50,000 in millions
                        opscost_increase_period = 5; % % every 5 years
                        opscost_interest = 0.01; % increase by 1% every 5 years

                        % intermediate - annual ops cost
                        annual_constellation_operations_cost = zeros(program_duration,1);

                        for i=1:program_duration
                            annual_constellation_operations_cost(i,1) =
annual_ops_cost_per_satellite*program_number_of_satellites(b)...
                                *(1+opscost_interest)^(ceil(i/opscost_increase_period)-1);
                        end
```

```
                        % Comment: expect ops cost for legacy systems to
                        % cost a lot more!

                        %% EXPENSE ATTRIBUTE 6 - Total Constellation Operations Cost

                        total_constellation_operations_cost =
sum(annual_constellation_operations_cost);
                        % STORE EXPENSE ATTRIBUTE 4

                        % Calculate contract cost - Only development cost
                        % and launch cost are one-time cost committments ie
                        % non-recurring. Labor is recurring for the
                        % development period while operations is recurring
                        % for the program duration. Retirement cost is
                        % within 1 year.

                        %% Contract Cost


                        % store intermediate
                        annual_development_cost = zeros(program_contract_length(f),1);
                        annual_launch_cost = zeros(program_contract_length(f),1);

                        for i=1:program_contract_length(f)
                            annual_development_cost(i) =
total_constellation_development_cost/program_contract_length(f);
                            annual_launch_cost(i) =
total_constellation_launch_cost/program_contract_length(f);
                        end

                        if program_contract_length(f)==4
                            constellation_contract_length = 4;
                        elseif program_contract_length(f)==7
                            constellation_contract_length = 7;
                        elseif program_contract_length(f)==10
                            constellation_contract_length = 10;
                        end

                        %% STORING VALUES

                        program_performance_attributes(count,1) = constellation_data_capacity;
                        program_performance_attributes(count,2) = constellation_science_value;
                        program_performance_attributes(count,3) = constellation_maneuverability;

                        program_expense_attributes(count,1) = total_constellation_development_cost;
                        program_expense_attributes(count,2) = total_constellation_launch_cost;
                        program_expense_attributes(count,3) = total_constellation_labor_cost;
                        program_expense_attributes(count,4) = total_constellation_operations_cost;
                        program_expense_attributes(count,5) = total_constellation_retirement_cost;
                        program_expense_attributes(count,6) = total_constellation_launch_time;
                        program_expense_attributes(count,7) = constellation_contract_length;

                        program_intermediate{count,1} = annual_development_cost;
                        program_intermediate{count,2} = annual_launch_cost;
                        program_intermediate{count,3} = annual_constellation_labor_cost;
                        program_intermediate{count,4} = annual_constellation_operations_cost;
                        program_intermediate{count,5} = total_constellation_retirement_cost;
                        program_intermediate{count,6} = constellation_development_time;
                        program_intermediate{count,7} = annual_constellation_data_capacity;
                        program_intermediate{count,8} = annual_constellation_science_value;
                        program_intermediate{count,9} = constellation_wetmass;
                        program_intermediate{count,10} = cost_penalty;
                        program_intermediate{count,11} = time_penalty;
                        program_intermediate{count,12} = data_capacity_penalty;
                        program_intermediate{count,13} = science_value_extra;
                        program_intermediate{count,14} = ops_cost_extra;
                        program_intermediate{count,15} = program_duration;

                        %% COUNTING
                        count = count + 1;
                    end
                end
            end
        end
    end
end

end
```

This section contains the MATLAB code written to **collectively perform the calculation of attributes of designs for every epoch, sort attributes for convenience of mathematical operations, generate weights and preferences, evaluate design points using embedded MAE and MAU functions, extract characteristic vectors of preferred program designs, and plot tradespaces** for the program level design of the FSS-enabled satellite constellation.

```
%% This code is written by Marcus Shihong Wu.
%% Copyright (c) 2014, SEARI MIT.
%% Generate performance and expense attributes for selected 7 Epochs

[Epoch1_0in00_Program_Performance_Attributes,Epoch1_0in00_Program_Expense_Attributes,Epoch1_0in00_Prog
ram_Intermediate, Epoch1_0in00_Program_DV]...
    =
Generate_Constellation_Design_For_Epoch(Epoch1_0in00_Expense_Attributes,Epoch1_0in00_Performance_Attri
butes,Epoch1_0in00_intermediate, Epoch1_0in00_DV);

[Epoch2_1in00_Program_Performance_Attributes,Epoch2_1in00_Program_Expense_Attributes,Epoch2_1in00_Prog
ram_Intermediate, Epoch2_1in00_Program_DV]...
    =
Generate_Constellation_Design_For_Epoch(Epoch2_1in00_Expense_Attributes,Epoch2_1in00_Performance_Attri
butes,Epoch2_1in00_intermediate, Epoch2_1in00_DV);

[Epoch3_2in00_Program_Performance_Attributes,Epoch3_2in00_Program_Expense_Attributes,Epoch3_02in00_Pro
gram_Intermediate, Epoch3_2in00_Program_DV]...
    =
Generate_Constellation_Design_For_Epoch(Epoch3_2in00_Expense_Attributes,Epoch3_2in00_Performance_Attri
butes,Epoch3_2in00_intermediate, Epoch3_2in00_DV);

[Epoch10_3in01_Program_Performance_Attributes,Epoch10_3in01_Program_Expense_Attributes,Epoch10_3in01_P
rogram_Intermediate, Epoch10_3in01_Program_DV]...
    =
Generate_Constellation_Design_For_Epoch(Epoch10_3in01_Expense_Attributes,Epoch10_3in01_Performance_Att
ributes,Epoch10_3in01_intermediate, Epoch10_3in01_DV);

[Epoch17_4in10_Program_Performance_Attributes,Epoch17_4in10_Program_Expense_Attributes,Epoch17_4in10_P
rogram_Intermediate, Epoch17_4in10_Program_DV]...
    =
Generate_Constellation_Design_For_Epoch(Epoch17_4in10_Expense_Attributes,Epoch17_4in10_Performance_Att
ributes,Epoch17_4in10_intermediate, Epoch17_4in10_DV);

[Epoch24_5in11_Program_Performance_Attributes,Epoch24_5in11_Program_Expense_Attributes,Epoch24_5in11_P
rogram_Intermediate, Epoch24_5in11_Program_DV]...
    =
Generate_Constellation_Design_For_Epoch(Epoch24_5in11_Expense_Attributes,Epoch24_5in11_Performance_Att
ributes,Epoch24_5in11_intermediate, Epoch24_5in11_DV);

[Epoch19_0in11_Program_Performance_Attributes,Epoch19_0in11_Program_Expense_Attributes,Epoch19_0in11_P
rogram_Intermediate, Epoch19_0in11_Program_DV]...
    =
Generate_Constellation_Design_For_Epoch(Epoch19_0in11_Expense_Attributes,Epoch19_0in11_Performance_Att
ributes,Epoch19_0in11_intermediate, Epoch19_0in11_DV);

%% Gather each attribute in a matrix across 7 Epochs
num_of_program_designs = 1080;
num_epochs = 7;

All7Epochs_Exp1_DevelopmentCost = zeros(num_of_program_designs,num_epochs);
All7Epochs_Exp2_LaunchCost = zeros(num_of_program_designs,num_epochs);
All7Epochs_Exp3_LaborCost = zeros(num_of_program_designs,num_epochs);
All7Epochs_Exp4_OperationsCost = zeros(num_of_program_designs,num_epochs);
All7Epochs_Exp5_RetirementCost = zeros(num_of_program_designs,num_epochs);
All7Epochs_Exp6_TimeToLaunch = zeros(num_of_program_designs,num_epochs);
All7Epochs_Exp7_ContractLength = zeros(num_of_program_designs,num_epochs);

% Gather all development cost
All7Epochs_Exp1_DevelopmentCost(:,1) = Epoch1_0in00_Program_Expense_Attributes(:,1)/10^4;
All7Epochs_Exp1_DevelopmentCost(:,2) = Epoch2_1in00_Program_Expense_Attributes(:,1)/10^4;
All7Epochs_Exp1_DevelopmentCost(:,3) = Epoch3_2in00_Program_Expense_Attributes(:,1)/10^4;
All7Epochs_Exp1_DevelopmentCost(:,4) = Epoch10_3in01_Program_Expense_Attributes(:,1)/10^4;
All7Epochs_Exp1_DevelopmentCost(:,5) = Epoch17_4in10_Program_Expense_Attributes(:,1)/10^4;
All7Epochs_Exp1_DevelopmentCost(:,6) = Epoch24_5in11_Program_Expense_Attributes(:,1)/10^4;
All7Epochs_Exp1_DevelopmentCost(:,7) = Epoch19_0in11_Program_Expense_Attributes(:,1)/10^4;

% Gather all launch cost
All7Epochs_Exp2_LaunchCost(:,1) = Epoch1_0in00_Program_Expense_Attributes(:,2)/10^6;
```

```
All7Epochs_Exp2_LaunchCost(:,2) = Epoch2_1in00_Program_Expense_Attributes(:,2)/10^6;
All7Epochs_Exp2_LaunchCost(:,3) = Epoch3_2in00_Program_Expense_Attributes(:,2)/10^6;
All7Epochs_Exp2_LaunchCost(:,4) = Epoch10_3in01_Program_Expense_Attributes(:,2)/10^6;
All7Epochs_Exp2_LaunchCost(:,5) = Epoch17_4in10_Program_Expense_Attributes(:,2)/10^6;
All7Epochs_Exp2_LaunchCost(:,6) = Epoch24_5in11_Program_Expense_Attributes(:,2)/10^6;
All7Epochs_Exp2_LaunchCost(:,7) = Epoch19_0in11_Program_Expense_Attributes(:,2)/10^6;

% Gather all labor cost
All7Epochs_Exp3_LaborCost(:,1) = Epoch1_0in00_Program_Expense_Attributes(:,3);
All7Epochs_Exp3_LaborCost(:,2) = Epoch2_1in00_Program_Expense_Attributes(:,3);
All7Epochs_Exp3_LaborCost(:,3) = Epoch3_2in00_Program_Expense_Attributes(:,3);
All7Epochs_Exp3_LaborCost(:,4) = Epoch10_3in01_Program_Expense_Attributes(:,3);
All7Epochs_Exp3_LaborCost(:,5) = Epoch17_4in10_Program_Expense_Attributes(:,3);
All7Epochs_Exp3_LaborCost(:,6) = Epoch24_5in11_Program_Expense_Attributes(:,3);
All7Epochs_Exp3_LaborCost(:,7) = Epoch19_0in11_Program_Expense_Attributes(:,3);

% Gather all operations cost
All7Epochs_Exp4_OperationsCost(:,1) = Epoch1_0in00_Program_Expense_Attributes(:,4);
All7Epochs_Exp4_OperationsCost(:,2) = Epoch2_1in00_Program_Expense_Attributes(:,4);
All7Epochs_Exp4_OperationsCost(:,3) = Epoch3_2in00_Program_Expense_Attributes(:,4);
All7Epochs_Exp4_OperationsCost(:,4) = Epoch10_3in01_Program_Expense_Attributes(:,4);
All7Epochs_Exp4_OperationsCost(:,5) = Epoch17_4in10_Program_Expense_Attributes(:,4);
All7Epochs_Exp4_OperationsCost(:,6) = Epoch24_5in11_Program_Expense_Attributes(:,4);
All7Epochs_Exp4_OperationsCost(:,7) = Epoch19_0in11_Program_Expense_Attributes(:,4);

% Gather all retirement cost
All7Epochs_Exp5_RetirementCost(:,1) = Epoch1_0in00_Program_Expense_Attributes(:,5);
All7Epochs_Exp5_RetirementCost(:,2) = Epoch2_1in00_Program_Expense_Attributes(:,5);
All7Epochs_Exp5_RetirementCost(:,3) = Epoch3_2in00_Program_Expense_Attributes(:,5);
All7Epochs_Exp5_RetirementCost(:,4) = Epoch10_3in01_Program_Expense_Attributes(:,5);
All7Epochs_Exp5_RetirementCost(:,5) = Epoch17_4in10_Program_Expense_Attributes(:,5);
All7Epochs_Exp5_RetirementCost(:,6) = Epoch24_5in11_Program_Expense_Attributes(:,5);
All7Epochs_Exp5_RetirementCost(:,7) = Epoch19_0in11_Program_Expense_Attributes(:,5);

% Gather all times to launch
All7Epochs_Exp6_TimeToLaunch(:,1) = Epoch1_0in00_Program_Expense_Attributes(:,6);
All7Epochs_Exp6_TimeToLaunch(:,2) = Epoch2_1in00_Program_Expense_Attributes(:,6);
All7Epochs_Exp6_TimeToLaunch(:,3) = Epoch3_2in00_Program_Expense_Attributes(:,6);
All7Epochs_Exp6_TimeToLaunch(:,4) = Epoch10_3in01_Program_Expense_Attributes(:,6);
All7Epochs_Exp6_TimeToLaunch(:,5) = Epoch17_4in10_Program_Expense_Attributes(:,6);
All7Epochs_Exp6_TimeToLaunch(:,6) = Epoch24_5in11_Program_Expense_Attributes(:,6);
All7Epochs_Exp6_TimeToLaunch(:,7) = Epoch19_0in11_Program_Expense_Attributes(:,6);

% Gather all contract length
All7Epochs_Exp7_ContractLength(:,1) = Epoch1_0in00_Program_Expense_Attributes(:,6);
All7Epochs_Exp7_ContractLength(:,2) = Epoch2_1in00_Program_Expense_Attributes(:,6);
All7Epochs_Exp7_ContractLength(:,3) = Epoch3_2in00_Program_Expense_Attributes(:,6);
All7Epochs_Exp7_ContractLength(:,4) = Epoch10_3in01_Program_Expense_Attributes(:,6);
All7Epochs_Exp7_ContractLength(:,5) = Epoch17_4in10_Program_Expense_Attributes(:,6);
All7Epochs_Exp7_ContractLength(:,6) = Epoch24_5in11_Program_Expense_Attributes(:,6);
All7Epochs_Exp7_ContractLength(:,7) = Epoch19_0in11_Program_Expense_Attributes(:,6);

All7Epochs_Perf1_DataCapacity = zeros(num_of_program_designs,num_epochs);
All7Epochs_Perf2_ScienceValue = zeros(num_of_program_designs,num_epochs);
All7Epochs_Perf3_Maneuverability = zeros(num_of_program_designs,num_epochs);

% Gather all data capacity
All7Epochs_Perf1_DataCapacity(:,1) = Epoch1_0in00_Program_Performance_Attributes(:,1);
All7Epochs_Perf1_DataCapacity(:,2) = Epoch2_1in00_Program_Performance_Attributes(:,1);
All7Epochs_Perf1_DataCapacity(:,3) = Epoch3_2in00_Program_Performance_Attributes(:,1);
All7Epochs_Perf1_DataCapacity(:,4) = Epoch10_3in01_Program_Performance_Attributes(:,1);
All7Epochs_Perf1_DataCapacity(:,5) = Epoch17_4in10_Program_Performance_Attributes(:,1);
All7Epochs_Perf1_DataCapacity(:,6) = Epoch24_5in11_Program_Performance_Attributes(:,1);
All7Epochs_Perf1_DataCapacity(:,7) = Epoch19_0in11_Program_Performance_Attributes(:,1);

% Gather all science value
All7Epochs_Perf2_ScienceValue(:,1) = Epoch1_0in00_Program_Performance_Attributes(:,2);
All7Epochs_Perf2_ScienceValue(:,2) = Epoch2_1in00_Program_Performance_Attributes(:,2);
All7Epochs_Perf2_ScienceValue(:,3) = Epoch3_2in00_Program_Performance_Attributes(:,2);
All7Epochs_Perf2_ScienceValue(:,4) = Epoch10_3in01_Program_Performance_Attributes(:,2);
All7Epochs_Perf2_ScienceValue(:,5) = Epoch17_4in10_Program_Performance_Attributes(:,2);
All7Epochs_Perf2_ScienceValue(:,6) = Epoch24_5in11_Program_Performance_Attributes(:,2);
All7Epochs_Perf2_ScienceValue(:,7) = Epoch19_0in11_Program_Performance_Attributes(:,2);

% Gather all maneuverability levels
All7Epochs_Perf3_Maneuverability(:,1) = Epoch1_0in00_Program_Performance_Attributes(:,3);
All7Epochs_Perf3_Maneuverability(:,2) = Epoch2_1in00_Program_Performance_Attributes(:,3);
All7Epochs_Perf3_Maneuverability(:,3) = Epoch3_2in00_Program_Performance_Attributes(:,3);
All7Epochs_Perf3_Maneuverability(:,4) = Epoch10_3in01_Program_Performance_Attributes(:,3);
All7Epochs_Perf3_Maneuverability(:,5) = Epoch17_4in10_Program_Performance_Attributes(:,3);
All7Epochs_Perf3_Maneuverability(:,6) = Epoch24_5in11_Program_Performance_Attributes(:,3);
```

```
All7Epochs_Perf3_Maneuverability(:,7) = Epoch19_0in11_Program_Performance_Attributes(:,3);

%% Construct Lifecycle Scenario using 7 Epochs

% We pick:
%   EPOCH 1
%   Mission 0 in Context 00 - Epoch 1 - We want to construct the FSS by
%   first designing a satellite constellation that will be able to achieve
%   all its objectives with equal preference to stakeholders. Laser and RLV
%   technology are still not mature yet.
%   Performance: Slightly more emphasis on science value than data
%   capacity. Relatively higher emphasis on maneuverability.
%   Expense: Equal emphasis on all expense attributes
%   Operations cost = 0.1
%   Retirement cost always constant = 0.05

%   EPOCH 2
%   Mission 1 in Context 00 - Epoch 2 - We want to send a second satellite
%   constellation to increase fulfillment of FSS objectives. However, this
%   epoch demands a higher demand on science value given the number of
%   space exploration missions to be conducted. Laser and RLV technology
%   are still not mature yet. Emphasis on Launch Cost.
%   Performance: Emphasis on Science Value more than data capacity. Relatively higher emphasis on
maneuverability.
%   Expense: Emphasis on lowering development and labor cost since we want greater science value
%   Operations cost = 0.1
%   Retirement cost always constant = 0.05

%   EPOCH 3
%   Mission 2 in Context 00 - Epoch 3
%   Mission 2 - FSS Datellite Development: Emphasis on Data Packet Available per
%   Satellite and ISL data rate, Emphasis on LOWER DEVELOPMENT COST since you
%   need to drive down cost to develop new technology in order to realize the
%   aforementioned attribures. Laser and RLV technology are still not
%   mature yet.
%   FSS Datellite Development: Emphasis on Data Packet Available per
%   Satellite and ISL data rate, Emphasis on LOWER DEVELOPMENT COST since you
%   need to drive down cost to develop new technology in order to realize the
%   aforementioned attribures
%   Performance: Emphasis on Data capacity more than science value. Relatively lower emphasis on
maneuverability.
%   Expense: Emphasis on lowering development cost, labor cost and development time
%   Operations cost = 0.1
%   Retirement cost always constant = 0.05

%   EPOCH 10
%   Mission 3 in Context 01 - Epoch 10
%   Mission 3 - FSS Initialization: Even more emphasis most on ISL Data Rate and Data packet
available, Emphasis on
%   lowering launch cost and development time since we need to launch as many
%   satellites as possible to meet market demand for FSS services. RLV
%   technology has matured
%   Performance: Even more emphasis on data capacity than science value
%   capacity. Relatively even lower emphasis on maneuverability.
%   Expense: Emphasis on lowering launch cost and time to launch
%   Operations cost = 0.1
%   Retirement cost always constant = 0.05

%   EPOCH 17
%   Mission 4 in Context 10 - Epoch 17
%   Mission 4 - FSS Discovery: Equal Emphasis on ISL Data Rate, Data Packet Available and
%   Science Value only, Equal emphasis on launch cost and development time
%   Performance: Equal emphasis on Data capacity and science value. Less
%   emphasis on maneuverability because satellites are rather stationary
%   relative to one another in other to maintain high data exchange rate,
%   and have min slant range and high access time
%   Expense: Emphasis on development cost, launch cost, labor cost and
%   launch time. Relatively high emphasis on maneuverability.
%   Operations cost = 0.1
%   Retirement Cost  = 0.05

%   EPOCH 24
%   Mission 5 in Context 11 - Epoch 24
%   Mission 5 - FSS Reconfiguration: Emphasis most on Delta_V, Emphasis on
%   lowering Development cost and Launch cost
%   Performance: Slightly more emphasis on data capacity since it is the beginning of
%   full FSS operation. Highest emphasis on maneuverability.
%   Expense: Emphasis on lowering development cost, labor cost and launch
%   time
%   Operations cost = 0.1
%   Retirement Cost  = 0.05
```

```
%   EPOCH 19
%   Mission 0 in Context 11 - Epoch 19
%   Normal FSS Operations - All Attributes equal
%   Performance: Both equal
%   Expense: All equal, except for Operations cost = 0.1 and Retirement
%   cost = 0.05

%% Create Weights for selected 7 Epochs

% [ Data_Capacity Science_Value Maneuverability]
Program_Performance_Weights = [0.25 0.45 0.3;
    0.2 0.6 0.2;
    0.5 0.3 0.2;
    0.6 0.2 0.2;
    0.35 0.35 0.3;
    0.3 0.2 0.5;
    0.35 0.3 0.35];

% [ Development_Cost Launch_Cost Labor_Cost Operations_Cost Retirement_Cost
% Time_to_Launch Contract Length]
Program_Expense_Weights = [0.2 0.2 0.2 0.1 0.05 0.2 0.05;
    0.25 0.125 0.25 0.1 0.05 0.125 0.1;
    0.2 0.1 0.2 0.1 0.05 0.25 0.1;
    0.1 0.3 0.1 0.1 0.05 0.25 0.1;
    0.2 0.15 0.15 0.1 0.05 0.2 0.15;
    0.2 0.1 0.2 0.1 0.05 0.2 0.15;
    0.2 0.15 0.15 0.1 0.05 0.2 0.15];

%% Create Preferences for attributes in selected 7 Epochs

Program_Performance_Preferences = {};

Program_Expense_Preferences = {};

% For Program Data Capacity
Program_Performance_Preferences{1,1} = [50000,0;100000,0.1;200000,0.3;300000,0.6;400000,0.8;500000,1];
Program_Performance_Preferences{2,1} = [50000,0;100000,0.1;200000,0.4;300000,0.7;400000,0.9;500000,1];
Program_Performance_Preferences{3,1} =
[50000,0;100000,0.1;200000,0.25;300000,0.5;400000,0.75;500000,1];
Program_Performance_Preferences{4,1} = [50000,0;100000,0.1;200000,0.2;300000,0.4;400000,0.7;500000,1];
Program_Performance_Preferences{5,1} = [50000,0;100000,0.1;200000,0.3;300000,0.6;400000,0.8;500000,1];
Program_Performance_Preferences{6,1} =
[50000,0;100000,0.1;200000,0.25;300000,0.5;400000,0.75;500000,1];
Program_Performance_Preferences{7,1} = [50000,0;100000,0.1;200000,0.3;300000,0.6;400000,0.8;500000,1];

% For Program Science Value
Program_Performance_Preferences{1,2} = [2000,0;10000,0.3;20000,0.6;30000,0.9;40000,1];
Program_Performance_Preferences{2,2} = [3000,0;10000,0.1;50000,0.4;100000,0.6;150000,0.9;190000,1];
Program_Performance_Preferences{3,2} = [2000,0;10000,0.35;20000,0.7;30000,0.9;40000,1];
Program_Performance_Preferences{4,2} = [2000,0;10000,0.4;20000,0.8;35000,1];
Program_Performance_Preferences{5,2} = [3000,0;10000,0.1;50000,0.4;100000,0.6;150000,0.9;190000,1];
Program_Performance_Preferences{6,2} = [2000,0;10000,0.4;20000,0.8;35000,1];
Program_Performance_Preferences{7,2} = [2000,0;10000,0.3;20000,0.6;30000,0.9;40000,1];

% For Program Maneuverability
Program_Performance_Preferences{1,3} = [0,0;1,0.4;2,0.7;3,1];
Program_Performance_Preferences{2,3} = [0,0;1,0.3;2,0.6;3,1];
Program_Performance_Preferences{3,3} = [0,0;1,0.4;2,0.7;3,1];
Program_Performance_Preferences{4,3} = [0,0;1,0.4;2,0.7;3,1];
Program_Performance_Preferences{5,3} = [0,0;1,0.3;2,0.6;3,0.8;4,1];
Program_Performance_Preferences{6,3} = [0,0;1,0.2;2,0.4;3,0.7;4,1];
Program_Performance_Preferences{7,3} = [0,0;1,0.3;2,0.6;3,0.8;4,1];


% For Development Cost
Program_Expense_Preferences{1,1} = [0,0;1E05,0.1;1.5E05,0.3;2E05,0.6;2.5E05,0.8;3E05,1];
Program_Expense_Preferences{2,1} = [0,0;1E05,0.2;1.5E05,0.5;2E05,0.8;3E05,0.9;4E05,1];
Program_Expense_Preferences{3,1} = [0,0;1E05,0.1;1.5E05,0.4;2E05,0.7;2.5E05,0.9;3E05,1];
Program_Expense_Preferences{4,1} = [0,0;1E05,0.1;1.5E05,0.35;2E05,0.65;2.5E05,0.85;3.5E05,1];
Program_Expense_Preferences{5,1} = [0,0;0.5E05,0.1;1E05,0.4;1.5E05,0.7;2E05,0.9;2.5E05,1];
Program_Expense_Preferences{6,1} = [0,0;0.5E05,0.1;1E05,0.5;1.5E05,0.8;2E05,0.9;2.5E05,1];
Program_Expense_Preferences{7,1} = [0,0;0.5E05,0.1;1E05,0.3;1.5E05,0.6;2E05,0.8;2.5E05,1];

% For Launch Cost
Program_Expense_Preferences{1,2} = [0,0;40E06,0.3;45E06,0.6;50E06,0.8;55E06,0.9;60E06,1];
Program_Expense_Preferences{2,2} = [0,0;40E06,0.25;45E06,0.5;50E06,0.75;55E06,0.85;60E06,1];
Program_Expense_Preferences{3,2} = [0,0;40E06,0.3;45E06,0.6;50E06,0.8;55E06,0.9;60E06,1];
Program_Expense_Preferences{4,2} = [0,0;40E06,0.4;45E06,0.7;50E06,0.8;55E06,0.9;60E06,1];
Program_Expense_Preferences{5,2} = [0,0;40E06,0.35;45E06,0.65;50E06,0.75;55E06,0.85;60E06,1];
```

```
Program_Expense_Preferences{6,2} = [0,0;40E06,0.3;45E06,0.6;50E06,0.8;55E06,0.9;60E06,1];
Program_Expense_Preferences{7,2} = [0,0;40E06,0.25;45E06,0.5;50E06,0.75;55E06,0.9;60E06,1];

% For Labor Cost
Program_Expense_Preferences{1,3} = [0,0;100,0.2;150,0.5;200,0.8;275,1];
Program_Expense_Preferences{2,3} = [0,0;50,0.2;100,0.4;150,0.6;200,0.8;275,1];
Program_Expense_Preferences{3,3} = [0,0;100,0.3;150,0.6;200,0.8;275,1];
Program_Expense_Preferences{4,3} = [0,0;100,0.35;200,0.7;300,1];
Program_Expense_Preferences{5,3} = [0,0;40,0.2;80,0.5;100,0.8;140,1];
Program_Expense_Preferences{6,3} = [0,0;40,0.25;60,0.5;80,0.7;100,0.85;140,1];
Program_Expense_Preferences{7,3} = [0,0;45,0.2;75,0.5;105,0.8;140,1];

% For Operations Cost
Program_Expense_Preferences{1,4} = [0,0;5,0.2;10,0.4;20,0.6;30,0.8;40,1];
Program_Expense_Preferences{2,4} = [0,0;5,0.2;10,0.4;20,0.6;30,0.8;40,1];
Program_Expense_Preferences{3,4} = [0,0;5,0.2;10,0.4;20,0.6;30,0.8;40,1];
Program_Expense_Preferences{4,4} = [0,0;5,0.2;10,0.4;20,0.6;30,0.8;40,1];
Program_Expense_Preferences{5,4} = [0,0;5,0.2;10,0.4;20,0.6;30,0.8;40,1];
Program_Expense_Preferences{6,4} = [0,0;5,0.2;10,0.4;20,0.6;30,0.8;40,1];
Program_Expense_Preferences{7,4} = [0,0;5,0.2;10,0.4;20,0.6;30,0.8;40,1];

% For Retirement Cost
Program_Expense_Preferences{1,5} = [0,0;4,0.2;6,0.4;8,0.6;10,0.8;12,0.1];
Program_Expense_Preferences{2,5} = [0,0;4,0.2;6,0.4;8,0.6;10,0.8;12,0.1];
Program_Expense_Preferences{3,5} = [0,0;4,0.2;6,0.4;8,0.6;10,0.8;12,0.1];
Program_Expense_Preferences{4,5} = [0,0;4,0.2;6,0.4;8,0.6;10,0.8;12,0.1];
Program_Expense_Preferences{5,5} = [0,0;4,0.2;6,0.4;8,0.6;10,0.8;12,0.1];
Program_Expense_Preferences{6,5} = [0,0;4,0.2;6,0.4;8,0.6;10,0.8;12,0.1];
Program_Expense_Preferences{7,5} = [0,0;4,0.2;6,0.4;8,0.6;10,0.8;12,0.1];

% For Time to Launch
Program_Expense_Preferences{1,6} = [4,0;10,0.4;20,0.8;36,1];
Program_Expense_Preferences{2,6} = [4,0;10,0.4;20,0.8;36,1];
Program_Expense_Preferences{3,6} = [4,0;10,0.5;20,0.8;36,1];
Program_Expense_Preferences{4,6} = [4,0;10,0.5;20,0.8;36,1];
Program_Expense_Preferences{5,6} = [4,0;6,0.4;8,0.8;12,1];
Program_Expense_Preferences{6,6} = [4,0;6,0.5;10,0.8;16,1];
Program_Expense_Preferences{7,6} = [4,0;6,0.5;10,0.8;16,1];

% For Contract Length
Program_Expense_Preferences{1,7} = [4,0;7,0.3;10,1];
Program_Expense_Preferences{2,7} = [4,0;7,0.3;10,1];
Program_Expense_Preferences{3,7} = [4,0;7,0.5;10,1];
Program_Expense_Preferences{4,7} = [4,0;7,0.5;10,1];
Program_Expense_Preferences{5,7} = [4,0;7,0.6;10,1];
Program_Expense_Preferences{6,7} = [4,0;7,0.6;10,1];
Program_Expense_Preferences{7,7} = [4,0;7,0.6;10,1];

%% Calculate MAE and MAU

Program_Performance_Attributes_For7Epochs = {Epoch1_0in00_Program_Performance_Attributes;
    Epoch2_1in00_Program_Performance_Attributes;
    Epoch3_2in00_Program_Performance_Attributes;
    Epoch10_3in01_Program_Performance_Attributes;
    Epoch17_4in10_Program_Performance_Attributes;
    Epoch24_5in11_Program_Performance_Attributes;
    Epoch19_0in11_Program_Performance_Attributes};

Program_Expense_Attributes_For7Epochs = {Epoch1_0in00_Program_Expense_Attributes;
    Epoch2_1in00_Program_Expense_Attributes;
    Epoch3_2in00_Program_Expense_Attributes;
    Epoch10_3in01_Program_Expense_Attributes;
    Epoch17_4in10_Program_Expense_Attributes;
    Epoch24_5in11_Program_Expense_Attributes;
    Epoch19_0in11_Program_Expense_Attributes};


%% EPOCH 1
Program_Epoch1_MAE = zeros(num_of_program_designs,num_epochs);
Program_Epoch1_MAU = zeros(num_of_program_designs,num_epochs);

for e = 1:size(Program_Performance_Preferences,1)

    [MAU_ForEachEpoch] =
Calculate_MAU(Epoch1_0in00_Program_Performance_Attributes,Program_Performance_Preferences(e,:),Program
_Performance_Weights(e,:));
    [MAE_ForEachEpoch] =
Calculate_MAE(Epoch1_0in00_Program_Expense_Attributes,Program_Expense_Preferences(e,:),Program_Expense
_Weights(e,:));
```

```
    Program_Epoch1_MAE(:,e) = MAE_ForEachEpoch;
    Program_Epoch1_MAU(:,e) = MAU_ForEachEpoch;

end

%% EPOCH 2
Program_Epoch2_MAE = zeros(num_of_program_designs,num_epochs);
Program_Epoch2_MAU = zeros(num_of_program_designs,num_epochs);

for e = 1:size(Program_Performance_Preferences,1)

    [MAU_ForEachEpoch] =
Calculate_MAU(Epoch2_1in00_Program_Performance_Attributes,Program_Performance_Preferences(e,:),Program
_Performance_Weights(e,:));
    [MAE_ForEachEpoch] =
Calculate_MAE(Epoch2_1in00_Program_Expense_Attributes,Program_Expense_Preferences(e,:),Program_Expense
_Weights(e,:));

    Program_Epoch2_MAE(:,e) = MAE_ForEachEpoch;
    Program_Epoch2_MAU(:,e) = MAU_ForEachEpoch;

end

%% EPOCH 3
Program_Epoch3_MAE = zeros(num_of_program_designs,num_epochs);
Program_Epoch3_MAU = zeros(num_of_program_designs,num_epochs);

for e = 1:size(Program_Performance_Preferences,1)

    [MAU_ForEachEpoch] =
Calculate_MAU(Epoch3_2in00_Program_Performance_Attributes,Program_Performance_Preferences(e,:),Program
_Performance_Weights(e,:));
    [MAE_ForEachEpoch] =
Calculate_MAE(Epoch3_2in00_Program_Expense_Attributes,Program_Expense_Preferences(e,:),Program_Expense
_Weights(e,:));

    Program_Epoch3_MAE(:,e) = MAE_ForEachEpoch;
    Program_Epoch3_MAU(:,e) = MAU_ForEachEpoch;

end

%% EPOCH 10
Program_Epoch10_MAE = zeros(num_of_program_designs,num_epochs);
Program_Epoch10_MAU = zeros(num_of_program_designs,num_epochs);

for e = 1:size(Program_Performance_Preferences,1)

    [MAU_ForEachEpoch] =
Calculate_MAU(Epoch10_3in01_Program_Performance_Attributes,Program_Performance_Preferences(e,:),Progra
m_Performance_Weights(e,:));
    [MAE_ForEachEpoch] =
Calculate_MAE(Epoch10_3in01_Program_Expense_Attributes,Program_Expense_Preferences(e,:),Program_Expens
e_Weights(e,:));

    Program_Epoch10_MAE(:,e) = MAE_ForEachEpoch;
    Program_Epoch10_MAU(:,e) = MAU_ForEachEpoch;

end

%% EPOCH 17
Program_Epoch17_MAE = zeros(num_of_program_designs,num_epochs);
Program_Epoch17_MAU = zeros(num_of_program_designs,num_epochs);

for e = 1:size(Program_Performance_Preferences,1)

    [MAU_ForEachEpoch] =
Calculate_MAU(Epoch17_4in10_Program_Performance_Attributes,Program_Performance_Preferences(e,:),Progra
m_Performance_Weights(e,:));
    [MAE_ForEachEpoch] =
Calculate_MAE(Epoch17_4in10_Program_Expense_Attributes,Program_Expense_Preferences(e,:),Program_Expens
e_Weights(e,:));

    Program_Epoch17_MAE(:,e) = MAE_ForEachEpoch;
    Program_Epoch17_MAU(:,e) = MAU_ForEachEpoch;

end
```

```
%% EPOCH 24
Program_Epoch24_MAE = zeros(num_of_program_designs,num_epochs);
Program_Epoch24_MAU = zeros(num_of_program_designs,num_epochs);

for e = 1:size(Program_Performance_Preferences,1)

    [MAU_ForEachEpoch] =
Calculate_MAU(Epoch24_5in11_Program_Performance_Attributes,Program_Performance_Preferences(e,:),Progra
m_Performance_Weights(e,:));
    [MAE_ForEachEpoch] =
Calculate_MAE(Epoch24_5in11_Program_Expense_Attributes,Program_Expense_Preferences(e,:),Program_Expens
e_Weights(e,:));

    Program_Epoch24_MAE(:,e) = MAE_ForEachEpoch;
    Program_Epoch24_MAU(:,e) = MAU_ForEachEpoch;

end

%% EPOCH 19
Program_Epoch19_MAE = zeros(num_of_program_designs,num_epochs);
Program_Epoch19_MAU = zeros(num_of_program_designs,num_epochs);

for e = 1:size(Program_Performance_Preferences,1)

    [MAU_ForEachEpoch] =
Calculate_MAU(Epoch19_0in11_Program_Performance_Attributes,Program_Performance_Preferences(e,:),Progra
m_Performance_Weights(e,:));
    [MAE_ForEachEpoch] =
Calculate_MAE(Epoch19_0in11_Program_Expense_Attributes,Program_Expense_Preferences(e,:),Program_Expens
e_Weights(e,:));

    Program_Epoch19_MAE(:,e) = MAE_ForEachEpoch;
    Program_Epoch19_MAU(:,e) = MAU_ForEachEpoch;

end

%%**************************************************************************
%% Apply Constraint Levels

Program_MAE_Constraint_Levels = zeros(num_epochs,1);
Program_MAU_Constraint_Levels = zeros(num_epochs,1);

% [ Data_Capacity Science_Value Maneuverability]

Program_Performance_Attribute_Constraints = [250000,25000,1;
    250000,100000,1;
    250000,25000,1;
    250000,20000,2;
    250000,50000,2;
    300000,15000,3;
    300000,25000,2];

% [ Development_Cost Launch_Cost Labor_Cost Operations_Cost Retirement_Cost
% Time_to_Launch Contract Length]

Program_Expense_Attribute_Constraints = [2.5E05,60E06,250,35,10,10,10;
    2.5E05,55E06,225,30,10,10,7;
    2.5E05,55E06,200,30,10,10,7;
    2.25E05,50E06,200,27.5,10,10,7;
    2.25E05,45E06,200,25,6,8,7;
    2.25E05,45E06,200,25,6,8,7;
    2E05,40E06,180,25,6,8,7];

for e = 1:size(Performance_Preferences_Constraints,1)

[MAU_Constraint] =
Calculate_MAU(Program_Performance_Attribute_Constraints(e),Program_Performance_Preferences(e,:),Progra
m_Performance_Weights(e,:));
[MAE_Constraint] =
Calculate_MAE(Program_Expense_Attribute_Constraints(e),Program_Expense_Preferences(e,:),Program_Expens
e_Weights(e,:));

Program_MAE_Constraint_Levels(e) = MAE_Constraint;
Program_MAU_Constraint_Levels(e) = MAU_Constraint;

end

%%**************************************************************************
```

```
%% Scatter plots - EPOCH 1

%
% store all the derived minimum expense levels
%
Program_Derived_Min_Expense = zeros(num_epochs,1);

scatter(Program_Epoch1_MAE(:,1),Program_Epoch1_MAU(:,1),'b','fill');
title('Program Analysis - Scatter Plot of MAU vs MAE for Epoch 1');
axis([0 1 0 1]);
xlabel('MAE');
ylabel('MAU');
hold on
plot([0 1],[Program_MAU_Constraint_Levels(1) Program_MAU_Constraint_Levels(1)],'--ro','LineWidth',3);
plot([Program_MAE_Constraint_Levels(1) Program_MAE_Constraint_Levels(1)],[0 1],'--mo','LineWidth',3);
% MAE = 0.4168 MAU = 0.4646
Program_Derived_Min_Expense(1) = 0.4168;
plot([Program_Derived_Min_Expense(1) Program_Derived_Min_Expense(1)],[0 1],'--go','LineWidth',3); %
observed from diagram

% High Risk Point = 0.6925 MAU= 0.9525% - Number 349
% [1,12,4,2,0,4]
% [495000,30000,3]
% [307578.240000000,58976294.4000000,251.557850710977,6.03000000000000,12,10,4]

% Medium Risk Point: MAE: 0.6288 MAU= 0.8447 - Number 283
% [1,10,4,3,0,4]
% [330000,28125,3]
% [272334.900000000,49146912,220.531286391406,5.02500000000000,10,9,4]

% Low Risk Point: MAE= 0.4876 MAU= 0.7725 - Number 1051
% [3,12,3,2,0,4]
% [495000,30000,1]
% [307578.240000000,44440056,143.236633136719,6.03000000000000,12,8,4]
Epoch1_High_Index = 349;
Epoch1_Med_Index = 283;
Epoch1_Low_index = 1051;

Epoch1_High_MAE = 0.6925;
Epoch1_High_MAU = 0.9525;
Epoch1_Med_MAE = 0.6288;
Epoch1_Med_MAU = 0.8447;
Epoch1_Low_MAE = 0.4876;
Epoch1_Low_MAU = 0.7725;

hold on
scatter(Epoch1_High_MAE,Epoch1_High_MAU,'o','k','LineWidth',10,'MarkerFaceColor','flat')
hold on
scatter(Epoch1_Med_MAE,Epoch1_Med_MAU,'s','r','LineWidth',10,'MarkerFaceColor','flat')
hold on
scatter(Epoch1_Low_MAE,Epoch1_Low_MAU,'^','c','LineWidth',10,'MarkerFaceColor','flat')

%% Scatter Plots - EPOCH 2

figure()
scatter(Program_Epoch2_MAE(:,2),Program_Epoch2_MAU(:,2),'b','fill');
title('Program Analysis - Scatter Plot of MAU vs MAE for Epoch 2');
axis([0 1 0 1]);
xlabel('MAE');
ylabel('MAU');

hold on
plot([0 1],[Program_MAU_Constraint_Levels(2) Program_MAU_Constraint_Levels(2)],'--ro','LineWidth',3);
plot([Program_MAE_Constraint_Levels(2) Program_MAE_Constraint_Levels(2)],[0 1],'--mo','LineWidth',3);

Program_Derived_Min_Expense(2) = 0.3923;
plot([Program_Derived_Min_Expense(2) Program_Derived_Min_Expense(2)],[0 1],'--go','LineWidth',3); %
observed from diagram

% High Risk Point = 0.5497 MAU= 0.886 - Number 331
% [1,12,3,2,0,4]
% [315000,150000,3]
% [393955.200000000,64453363.2000000,122.130126796875,6.03000000000000,12,7,4]

% Medium Risk Point: MAE: 0.472 MAU= 0.6856 - Number 181
% [1,8,3,1,0,4]
% [126000,125000,3]
% [295466.400000000,48340022.4000000,102.028692187500,4.02000000000000,8,6,4]
```

```
% Low Risk Point: MAE= 0.3923 MAU= 0.5628 - Number - Number 109
% [1,6,3,1,0,4]
% [94500,93750,3]
% [233910.900000000,40283352,102.028692187500,3.01500000000000,6,6,4]

Epoch2_High_Index = 331;
Epoch2_Med_Index = 181;
Epoch2_Low_index = 109;

Epoch2_High_MAE = 0.5497;
Epoch2_High_MAU = 0.886;
Epoch2_Med_MAE = 0.472;
Epoch2_Med_MAU = 0.6856;
Epoch2_Low_MAE = 0.3923;
Epoch2_Low_MAU = 0.5628;

hold on
scatter(Epoch2_High_MAE,Epoch2_High_MAU,'o','k','LineWidth',10,'MarkerFaceColor','flat')
hold on
scatter(Epoch2_Med_MAE,Epoch2_Med_MAU,'s','r','LineWidth',10,'MarkerFaceColor','flat')
hold on
scatter(Epoch2_Low_MAE,Epoch2_Low_MAU,'^','c','LineWidth',10,'MarkerFaceColor','flat')


%% Scatter Plots - EPOCH 3

figure()
scatter(Program_Epoch3_MAE(:,3),Program_Epoch3_MAU(:,3),'b','fill');
title('Program Analysis - Scatter Plot of MAU vs MAE for Epoch 3');
axis([0 1 0 1]);
xlabel('MAE');
ylabel('MAU');

hold on
plot([0 1],[Program_MAU_Constraint_Levels(3) Program_MAU_Constraint_Levels(3)],'--ro','LineWidth',3);
plot([Program_MAE_Constraint_Levels(3) Program_MAE_Constraint_Levels(3)],[0 1],'--mo','LineWidth',3);

Program_Derived_Min_Expense(3) = 0.3399;
plot([Program_Derived_Min_Expense(3) Program_Derived_Min_Expense(3)],[0 1],'--go','LineWidth',3); %
observed from diagram

% High Risk Point = 0.6396 MAU= 0.9638 - Number 349
% [1,12,4,2,0,4]
% [495000,30000,3]
% [307578.240000000,58976294.4000000,251.557850710977,6.03000000000000,12,10,4]

% Medium Risk Point: MAE: 0.4785 MAU= 0.8438 - Number 1069
% [3,12,4,2,0,4]
% [495000,30000,1]
% [307578.240000000,44440056,162.840169062500,6.03000000000000,12,7,4]

% Low Risk Point: MAE= 0.4438 MAU= 0.5775 - Number 889
% [3,8,2,2,0,4]
% [330000,20000,1]
% [230683.680000000,44440056,110.265643195703,4.02000000000000,8,9,4]

Epoch3_High_Index = 349;
Epoch3_Med_Index = 1069;
Epoch3_Low_index = 889;

Epoch3_High_MAE = 0.6396;
Epoch3_High_MAU = 0.9638;
Epoch3_Med_MAE = 0.4785;
Epoch3_Med_MAU = 0.8438;
Epoch3_Low_MAE = 0.4438;
Epoch3_Low_MAU = 0.5775;

hold on
scatter(Epoch3_High_MAE,Epoch3_High_MAU,'o','k','LineWidth',10,'MarkerFaceColor','flat')
hold on
scatter(Epoch3_Med_MAE,Epoch3_Med_MAU,'s','r','LineWidth',10,'MarkerFaceColor','flat')
hold on
scatter(Epoch3_Low_MAE,Epoch3_Low_MAU,'^','m','LineWidth',10,'MarkerFaceColor','flat')


%% Scatter Plots  - EPOCH 10
figure()
scatter(Program_Epoch10_MAE(:,4),Program_Epoch10_MAU(:,4),'b','fill');
title('Program Analysis - Scatter Plot of MAU vs MAE for Epoch 10');
```

```
axis([0 1 0 1]);
xlabel('MAE');
ylabel('MAU');

hold on
plot([0 1],[Program_MAU_Constraint_Levels(4) Program_MAU_Constraint_Levels(4)],'--ro','LineWidth',3);
plot([Program_MAE_Constraint_Levels(4) Program_MAE_Constraint_Levels(4)],[0 1],'--mo','LineWidth',3);
% MAE = 0.485 MAU = 0.485 - choose a line slightly below 0.488
Program_Derived_Min_Expense(4) = 0.3735;
plot([Program_Derived_Min_Expense(4) Program_Derived_Min_Expense(4)],[0 1],'--go','LineWidth',3); %
observed from diagram

% High Risk Point = 0.5691 MAU= 0.9777 - Number 349
% [1,12,4,2,0,4]
% [495000,30000,3]
% [307578.240000000,46426716,284.135743246525,6.03000000000000,12,11,4]

% Medium Risk Point: MAE: 0.4422 MAU= 0.8577 - Number 1069
% [3,12,4,2,0,4]
% [495000,30000,1]
% [307578.240000000,44440056,162.840169062500,6.03000000000000,12,7,4]

Epoch10_High_Index = 349;
Epoch10_Med_Index = 1069;
Epoch10_Low_index = 0;

Epoch10_High_MAE = 0.5691;
Epoch10_High_MAU = 0.9777;
Epoch10_Med_MAE = 0.4422;
Epoch10_Med_MAU = 0.8577;
Epoch10_Low_MAE = 0;
Epoch10_Low_MAU = 0;

hold on
scatter(Epoch10_High_MAE,Epoch10_High_MAU,'o','k','LineWidth',10,'MarkerFaceColor','flat')
hold on
scatter(Epoch10_Med_MAE,Epoch10_Med_MAU,'s','r','LineWidth',10,'MarkerFaceColor','flat')
% hold on
% scatter(0.488,0.85667,'^','c','LineWidth',10,'MarkerFaceColor','flat')

%% Scatter Plots - EPOCH 17

% Points in this epoch have utility of 0.6 and below because
% maneuverability level is always at 1 given the lower deltaVs of all the
% satellites. The performance attributes in Epoch 17 is close to that of
% Epoch 2, but there is a greater

figure()
scatter(Program_Epoch17_MAE(:,5),Program_Epoch17_MAU(:,5),'b','fill');
title('Program Analysis - Scatter Plot of MAU vs MAE for Epoch 17');
axis([0 1 0 1]);
xlabel('MAE');
ylabel('MAU');

hold on
plot([0 1],[Program_MAU_Constraint_Levels(5) Program_MAU_Constraint_Levels(5)],'--ro','LineWidth',3);
plot([Program_MAE_Constraint_Levels(5) Program_MAE_Constraint_Levels(5)],[0 1],'--mo','LineWidth',3);
% MAE = 0.485 MAU = 0.485 - choose a line slightly below 0.488
Program_Derived_Min_Expense(5) = 0.4038;
plot([Program_Derived_Min_Expense(5) Program_Derived_Min_Expense(5)],[0 1],'--go','LineWidth',3); %
observed from diagram

% THERE ARE ONLY 2 LOGICAL POINT IN THIS EPOCH
% High Risk Point = 0.4483 MAU= 0.6255 - Number 331
% [1,12,3,2,0,4]
% [315000,150000,1]
% [207746.784000000,50682753,82.8844687500000,6.03000000000000,12,5,4]

% Medium Risk Point: MAE: 0.4038 MAU= 0.47 - Number - Number 1051
% [3,12,3,2,0,4]
% [526500,9000,1]
% [207746.784000000,41949090,82.8844687500000,6.03000000000000,12,5,4]

Epoch17_High_Index = 331;
Epoch17_Med_Index = 1051;
Epoch17_Low_index = 0;

Epoch17_High_MAE = 0.4483;
Epoch17_High_MAU = 0.6255;
```

```
Epoch17_Med_MAE = 0.4238;
Epoch17_Med_MAU = 0.47;
Epoch17_Low_MAE = 0;
Epoch17_Low_MAU = 0;

hold on
scatter(Epoch17_High_MAE,Epoch17_High_MAU,'o','k','LineWidth',10,'MarkerFaceColor','flat')
hold on
scatter(Epoch17_Med_MAE,Epoch17_Med_MAU,'s','r','LineWidth',10,'MarkerFaceColor','flat')


%% Scatter Plots - EPOCH 24

figure()
scatter(Program_Epoch24_MAE(:,6),Program_Epoch24_MAU(:,6),'b','fill');
title('Program Analysis - Scatter Plot of MAU vs MAE for Epoch 24');
axis([0 1 0 1]);
xlabel('MAE');
ylabel('MAU');

hold on
plot([0 1],[Program_MAU_Constraint_Levels(6) Program_MAU_Constraint_Levels(6)],'--ro','LineWidth',3);
plot([Program_MAE_Constraint_Levels(6) Program_MAE_Constraint_Levels(6)],[0 1],'--mo','LineWidth',3);

Program_Derived_Min_Expense(6) = 0.4863;
plot([Program_Derived_Min_Expense(6) Program_Derived_Min_Expense(6)],[0 1],'--go','LineWidth',3); %
observed from diagram

% High Risk Point = 0.5401 MAU= 0.9829 - Number 349
% [1,12,4,2,0,4]
% [495000,30000,4]
% [207746.784000000,82706184,110.512625000000,6.03000000000000,12,5,4]

% Medium Risk Point: MAE: 0.5065 MAU= 0.87 - Number 691
% [2,12,3,2,0,4]
% [526500,9000,4]
% [207746.784000000,80994600,82.8844687500000,6.03000000000000,12,5,4]

Epoch24_High_Index = 349;
Epoch24_Med_Index = 691;
Epoch24_Low_index = 0;

Epoch24_High_MAE = 0.5401;
Epoch24_High_MAU = 0.9829;
Epoch24_Med_MAE = 0.5065;
Epoch24_Med_MAU = 0.87;
Epoch24_Low_MAE = 0;
Epoch24_Low_MAU = 0;

hold on
scatter(Epoch24_High_MAE,Epoch24_High_MAU,'o','k','LineWidth',10,'MarkerFaceColor','flat')
hold on
scatter(Epoch24_Med_MAE,Epoch24_Med_MAU,'s','r','LineWidth',10,'MarkerFaceColor','flat')


%% Scatter Plots - EPOCH 19
figure()
scatter(Program_Epoch19_MAE(:,7),Program_Epoch19_MAU(:,7),'b','fill');
title('Program Analysis - Scatter Plot of MAU vs MAE for Epoch 19');
axis([0 1 0 1]);
xlabel('MAE');
ylabel('MAU');

hold on
plot([0 1],[Program_MAU_Constraint_Levels(7) Program_MAU_Constraint_Levels(7)],'--ro','LineWidth',3);
plot([Program_MAE_Constraint_Levels(7) Program_MAE_Constraint_Levels(7)],[0 1],'--mo','LineWidth',3);

Program_Derived_Min_Expense(7) = 0.4224;
plot([Program_Derived_Min_Expense(7) Program_Derived_Min_Expense(7)],[0 1],'--go','LineWidth',3); %
observed from diagram

% High Risk Point = 0.5879 MAU= 0.8965 - Number 349
% [1,12,4,2,0,4]
% [495000,30000,3]
% [207746.784000000,58659957,136.038256250000,6.03000000000000,12,6,4]

% Medium Risk Point: MAE: 0.5844 MAU= 0.7938 - 277
% [1,10,4,2,0,4]
```

```
% [412500,25000,3]
% [183942.465000000,48883297.5000000,136.038256250000,5.02500000000000,10,6,4]

% Low Risk Point: MAE=0.4224 MAU=0.7215 - Number 1061
% [3,12,3,3,1,7]
% [396000,33750,1]
% [207746.784000000,55083969,82.8844687500000,38.0475090395909,0,5,7]

Epoch19_High_Index = 349;
Epoch19_Med_Index = 277;
Epoch19_Low_index = 1051;

Epoch19_High_MAE = 0.5879;
Epoch19_High_MAU = 0.8965;
Epoch19_Med_MAE = 0.558;
Epoch19_Med_MAU = 0.7938;
Epoch19_Low_MAE = 0.4224;
Epoch19_Low_MAU = 0.7215;

hold on
scatter(Epoch19_High_MAE,Epoch19_High_MAU,'o','k','LineWidth',10,'MarkerFaceColor','flat')
hold on
scatter(Epoch19_Med_MAE,Epoch19_Med_MAU,'s','r','LineWidth',10,'MarkerFaceColor','flat')
hold on
scatter(Epoch19_Low_MAE,Epoch19_Low_MAU,'^','m','LineWidth',10,'MarkerFaceColor','flat')
```

APPENDIX 3:
PORTFOLIO LEVEL ANALYSIS

The MATLAB code for the entire FSS tradespace model will not be included in this appendix. Only important sections are shown. Contact MIT SEAri or the author of this thesis for access to the full code.

This section contains the MATLAB code written to **generate all possible portfolio designs** for the FSS portfolio.

```
%% This code is written by Marcus Shihong Wu.
%% Copyright (c) 2014, SEARI MIT.


%% 1322123

%% 1. Get program designs from Epoch 1 - Only 1 design
% High Risk Point = 0.6925 MAU= 0.9525% - Number 349
% [1,12,4,2,0,4]
% [495000,30000,3]
% [307578.240000000,58976294.4000000,251.557850710977,6.03000000000000,12,10,4]

Epoch1_High_Index = 349;

Selected_Program_Epoch1_Performance_Attributes(1,:) =
Epoch1_0in00_Program_Performance_Attributes(Epoch1_High_Index,:);

Selected_Program_Epoch1_Expense_Attributes(1,:) =
Epoch1_0in00_Program_Expense_Attributes(Epoch1_High_Index,:);

Selected_Program_Epoch1_Intermediate(1,:) = Epoch1_0in00_Program_Intermediate(Epoch1_High_Index,:);

Selected_Program_Epoch1_DV(1,:) = Epoch1_0in00_Program_DV(Epoch1_High_Index,:);

%% 2. Get program designs from Epoch 2 - All 3 designs
% High Risk Point = 0.5497 MAU= 0.886 - Number 331
% [1,12,3,2,0,4]
% [315000,150000,3]
% [393955.200000000,64453363.2000000,122.130126796875,6.03000000000000,12,7,4]

% Medium Risk Point: MAE: 0.472 MAU= 0.6856 - Number 181
% [1,8,3,1,0,4]
% [126000,125000,3]
% [295466.400000000,48340022.4000000,102.028692187500,4.02000000000000,8,6,4]

% Low Risk Point: MAE= 0.3923 MAU= 0.5628 - Number - Number 109
% [1,6,3,1,0,4]
% [94500,93750,3]
% [233910.900000000,40283352,102.028692187500,3.01500000000000,6,6,4]

Epoch2_High_Index = 331;
Epoch2_Med_Index = 181;
Epoch2_Low_Index = 109;

Selected_Program_Epoch2_Performance_Attributes(1,:) =
Epoch2_1in00_Program_Performance_Attributes(Epoch2_High_Index,:);
Selected_Program_Epoch2_Performance_Attributes(2,:) =
Epoch2_1in00_Program_Performance_Attributes(Epoch2_Med_Index,:);
Selected_Program_Epoch2_Performance_Attributes(3,:) =
Epoch2_1in00_Program_Performance_Attributes(Epoch2_Low_Index,:);

Selected_Program_Epoch2_Expense_Attributes(1,:) =
Epoch2_1in00_Program_Expense_Attributes(Epoch2_High_Index,:);
Selected_Program_Epoch2_Expense_Attributes(2,:) =
Epoch2_1in00_Program_Expense_Attributes(Epoch2_Med_Index,:);
Selected_Program_Epoch2_Expense_Attributes(3,:) =
Epoch2_1in00_Program_Expense_Attributes(Epoch2_Low_Index,:);

Selected_Program_Epoch2_Intermediate(1,:) = Epoch2_1in00_Program_Intermediate(Epoch2_High_Index,:);
Selected_Program_Epoch2_Intermediate(2,:) = Epoch2_1in00_Program_Intermediate(Epoch2_Med_Index,:);
Selected_Program_Epoch2_Intermediate(3,:) = Epoch2_1in00_Program_Intermediate(Epoch2_Low_Index,:);
```

```
Selected_Program_Epoch2_DV(1,:) = Epoch2_1in00_Program_DV(Epoch2_High_Index,:);
Selected_Program_Epoch2_DV(2,:) = Epoch2_1in00_Program_DV(Epoch2_Med_Index,:);
Selected_Program_Epoch2_DV(3,:) = Epoch2_1in00_Program_DV(Epoch2_Low_Index,:);

%% 3. Get program designs from Epoch 3 - only 2 designs
% High Risk Point = 0.6396 MAU= 0.9638 - Number 349
% [1,12,4,2,0,4]
% [495000,30000,3]
% [307578.240000000,58976294.4000000,251.557850710977,6.03000000000000,12,10,4]
% LEGACY OPTION: 352

% Medium Risk Point: MAE: 0.4785 MAU= 0.8438 - Number 1069
% [3,12,4,2,0,4]
% [495000,30000,1]
% [307578.240000000,44440056,162.840169062500,6.03000000000000,12,7,4]
% LEGACY OPTIO: 1072

Epoch3_High_Index = 349;
Epoch3_Med_Index = 1069;

Selected_Program_Epoch3_Performance_Attributes(1,:) =
Epoch3_2in00_Program_Performance_Attributes(Epoch3_High_Index,:);
Selected_Program_Epoch3_Performance_Attributes(2,:) =
Epoch3_2in00_Program_Performance_Attributes(Epoch3_Med_Index,:);

Selected_Program_Epoch3_Expense_Attributes(1,:) =
Epoch3_2in00_Program_Expense_Attributes(Epoch3_High_Index,:);
Selected_Program_Epoch3_Expense_Attributes(2,:) =
Epoch3_2in00_Program_Expense_Attributes(Epoch3_Med_Index,:);

Selected_Program_Epoch3_Intermediate(1,:) = Epoch3_02in00_Program_Intermediate(Epoch3_High_Index,:);
Selected_Program_Epoch3_Intermediate(2,:) = Epoch3_02in00_Program_Intermediate(Epoch3_Med_Index,:);

Selected_Program_Epoch3_DV(1,:) = Epoch3_2in00_Program_DV(Epoch3_High_Index,:);
Selected_Program_Epoch3_DV(2,:) = Epoch3_2in00_Program_DV(Epoch3_Med_Index,:);

%% 4. Get program designs from Epoch 10 - Only 2
% High Risk Point = 0.5691 MAU= 0.9777 - Number 349
% [1,12,4,2,0,4]
% [495000,30000,3]
% [307578.240000000,46426716,284.135743246525,6.03000000000000,12,11,4]

% Medium Risk Point: MAE: 0.4422 MAU= 0.8577 - Number 1069
% [3,12,4,2,0,4]
% [495000,30000,1]
% [307578.240000000,44440056,162.840169062500,6.03000000000000,12,7,4]

Epoch10_High_Index = 349;
Epoch10_Med_Index = 1069;

Selected_Program_Epoch10_Performance_Attributes(1,:) =
Epoch10_3in01_Program_Performance_Attributes(Epoch10_High_Index,:);
Selected_Program_Epoch10_Performance_Attributes(2,:) =
Epoch10_3in01_Program_Performance_Attributes(Epoch10_Med_Index,:);

Selected_Program_Epoch10_Expense_Attributes(1,:) =
Epoch10_3in01_Program_Expense_Attributes(Epoch10_High_Index,:);
Selected_Program_Epoch10_Expense_Attributes(2,:) =
Epoch10_3in01_Program_Expense_Attributes(Epoch10_Med_Index,:);

Selected_Program_Epoch10_Intermediate(1,:) = Epoch10_3in01_Program_Intermediate(Epoch10_High_Index,:);
Selected_Program_Epoch10_Intermediate(2,:) = Epoch10_3in01_Program_Intermediate(Epoch10_Med_Index,:);

Selected_Program_Epoch10_DV(1,:) = Epoch10_3in01_Program_DV(Epoch10_High_Index,:);
Selected_Program_Epoch10_DV(2,:) = Epoch10_3in01_Program_DV(Epoch10_Med_Index,:);

%% 5. Get program designs from Epoch 17 - Only 1
% High Risk Point = 0.4483 MAU= 0.6255 - Number 331
% [1,12,3,2,0,4]
% [315000,150000,1]
% [207746.784000000,50682753,82.8844687500000,6.03000000000000,12,5,4]

Epoch17_High_Index = 331;

Selected_Program_Epoch17_Performance_Attributes(1,:) =
Epoch17_4in10_Program_Performance_Attributes(Epoch17_High_Index,:);
```

```
Selected_Program_Epoch17_Expense_Attributes(1,:) =
Epoch17_4in10_Program_Expense_Attributes(Epoch17_High_Index,:);

Selected_Program_Epoch17_Intermediate(1,:) = Epoch17_4in10_Program_Intermediate(Epoch17_High_Index,:);

Selected_Program_Epoch17_DV(1,:) = Epoch17_4in10_Program_DV(Epoch17_High_Index,:);

%% 6. Get program designs from Epoch 24 - All 2
% High Risk Point = 0.5401 MAU= 0.9829 - Number 349
% [1,12,4,2,0,4]
% [495000,30000,4]
% [207746.784000000,82706184,110.512625000000,6.03000000000000,12,5,4]

% Medium Risk Point: MAE: 0.5065 MAU= 0.87 - Number 691
% [2,12,3,2,0,4]
% [526500,9000,4]
% [207746.784000000,80994600,82.8844687500000,6.03000000000000,12,5,4]

Epoch24_High_Index = 349;
Epoch24_Med_Index = 691;

Selected_Program_Epoch24_Performance_Attributes(1,:) =
Epoch24_5in11_Program_Performance_Attributes(Epoch24_High_Index,:);
Selected_Program_Epoch24_Performance_Attributes(2,:) =
Epoch24_5in11_Program_Performance_Attributes(Epoch24_Med_Index,:);

Selected_Program_Epoch24_Expense_Attributes(1,:) =
Epoch24_5in11_Program_Expense_Attributes(Epoch24_High_Index,:);
Selected_Program_Epoch24_Expense_Attributes(2,:) =
Epoch24_5in11_Program_Expense_Attributes(Epoch24_Med_Index,:);

Selected_Program_Epoch24_Intermediate(1,:) = Epoch24_5in11_Program_Intermediate(Epoch24_High_Index,:);
Selected_Program_Epoch24_Intermediate(2,:) = Epoch24_5in11_Program_Intermediate(Epoch24_Med_Index,:);

Selected_Program_Epoch24_DV(1,:) = Epoch24_5in11_Program_DV(Epoch24_High_Index,:);
Selected_Program_Epoch24_DV(2,:) = Epoch24_5in11_Program_DV(Epoch24_Med_Index,:);

%% 7. Get program designs from Epoch 19
% [1,12,4,2,0,4]
% [495000,30000,3]
% [207746.784000000,58659957,136.038256250000,6.03000000000000,12,6,4]

% Medium Risk Point: MAE: 0.5844 MAU= 0.7938 - 277
% [1,10,4,2,0,4]
% [412500,25000,3]
% [183942.465000000,48883297.5000000,136.038256250000,5.02500000000000,10,6,4]

% Low Risk Point: MAE=0.4224 MAU=0.7215 - Number 1061
% [3,12,3,3,1,7]
% [396000,33750,1]
% [207746.784000000,55083969,82.8844687500000,38.0475090395909,0,5,7]

Epoch19_High_Index = 349;
Epoch19_Med_Index = 277;
Epoch19_Low_Index = 1051;

Selected_Program_Epoch19_Performance_Attributes(1,:) =
Epoch19_0in11_Program_Performance_Attributes(Epoch19_High_Index,:);
Selected_Program_Epoch19_Performance_Attributes(2,:) =
Epoch19_0in11_Program_Performance_Attributes(Epoch19_Med_Index,:);
Selected_Program_Epoch19_Performance_Attributes(3,:) =
Epoch19_0in11_Program_Performance_Attributes(Epoch19_Low_Index,:);

Selected_Program_Epoch19_Expense_Attributes(1,:) =
Epoch19_0in11_Program_Expense_Attributes(Epoch19_High_Index,:);
Selected_Program_Epoch19_Expense_Attributes(2,:) =
Epoch19_0in11_Program_Expense_Attributes(Epoch19_Med_Index,:);
Selected_Program_Epoch19_Expense_Attributes(3,:) =
Epoch19_0in11_Program_Expense_Attributes(Epoch19_Low_Index,:);

Selected_Program_Epoch19_Intermediate(1,:) = Epoch19_0in11_Program_Intermediate(Epoch19_High_Index,:);
Selected_Program_Epoch19_Intermediate(2,:) = Epoch19_0in11_Program_Intermediate(Epoch19_Med_Index,:);
Selected_Program_Epoch19_Intermediate(3,:) = Epoch19_0in11_Program_Intermediate(Epoch19_Low_Index,:);

Selected_Program_Epoch19_DV(1,:) = Epoch19_0in11_Program_DV(Epoch19_High_Index,:);
Selected_Program_Epoch19_DV(2,:) = Epoch19_0in11_Program_DV(Epoch19_Med_Index,:);
Selected_Program_Epoch19_DV(3,:) = Epoch19_0in11_Program_DV(Epoch19_Low_Index,:);
%% TOTAL NUMBER OF PORTFOLIO DESIGNS = 72
```

```
num_portfolio_designs = 72;

Portfolio_DV = zeros(num_portfolio_designs,7);
Portfolio_Performance_Attributes = zeros(num_portfolio_designs,3); % 3 Performance Attributes
Portfolio_Expense_Attributes = zeros(num_portfolio_designs,6);% 7 Expense Attributes

count  = 1;

for a = 1:size(Selected_Program_Epoch1_DV,1)
    for b = 1: size(Selected_Program_Epoch2_DV,1)
        for c = 1:size(Selected_Program_Epoch3_DV,1)
            for d = 1:size(Selected_Program_Epoch10_DV,1)
                for e = 1:size(Selected_Program_Epoch17_DV,1)
                    for f = 1:size(Selected_Program_Epoch24_DV,1)
                        for g = 1:size(Selected_Program_Epoch19_DV,1)

                            %% store all DV
                            Portfolio_DV(count,1) = a;
                            Portfolio_DV(count,2) = b;
                            Portfolio_DV(count,3) = c;
                            Portfolio_DV(count,4) = d;
                            Portfolio_DV(count,5) = e;
                            Portfolio_DV(count,6) = f;
                            Portfolio_DV(count,7) = g;

                            %% PERFORMANCE

                            % Sum annual free data packet capacity
                            Portfolio_Performance_Attributes(count,1) =
Selected_Program_Epoch1_Performance_Attributes(a,1)+...
                                + Selected_Program_Epoch2_Performance_Attributes(b,1)+...
                                + Selected_Program_Epoch3_Performance_Attributes(c,1)+...
                                + Selected_Program_Epoch10_Performance_Attributes(d,1)+...
                                + Selected_Program_Epoch17_Performance_Attributes(e,1)+...
                                + Selected_Program_Epoch24_Performance_Attributes(f,1)+...
                                + Selected_Program_Epoch19_Performance_Attributes(g,1);

                            % Sum annual science value
                            Portfolio_Performance_Attributes(count,2) =
Selected_Program_Epoch1_Performance_Attributes(a,2)+...
                                + Selected_Program_Epoch2_Performance_Attributes(b,2)+...
                                + Selected_Program_Epoch3_Performance_Attributes(c,2)+...
                                + Selected_Program_Epoch10_Performance_Attributes(d,2)+...
                                + Selected_Program_Epoch17_Performance_Attributes(e,2)+...
                                + Selected_Program_Epoch24_Performance_Attributes(f,2)+...
                                + Selected_Program_Epoch19_Performance_Attributes(g,2);

                            % Sum maneuverability
                            Portfolio_Performance_Attributes(count,3) =
Selected_Program_Epoch1_Performance_Attributes(a,3)+...
                                + Selected_Program_Epoch2_Performance_Attributes(b,3)+...
                                + Selected_Program_Epoch3_Performance_Attributes(c,3)+...
                                + Selected_Program_Epoch10_Performance_Attributes(d,3)+...
                                + Selected_Program_Epoch17_Performance_Attributes(e,3)+...
                                + Selected_Program_Epoch24_Performance_Attributes(f,3)+...
                                + Selected_Program_Epoch19_Performance_Attributes(g,3);

                            %% EXPENSE

                            % Sum development cost - in millions
                            % times 100 then divide by 1000,000
                            Portfolio_Expense_Attributes(count,1) = 10E-
04*(Selected_Program_Epoch1_Expense_Attributes(a,1)+...
                                + Selected_Program_Epoch2_Expense_Attributes(b,1)+...
                                + Selected_Program_Epoch3_Expense_Attributes(c,1)+...
                                + Selected_Program_Epoch10_Expense_Attributes(d,1)+...
                                + Selected_Program_Epoch17_Expense_Attributes(e,1)+...
                                + Selected_Program_Epoch24_Expense_Attributes(f,1)+...
                                + Selected_Program_Epoch19_Expense_Attributes(g,1));

                            % Sum launch cost - in millions
                            Portfolio_Expense_Attributes(count,2) = 10E-
06*(Selected_Program_Epoch1_Expense_Attributes(a,2)+...
                                + Selected_Program_Epoch2_Expense_Attributes(b,2)+...
                                + Selected_Program_Epoch3_Expense_Attributes(c,2)+...
                                + Selected_Program_Epoch10_Expense_Attributes(d,2)+...
                                + Selected_Program_Epoch17_Expense_Attributes(e,2)+...
                                + Selected_Program_Epoch24_Expense_Attributes(f,2)+...
                                + Selected_Program_Epoch19_Expense_Attributes(g,2));
```

```
                            % Sum labor cost - in millions
                            Portfolio_Expense_Attributes(count,3) =
Selected_Program_Epoch1_Expense_Attributes(a,3)+...
                                + Selected_Program_Epoch2_Expense_Attributes(b,3)+...
                                + Selected_Program_Epoch3_Expense_Attributes(c,3)+...
                                + Selected_Program_Epoch10_Expense_Attributes(d,3)+...
                                + Selected_Program_Epoch17_Expense_Attributes(e,3)+...
                                + Selected_Program_Epoch24_Expense_Attributes(f,3)+...
                                + Selected_Program_Epoch19_Expense_Attributes(g,3);

                            % Sum operations cost - in millions
                            Portfolio_Expense_Attributes(count,4) =
Selected_Program_Epoch1_Expense_Attributes(a,4)+...
                                + Selected_Program_Epoch2_Expense_Attributes(b,4)+...
                                + Selected_Program_Epoch3_Expense_Attributes(c,4)+...
                                + Selected_Program_Epoch10_Expense_Attributes(d,4)+...
                                + Selected_Program_Epoch17_Expense_Attributes(e,4)+...
                                + Selected_Program_Epoch24_Expense_Attributes(f,4)+...
                                + Selected_Program_Epoch19_Expense_Attributes(g,4);

                            % Sum retirement cost - in millions
                            Portfolio_Expense_Attributes(count,5) =
Selected_Program_Epoch1_Expense_Attributes(a,5)+...
                                + Selected_Program_Epoch2_Expense_Attributes(b,5)+...
                                + Selected_Program_Epoch3_Expense_Attributes(c,5)+...
                                + Selected_Program_Epoch10_Expense_Attributes(d,5)+...
                                + Selected_Program_Epoch17_Expense_Attributes(e,5)+...
                                + Selected_Program_Epoch24_Expense_Attributes(f,5)+...
                                + Selected_Program_Epoch19_Expense_Attributes(g,5);

                            % Sum total time required to launch all
                            % constellations - in years
                            Portfolio_Expense_Attributes(count,6) =
Selected_Program_Epoch1_Expense_Attributes(a,6)+...
                                + Selected_Program_Epoch2_Expense_Attributes(b,6)+...
                                + Selected_Program_Epoch3_Expense_Attributes(c,6)+...
                                + Selected_Program_Epoch10_Expense_Attributes(d,6)+...
                                + Selected_Program_Epoch17_Expense_Attributes(e,6)+...
                                + Selected_Program_Epoch24_Expense_Attributes(f,6)+...
                                + Selected_Program_Epoch19_Expense_Attributes(g,6);

                            %% INTERMEDIATE

                            % 1. Sum all annual development cost

                            count = count + 1;
                        end
                    end
                end
            end
        end
    end
end
```

This section contains the MATLAB code written to **generate weights and preference sets and evaluate all portfolio designs**.

```
%% This code is written by Marcus Shihong Wu.
%% Copyright (c) 2014, SEARI MIT.
%% EVALUATE PORTFOLIO DESIGNS

% [ Data_Capacity Science_Value Maneuverability]
Portfolio_Performance_Weights = [0.4 0.2 0.4];

% [ Development_Cost Launch_Cost Labor_Cost Operations_Cost Retirement_Cost
% Time_to_Launch Contract Length]
Portfolio_Expense_Weights = [0.2 0.2 0.2 0.1 0.1 0.2 ];

%% Create Preferences for attributes in selected 7 Epochs

Portfolio_Performance_Preferences = {};

Portfolio_Expense_Preferences = {};

% For Portfolio Data Capacity: 2802000 - 3136500
Portfolio_Performance_Preferences{1,1} =
[2800000,0;2850000,0.2;2900000,0.4;3000000,0.6;3100000,0.8;3200000,1];

% For Portfolio Science Value: 367750 - 450000
Portfolio_Performance_Preferences{1,2} =
[365000,0;370000,0.1;390000,0.3;410000,0.5;430000,0.8;450000,1];

% For Portfolio Maneuverability: 14-20
Portfolio_Performance_Preferences{1,3} = [13,0;14,0.25;16,0.5;18,0.75;20,1];

% For Portfolio Development Cost: 1756.1 - 1939.9
Portfolio_Expense_Preferences{1,1} = [0,0;1700,0.1;1750,0.3;1800,0.5;1850,0.7;1900,0.9;1950,1];

% For Portfolio Launch Cost: 3687 - 4208.8
Portfolio_Expense_Preferences{1,2} = [0,0;3500,0.2;3700,0.4;3900,0.6;4100,0.8;4250,1];

% For Portfolio Labor Cost: 927.9 - 1238.8
Portfolio_Expense_Preferences{1,3} = [0,0;900,0.1;950,0.3;1000,0.5;1100,0.8;1250,1];

% For Portfolio Operations Cost: 38.19 - 42.21
Portfolio_Expense_Preferences{1,4} = [0,0;38,0.2;39,0.4;40,0.6;41,0.8;42,0.9;42.5,1];

% For Portfolio Retirement Cost: 76 - 84
Portfolio_Expense_Preferences{1,5} = [0,0;76,0.2;78,0.4;80,0.6;82,0.8;84,1];

% For Portfolio total Time to Launch: 45-54
Portfolio_Expense_Preferences{1,6} = [0,0;45,0.2;47,0.4;49,0.6;51,0.8;53,0.9;55,1];

Portfolio_MAU =
Calculate_MAU(Portfolio_Performance_Attributes,Portfolio_Performance_Preferences,Portfolio_Performance
_Weights);
Portfolio_MAE =
Calculate_MAE(Portfolio_Expense_Attributes,Portfolio_Expense_Preferences,Portfolio_Expense_Weights);

%scatter(Portfolio_MAE,Portfolio_MAU);

[Portfolio_pareto_set_MAE, Portfolio_pareto_set_MAU, Portfolio_pareto_count, Portfolio_p] =
PlotandFind_Pareto_For_System(Portfolio_MAE,Portfolio_MAU);
hold on
scatter(Portfolio_pareto_set_MAE(43),Portfolio_pareto_set_MAU(43),'*','g','LineWidth',12,'MarkerFaceCo
lor','flat')
scatter(Portfolio_pareto_set_MAE(19),Portfolio_pareto_set_MAU(19),'*','m','LineWidth',12,'MarkerFaceCo
lor','flat')
scatter(Portfolio_pareto_set_MAE(1),Portfolio_pareto_set_MAU(1),'*','k','LineWidth',12,'MarkerFaceColo
r','flat')
% There are 14 Pareto Optimal designs:
% 4,10,22,24,43,44,45,46,67,68,69,70,71,72
% 1. X=0.8419 Y=0.8923 - Design No: 10 [1,1,1,2,1,2,1]
% 2. X=0.6948 Y=0.7923 - Design No: 22 [1,1,2,2,1,2,1]
% 3. X=0.6254 Y=0.6923 - Design No: 24 [1,1,2,2,1,2,3]
% 4. X=0.5389 Y=0.5711 - Design No: 43 [1,2,2,2,1,1,1]
% 5. X=0.4709 Y=0.4711 - Design No: 45 [1,2,2,2,1,1,3]
% Design No: 4  X=0.9329 Y=0.9923 [1,1,1,1,1,2,1]
% Design No: 45 [1,2,2,2,1,1,3]
% Design No: 46 [1,2,2,2,1,2,1]
% Design No: 47 [1,2,2,2,1,2,2]
```