Increased Confidence in Concept Design through Trade Space Exploration and Multiobjective Optimization

by

Ryan Glenn Odegard

B.S., Mechanical Engineering
University of Arizona, 2006

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of
Master of Science in Aeronautics and Astronautics
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2008

Author ....
Department of Aeronautics and Astronautics
May 23, 2008

Certified by ....
David W. Miller
Professor of Aeronautics and Astronautics
Thesis Supervisor

Certified by ....
Jana L. Schwartz
Senior Member of the Technical Staff
The Charles Stark Draper Laboratory, Inc.
Thesis Supervisor

Accepted by ....
David L. Darmofal
Associate Department Head
Chair, Committee on Graduate Students

Increased Confidence in Concept Design through Trade Space Exploration and Multiobjective Optimization

by

Ryan Glenn Odegard

Submitted to the Department of Aeronautics and Astronautics
on May 23, 2008, in partial fulfillment of the
requirements for the degree of
Master of Science in Aeronautics and Astronautics

Abstract

The growing size, complexity and demands of engineering systems requires paying greater attention to the initial design of the system concept. To improve the process by which concept design is carried out, this thesis develops an Engineering Framework for Concept Development. The Engineering Framework is applicable to a diverse range of concept design problems. It helps guide the otherwise haphazard process of the early stages of design to provide confidence that the chosen concept is superior to a large set of alternatives. Accompanying the Engineering Framework is a collection of tools which aid the designer in analyzing different options. Two tools in particular are demonstrated for their mutually beneficial characteristics: 1) Object-Process Network is used to explore the full space of options, revealing the relationships among design decisions and system performance, and 2) a particle swarm optimization algorithm is implemented to efficiently search through the design space. The use of such an optimization algorithm becomes especially advantageous when higher fidelity models are included in the analysis because it is able to quickly identify the most favorable families of designs. The complementary approaches of exploring the entire trade space and then efficiently searching for the best groups of designs are shown to provide valuable insights in concept design problems. Two case study examples are presented as applications of the Engineering Framework and design tools. The first is an air-launched sounding rocket propulsion system design. The second is the design of a responsive disaster monitoring system. In each case, the use of the Engineering Framework and concept design tools give the designer increased confidence that quality concept designs have been identified.

Thesis Supervisor: David W. Miller
Title: Professor of Aeronautics and Astronautics

Thesis Supervisor: Jana L. Schwartz
Title: Senior Member of the Technical Staff
The Charles Stark Draper Laboratory, Inc.

Acknowledgements

I would like to thank a number of people who made this thesis possible, and who made my time at MIT so rewarding. First of all, I would like to thank my Draper advisor, Jana Schwartz, for her guidance and assistance through many aspects of my graduate school experience. She spent countless hours teaching me about aerospace and systems engineering, stimulating ideas about avenues of research, challenging me to think about things I hadn't previously and in new ways, giving me advice on career possibilities, recommending things to do in and around Boston, and the list goes on. I am extremely grateful.

I'd like to thank Professor Dave Miller for taking the time to listen to what we were doing across the street, and for providing many interesting ideas that we had not thought about. I greatly appreciate him sharing his extensive knowledge on space systems with me.

Another person I want to thank is Nick Borer. His seemingly endless knowledge on aircraft and design methods never ceased to amaze me. He was always willing to take time for me whenever I stuck my head in his office, and was exceedingly helpful when I did.

I was fortunate to make great friends during my time at MIT. Jaime Mateus and Luc Brunet made MIT much more fun than I thought was possible. Thanks for always giv'n'r. Thanks to Jim Keller, Phillip Cunio, and Zahra Khan for elevating the nerdiness of random conversations to a new all-time high. My fellow Fellow Gene Cook, trombone player extraordinaire, is one of the smartest and humblest people I know, and a mighty fine sailor to boot. And to the rest of the crew — Brandon Luders, Josh Krakos, and Josh Joseph.

Thanks also go to my roommates, Long Hua Lee and Andrew Mastin. Long Hua, as generous a person as any I know, expanded my understanding of the world and its cultures. His work ethic and ability to function on such little sleep are inspirational. Thanks to Andrew for late-night ping pong matches, ever-entertaining Wendy's runs, musical inspirations, and getting me through studying for the 6.255 final.

And I couldn't have gotten to where I am without the love and support of my family. Mom, Dad, Kyle, Julie — I am lucky to have you in my life. In expressing my gratitude for all you do and all you mean to me, words fall short. But I thank you.

May 21, 2008

This thesis was prepared at The Charles Stark Draper Laboratory, Inc., under Internal Company Sponsored Research Projects 21803-001 and 21136-001.

Ryan Odegard

Contents

List of Figures

List of Tables

Chapter 1

Introduction

"You can use an eraser on the drafting table or a sledge hammer on the construction site."

— Frank Lloyd Wright

1.1 Motivation

Today's engineering systems are becoming larger, more complex, and more challenging to design. As the demands placed on a wide variety of systems continue to increase, it is important to understand the design implications early on. Even the small but sophisticated devices that fit in our pockets are becoming more complex — they send messages via the Internet to users on the opposite side of the world, all with the aid of communications systems that span the globe and extend into space. With additional functionality and interconnections between systems, the list of requirements for many large, complex systems is growing.

The design of these new systems can be a difficult and inexact undertaking. For those systems that are direct descendants of another, it may be straightforward to design an alternate version that will be successful. For more innovative or complex designs, the risk involved in the task increases significantly. This is mainly due to the uncertainties that exist, both in how the concept will be implemented (what it looks like and what it does) and the environments that will exist when it is finally developed (physical, dynamic, social, economic, etc.). Furthermore, the interactions with existing systems adds another layer of complexity.

In addition to being the most uncertain, the beginning stages of concept design are also the most important. It is at this stage that the fundamental aspects of the system are decided and developed. If the initial concept is flawed, it is likely that implementing a successful system will prove difficult. The importance of early decisions is illustrated in Figure 1-1. This graph shows the classic relation between up-front decision making and total life-cycle costs. When a problem is first defined, the number of possible solutions is the largest it will ever be; that is, there is a large amount of design freedom. Exploring this large set of design choices occurs before committing to one particular design solution. Eventually decisions are made and the design is refined. As decisions are made in determining what the system design will be and how it will work, the freedom in design diminishes. Simultaneously, the commitment of the overall life-cycle costs is established. This loss of design freedom and commitment to life-cycle costs make the early stages of concept design among the most important phases of a program. This is especially true for large systems, where 80–90%

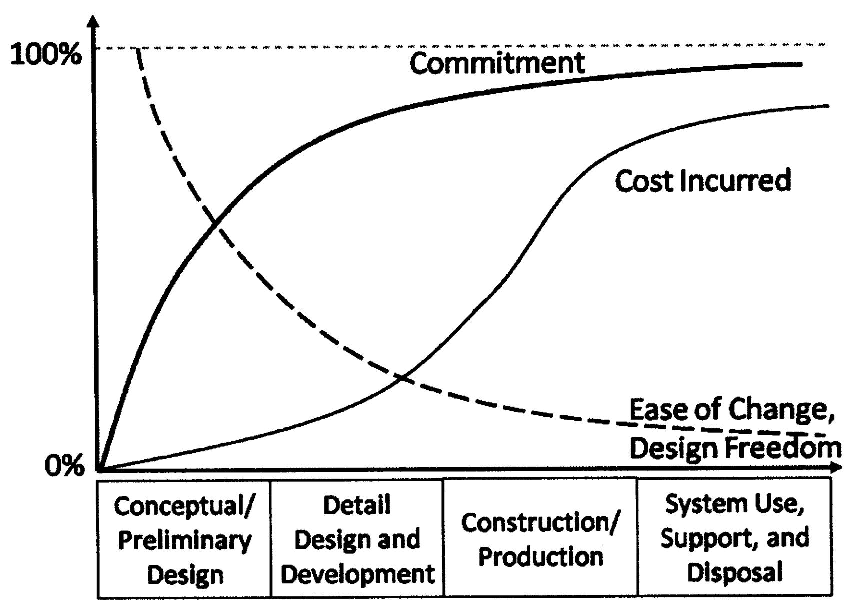

Figure 1-1: Life-cycle costs committed and incurred throughout phases of the design. [5]

of the development cost is predetermined by the time 5–10% of the development effort has been completed [27]. Therefore, an efficient, orderly process for defining and designing large systems is essential.

1.1.1 Apollo LOR Decision

The Apollo space program of the 1960s and early 1970s provides an example of the importance of early design decisions on the success of a large program. Tasked in 1961 by President John F. Kennedy to land a man on the Moon and return him safely to the Earth, NASA had many crucial decisions to make under uncertain circumstances. Up to that point little or no previous knowledge existed for many of the challenges that the program faced. One of the most critical decisions was made in 1962, when less than 1% of what would be the total cost of the program had been spent. It was then that NASA decided to use lunar orbit rendezvous (LOR) as the mission mode. Occurring seven years before Neil Armstrong set foot on the lunar surface, the selection of LOR is still today a hallmark decision of the program [12, 40].

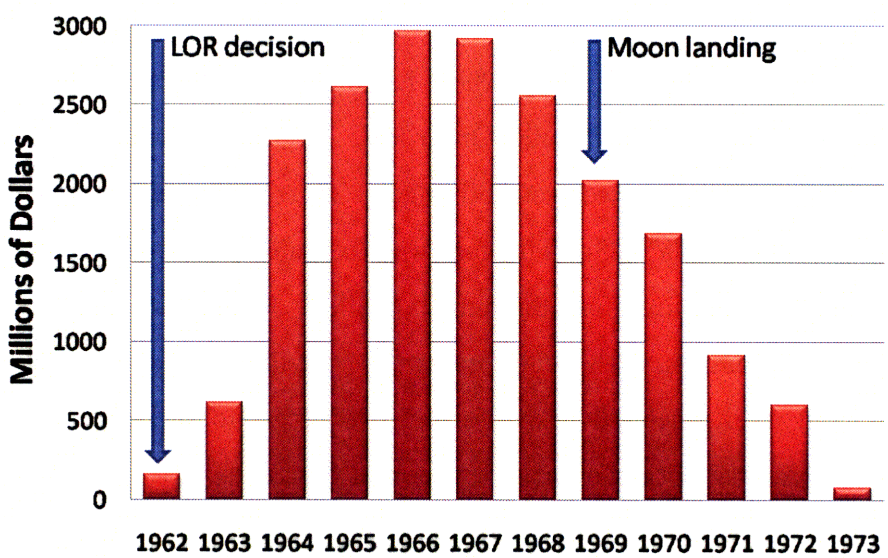

Figure 1-2: Annual breakdown of Apollo program funding. [80]

1.1.2 Space Shuttle Requirements

There are also historical cases in which poor decisions early in concept design caused unnecessary growth in life-cycle costs. A good example is taken from the U.S. Space Shuttle program. Initial requirements for the Shuttle led to the design of a two-stage reusable spacecraft with a payload capacity of 25,000 lb to orbit. Additional requirements were then adopted from the Air Force, which demanded a 40,000 lb polar orbit capability from Vandenburg Air Force Base and a 1,100 nautical mile crossrange capability. As a result of the adoption of these new requirements, the redesigned concept was a delta wing configuration, heavier and more expensive than the initial design. In the end the Shuttle never utilized Vandenburg AFB as a launch platform, nor did it ever fulfill the design mission envisioned by the Air Force. By avoiding the additional burdensome requirements that were never actually required, the program could have saved billions of dollars over its lifetime [31].

1.1.3 Orion Design Changes

More recent programs continue to display the need for thorough early examination of design decisions. The Orion Crew Exploration Vehicle is the spacecraft NASA is currently designing to replace the Shuttle fleet. In August of 2006, Lockheed Martin was selected as the prime contractor to design, develop, and build Orion [36]. About a year into the contract, the vehicle was exceeding its allocated weight budget. To counter this problem, the project created a "Zero Base Vehicle" [28]. The goal was to remove all of the components of the spacecraft except those required for minimum capability. Any features not essential had to demonstrate their worthiness, with weight being the overriding objective. The Orion design was thus reworked with redefined priorities and loss of capabilities. It continues to face challenges that originated with the initial conceptual design.

1.1.4 Emerging Engineering Design Approaches

In addition to the importance of early decisions on the success of programs, there is also the need to be able to design complex and collaborative systems. There is general agreement that the increasing complexity of today's systems is a pivotal issue and the source of the toughest challenges for systems architects and engineers [51]. As more is demanded of the next generation of complex engineering systems, quality up-front design becomes paramount.

The advances of computer technology in recent decades have revolutionized engineering design. In the span of a generation, computers have shrunk from room-sized pieces of equipment down to devices that fit in our pockets and contain orders of magnitude improvements in performance. With this rapid advancement of technology comes the ability of engineers to more effectively and more quickly analyze designs. Instead of relying on tedious hand calculations to verify the feasibility of a design, today's engineers have access to design tools that allow rapid simulations. Plus, the sophistication of analysis tools allows much more innovative designs to be considered. For instance, the advent of finite element programs has facilitated in depth analysis of complicated geometries in structure design that were extremely arduous in the past.

With the time required for design iterations continually decreasing, it is easier for the designer to explore many different design options. Rather than selecting a baseline or reference design and trying to improve individual aspects of the design, the designer should examine a large set of alternatives to understand the effects of design decisions across the design space. Often times, a design team shrinks the range of design options by falling back on past design experience [26]. This is limiting as these options may be sub-optimal for the current design problem. Furthermore, engineers tend to latch on to concepts, spending large amounts of time trying to rework a design rather than considering alternative solutions [6]. Therefore, it is important to consider as many alternatives as possible early in the design process to avoid potential redesign work, cost overruns, and schedule delays later on.

In the conceptual design phase, computer power has been increasingly leveraged to manage not only the subsystem engineering design, but also system-level optimization. Multidisciplinary Design Optimization (MDO) approaches have shown great value in concentrating on optimization at the system level [59, 60]. As a complementary approach to trade space exploration in concept design, optimization allows the designer to efficiently and quickly identify families of designs that exhibit superior features. The tandem utilization of trade space exploration and optimization in concept design is a powerful approach to identifying the best families of designs.

1.2 Objectives

In an effort to improve the way concept design is carried out, the goal of this thesis is to develop an Engineering Framework for Concept Development and to demonstrate a set of design tools that work collaboratively within that Engineering Framework. The use of the Engineering Framework and concept design tools increase the confidence of the designer that high-quality concepts are identified early in the process. Specifically, the Engineering Framework for Concept Development is demonstrated with the use of the following concept design tools:

- Object-Process Network (OPN) is used to explore a full range of possible concept alternatives. Its ability to visually represent the design space aids in understanding the relationships among design options. Additionally, OPN incorporates mathematical features to model the space of feasible combinations of alternatives.

- Particle swarm optimization (PSO) is a heuristic search algorithm that efficiently identifies regions in the design space that contain the best designs, and — with the incorporation of higher fidelity models — provides the designer a means of reducing computational effort without limiting the search space.

Further knowledge of the trade space is gained with the use of two additional tools, which in this thesis supplement the trade space exploration and optimization tools:

- Pugh analysis is a qualitative method that involves pair-wise comparisons between different concepts and provides a systematic way to make subjective tradeoffs early in concept design.

- Analysis of variance (ANOVA) and trade space sensitivity techniques are used in the later phases of concept design. ANOVA is a statistical means of identifying the most important design variables in a set of design experiments. Sensitivity studies involve the post-processing of results to better understand the underlying uncertainties in the models and parameters.

1.3 Overview

The remainder of this thesis is organized as follows. Chapter 2 is a review of the literature relevant to trade space exploration and optimization in concept design. Chapter 3 discusses the Engineering Framework for Concept Development, Pugh, OPN, PSO, and ANOVA. Chapter 4 contains two examples of applications in demonstrating the value of the Engineering Framework and concept design tools in the early stages of design. Chapter 5 provides a summary and conclusion on the topics and application examples, as well as a discussion on future work that can be pursued in this area.

Chapter 2

Literature Review

"We are searching for some kind of harmony between two intangibles: a form which we have not yet designed, and a context which we cannot properly describe."

— Christopher Alexander

This chapter presents a review of the literature relevant to concept design, trade space exploration, multiobjective optimization, and trade space sensitivity analysis. Prominent design processes are presented for context to the Engineering Framework. Then strategies for exploring trade spaces are discussed. The role of optimization in concept design and the development of particle swarm optimization is introduced. Lastly a review of sensitivity analysis for understanding uncertainty in trade spaces is presented.

2.1 Design Processes

Design, as indicated by the opening quote from Christopher Alexander, takes vague ideas and notions of something novel, and transforms them into a useful product. The process of taking a concept through design, development, and implementation has different names depending on the field. These terms include the system acquisition process, the product development process, and the product design process. Besides having different names, the processes also have specific phases that can be domain-dependent. In general, the design process consists of the steps shown in Figure 2-1. These major steps are: conceptual design, preliminary design, detail design and development, production/construction, operational use and system support, and retirement. Other generic models for describing the design process are the "vee" process model, the spiral model, and the waterfall model [5]. The "vee" model, shown in Figure 2-2, represents how the focus of work goes from the top-level design of the entire system, down to the details of the components, and then up again when the components are integrated together to constitute the full system. Figure 2-3 shows the spiral process model. This representation emphasizes the iterative nature of design, where the same steps are visited multiple times. In each cycle the design becomes more refined. The waterfall model in Figure 2-4 shows the forward path of steps to take in designing and implementing a new system. The feedback portion of the model allows iterations and refinement as necessary.

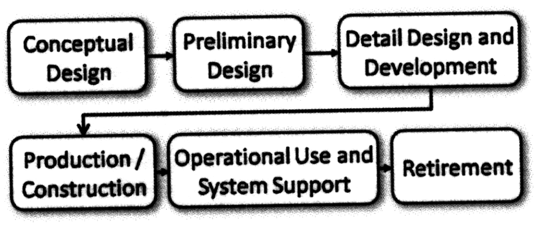

Figure 2-1: General process [5]

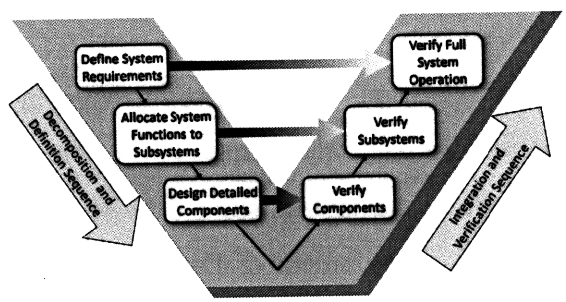

Figure 2-2: "Vee" Process Model [5]

Over the years different people and organizations have adopted their own tailored versions of the design process. For spacecraft design, Larson and Wertz [47] define a space mission analysis and design process that consists of the following steps: define objectives, characterize the mission, evaluate the mission, define requirements. Raymer [67] gives an overview of the aircraft design process, which has three phases — conceptual design, preliminary design, and detail design. The iterations on design include evaluating the requirements, developing design concepts to meet the requirements, conducting design analysis, and doing sizing and trade studies.

Many organizations have developed processes for the design of large systems and missions. The NASA systems engineering process consists of the following phases: mission feasibility, mission definition, system definition, preliminary design, final design, fabrication

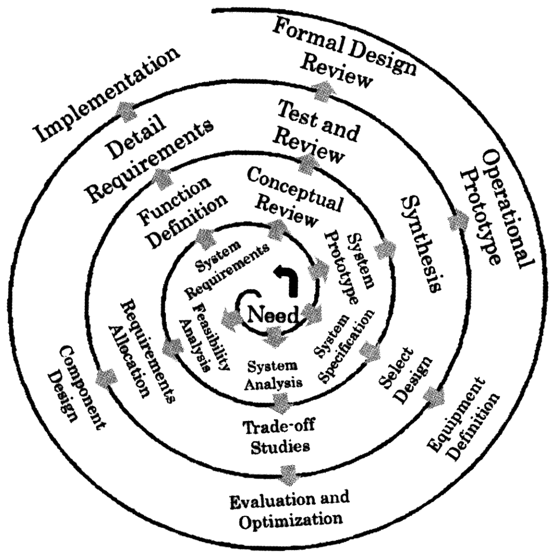

Figure 2-3: Spiral Process Model [5]

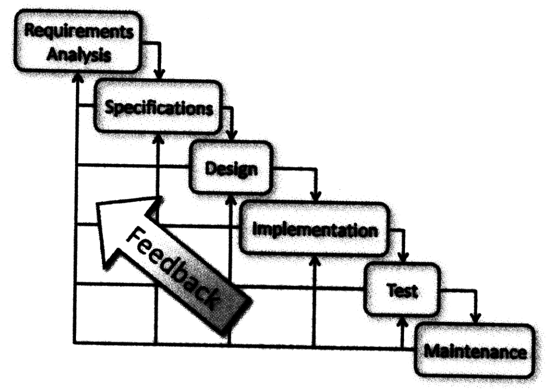

Figure 2-4: Waterfall Process Model [5]

and integration, preparation for deployment, deployment and operational verification, mission operations, and disposal [76]. For NASA's programs, the mission design is the early conceptual work, followed by deeper levels of design before the system is built and flown. The U.S. Department of Defense (DoD) has a process that includes the following phases: Pre-Phase 0 (determination of mission need), Phase 0 (concept exploration), Phase I (program definition and risk reduction), Phase II (engineering and manufaturing development), and Phase III (production, fielding/deployment and operational support) [27].

In addition, modern systems engineering is adopting improved process management techniques initially developed in software engineering, including the CMMI (Capability Maturity Model® Integration) approach [55]. In Ref. [48], Lee uses a top-down approach to architecting software systems. Lee considers a two-phase process. This process first addresses the quality goal requirements and maps those requirements to strategic architectural decisions. Then an architectural tradeoff analysis assesses correlations between architectural decisions and quality goal requirements. The next phase consists of determining the quality factor requirements, which is a refinement of the quality goal requirements. These requirements are mapped to tactical decisions (rather than the more general strategic decisions made in the first phase). Finally, a tradeoff analysis is performed in the same manner as the first phase, concluding the software system design process.

Many of these processes have specific gates and milestones that must be met before proceeding to the next phase. Unfortunately, each of these customized processes may not translate across domains, which can pose a problem during the design of very innovative concepts or large, interacting systems. The Engineering Framework for Concept Development proposed in this thesis, explained further in Chapter 3, is an abstraction of the specific concept design processes. It serves as a guide for constructing an appropriate process suitable to a specific design problem, regardless of maturity or domain.

2.2 Trade Space Exploration

At the outset of design, there are many possible ways to meet the needs of the customer, user, and stakeholders. In essence, there are different tradeoffs to consider when selecting a concept to design and develop. The set of all possible design options is referred to as the design space, and choosing the most favorable design based on its attributes is carried out in the trade space. To give an example of the difference between the design space and the trade space, consider options for getting to work. The design space might include traveling by car, train, plane, bike, or by walking. The trade space (also referred to as the objective space or attribute space) would show how each of these options performs — walking is slower than driving, but also cheaper; taking the train is safer than driving, but does not offer as much flexibility in picking a final destination.

Gries [26] gives an overview of methods available for use in exploring the space of design options. He provides as motivation for searching through many possible designs (opposed to conducting a limited search of only a few options) the following future trends: increasing complexity of designs, heterogeneous architectures, and decreasing time to market. Since an exhaustive search of the design space is generally prohibitive, Gries provides examples of the tradeoffs that must be considered in choosing which designs to evaluate. These tradeoffs include the total number of designs to evaluate, the accuracy of the evaluation, the analysis time for each, the precision or granularity of the design space coverage, and automation in trade space exploration.

These tradeoffs are discussed in subsequent subsections. To illustrate two of the tradeoffs that are present in trade space exploration, Figure 2-5 shows the balance between the objectives of obtaining highly accurate results and doing so in a time-efficient manner.

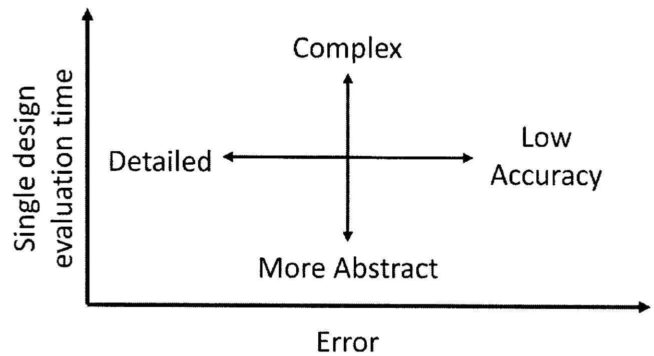

Figure 2-5: Tradeoff between evaluation accuracy and analysis time.

2.2.1 Point design

The quickest but most limited exploration of the design space is done via single point design optimization. Often an existing concept is used as a baseline and the subsystems then optimized for new requirements. This approach may demonstrate the feasibility of the new concept, but it does not provide information on whether it is the best option or even how to find a more optimal system-level solution. Benchmarks, simulations and analytical approaches can all be used in evaluating a single design [26]. Analytical approaches often provide less precision than simulation, but simulations require an executable model. Unlike the methods of this thesis that demonstrate the value of trade space exploration, point design analysis does not provide the knowledge that is gained by analyzing many different design options.

2.2.2 Random search

A Monte Carlo simulation randomly samples the design space [26]. This allows the designer to look at more options and obtain a better understanding of the space than with a single point design evaluation. Random searches are hindered by the fact that there is not, in general, evenly distributed coverage of the design space, and local regions may be over- or under-explored [14]. The principles of Tabu search [20, 21] improve upon random searches. The Tabu search technique enforces diversification into unexplored regions of the design space by avoiding evaluation of the same design twice, and also incorporates mechanisms to explore the vicinity around promising design points [26, 14].

2.2.3 Design of Experiments

A collection of statistical techniques that provide a more structured approach to exploring the design space falls under the heading of Design of Experiments (DoE). Originally developed by R.A. Fischer in the 1920s and later employed by Genichi Taguchi in his revolutionary approaches to Quality Engineering, experimental design aims to reduce the number of design evaluations needed to obtain information on the design decisions [93]. Without having to exhaustively search the design space, DoE has the advantages of being able to study the effects of multiple input variables on one or more output parameters, identify key drivers among potential design drivers variables, identify appropriate design variable ranges, identify achievable objective function values, and study the robustness and sensitivity in the context of uncertainty [14].

Full Factorial Design of Experiments

A full factorial run of DoE evaluates every level for every factor in the design space. A factor, also called a design variable, is something the designer controls in evaluating different designs. An example might be the size of a component. Examples of the levels associated with a factor may be dimensions such as 1 m or 2 m. A two-level full factorial DoE includes a high and low level for each factor. A three-level full factorial DoE has a low, moderate, and high level for each factor. The full factorial DoE technique provides the coverage of the vertices of the design space. It is therefore able to capture the main effects (averaged effects due to individual factors) as well as interaction effects (when the effect of one factor depends on the level of another) [14].

The Object-Process Network (OPN) tool [44] provides this full coverage of the space. OPN has been used to evaluate the complete space of architectural options for NASA's return to the Moon [77]. Simmons describes the use of OPN to enumerate the space of design options, which include the number of crew, propulsion type, trajectories, types of vehicles, and the ultimate destination. OPN can be used to enumerate all of the design options in a full factorial experiment while providing additional capabilities not explicit in DoE. These capabilities, explained further in Section 3.3, include logical rules and constraints to eliminate infeasible options. In Ref. [78], Simmons and Koo present an application of OPN for enumerating the design space for a Shuttle-derived launch vehicle. They show how searching the space using OPN to enumerate all the feasible combinations allows greater information on the trade space than if a small set of designs were evaluated.

Gralla *et. al* [24] use a full factorial search to examine metrics for the optimal launch vehicle size for Moon and Mars transportation architectures. The full factorial search is used in combination with an integer optimization formulation to minimize cost. This approach utilizes DoE for its trade space exploration perspective and optimization for the benefit of lower total evaluation time. Lamassoure and Hastings [45] also search an entire trade space using generalized metrics and first-order models to examine the cost-effectiveness of on-orbit servicing for several types of space systems. Lamassoure, Wall and Easter [46] utilize a full-factorial search in the design of deep space missions, paying particular attention to capturing sensitivities to key requirements early in the design.

Fractional Factorial Design of Experiments

One disadvantage of the full factorial Design of Experiments approach is that it leads to combinatorial explosion [14]. Consider a problem with 7 factors with 3 levels each. This requires $3^7 = 2187$ evaluations; adding 3 factors increases the number of evaluations to 59,049. Fractional factorial design aims to reduce the number of evaluations while maintaining coverage of the space and the ability to capture the effects of the factors. Some fractional factorial experimental design techniques include orthogonal arrays, half-fractional, Latin hypercubes, parameter study, and the one-at-a-time method [58, 14].

2.2.4 Trade Space Visualization

The idea of exploring a large trade space implies the need to intelligently interpret the results. These results can include hundreds or thousands of unique designs. This extensive amount of information is only useful if it can be understood. In single point design methods, the numerical data can be interpreted in a tabular format. With many designs and many variables to track, however, this approach becomes impractical. Instead, visualization techniques can be used to gain understanding into the problem.

The approach of exploring and visualizing the trade space before the designer expresses his or her design preferences is the design strategy used in this thesis. First outlined by Balling [1], the idea of "design by shopping" refers to this *a posteriori* articulation of preference. Likened to buying a car, the design by shopping paradigm advocates providing a "car lot" of engineering designs from which the customer can form opinions and choose the design he or she feels is the best. This approach is the opposite of *a priori* articulation of preferences in which the customer expresses up-front priorities in what design attributes are most important. If the customer knows what these attributes of the design are, relative importance weightings on the design objectives can be used to find a design that is optimized to those preferences. On the other hand, the design by shopping paradigm relies on forming opinions subsequent to seeing the trade space of designs. This adds flexibility in making decisions and offers additional insights to the design problem.

With the design by shopping paradigm as motivation, Stump and colleagues at the University of Pennsylvania use glyph visualization to find key trends and relationships in multidimensional data sets. They present applications involving the visualization of trade spaces that include Mars satellite missions [83, 84] and autonomous underwater vehicle models [83, 82]. The glyph visualization technique that they use is one in which designs are presented in a trade space with different properties representing certain parameters or variables. These properties include color, shape, size, transparency, etc. Ward [92] provides a thorough summary of the use of glyphs in multidimensional data visualization.

A number of other sources in the literature utilize visualization techniques to interpret the trade space data. The work by Cornford *et. al.* [11] encompasses both design and development choices in the design space. They argue that making decisions based on *a posteriori* preferences enhances the selection of preferred designs, provides the means to evaluate "what-if" scenarios for revised requirements, and allows a visual examination of the sensitivities of designs. Kanukolanu, Lewis and Winer [41] use visualization techniques such as blocking together portions of the trade space and multidimensional scatterplots to assess tradeoff decisions for coupled systems. Another example showing the value of representing trade space information visually is given in Ref. [8]. Here Cohan *et. al.* use different colored glyphs in two-dimensional trade space plots for each design variable of a modular optical space telescope. In Ref. [54], McManus and Schuman explore a utility-versus-cost trade space for an orbital transfer vehicle. They highlight distinct propulsion options to visually divide the space according to different types of vehicles.

These examples all exhibit deliberate use of visualization techniques to understand the trade space implications of decisions. There remain challenges in presenting multi-dimensional information in a way that proves useful and understandable to the designer and decision-maker, especially with media restricted to two dimensions. Yet the previous research and the work in this thesis demonstrate the value in visualizing trade spaces to build insight into the design problem.

2.3 Optimization in Concept Design

The words "optimization" and "concept design" do not seem suitable to appear in the same sentence. When a concept is initially designed, the large amount of uncertainty in its description prohibits it from being optimized in the traditional sense of the word. The role of optimization in concept design is not to "optimize" the concept. Rather, there are many techniques from the operations research and optimization communities that can be leveraged in the search for good concept designs.

Much of the work involving optimization early in the design process has been in the field of Multidisciplinary Design Optimization (MDO). Two of the goals outlined in a white paper issued by the AIAA Technical Committee on MDO in 1991 [9] are: 1) improve optimization algorithms for effective handling of very large numbers of design variables, disjoint and nonconvex design spaces, multiple extrema, and multiobjectives; and 2) improve post-optimum sensitivity analysis for greater computational efficiency. While progress has been made since that time, there is still much work that can be done.

As discussed further in Chapter 3, many types of conceptual engineering design problems, especially for large, integrated systems, prove challenging for optimization algorithms. The first goal mentioned from the AIAA MDO white paper addresses this point directly. The most successful to date have been: gradient-based algorithms with backtracking mechanisms (to avoid local extrema) and evolutionary search algorithms such as genetic algorithms (GAs) [32]. The heuristic optimization method of simulated annealing (SA) has also been shown to work well [43]. Jilla [39] outlines a methodology for conducting trade space exploration of distributed satellite systems that utilizes simulated annealing to intelligently search the space. Schuman and de Weck [71] use system-level optimization with both neural networks and genetic algorithms for the multiobjective design of a Space Shuttle external tank model. These cases are examples of the successful implementation of optimization in the context of concept design.

The following subsections discuss the two points taken from the AIAA MDO white paper further. First lessons learned from the MDO community are presented to validate the use of optimization in concept design. Then a review of the literature for a recent optimization algorithm — particle swarm optimization — is discussed. The following section on sensitivity in concept design trade spaces addresses the second goal from the AIAA MDO white paper.

2.3.1 Multidisciplinary Design Optimization

In recent years the advances in computing power have ushered in the growth in popularity of MDO. The field of MDO first gained traction in the aerospace industry. The physics dictating aircraft design inherently involves many coupling characteristics. For instance, in both rotary and fixed wing aircraft, there are strong interactions between the aerodynamics and complex structural dynamics that cause the phenomenon known as flutter. The flutter of the aerodynamic surfaces is further complicated by direct coupling with the flight control system used to trim the aircraft [9]. There is a necessary balance that must occur, and finding the optimal design requires full consideration of all the interacting elements.

The demand for multidisciplinary optimization in aerospace engineering stems from two primary sources [9]:

1. Major new aircraft designs occur less often, limiting the role of past experience in the decision making process.

2. Advanced aircraft tend to be enormously complex systems, and finding an optimal design (or knowing how to get there) is beyond the power of human judgment when "everything influences everything else."

These motivations for MDO from the aircraft industry are justifying its application to other fields as well. Mosher [59] explains how, in the case of spacecraft design, a manually optimized design does not guarantee a system level focus, and the collection of high-performance subsystems may not integrate into a highly efficient system implementation. MDO techniques, on the other hand, combine a multidisciplinary approach with optimization methods to direct the search through the design space in an intelligent manner. This reduces the set of candidate designs that must be evaluated, focusing effort on better areas of the space and sometimes leading to counterintuitive results.

2.3.2 Particle Swarm Optimization

Particle swarm optimization is one of the more recent heuristic search algorithms. It was conceived in 1995 by Kennedy and Eberhart [42, 15] with the goal of modeling the social interactions of flocks of birds. Subsequent work by Eberhart and Shi [16], as well as Venter and Sobieszczanski-Sobieski [89], tested the algorithm's efficacy and input parameters, and verified the algorithm's advantages over gradient-based optimization techniques.

Hassan and Cohanim [29] present a comparison between genetic algorithms and particle swarm optimization. In addition to several test cases, they examine applications for spacecraft reliability design and radio telescope array configuration optimization. Their findings indicate that the particle swarm algorithm performs as well in its effectiveness (finding the global optimum) as the genetic algorithm, and that it requires fewer function evaluations (better computational efficiency).

In trade space exploration, there is the implicit need for the problem to have multiple objectives. For instance, if the only goal is to minimize cost, then the design with the lowest cost should be chosen. Tradeoffs arise when other characteristics of the design are factored into the decisions. There is the need then for optimization algorithms that can solve multiobjective problems.

Coello Coello and Lechuga [7] introduce a multiobjective formulation of the particle swarm algorithm. This algorithm uses a repository that divides the population of particles and guides them toward multiple objectives. The space is divided into hypercubes where selection of the "best" particles is inversely proportional to the number of non-dominated solutions in the cube.

Grant and Mendeck [25] apply a multiobjective particle swarm technique to trajectory optimization for atmospheric entry of the Mars Science Laboratory. The algorithm handles both two-objective and three-objective cases. Their multiobjective particle swarm optimization (MOPSO) algorithm uses an external archive (similar to the repository in Ref. [7]) of non-dominated solutions to communicate information between particles of the swarm for finding the Pareto front in multiobjective problems.

Other multiobjective approaches have been developed as well. Mostaghim *et. al.* [61] introduce the sigma method, which directs particles along different directions based on relative importances of the objectives. Fieldsend *et. al.* [18] proposes a dominated tree for storing particles and moving in the direction of the Pareto front. Lastly, Hu *et. al.* [33] develop a dynamic neighborhood strategy for picking the best particle in multiobjective problems. This approach is simple in its implementation but requires the selection of one objective as the optimization objective. It therefore suffers from sensitivity to that selection.

A variety of hybrid algorithms that incorporate other optimization methods have been investigated in the literature. These applications are numerous and will not be listed here, but hybrid PSO examples related to concept design include the use of sequential quadratic programming and neural networks [66]; combination of dynamic niche technology with PSO [49]; and a hybrid of simulated annealing, Kriging meta-modeling, and PSO [37].

2.4 Sensitivity in Trade Spaces

The large amount of uncertainty early in a design prompts the need for ways to measure how sensitive the design options are to changes. Similar in their application to exploring a design space, Taguchi methods are often used for achieving robust designs [93]. These techniques make early designs more reliable and more robust to uncertain conditions in manufacturing and the marketplace.

Walton and Hastings [91] outline a framework that explores the implications of uncertainty in conceptual design architectures. The first step is to identify and quantify the uncertainty sources. The next step is to develop distributions of outcomes. This is done by taking the best, the worst and the expected states and evaluating each one in a full factorial Design of Experiments.

Uebelhart [86] examines uncertainty in conceptual design by using Design of Experiments tools and propagating parametric uncertainties through analytic models. He is then able to place bounds on the resulting designs in the trade space. Comparisons are made between the vertex and Monte Carlo methods for bounding the uncertainty range, where the vertex method reproduced the bounds of the Monte Carlo analysis in 99% of 700 trials. Each of these has particular applications, and both are demonstrated in this thesis.

Masterson [52] describes a methodology for developing robust designs in the face of high levels on uncertainty. Her approach of Robust Performance Tailoring and Tuning uses optimization and isoperformance analysis techniques to make adjustments during design as well as on the physical hardware. This allows better compensation for uncertainties. The resulting design is more robust, thus less sensitive to variations.

2.5 Literature Review Conclusion

This chapter presented a review of the literature related to concept design processes, trade space exploration, multiobjective optimization, and trade space sensitivity techniques.

Many versions of the design process exist, and they have evolved over time to meet the needs of various industries and organizations. Some particular models are the "vee" process model, the spiral process model, and the waterfall process model. Particular design processes have been promoted in different vehicle design strategies (satellites, aircraft), in various organizations (NASA, DoD), and in different fields (CMMI from software engineering).

Trade space exploration can be carried out in a number of ways, including random search, full factorial experimental design and fractional factorial experimental design. To help interpret the resulting trade space(s), various visualization techniques have been developed to allow the designer to make decisions based on *a posteriori* preference information.

Optimization in the context of concept design has seen increasing popularity with the development of the field of MDO. As systems become more complex and the interactions among subsystems unpredictable, a systems-level optimization is increasingly important. Particle swarm optimization is an algorithm that is well-suited to handle the types of problems often seen in concept design stages. Multiobjective algorithms have been developed that enable an efficient search for Pareto front designs.

Sensitivity is an important consideration in concept design because of the large amount of uncertainty present. The previous work of several authors was reviewed for their relevance to trade space uncertainty and sensitivities.

Chapter 3

Engineering Framework and Concept Design Tools

"To dismiss front-end design as mere icing is to jeopardize the success of any site."

— Curt Cloninger

This chapter describes the Engineering Framework for Concept Development, as well as the concept design tools of Pugh analysis, Object-Process Network, particle swarm optimization, and analysis of variance.

3.1 Engineering Framework for Concept Development

The opening quote to this chapter, though referring to website design, is representative of the importance of the initial concept design in many different types of projects. This is where the advantages of a guide for approaching the early stages of design becomes most useful.

In Chapter 2, the design process and several tailored versions of it were discussed. At the beginning of all of those processes was a period dedicated to determining the design concept. Then a series of iterations refine this concept and add detail to its description. The numerous customized design processes, however, contain domain-dependent steps for carrying this out. An Engineering Framework for Concept Development has been proposed at The Charles Stark Draper Laboratory, Inc., as a guide for conducting the difficult and important early design stages, regardless of concept domain. This Engineering Framework is an abstraction of the more specific concept design processes. It enables a user to create a process tailored to the specific problem at hand.

3.1.1 Engineering Framework Activities

The Engineering Framework* is defined by four problem-solving *activities*. The successful execution of these activities constitutes a *phase*. Each phase requires all four activities, but the number of phases and the type of data considered in each phase differ depending on the domain and resolution of the problem.

The four activities, described below, are necessary to successfully complete a phase. They should be performed in the order listed since downstream activities often rely on those upstream, and jumping ahead can result in hasty decisions that diminish the quality of the final design. Issues that arise in any one or more activities can be resolved by revisiting upstream activities if necessary; as in most iterative design, the flow is not intended to be purely linear, but rather a guide for progression.

1. **Problem Characterization** involves gathering necessary information on the problem to correctly characterize it. This can include descriptors of value, external influences and constraints, distinguishing attributes, stakeholders and their objectives, and *a priori* preference information.

2. **Alternative Generation** is the activity in which distinct alternative solutions to the problem are generated. Cataloging of alternatives can occur by directly specifying a potential solution, or by enumerating and combining design options into feasible

*The Engineering Framework for Concept Development is currently being developed at Draper Laboratory, with collaboration taking place through http://collab.draper.com/projteams/cdmc alternatives. The design options can take discrete values (such as technology choices and binary decisions) or have continuous ranges (e.g., dimensions, power levels, mass distributions).

3. **Model Development and Evaluation** consists of mapping the alternatives to the criteria from the *Problem Characterization* activity. This activity measures how well the alternatives deliver value while not violating the constraints and meeting the objectives of the stakeholders. Depending on the type of data available, this can be done with expert opinion, empirical evidence, or simulation models to represent system performance.

4. **Decision Analysis** is the last activity and constitutes building intuition and insight about the alternatives. This is done by evaluating the data on each alternative with respect to the problem criteria. The goal is to learn about the important tradeoffs and how design decisions affect the achievement of the objectives. Insights can include *a posteriori* preference modification (what was considered important may change), assessment of feasibility of alternatives (non-viable solutions are discarded), and comparison to external references (benchmarks). If a subset of alternatives appears to satisfy constraints, not conflict with external influences, and achieve acceptable value, then this subset can be carried through a more detailed iteration of the activities until a sufficient level of detail has been reached that adequately describes a solution to the original problem.

3.1.2 Engineering Framework Phase Classes

The set of four Engineering Framework activities constitutes one iteration of design. While the activities can be repeated any number of times until a fully described concept emerges, it is convenient to include a description of the level of detail one might be working at for a given iteration through the problem. The Engineering Framework incorporates *phase classes* that describe and classify levels of refinement in the design. The phase class descriptions are:

1. **Exploration** involves the least precise data. In this phase class the problem is not well-defined in terms of measures of value, external influences, or distinguishing attributes. Idea generation is highly creative and identifies high-level classes of solutions. Mission options are far-ranging and include few specifics. Relationship evaluation is imprecise and results in highly qualitative comparisons. Decision analysis is based on subjective assessments and user judgment. If analysis is used, it is often heuristic, based on empirical data, or based on a limit analysis of fundamental theorems. The results should indicate which concepts are clearly inferior or infeasible, and should point to those which merit further investigation.

2. **Selection** involves a deeper understanding of a smaller set of candidate solutions. The overall problem is understood, and the external influences, including outside constraints, limitations, and competition should come into play. Alternative generation focuses on combining different components that constitute the design. Mission operations are explored in greater detail. Relationships may be established based on fundamental theorems, empirical data, and expert opinion, and will result in mixed quantitative-qualitative values. Decisions require more rigorous ranking of the alternatives based on the data. The results should describe how alternatives perform with respect to one another and indicate which to examine in more depth.

3. **Refinement** involves the most detailed analysis of alternatives. The problem is clearly defined, complete with a concept of operations, identified stakeholders, constraints, and competitors. The concept is parameterized and the parameters explored within reasonable ranges to generate alternatives. Relationships are built on physics-based, detailed models, and may require the integration of subsystems. The mission or mission envelope is fully quantified. The data are highly quantitative and held to a fair degree of precision. The decisions should be more focused and may rely on numerical or graphical techniques. The end result describes a feasible system concept design.

One other note to the phase classes is that there may be more than one iteration necessary in a phase class. The descriptions of the phase classes, however, remain sufficient for classifying the level of detail and the type of data with which one is working.

3.1.3 Engineering Framework Design Tools

The Engineering Framework for Concept Development makes it possible to organize and classify various design tools. The arrangement of the Engineering Framework in a matrix structure provides a clear representation of this, as shown in Table 3.1. Pugh analysis, Object-Process Network, particle swarm optimization, and analysis of variance are included in the cells in the matrix in which they are applied.

Table 3.1: Matrix representation of the Engineering Framework for Concept Development with Pugh, OPN, PSO, and ANOVA included.

	Problem Characterization	Alternative Generation	Model Development & Evaluation	Decision Analysis
Exploration			Pugh	
Selection		OPN	OPN	ANOVA
Refinement				PSO ANOVA

Having concept design tools organized with respect to the Engineering Framework makes it easy to identify what tools are available at different stages of design. Pugh analysis, for example, is best used when concept evaluation is necessary at a qualitative level. On the other hand, OPN is well suited for developing models and evaluating the results when the type of data is a mix between qualitative and quantitative. Particle swarm optimization and analysis of variance are useful for examining the results, extracting useful information, and building insight into the problem when the models have been refined.

It is worthwhile to highlight some nuances to concept design tool placement in the Engineering Framework. Optimization techniques can be considered a nested iteration of the activities of *Alternative Generation*, *Model Development & Evaluation*, and *Decision Analysis*. A particular optimization algorithm will pick, evaluate, and assess one or more designs with respect to the criteria from the *Problem Characterization* activity, and then repeat the final three activities based on the result found, attempting to find a better solution. Because the ultimate goal of the algorithm will be to find good results on which to make decisions, it is best classified under *Decision Analysis*. As an example of a tool that spans phase classes, ANOVA is applicable to the *Selection* phase class as well as the *Refinement* phase class. OPN is well suited for generating combinations of feasible design options, an *Alternative Generation* activity, and also evaluating the design combinations, which is *Model Development and Evaluation.*

Other design tools can also be organized to fit in the Engineering Framework. Examples of tools applicable in *Problem Characterization* include the Delphi method [22], the Analytical Hierarchy Process [70], and stakeholder analysis [13]. Some design tools for *Alternative Generation* are affinity diagrams [22], morphological matrices [13], trade trees [22], and Design of Experiments techniques [90]. For *Model Development and Evaluation*, useful design tools include Design Structure Matrices [81], axiomatic design [79], regression analysis, and risk assessment matrices [22]. Finally, some examples that fall in the *Decision Analysis* activity are utility theory [85], Physical Programming [56, 57], as well as a variety of other optimization and statistical tools.

By selecting appropriate design tools to use in the Engineering Framework, a tailored design process can be developed for specific applications. An example of a custom process that fits this description is the design process formulated by Ross *et. al.* [68]. This process for designing space systems, Multi-Attribute Tradespace Exploration (MATE), involves tools such as stakeholder analysis, full factorial DoE, utility theory based on formal interviewing techniques, and sensitivity analysis. The MATE process also includes an iteration that involves a detailed, concurrent engineering session. The phase classes of the Engineering Framework support this tailoring of the design process to particular applications.

3.2 Pugh Analysis

The first design tool demonstrated as part of the Engineering Framework is Pugh analysis [65]. This is an *Exploration* phase class design technique used to compare different concepts. It therefore falls in the *Model Development and Evaluation* activity. Pugh analysis helps guide subjective, qualitative pair-wise comparisons of various concepts. The result is a relative ranking of the concepts, which can be used to select a smaller set of concepts to explore in greater detail. In addition to the ranking of alternatives that Pugh analysis provides, the other important aspect is the dialogue that is stimulated when team members collaborate to discuss the merits of competing concepts. When the problem, the concept alternatives, and the goals are vague, structuring the discussion to clarify these ideas is very useful.

3.2.1 Pugh Implementation

The necessary inputs to conduct a Pugh analysis are the attributes of the design and a set of concept alternatives. The attributes are what the stakeholders of the system care about. These are used to measure the "goodness" of the concept alternatives. The attributes and concepts are arranged in a matrix, with the attributes in the first column and the concepts along the top row. An example matrix is shown in Table 3.2.

Table 3.2: Example Pugh Matrix

	Concept 1 Baseline	Concept 2	Concept 3	Concept 4
Attribute 1	0	1	-1	0
Attribute 2	0	1	-1	0
Attribute 3	0	0	0	1
Attribute 4	0	1	1	0
Attribute 5	0	1	-1	0
Score	0	4	-2	1

Ref. [65] outlines the procedure for using the Pugh matrix. It is summarized here. The first step in the analysis is to select a baseline concept. This concept will be the datum against which the others are compared. For each attribute, the concepts are compared to the baseline concept and given a score of (+1) if it performs better in that attribute, (-1) if it performs worse, or (0) if there is negligible difference between it and the baseline concept. By comparing concepts over all of the stakeholder attributes, it becomes evident which concepts are the "winners" and which are the "losers." The simplest way to quantitatively measure the goodness of each concept is to sum the numbers in each concept's column. The highest score represents the best rankings relative to the baseline concept.

In the example matrix shown in Table 3.2, there are four concepts being evaluated for five attributes. The scores indicate that Concept 2 is the best choice compared to the baseline Concept 1. Concept 3 has the worst relative performance, while Concept 4 tends to be about the same as the baseline.

To enhance the Pugh approach, this thesis implements a cumulative scoring technique in which multiple baselines are selected and Pugh matrices completed for each baseline. By selecting more than one baseline concept, the scores and rankings represent a fuller description of the tradeoffs among the concepts. This is evident by considering an example case in which two concepts that both score better than the baseline have different scores when compared to one another. Carrying out analyses with multiple baselines reveals additional information that better ranks the relative attractiveness of the each of the concepts.

Completing a Pugh analysis results in [65]:

- Greater insight into the requirements,
- Greater understanding of the problem,
- Greater understanding of the potential solutions,
- An understanding of the interaction between the proposed solutions, which can give rise to additional solutions,
- A knowledge of the reasons why one concept is stronger or weaker than another.

Also, the ranking of the concepts allows the designer to prioritize which concepts are most promising. The Pugh method formalizes a concept comparison procedure that works effectively in practice. Pugh argues that it avoids the false confidence instilled by weighting matrices of other methods, and stimulates creative, unconstrained thinking as well [65].

3.3 Object-Process Network

The Object-Process Network software tool was developed by Ben Koo as a domain-neutral, executable meta-language designed to represent, generate, and manipulate simulation models [44]. What it is best used for is exploring a large space of design alternatives. In the manner of a full factorial design of experiments, OPN generates and evaluates all the possible combinations of design alternatives. User-defined mathematical functions calculate the attributes for all of the design combinations. OPN also has the ability to use the internal functions to reduce the number of alternative possibilities. This is useful for constrained problems in which some combinations of alternatives result in infeasible designs. This unique aspect of OPN makes it superior to full factorial design since only feasible alternatives are evaluated. Finally, the object-oriented, visual aspect of OPN makes it easy for the designer to understand the design space, relationships between design options, and the constraints that exist.

An example of how a model can be set up in OPN is shown in Figure 3-1, which shows an OPN model representing decisions that accompany designing a car. The decisions are represented by the boxes and the alternatives by ellipses.

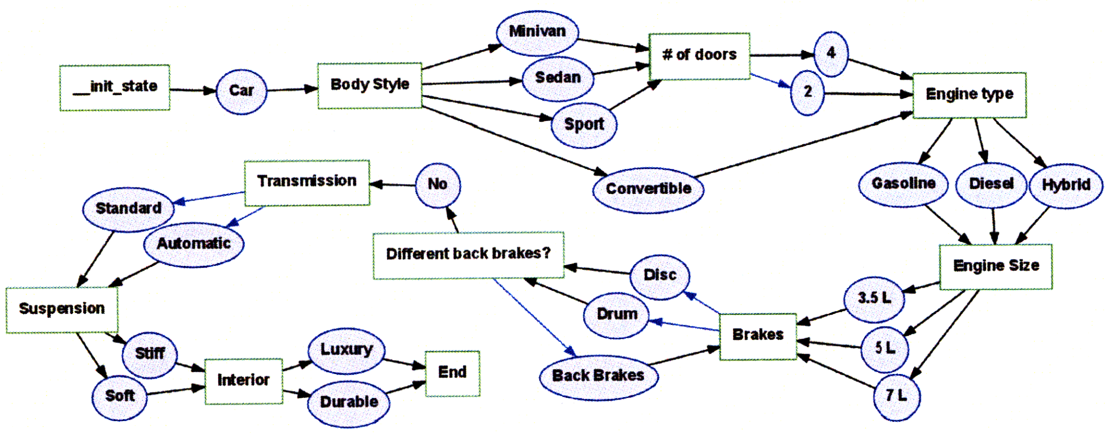

Figure 3-1: An example OPN model that enumerates and models all of the combinations of alternatives that produce feasible car designs.

The execution of an OPN model can be thought of as a set of tokens propagating through the network [77]. In the car OPN model example, an initial token starts at the upper left with initialization of the model. The token duplicates itself whenever it encounters a decision with more than one alternative. Here the first token moves to the first decision — "Body Style." At this point the token duplicates itself to create four instances of design possibilities, one each for "Minivan," "Sedan," "Sport," and "Convertible." This split represents a tradeoff in the design space. The tokens then repeat this duplication for each decision; thus the set of tokens that pass through the OPN model represents all the possible combinations of alternatives.

3.3.1 Logic and Model Relationships

An interesting feature of OPN is its ability to to handle interrelated decisions and rules of logic to prune the feasible space. Alternatives may be dependent on other decisions, and OPN provides a visual representation in which to view these relationships. In the car OPN model, the "Body Style" alternative of "Convertible" is linked to the "Engine Type" decision, bypassing the "Number of Doors" decision. This is because convertibles are modeled as having two doors. There is no decision to make at that point in the model, and the token entering the "Convertible" ellipse proceeds directly to the decision for choosing the "Engine Type."

The logic rules built into OPN models are contained in the arrows connecting decisions and alternatives. When these Boolean expressions are codified, the arrow changes color to blue instead of the default black. An implementation of this is exemplified by the arrow pointing from the decision for "Number of Doors" to the alternative of "2." In this case the alternative selection of "Minivan" requires four doors, so the token proceeding from "Minivan" cannot propagate through the choice of "2" doors, and the space is trimmed of infeasible combinations.

After enumerating the possibilities for "Engine Type" and "Engine Size," the tokens reach the "Brakes" decision in the model. A loop in the OPN models at this decision demonstrates another unique feature. Here the decision for type of brakes is made as either "Disc" or "Drum." After this decision there is the option to choose back brakes that are different from the front. The token will proceed through the arrow pointing to "Back Brakes" and then choose the set of brakes different from its initial decision. The logical rules contained in the arrows allow these choices to be made.

3.3.2 Resulting Design Space Combinations

A particular path through the network of decisions and alternatives yields a design concept. An example of one such car design resulting from the OPN model is shown in Figure 3-2. This set of alternatives represents a sedan which has a 3.5 L gasoline engine, standard transmission, and a luxurious interior. A different feasible combination of alternatives is shown in Figure 3-3, and represents a convertible design with different sets of front and back brakes, a stiff suspension and a luxurious interior to accompany the standard transmission. Depending on user-defined measures of "goodness," the OPN model can also be used to rate and rank each of the possible design combinations.

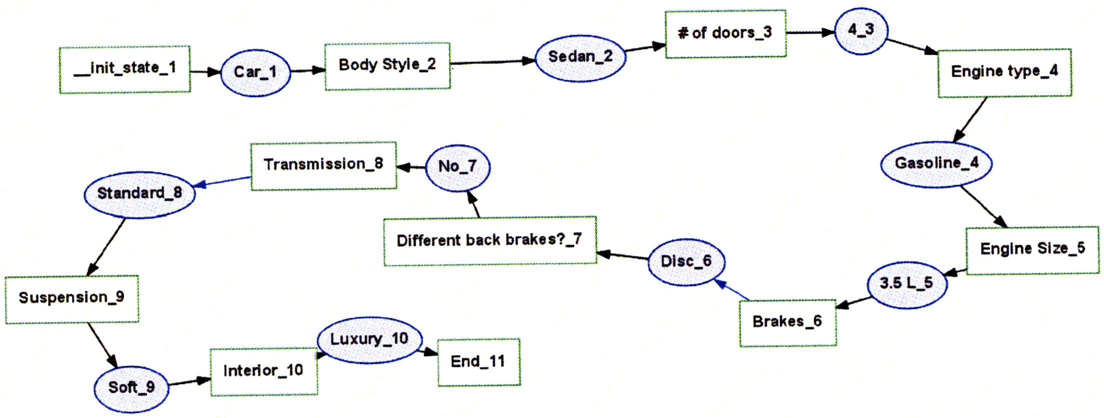

Figure 3-2: Example car design of a sedan obtained as a path through the OPN model.

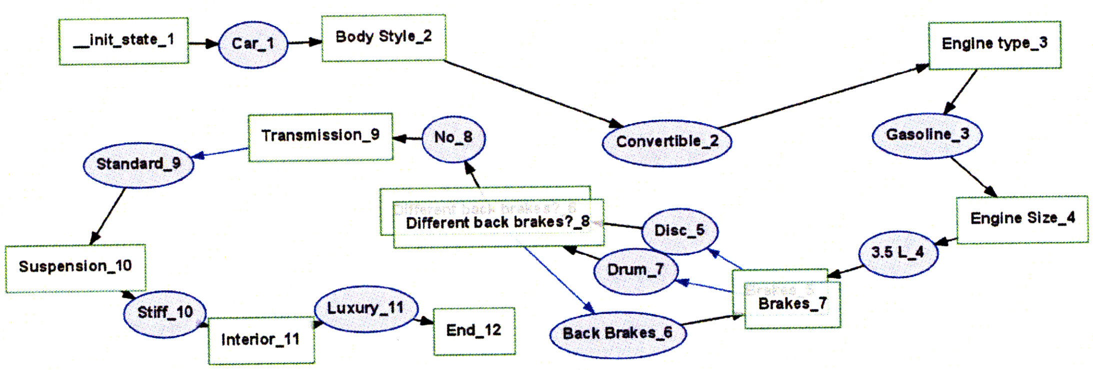

Figure 3-3: Example car design of a convertible obtained as a path through the OPN model, with the unique use of OPN features such as skipping decisions and looping to visit a decision again.

3.3.3 Advantages and Disadvantages

OPN is tool that has many unique and advantageous features, as well as some disadvantages. Traditional Design of Experiment techniques enumerate all (full factorial) or some (fractional factorial) of the combinations of alternatives. Any constraints, coupling or interrelated decisions are handled after all the combinations have been enumerated. OPN provides a means of modeling these interactions before the full enumeration and execution are complete. This allows only feasible combinations to be evaluated. The other appeal of OPN is its visual representation of the decision space. OPN gives the designer the ability to visually identify what alternatives match with others, in addition to being able to execute the model and enumerate the possibilities. In a vein similar to the usefulness of using Simulink instead of writing m-files in MATLAB®, OPN provides a user-friendly interface with which to build models for conducting trades.

One of the disadvantages of OPN is its inability to perform detailed mathematical operations. Granted many "architecture"-level decisions may be based entirely on analytical correlations, when the need for higher fidelity evaluations of concepts is required, other tools are better suited to this than OPN. The mathematical capabilities of OPN consist of closed-form analytic expressions, which may or may not be sufficient.

The biggest hindrance currently to the implementation of OPN models is the need for improved memory management. Models which generate a total number of combinations on the order of 1000 use about 1 gigabyte of RAM, depending on the particular specifications in the model. For applications in which the number of combinations in the space exceeds this, significant time and effort is required to manipulate the models to prevent memory overflow. In addition, OPN models take longer to run than with other programs. While the features of OPN such as its visual representation of interrelated decisions and rules to prune the design space are quite useful, its limitations motivate the incorporation of optimization techniques.

3.4 Optimization Techniques

Trade space exploration with OPN provides a systematic approach to exploring many trade-offs. This full view of the space is important in understanding the drivers of a system design. Instead of selecting a point design and optimizing to what could be merely a local optimum, searching a larger set of options and seeing how coupling between combinations affects the quality of the design gives additional insight to the concept design process. As the fidelity of the models used to evaluate candidate designs necessarily improves as more knowledge is gained, running a full-factorial search can become unmanageable. It is then desirable to implement a way of conducting trades that identifies the better designs more quickly, hopefully without narrowing the search space.

Optimization methods are, in general, founded on rigorous mathematical principles, where ideas such as global optimality, uniqueness, convergence, and the number of function evaluations are important [4]. When considering the incorporation of optimization in the context of concept design, however, these notions are blurred and lose their strict interpretation due to the fact that there is so much uncertainty. In designing a new car, for instance, an "optimized" design early in the process may turn out to be clearly inferior if additional information is gained on the engine performance, overall weight, or body design. Thus, the use of optimization in the early concept design stages is not intended to find the single best design, but to identify groups or families of good designs in an efficient manner.

For problems with more than one objective, the idea of Pareto optimality dictates how to identify which solutions are the best. While Chair of Political Economy at the University of Lausanne in Switzerland, Vilfredo Pareto introduced his eponymous concept: "The optimum allocation of the resources of a society is not attained so long as it is possible to make at least one individual better off in his own estimation while keeping others as well off as before in their own estimation." [63] In a multiobjective problem, a Pareto optimal solution is one in which improvement in one objective cannot be achieved without a simultaneous worsening in another objective. The solutions that are Pareto optimal are referred to as non-dominated solutions since any other solutions are worse in at least one other objective. These non-dominated solutions form what is called the Pareto front. Visual examples of Pareto fronts and non-dominated solutions are shown in Figure 3-10 and in Chapter 4.

The Pareto front comprises many "optimal" solutions, and choosing one depends on which objective(s) is (are) considered more important. When the problem is fully characterized in the problem formulation, a dominated solution (one not on the Pareto front) should not be preferred to a non-dominated one, since at least one of the objectives can be improved by selecting a different solution. It is worth noting, however, that in the presence of uncertainty, the Pareto front may not in fact accurately describe all of the "best" solutions. Changes in assumptions or conditions may cause other solutions to become constituents of the Pareto optimal set of solutions, and others to leave the set. For this reason, the optimization techniques used in this thesis focus on the history of the solutions found that cover the space and on the region around the Pareto front.

3.4.1 Problem Formulation

Though sometimes trivialized when doing optimization exercises, problem formulation is one of the most important steps in setting up an optimization algorithm. The *Problem Characterization* activity from the Engineering Framework, as discussed in Section 3.1, is used to directly formulate the problem in mathematical terms. The standard form for a problem can be expressed as shown in Equation 3.1 [4, 14],

$$\begin{aligned}
\text{minimize} \quad & \mathbf{J}(\mathbf{x}) & \\
\text{subject to} \quad & g_j(\mathbf{x}) \leq 0 & j = 1, \ldots, m_1 \\
& h_k(\mathbf{x}) = 0 & k = 1, \ldots, m_2 \\
& x_i^l \leq x_i \leq x_i^u & i = 1, \ldots, n
\end{aligned} \tag{3.1}$$

where $\mathbf{x}$ is a vector of the design variables (called the design vector), $\mathbf{J}$ is a vector of objective functions, g_j is the j^{th} inequality constraint, h_k is the k^{th} equality constraint, x_i^l is the lower bound of the i^{th} variable in the design vector, and x_i^u is the upper bound of the i^{th} variable in the design vector. The bounds on the design vector are called side constraints. The goal is to minimize the objective function(s), usually directly related to the attributes of the system, such as the performance, cost, or risk. If the desire is to maximize an objective, inserting a negative sign into the objective function easily handles this situation. There are n design variables and a total of $m = m_1 + m_2$ constraints (not including the design vector side constraints).

The design vector is a set of design choices. From the car example in Section 3.3, the design vector representing the sedan shown in Figure 3-2 would be $\mathbf{x}$ = [Sedan, Gasoline, 3.5 L, Disc, Standard, Soft, Luxury]^T. An example of a constraint may be that the car needs to get at least 20 miles per gallon fuel efficiency, which could be expressed as: $g_1(\mathbf{x}) = MPG \geq 20$. It may be useful as well to define another vector of inputs to the mathematical models, but whose values do not change from design to design. This *constants vector*, $\mathbf{k}$, consists of values that are defined to be nonvarying, or are assumed fixed for the duration of the simulation.

Many engineering and concept design problems prove difficult for optimization techniques due to two main reasons: nonconvexity and nondifferentiability [14]. A convex set is one in which any two points in the set can be connected by a line which is also completely within the set. Figures 3-4 and 3-5 illustrate convex sets and functions. Formally, the set $S \in \Re^n$ is convex [4] if for $\mathbf{x^1} \in S$ and $\mathbf{x^2} \in S$,

$$\lambda \mathbf{x^1} + (1-\lambda)\mathbf{x^2} \in S, \qquad \forall \lambda \in (0,1) \tag{3.2}$$

and a function $f(\mathbf{x})$ bounding a convex set is convex if

$$f\left(\lambda \mathbf{x^1} + (1-\lambda)\mathbf{x^2}\right) \leq \lambda f(\mathbf{x^1}) + (1-\lambda) f(\mathbf{x^2}), \qquad \forall \lambda \in (0,1) \tag{3.3}$$

Many (or most) engineering problems are naturally non-convex [14].

Commonly used optimization methods such as the Simplex method, Newton's method, and the conjugate gradient method utilize gradient information to determine direction of improvement in the problem [87, 4]. Analytic functions allow straightforward determination of derivatives, but for problems without closed-form expressions, finite difference approximations can be costly. Furthermore, discontinuity is another problem gradient-based search

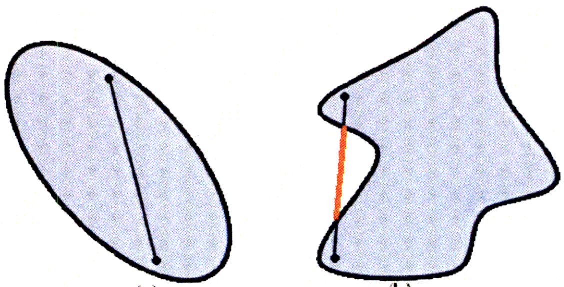

Figure 3-4: (a) Convex set: a line connecting any two points in the set is in the set. (b) Nonconvex set: the orange segment of the line connecting two points in the set is outside the set.

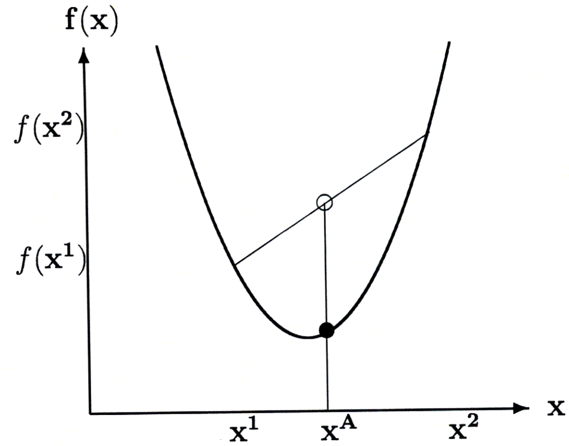

Figure 3-5: Convex function: between any two points $\mathbf{x^1}$ and $\mathbf{x^2}$, the value of the function is not greater than the point on the line connecting $f(\mathbf{x^1})$ and $f(\mathbf{x^2})$.

algorithms face [87]. This can surface in the form of discontinuous functions, or with the presence of integer variables. Techniques such as branch and bound exist for handling integer programming problems [4]. This method systematically enumerates every candidate integer solution, or stops after a specified number of branches have been explored. Problems that incorporate look-up tables pose the problem of discontinuity and may not fit an integer programming formulation.

Optimization problems with the characteristics of nonconvexity, discontinuity, and/or nondifferentiability usually fall under the class of NP-hard (nondeterministic polynomial-time hard) problems, and many engineering challenges have these traits [14]. Methods to handle these types of problems are limited, but the most promising approach is often heuristic-based optimization, of which particle swarm optimization is an example. These heuristic search algorithms generally rely on stochastic parameters that perturb a guided search in order to avoid local extrema and span nonconvex and discontinuous spaces. The most popular heuristic search technique to date has been Genetic Algorithms (GAs), which use a population of solutions that is refined based on ideas borrowed from biological evolution to search through the space for better solutions [23].

To quickly arrive at "good" solutions in concept design, the formal components of optimization may not be necessary, and heuristic searches are viable alternatives. By definition heuristic search algorithms are not guaranteed to find the optimal solution(s). Instead, they are methods to guide the search for better solutions, and can be used to handle the hard problems that other optimization methods cannot.

3.5 Particle Swarm Optimization

Particle swarm optimization is a heuristic search algorithm originally based on modeling the social interaction of a flock of birds in flight [42, 15]. Similar to how a population of birds will hone in on a search (for food, for example), the PSO algorithm consists of a set of particles that "fly" through the design space searching for the best solution. By incorporating the exchange of information between particles, the population as a whole can find better regions in which to look and avoid getting trapped in local extrema.

3.5.1 Single Objective Formulation

Each particle is equivalent to a design vector, $\mathbf{x}$, and the design solution that it describes is known in PSO parlance as a "position." How the particles search through the space is determined by each particle's "velocity," which is influenced by three factors. The position of each particle is then updated based on this movement. For a single objective problem, this movement is described mathematically in Equation 3.4 [33, 89].

$$\mathbf{v}_i^{k+1} = \underbrace{w^k \mathbf{v}_i^k}_{\text{inertial component}} + \underbrace{c_1 r_1^k \frac{\mathbf{p}_i^k - \mathbf{x}_i^k}{\Delta t}}_{\text{cognitive component}} + \underbrace{c_2 r_2^k \frac{\mathbf{p}_g^k - \mathbf{x}_i^k}{\Delta t}}_{\text{social component}} \tag{3.4}$$

The new velocity of particle i at step $k+1$ is a sum of its current velocity, $\mathbf{v}_i^k$, multiplied by an inertia weighting coefficient w^k; its relative position compared to the best position visited by particle i; and its relative position compared to the best position in the entire swarm. These three components are referred to as the inertial component, cognitive component, and social component, respectively. The variables are defined in Table 3.3. An illustration of how these three components contribute to the velocity update for each particle is shown in Figure 3-6 [29].

Table 3.3: PSO variables

$\mathbf{v}_i^k$:	velocity of particle i at step k	w^k:	inertia weighting coefficient at step k
$\mathbf{x}_i^k$:	position of particle i at step k	c_1:	cognitive constant
$\mathbf{p}_i^k$:	best position visited by particle i up to step k	c_2:	social constant
$\mathbf{p}_g^k$:	best (global) position from all particles in population up to step k	r_1^k, r_2^k:	random values at step k selected from uniform distribution between 0 and 1
		Δt	$\equiv 1$

The Inertial Component, first introduced by Shi and Eberhart [73], acts to move the particle in the same direction as the last step. The suggested value of the weighting coefficient, w^k, has varying interpretations in the literature. Shi and Eberhart [73] originally tested a constant weighting coefficient between $w = 0$ and $w = 1.4$, and Shi and Eberhart [75] suggest using $0.8 < w < 1.4$. Shi and Eberhart [75] show improved convergence using an adaptive inertia weight. They suggest that a decreasing w^k provides large initial velocities for a global search of the space to start followed by localized search. Others, such as Zheng [94], make a case for the opposite: an initially randomly distributed swarm can adequately cover the space with a low w^k, but as the particles converge, there is a greater need to increase the inertia to prevent premature convergence in a local minimum. Their tests indicate that an increasing inertia weight outperforms the decreasing inertia weight in convergent speed. The work in this thesis has found a constant weighting coefficient to be sufficient.

The Cognitive Component refers to the memory of each individual particle. Like an individual bird remembering where it found its last meal, this component of the velocity is a measure of how far away the current position is from the best position the particle has found up to the current step, k. The coefficient is comprised of a constant, c_1, and a random variable, r_1, uniformly distributed between 0 and 1. Kennedy and Eberhart [42] originally proposed using $c_1 = c_2 = 2$, so that the mean of the stochastic coefficient, $r_1 c_1$, is equal to 1. Venter and Sobieszczanski-Sobieski [89] proposed giving more trust to the social parameter, using $c_1 = 1.5$ and $c_2 = 2.5$. The physical interpretation of these parameters is explained with the aid of Figure 3-7 [25]. If $c_1 = 2$, the maximum value of the stochastic coefficient, r_1c_1, is 2, the mean is 1, and the minimum is 0. This sets the range over which the cognitive component of the velocity vector can vary. The mean value places the position at $\mathbf{p}_i^k$, the best position in the particle's memory.

The Social Component of the velocity is the contribution of the current globally optimal position — the best position found by all the particles in all the previous iteration steps. This is analogous to the birds sharing their food discovery with all the others so they can move toward that location. It varies with each time step, k, as the swarm flies around, but ideally converges to the global optimum. The same reasoning behind the coefficient c_2r_2 holds for the social component as for the cognitive component, and is also shown in Figure 3-7.

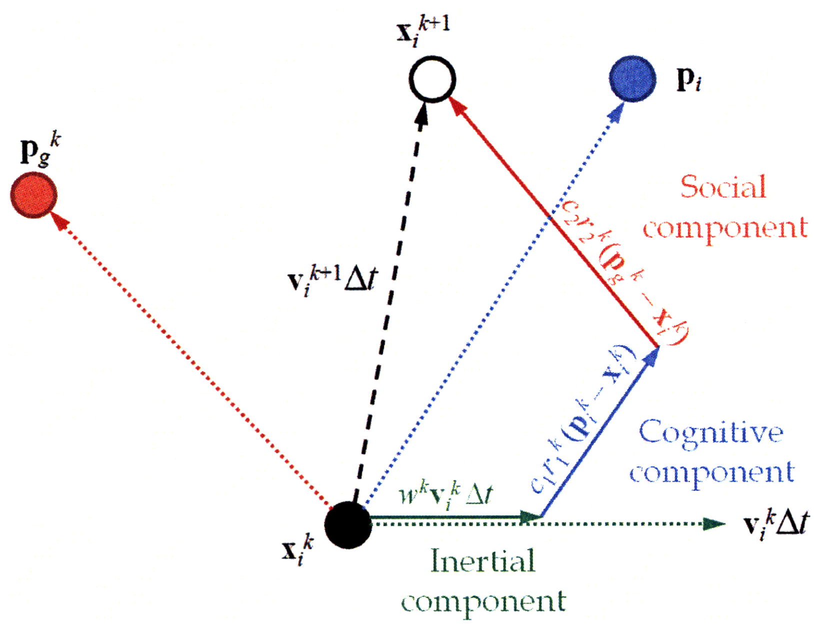

Figure 3-6: Illustration of the velocity components in the PSO algorithm (Eq. 3.4), showing the inertial, cognitive and social influences on the particle's movement through the design space [29].

Because the velocities and positions are vectors, the updates in movement are done for each design variable in the design vector. Thus the diagrams in Figures 3-6 and 3-7 would need to be extended to n dimensions.

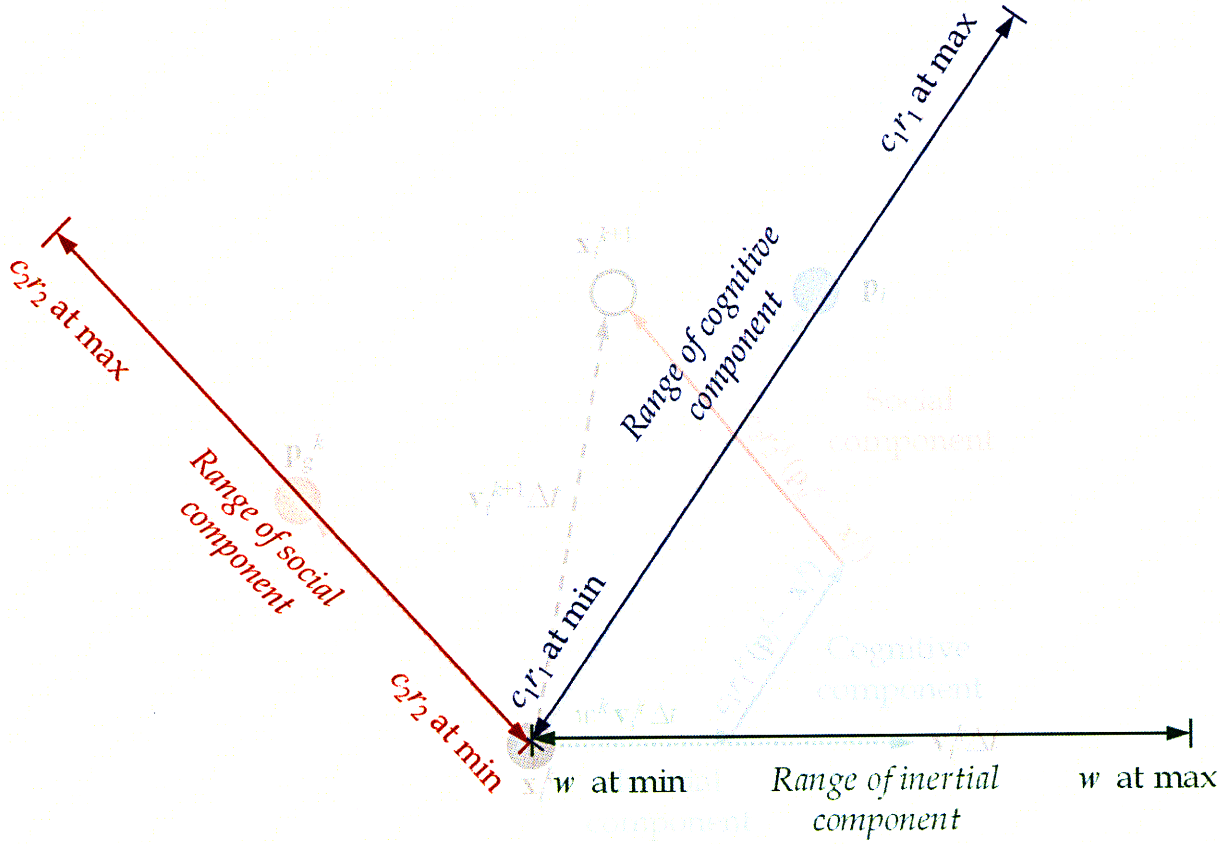

Figure 3-7: Illustration of how the coefficients of the velocity terms in Eq. 3.4 affect the magnitude of each velocity component.

Constrained optimization problems are more difficult than unconstrained ones. Several approaches to side constraint violation exist. For handling the position of the particles in the design space, there are several approaches. When a side constraint on the design variable is violated, the particle can be reset to its last feasible position [17]. Alternately, only the feasible solutions are retained in the population [34], or a penalty function can be applied that degrades the objective value of the particle so it does not gain favor in the swarm [64]. The approach found to be most effective in this thesis is resetting the particle's position to its last feasible position.

The other issue for particles that violate the side constraints is their velocities. The first method is to set the velocity equal to zero for the design variable that violated the constraint. This has the effect of stopping the motion which sent the particle toward the infeasible region, and the next step would return the particle in the direction of the global best and/or the particle's best position. The other approach is to set the velocity equal to its negative upon side constraint violation. This effect would turn the particle around 180 degrees, avoiding further exploration of the infeasible region. Both approaches are used in the case studies of Chapter 4, where either the "rest" or "bounce" method is chosen with an equal probability [25].

What is referred to as "velocity explosion" is a drawback of the algorithm that is related to how constrained the problem is [14]. This is exhibited by velocities that grow increasingly large with respect to the expected design space range; divergence characteristics prevent the particles from finding good solutions. A quadratic exterior penalty function on the objective function is implemented by Venter and Sobieszczanski-Sobieski [88] to reduce this effect. The work in this thesis found that resetting the positions of particles that violated the constraints to the boundaries of the design space was sufficient in preventing velocity explosion.

Whereas the genetic algorithm is inherently a discrete algorithm, PSO is inherently continuous. Handling discrete problems is easily and successfully done by rounding those design variables whose values must take integer quantities. Venter and Sobieszczanski-Sobieski [89] show that the integer version of PSO runs faster than the continuous one. The variables in this thesis include both continuous and discrete values. Therefore, only the discrete variables, which represent choices among technology options, are rounded.

3.5.2 Multiobjective Formulation

The particle swarm algorithm has inherent advantages for problems with multiple objectives. These derive from its property of having a population of particles. An example of a multiobjective problem is one in which a design must maximize performance while simultaneously minimizing cost or risk. The goal in multiobjective optimization problems is to identify the Pareto front. A common approach to multiobjective problems is to form a (single) composite or aggregate objective function, which is generally a convex combination of the objective functions. That is, each objective is given a relative importance, or weight, and the composite objective function is optimized. By sweeping through different relative importance values, the Pareto front can be approximated. This technique however, suffers from the inability to handle non-convex problems, as described in Section 3.5.3.

There are different approaches for implementing a multiobjective PSO. Most consist of either handling separate objectives independently or by using the non-dominated solutions to alter what particles are considered the cognitive and social bests.

One approach, first introduced by Coello Coello and Lechuga [7] is to establish a repository of non-dominated solutions with the design space separated into hypercubes. For each simulation step, the global optimum, $\mathbf{p_g^k}$, is chosen from this repository using tournament selection, which is a random selection criterion. As described further by Grant and Mendeck [25], $\mathbf{p_g^k}$ is determined based on a particle's crowdedness distance, a measure of proximity to other particles in the objective space. This directs the exploration toward sparsely populated regions of the Pareto front.

Another approach, which has not been utilized in the literature, to the best knowledge of the author, is to divide the swarm population evenly into subswarms. The performance of the particles in each subswarm is measured according to separate objectives. For instance, one subswarm may attempt to minimize the mass of the system while the objective of another is to maximize stiffness. If left separate they would both carry out independent problems. By manipulating the social components of the particles, they receive competing information that tends to move them in a direction toward the Pareto front of the multiple objectives. The social exchange is accomplished by substituting the $\mathbf{p_g^k}$ of each subswarm i (or objective i) with that of subswarm/objective $(i + 1)$ — where the final subswarm uses the $\mathbf{p_g^k}$ of the first subswarm.

In tests conducted in this thesis, the multiobjective algorithm using the repository suffers from increased complexity and computational expense as compared to the subswarm formulation. For the ease of formulation and speed of execution, the subswarm algorithm is implemented with excellent results.

3.5.3 Algorithm Test Cases

To verify the effectiveness of the particle swarm algorithm, two test cases are conducted: a single-objective problem and a multiobjective problem. Both test functions have multiple local extrema and thus are non-convex, a traditionally difficult characteristic to overcome. The single-objective formulation is:

$$\begin{aligned} \text{maximize} \quad & J(x_1, x_2) = \\ & 3(1-x_1)^2 e^{-x_1^2-(x_2+1)^2} - 10\left(x_1/5 - x_1^3 - x_2^5\right) e^{-x_1^2-x_2^2} - \tfrac{1}{3} e^{-(x_1+1)^2-x_2^2} \\ \text{subject to} \quad & -3 \le x_1 \le 3 \\ & -3 \le x_2 \le 3 \end{aligned} \tag{3.5}$$

where J is the objective function that is to be maximized, and x_1 and x_2 are the design variables with side constraints limiting the range of each to the interval [-3, 3]. This function is also the peaks function in MATLAB®.

Figures 3-8 and 3-9 show contour plots and surface plots, respectively, of the objective function. The blue points are the particles of the PSO algorithm. The sequence of plots represent the procession of the algorithm through time steps as the population searches the space for the best values of the objective function. Examination of the particles through the steps reveals how the swarm adjusts its motion based on its most current information. In Time Step 1, the population is randomly distributed throughout the design space. By Time Step 7, the swarm is concentrating on the local maximum around $(x_1, x_2) = (1.5, 0)$. In this time step, however, there is one particle that has migrated to the region of $(x_1, x_2) = (-0.5, -0.5)$ where another local maximum exists, except this one is better than $(x_1, x_2) = (1.5, 0)$. By Time Step 10, the majority of the population has moved in the vicinity of this better solution. The population again finds and migrates to the local (and global) maximum close to $(x_1, x_2) = (0, 2)$. Time Step 13 shows slight overshooting of a number of the particles due to their inertia, but they all begin to converge on this globally optimal solution through the remaining time steps.

The second test case is a multiobjective problem. Instead of the objective, J, being a single function, it is a vector of two functions for which there is not a single optimal solution.

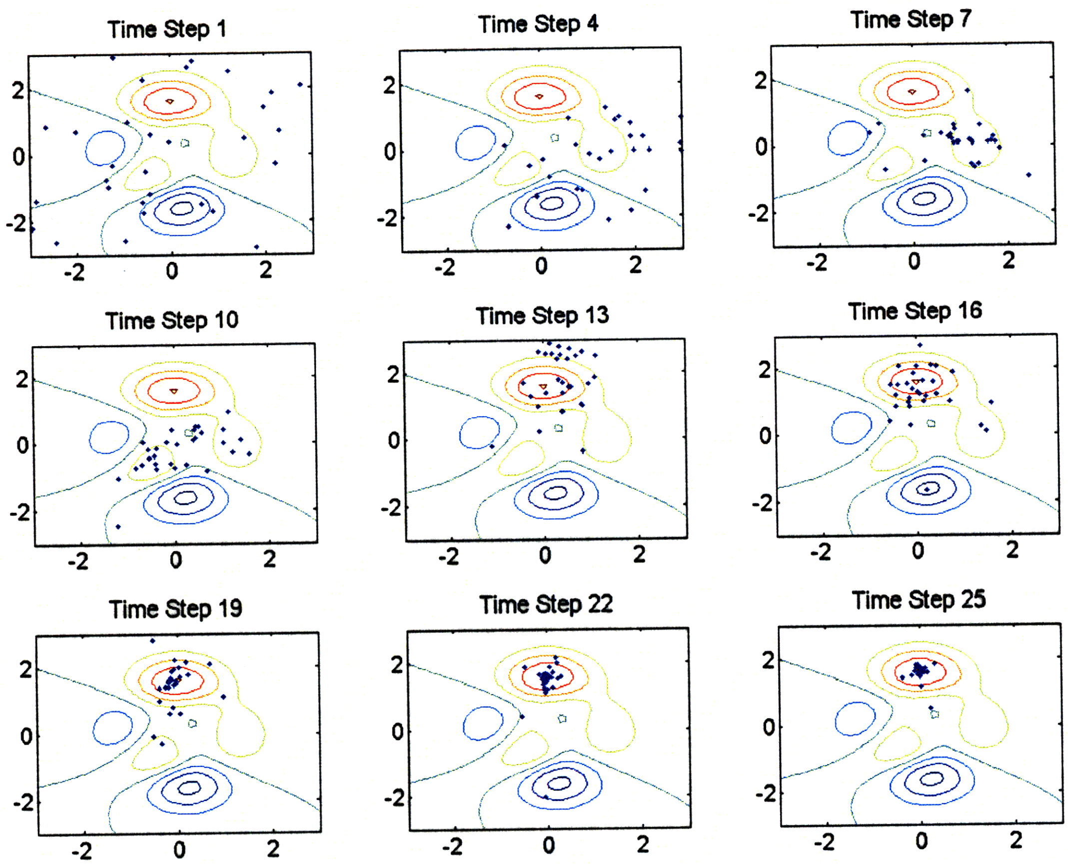

Figure 3-8: Contour plots of the single objective test case for the PSO algorithm

In fact, for most regions of the design space, the values of the objective functions conflict with one another. That is, for many values of (x_1, x_2), the value of J_1 is high while the value of J_2 is low. The problem formulation is:

$$\text{minimize} \quad \mathbf{J}(\mathbf{x}) = \begin{bmatrix} J_1(x_1, x_2) \\ J_2(x_1, x_2) \end{bmatrix} =$$
$$\begin{bmatrix} 3(1-x_1)^2 e^{-x_1^2-(x_2+1)^2} - 10(x_1/5 - x_1^3 - x_2^5)e^{-x_1^2-x_2^2} \\ \quad - 3e^{-(x_1+2)^2-x_2^2} + \frac{1}{2}(2x_1 + x_2) \\ 3(2-x_2)^2 e^{-x_2^2-(x_1+1)^2} - 5(-x_2/5 - x_2^4 - x_1^5)e^{-x_2^2-x_1^2} \\ \quad - 10e^{-(1-x_2)^2-x_1^2} \end{bmatrix} \tag{3.6}$$
$$\text{subject to} \quad -3 \leq \mathbf{x} \leq 3$$

where $\mathbf{J} = [J_1, J_2]^T$ is a vector of the first and second objective functions (J_1 and J_2, respec-

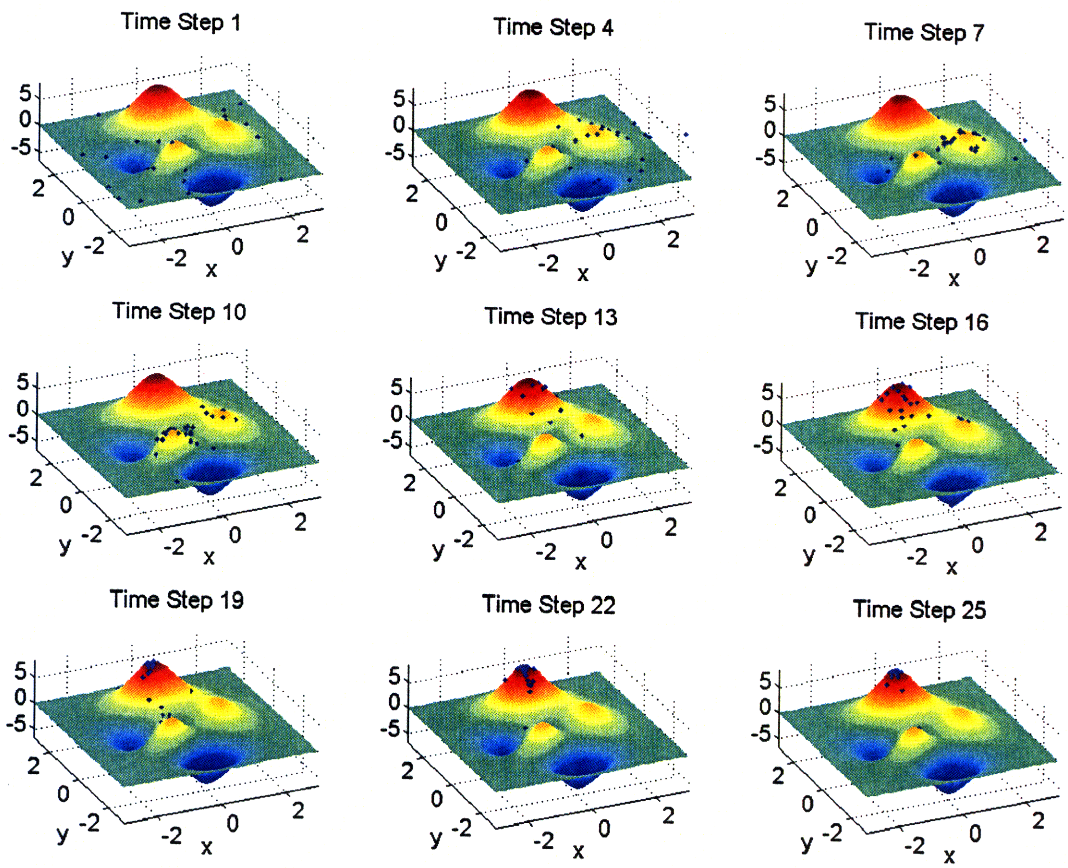

Figure 3-9: Surface plots of the single objective test case for the PSO algorithm

tively), both of which are minimized, and $\mathbf{x} = [x_1, x_2]^T$ is a vector of the design variables, x_1 and x_2.

Figure 3-10 shows the design and objective spaces. The top two plots show the contours of J_1 and J_2 as functions of x_1 and x_2, along with the whole history (all time steps) of the PSO particles. The bottom plot is the objective or trade space, where J_1 is plotted against J_2. To visualize how the true design space maps to the objective space, a grid of points along the x_1 and x_2 directions at equally spaced intervals of 0.05 is evaluated for J_1 and J_2 and plotted as the black dots in the objective space. It is clear that this multiobjective space is non-convex. This non-convexity proves challenging for many optimization techniques. For instance, the green squares represent the optimized solution to a series of convex combinations of the two objectives. They are found by creating a single objective function by using a series of weighted summations of the two objective functions. For increments of 0.05 for λ between 0 and 1, the green solutions are minimizations of:

$$J_\lambda = \lambda J_1 + (1 - \lambda) J_2 \tag{3.7}$$

These solutions of J_λ are valid, but it is evident that the use of the convex combination of J_1 and J_2 to obtain the single J_λ misses a lot of interesting, non-dominated solutions. In fact this approach cannot find any of the non-convex solutions. Plus there is a clear lack of uniformity of solutions across the space in this problem, evidenced by the large jump between solutions at the bottom right and top left portions of the objective space.

The blue points represent the history of the population of PSO particles, both in the design spaces and the objective space. There are three characteristics of interest: 1) the initial random distribution helps establish a wide, global search of the whole space, 2) the particles then tend to search the regions that perform well in both objectives, and 3) the efficiency with which the PSO algorithm finds non-dominated solutions is evident by the number of PSO function evaluations compared to evaluation of the full space with a grid. The pink stars represent the non-dominated solutions of the PSO set. The yellow points (some of which are below other points) are the non-dominated solutions of the full grid of the design space (black points). The cyan stars are the points in the PSO population history that are non-dominated with respect to the true, full grid space of points. For the full grid, 14,641 evaluations of each of the functions are required (one for each point in the design space) in order to find the 126 non-dominated solutions over the full grid space. Compare that with the particle swarm algorithm, which identifies 102 non-dominated solutions with respect to the full grid, needing just 1500 function calls (50 time steps for a population of 30 particles). This is an order of magnitude improvement in the number of function evaluations to find 81% as many non-dominated solutions.

3.5.4 Advantages and Disadvantages

As described previously, heuristic search algorithms are very appropriate for finding "good" solutions and identifying promising regions of trade spaces, which is the focus in conceptual

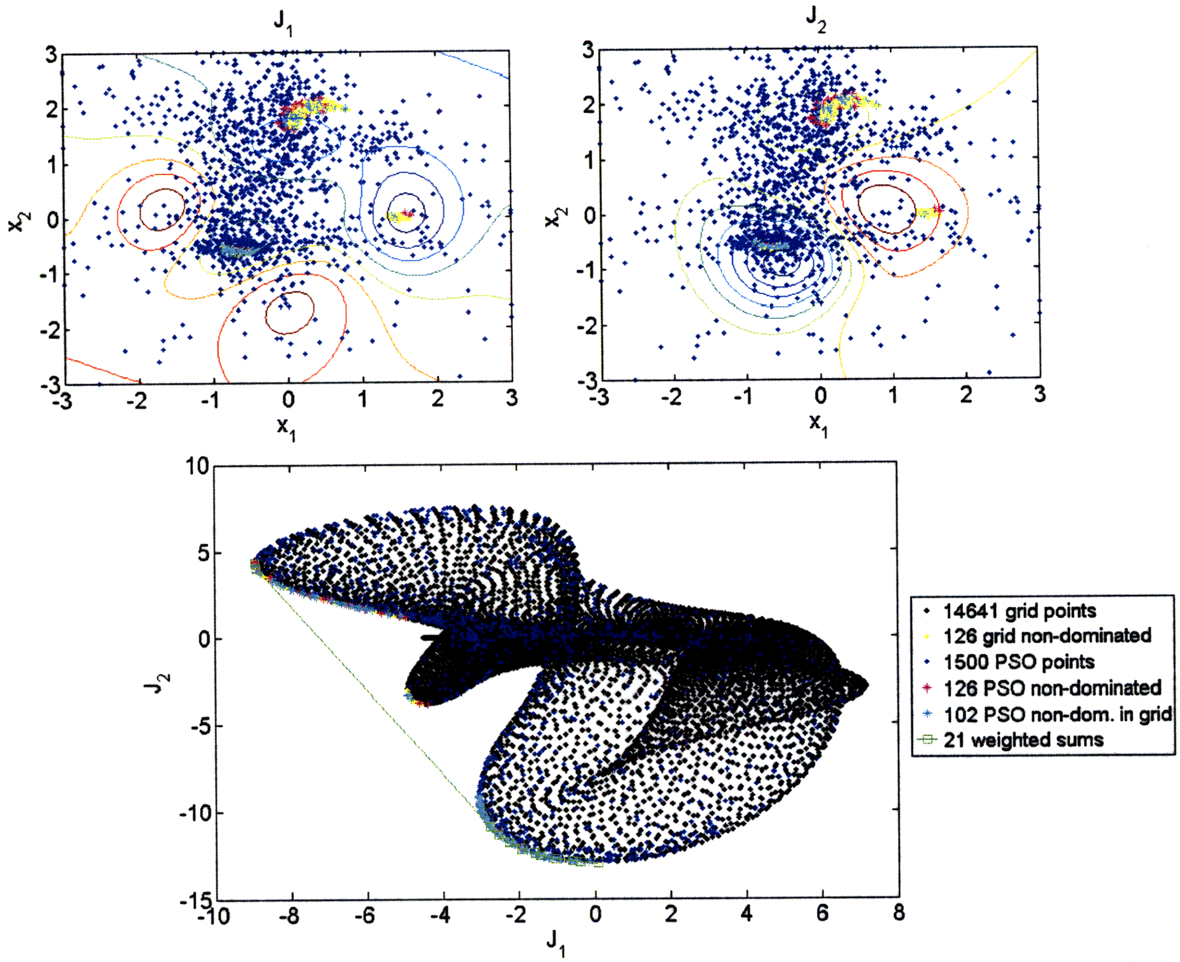

Figure 3-10: Multiobjective test case for the PSO algorithm

design. Specifically, PSO has been shown to outperform the popular Genetic Algorithm [29] in the number of required function evaluations, and still finds the optimal solution with the same frequency. In addition, GAs have many more parameters which the designer must control, sometimes a burdensome task as much as a guessing game. Granted PSO has been implemented in "hybrid" algorithms that incorporate other methods, thereby increasing its complexity, the essence of the algorithm is simple conceptually and there are few parameters to control. PSO is also ideally suited for parallel computing applications [89]. Since the evaluation for each particle is done independently, multiple processors can be used to reduce the run-time.

The disadvantages of PSO also derive mostly from its status as an heuristic algorithm.

It is not guaranteed to find the optimal solution(s), nor converge in any strict interpretation of the term. The literature reports excellent repeatability in finding global optima for benchmark functions [74], but its performance in problems whose solutions are unknown *a priori* is unverifiable. Highly constrained problems are a challenge for many optimization algorithms is, including particle swarm optimization. However, constrained optimization is a common application area for PSO [72].

PSO is able to effectively search for optimal (or close to optimal) solutions — which correspond to the most promising designs in engineering a new system — for problems that are traditionally hard to solve. It is has much merit in concept design, as will be demonstrated in Chapter 4.

3.6 Sensitivity Analysis

The goal of a sensitivity analysis is to identify which aspects of a system are most susceptible to variation in the inputs, and to measure the degree to which each aspect is influenced by each source of variation.

3.6.1 Analysis of Variance

Analysis of variance (ANOVA) is a statistical method that decomposes the contributions in the output to variation among the inputs. In combination with an organized Design of Experiments, ANOVA indicates which are the most important factors affecting the performance characteristics [90]. In examining concept design trade spaces, ANOVA techniques are used to identify how observed variations in the attributes correspond to choices of design variables. The method works by apportioning the total variation in the outputs to individual design variables. It is based on summing the squares of the deviation from the mean for each design variable.

Recall that a design vector, $\mathbf{x}$, is composed of design variables such that $\mathbf{x} = [x_1, x_2, x_3 \ldots x_n]^T$. Each x_i represents a design choice, such as a 3 L engine size (e.g., $x_1 = 3$). The first calculation in using ANOVA is the total sum of squares, S_T, which gives the sum of the squares of the differences between each observation y_i (an attribute of one particular design) and the mean of the entire set of observations (total number of designs evaluated):

$$S_T = \sum_{i=1}^{T} (y_i - \mathrm{E}(y))^2 = \sum_{i=1}^{T} y_i^2 - \frac{1}{T}\left(\sum_{i=1}^{T} y_i\right)^2 \tag{3.8}$$

where $\mathrm{E}(y) = \frac{1}{T}\sum_{i=1}^{T} y_i$ is the mean of all observations and T is the total number of observations.

Next the variation from the overall mean for each setting of the design variables is calculated. For each design variable, its sum of squares, S_{x_i}, is given by:

$$S_{x_i} = \sum_{j=1}^{p} n(x_i|x_i = x_j)\Big[\mathrm{E}(y|x_i = x_j) - \mathrm{E}(y)\Big]^2 \tag{3.9}$$

where p is the total number of possible settings for design variable x_i, $n(x_i|x_i = x_j)$ is the number of observations for which the design variable x_i is at setting x_j, and $\mathrm{E}(y|x_i = x_j)$ is the mean of the observations at which the design variable x_i is at setting x_j. This is calculated for each design variable in the design vector $(i = 1, \ldots, n)$.

Given the variation for each design variable, the ratio of the sums of squares tells how each design variable influences the overall variation. The relative influence, RI_{x_i}, gives how much variation is attributable to each design variable:

$$RI_{x_i} = \frac{S_{x_i}}{S_T} \tag{3.10}$$

For multiple attributes, this procedure would be repeated for other sets of observations, such as for z_i, $i = 1, \ldots, T$.

3.6.2 Uncertainty Characterization

Identifying important design variables with ANOVA is critical if those design variables are also sensitive to uncertainty, such that design decisions have the possibility of being greatly affected by future changes. There are a number of ways that uncertainty comes into play in system design. Drawn largely from Ref. [30], Table 3.4 lists a number of sources of uncertainties, possible mitigation strategies, and desired outcomes for the system when uncertainties are reduced early in the design and development.

Table 3.4: Uncertainty sources, mitigation strategies, and desired outcomes

Sources of uncertainty	Statistical variation Systematic error and subjective judgment Linguistic imprecision Variability Randomness Disagreement in interpretation Approximations
Mitigation strategies	Margins Redundancy Verification and Testing Generality (multi-functionality, commonality) Serviceability/Upgradeability Modularity Trade Space Exploration Portfolio and Real Options
Desired outcomes	Reliability (probability system will work) Robustness (ability to perform under variety of circumstances) Versatility (ability to do unintended functions) Flexibility (ability to be modified to do unintended functions) Evolvability (ability to serve as basis to meet new needs) Interoperability

3.6.3 Sensitivity Analysis Techniques

To better understand the sensitivities of a design to possible uncertainties, the results from the analysis of variance can be combined with a sensitivity sweep in which the important design parameters are varied and the resulting change in output is examined. The ANOVA tells the designer which design elements contribute most to the variation for a particular attribute. A sensitivity analysis on this design variable gives the expected variation in performance for a range of inputs, based on the extent of the uncertainty for that design variable.

There are two main approaches to conducting the sensitivity study, both closely related to methods used for trade space exploration. The first is to use a Monte Carlo simulation to select values from a given probability distribution. This would be most appropriate for a parameter whose variation is well characterized by a given distribution. However, it often requires many simulation runs to obtain adequately distributed and statistically significant results. The other alternative is to utilize a Design of Experiments approach. With this method, extreme cases are chosen to represent worst-case scenarios; for example, a 3-σ deviation from the nominal mean could be the value used for the sensitivity analysis. This would require far fewer runs than the Monte Carlo, but may not be as representative of real world cases [26]. In this thesis, the two approaches are used for the two different case studies. The Air-Launched Sounding Rocket problem, described in Section 4.1, has a short simulation time, which makes a Monte Carlo run reasonable. The disaster monitoring problem, described in Section 4.2 has a much longer simulation time, thus motivating the use of the DoE method. In general, many concept design problems will not have verifiable probability distributions on various parameters, and with the exception of assuming a uniform or normal (or other) distribution, the Design of Experiments approach should be sufficient [86].

Chapter 4

Case studies

"However beautiful the strategy, you should occasionally look at the results."

— Winston Churchill

"A lot of times, people don't know what they want until you show it to them."

— Steve Jobs

Two cases are presented to demonstrate the value of using the Engineering Framework for concept design problems. In particular, Pugh analysis, Object-Process Network, particle swarm optimization, and analysis of variance are used together to effectively and systematically approach these concept design examples. The greatest value in using the Engineering Framework and the design tools is the ability of the designer to find a concept that is superior to a large set of alternatives. Accompanying the final concept design selection is the aggregation of insight and better understanding of the problem and the design space. The interplay between the designer's reasoning abilities and the computational advantages of the concept design tools is a key aspect of the Engineering Framework. The tools do not create the design, and they should not choose the design either. Rather, intelligent decision-making through the design process is aided by the structure of the Engineering Framework and the computational resources of the design tools.

The Engineering Framework is applicable to problems of varying type and scope. Both of the case studies presented are conceptual designs, but they have distinct differences that demonstrate the generality of the approach outlined in this thesis to solving other concept design problems. The first case study is the design of the propulsion system for an Air-Launched Sounding Rocket (ALSR). The emphasis in this example is on the *Selection* and *Refinement* phase classes of the Engineering Framework. The design tools linked together are OPN, PSO, and ANOVA. These enable exploration of the trade space and knowledge about the tradeoffs among different design choices. The ALSR propulsion system design involves choices in geometry, propellant options, and staging choices, and is generally a tightly defined problem.

The second problem considered is the design of a multi-vehicle system for providing surveillance in a disaster response scenario. The proposed problem is to design a system that provides high-quality, rapid coverage of an area affected by a recent disaster. This open-ended problem statement is given in such a way as to not bias a particular solution. Thus there is a large amount of design freedom. At the same time there is little knowledge on the problem and how to solve it. Therefore, a significant amount of work in the *Exploration* phase class is needed, utilizing Pugh analysis for selecting among high-level concept alternatives. The *Selection* phase class analysis incorporates preliminary modeling of a system design using OPN. Then refined models are evaluated with the use of PSO. Finally, the most important aspects of the design are identified using ANOVA, and sensitivity studies reveal how to select a design that is robust to uncertainty.

4.1 Air-Launched Sounding Rocket

The first application problem to demonstrate the Engineering Framework and concept design tools is an Air-Launched Sounding Rocket (ALSR). Focusing on the elements of the propulsion system, this design study evaluates at a total of 13,488 designs to identify which are the most promising and what are the main design drivers and their effects.

Also referred to as research rockets, sounding rockets carry scientific instruments to almost any altitude, ranging from a few kilometers to a few thousand kilometers. They are especially useful for collecting experimental data in regions of the atmosphere above the heights balloons can reach (about 40 km) and below the orbits of satellites (about 160 km). Other advantages of sounding rockets include their simplicity, low costs, short lead times, and flexibility in launch locations and launch dates [10].

Research groups in the 1950s started firing sounding rockets from airborne platforms. Many "rockoons" — named because they consist of a rocket lifted by a balloon to launch above the thickest part of the atmosphere — were fired between 1952 and 1960. James Van Allen led one of the first teams to put rockoons to practical use when his group lofted an 11-kg payload to over 80 km. With the goal of detecting radiation at high altitudes, Van Allen's rockoon experiments ultimately helped him better understand and explain the characteristics of Earth's magnetic field. With the successes of rockets launched from balloons, the Air Force and Navy started looking into using high-altitude aircraft. This concept took on the name "rockaire." In 1955, the Navy successfully tested launching a rocket from an F2H2 aircraft. The Air Force conducted a test in 1956. No important scientific experiments were done using rockaires, however, and the rockaire concept never garnered as much attention as the rockoon did during the 1950s [10].

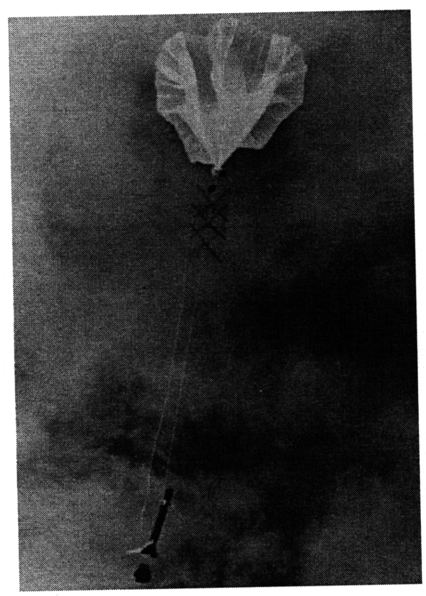

Figure 4-1: A Navy rockoon just after a shipboard launch [10].

These previous research endeavors with sounding rockets launched from airborne platforms all come from the nascent days of the space age. Since that time, improvements have been made in rocket technology, as well as great advances in the computational capabilities of computers. There is interest, then, in investigating the benefits of air-launched sounding rockets using today's technologies and computational resources. The automated features of computer-based design can shed additional light on the possibilities for a new ALSR design.

4.1.1 ALSR Selection

Because the Engineering Framework for Concept Development is designed to be a guide for solving various concept design problems, each problem will naturally adopt certain features to customize an appropriate design strategy. Examination of the information on the problem determines the most suitable starting point in the Engineering Framework. The *Exploration* phase class deals with problems that are ill-defined and includes qualitative notions of concept alternatives. The ALSR problem definition is beyond this level. For example, if the need was to obtain data on the upper atmosphere, this problem statement would require starting in the *Exploration* phase class. Since the design class is given as an air-launched rocket, this leads to the placement of the ALSR problem in the *Selection* phase class.

ALSR Selection: Problem Characterization

Despite the fact that different problems may start in different phase classes, each problem starts with the *Problem Characterization* activity. Even if the criteria of the problem are already defined, it is crucial to verify these are correct by reviewing what are the important characteristics of the problem. In the *Problem Characterization* activity for the ALSR problem, the important attributes are identified as altitude at apogee and per unit cost. Certainly other attributes of the system can be considered (such as risk, reliability, or manufacturability), but these two are sufficient in differentiating the concepts in this study. Altitude at apogee, or the highest point the rocket reaches in its flight, is an indication of the performance of the vehicle and determines which types of scientific experiments can be flown. Cost is a common metric to track, and here is modeled as a per unit cost based on mass. In addition to the identification of the attributes, constraints are also defined in the *Problem Characterization* activity. In the ALSR case, the primary design constraint is that the rocket is launched from the air, rather than the ground.

ALSR Selection: Alternative Generation

The second activity consists of generating design options for the rocket. Through the study of existing concepts as well as iterative brainstorming sessions, a set of design variables and alternatives are selected, as shown in Table 4.1.

Table 4.1: ALSR *Selection* Design Variables and Alternatives

Design Variable		Design Alternatives		
Release velocity, m/s		0	20	40
Body length, ft		15	17	
Diameter, in		10	14	
Propellant	liquid:	$RP\text{-}1/O_2$	HAN	MMH/N_2O_4
	solid:	ORBUS 6E	Castor IVA	PBAN
Number of stages		1	2	
Length distribution of stages		1:1	3:2	2:1

The release velocity represents what different airborne platforms could potentially provide. The propellant options are taken from Ref. [35]. The length distribution of the two stages refers to the first being the same length as the second, 1.5 times as long, or twice as long.

ALSR Selection: Model Development and Evaluation

The set of alternatives is then used in the construction of an OPN model. The OPN model includes the series of design decisions between the alternatives, as well as the mathematical formulation and logic necessary to evaluate the various combinations of alternatives. The OPN model for the Air-Launched Sounding Rocket is shown in Figure 4-2.

The ALSR OPN model begins with initialization of the model. Then OPN enumerates all of the combinations for the first three decisions (Release Velocity, Body Length, and Diameter). At the Propellant decision, there is some logic to determine for which of the

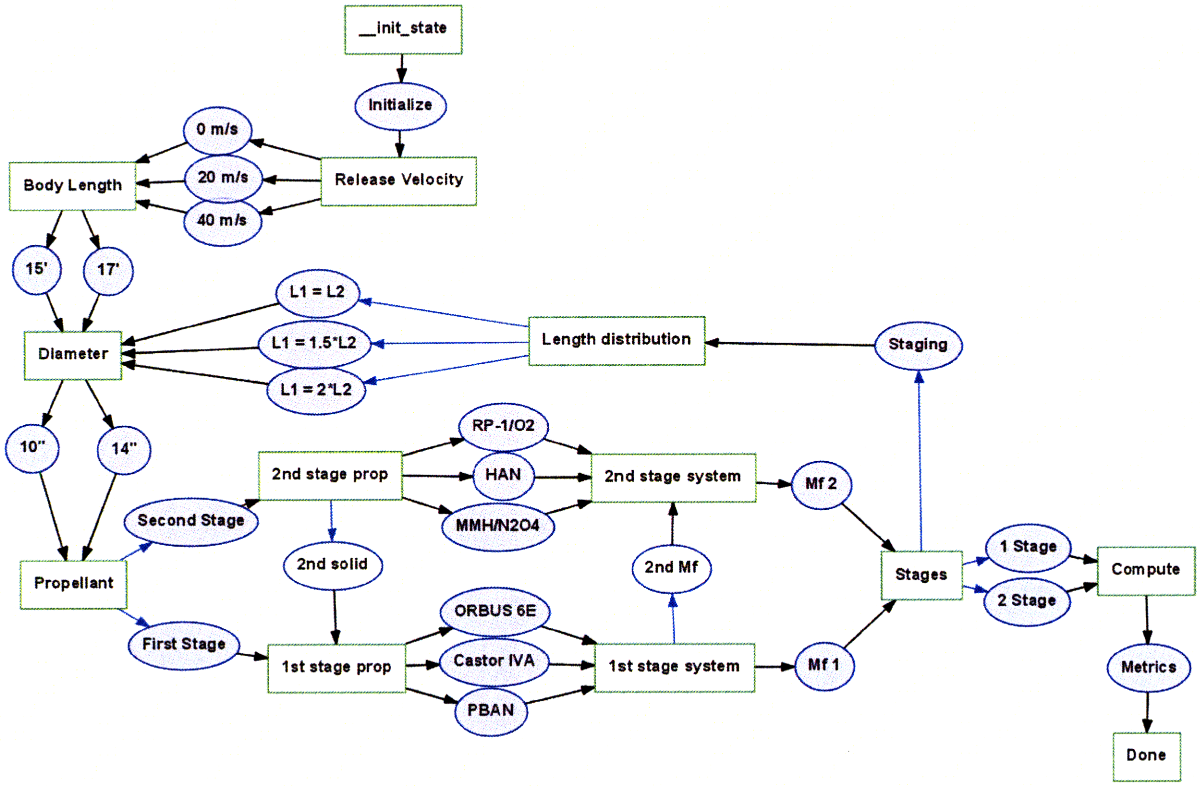

Figure 4-2: Air-Launched Sounding Rocket OPN model

two stages the propellant is being decided. The sounding rocket can either have one or two stages for propulsion, but there is a constraint encoded — the first stage can only be a solid propellant choice. The second can be either solid or liquid. OPN has the ability to model these constraints so enumerating all the possible first-stage liquid combinations is not necessary. This ability of OPN is beneficial as additional time and effort would be required to model these designs and then sort through and remove them from the data. After the propellant is chosen, a mass fraction is assigned (Mf1 = 0.2 or Mf2 = 0.3) based on the propellant type chosen. At the Stages decision, the single-stage designs proceed through the 1 Stage ellipse to calculate the attributes. The two-stage designs continue upward through the Staging ellipse where a loop enables the second stage properties to be enumerated. The ratio of first stage length to second stage length is decided at the Length Distribution decision, and then the choices are made for the second stage Diameter and second stage Propellant. These two-stage designs continue through the 2 Stage ellipse to compute the attributes of altitude at apogee and per unit cost.

The calculation of the attributes for the OPN modeling is done using analytic expressions with some simplifying assumptions on the trajectory parameters. Based on the dimensions, propellant types, mass fractions, and stage distribution, the rocket equation is used to calculate the delta-V (ΔV). The trajectory is modeled as a vertical ballistic flight starting from 45,000 feet. The initial velocity is the release velocity plus 65% of its ΔV, where the reduction is to account for drag and gravity losses. The per unit cost is modeled as a function of total mass.

ALSR Selection: Decision Analysis

The outputs of the OPN model are the attributes for 11,988 ALSR designs, which represent all of the feasible combinations of alternatives. At this point the challenge is to understand the differences, drivers, and distinguishing characteristics within the design space. The process of building the insight that comes from understanding these facets falls under the *Decision Analysis* activity.

To better understand the designs generated using OPN, the results are plotted in Figure 4-3. The arrows next to the axis labels point in the direction of improvement. The non-dominated (best) designs are therefore toward the upper-left portion of each plot. Where the designs fall in the trade space is itself interesting, but more useful information is extracted by identifying which design variables affect which designs. This is accomplished by highlighting these options in the trade space.

In the plot highlighted by Stage Diameter in Figure 4-3, the most noticeable feature is how the single stage designs are the worst performing designs, being dominated by the rest of the space of options. These are the blue (10" diameter) and green (14" diameter) designs along the bottom of the plot (lowest altitude). (These designs are also evident in the Length Division plot.) The cyan designs with a 10" first stage and 14" second stage do not have any non-dominated designs. The other options, however, have points that are non-dominated at different portions of the Pareto front. In the Total Length plot, the longer rockets are shifted up and to the right of the shorter rockets, which means they go higher but also cost more. For the First Stage Propellant design highlighting, there is a distinct region associated with the

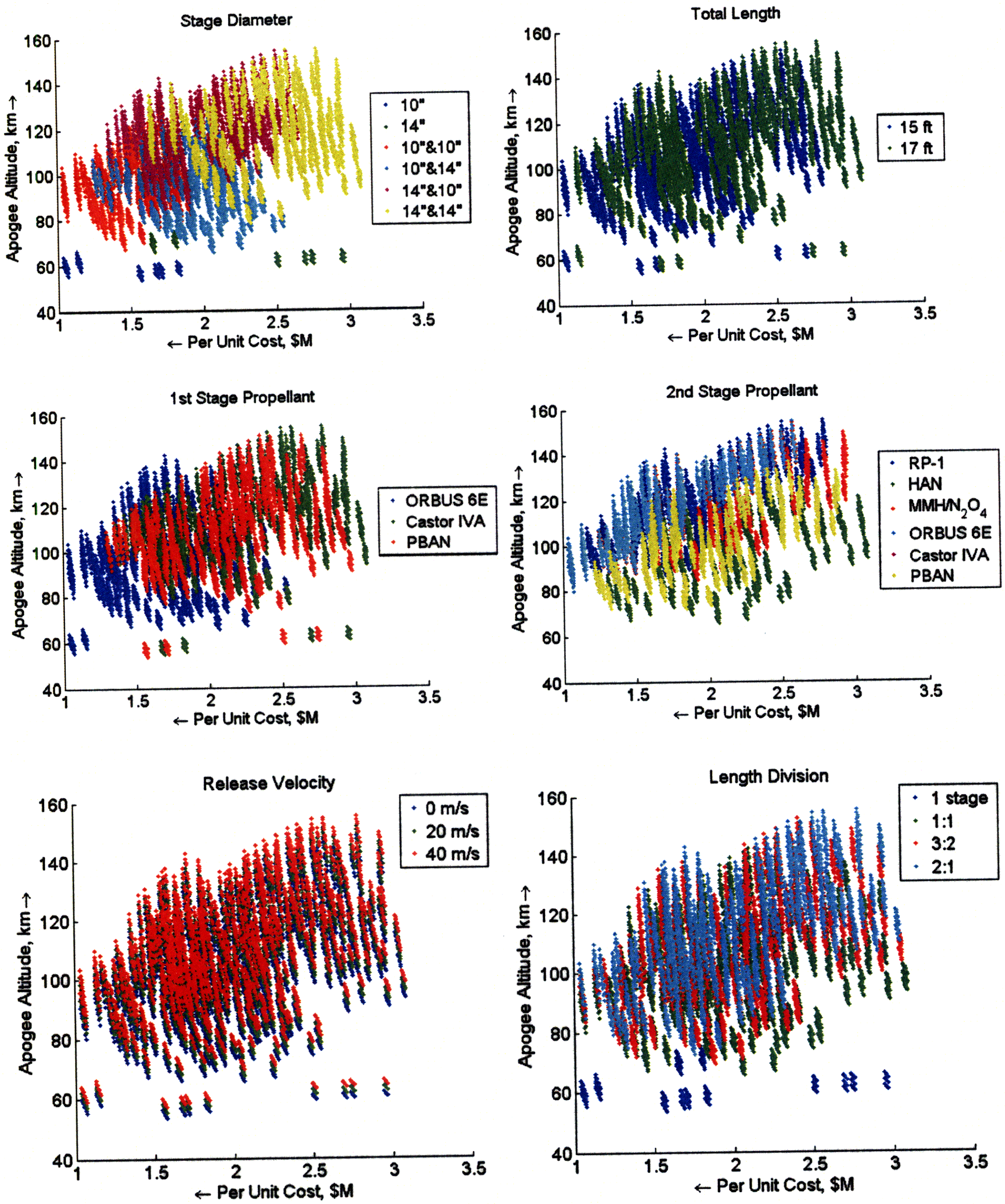

Figure 4-3: ALSR *Selection* OPN Trade Spaces

ORBUS 6E solid propellant, toward the lower-flying and cheaper region of the trade space. The other two options show little difference in their attributes. The Second Stage Propellant plot shows three propellant options that perform better than the rest: ORBUS 6E, RP-1, and MMH/N_2O_4, which is hidden behind the ORBUS 6E designs. Interestingly, ORBUS 6E has shown good performance as both a first-stage and second-stage propellant choice. In the final two plots, there are slight differences in the design options. Faster release velocities give higher apogee altitudes, which is intuitive. Higher ratios between the first stage length and the second stage length gain higher altitudes than lower ratios.

Identifying the effects of different design choices in the trade space informs the designer as to what design choices impact the attributes the most, as well as which designs perform the best. To help sort through the numerous design tradeoffs in a statistical fashion, ANOVA reveals which design choices are the most important contributors to variation in the attributes. Figure 4-4 shows the ANOVA results for *Selection* phase class analysis of the ALSR problem.

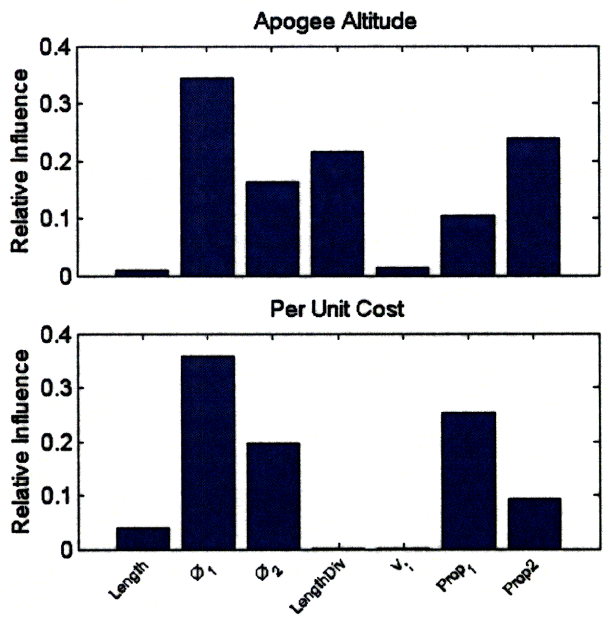

Figure 4-4: ALSR *Selection* ANOVA

The ANOVA results show the importance of each design variable for the two attributes, where the order of the design variables is total length, first stage diameter, second stage diameter, length division between the stages, release velocity, first stage propellant, and second stage propellant. The apogee altitude attribute, for example, shows the first stage diameter having the largest relative influence. This means that the choice of the first stage diameter largely determines where in the trade space the design will be. This can be seen in the trade space plot by noticing the distinct regions that stand out for each diameter choice. Note that the first stage diameter also has the highest relative influence in the per unit cost attribute. The designer therefore must closely examine the tradeoffs when choosing the diameter of the first stage, since it an important factor in determining both how high the sounding rocket will fly and how much it will cost. An example of a design variable exhibiting low relative influence is the release velocity. The release velocity does not factor into the cost calculation, and it results in a small shift in apogee altitude.

The information and insights from both the trade space exploration and the analysis of variance results are used to select a set of design options to carry forward to the next phase class. That knowledge is also used to reduce the number of design variables the optimization algorithm must search through, and is done by setting the less important ones to constants (moved to the constants vector, $\mathbf{k}$). These include the number of stages (set to two), the total length (set to 15 ft.), the length division (set to 1:1), and the release velocity (set to 40 m/s). The remaining design variables are carried forward in the detailed analysis of the *Refinement* phase class.

4.1.2 ALSR Refinement

After completing the four *Selection* phase class activities, the next iteration of analysis is in the *Refinement* phase class. The additional modeling detail comes in the form of a two degree-of-freedom trajectory simulation. The increased fidelity is important when the accuracy of the models must be verified with respect to more advanced simulations or empirical data. Particle swarm optimization is implemented to enhance the efficiency in searching the design space for good families of designs.

ALSR Refinement: Problem Characterization

Re-examination of the problem definition, attributes, and constraints results in no modification from the *Selection* characterization of the problem. In other cases, this step might include redefining the key attributes. For example, if in *Selection* the downrange distance from the release point were also calculated as an attribute and found to be highly correlated with altitude, then the designer may choose to drop downrange distance from the set of attributes.

ALSR Refinement: Alternative Generation

While there may be instances where additional alternatives are added to the design space in the *Refinement* phase, most cases will involve narrowing the space to reduce the number of model runs needed. In the ALSR case, the single stage designs are eliminated from the design space, and the length, stage length division, and release velocity are set to constants. Since the model evaluation in this phase is done with the aid of PSO, the design variables that are inherently continuous values are given a range rather than discrete choices.

ALSR Refinement: Model Development and Evaluation

The analysis of the sounding rocket design now moves from the analytic, closed-form solutions implemented in OPN to a more detailed, flyout simulation. A two degree-of-freedom (DoF) — horizontal and vertical components of motion — trajectory code is written in MATLAB® to simulate the flight. It numerically integrates the equations of motion (Eqs. 4.1 and 4.2) via 4^{th} order Runge-Kutta integration.

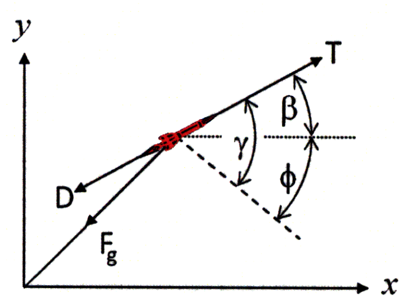

Figure 4-5: ALSR geometry for equations of motion

$$\sum F_x = m\ddot{x} = (T - D)\cos(\beta) - F_g \sin(\phi) \tag{4.1}$$

$$\sum F_y = m\ddot{y} = (T - D)\sin(\beta) - F_g \cos(\phi) \tag{4.2}$$

Figure 4-5 shows the geometry for the equations of motion, and the variables are defined as: T is the thrust, D is the drag, γ is the flight path angle with respect to the local horizontal, ϕ is the angle between the local horizontal and the inertial x-direction, $\beta = \gamma - \phi$, and F_g is the gravitational force.

In the simulation, the rocket launches vertically at the prescribed release velocity and its velocity and position are calculated as a function of time. The flight profile assumes a gravity turn.

ALSR Refinement: Decision Analysis

As mentioned in Section 3.1.3, optimization methods encompass a repetition of the three activites of *Alternative Generation*, *Model Evaluation*, and *Decision Analysis*. The optimization algorithm takes the attributes and preferences from the *Problem Characterization* activity and iterates on the other three activities. It starts with a design, evaluates that design using the models, identifies how that design ranks in terms of the *a priori* preferences, and then repeats the process to find better solutions. This is the approach used for the ALSR *Refinement* PSO algorithm.

The multiobjective particle swarm optimization algorithm described in Section 3.5.2 is used to search the design space of the refined, 2-DoF ALSR flyout simulation. The attributes are used as the two objectives. The particle swarm population size is 30 and the maximum number of time step iterations is set to 50.

Figure 4-6 shows the trade space of designs found with the PSO algorithm. The top left plot shows the designs evaluated by the PSO algorithm over its entire history (30 particles over 50 iterations) with the non-dominated solutions highlighted. The other three plots highlight the designs by their design variables. Since the stage diameter was a continuous variable and not selected from discrete choices, the colors are for dimensions either less than 12" or greater than or equal to 12." The smaller first stage designs (blue and green points) occupy almost the entire Pareto front. Note that this is slightly different from the analytic results of the *Selection* phase class. In that analysis, the smaller first stage designs only were non-dominated on the low-cost end of the Pareto front. The more detailed analysis was

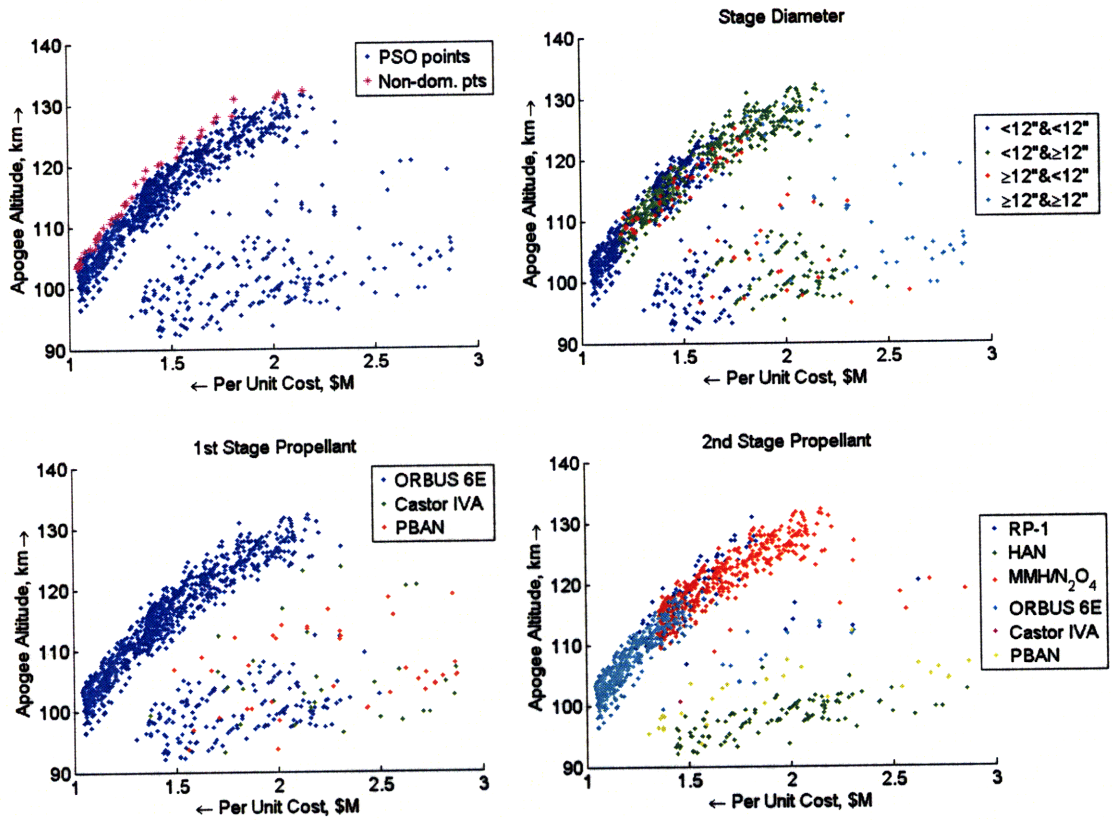

Figure 4-6: ALSR *Refinement* PSO Trade Spaces

necessary to differentiate the aerodynamic effects on the smaller configurations. In the first stage propellant variable, the optimizer clearly identified ORBUS 6E as the best. The large gap between the points near the Pareto front and those scattered throughout the rest of the trade space is due to the dominance of ORBUS 6E as the first stage propellant. The other propellant choices are entirely dominated and the optimization algorithm does not spend time populating those choices. The non-dominated designs with respect to the second stage propellant are RP-1, MMH/N_2O_4, and ORBUS 6E. Both of these propellant results are in agreement with the *Selection* results.

To see how the particle swarm optimization algorithm performed relative to the full set of design combinations generated in the *Selection* phase, the 11,988 designs generated in the manner of a full factorial design of experiments are analyzed with the more detailed 2-DoF trajectory code. Figure 4-7 shows these designs and the 1500 PSO designs. It is evident that the PSO algorithm quickly populated designs along the Pareto front. The PSO algorithm needed far fewer function evaluations to find almost the exact same set of non-dominated designs. Furthermore, the presence of PSO points distributed throughout the trade space means that it conducted a global search. In this case the length of the simulation time was short enough that the full space of 11,988 designs could be evaluated with the higher fidelity simulation. With more time-intensive and computationally expensive models, this may not be practical. This ALSR case, however, verifies that the PSO algorithm is efficient in its search to find the Pareto front.

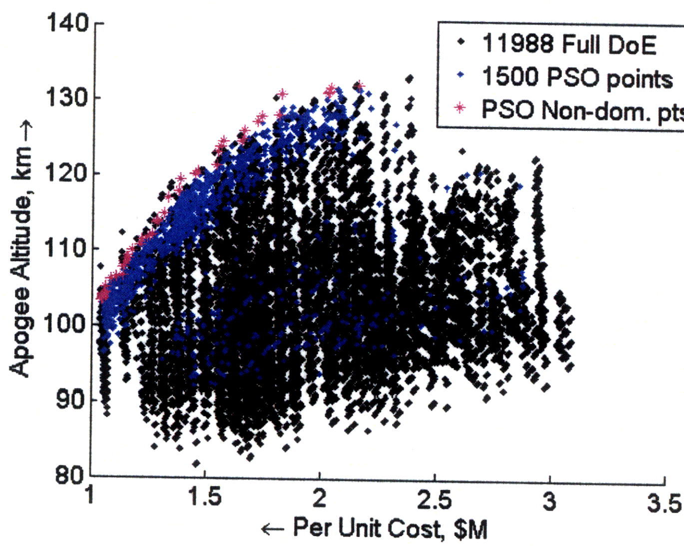

Figure 4-7: ALSR *Refinement* Comparison between DoE Full Factorial and PSO

The next important consideration for evaluating the ALSR designs is to examine the sensitivity to uncertainties. Depending on the particular study, a number of different factors and parameters can be varied to determine the sensitivities to those changes. This can be done for values in the constants vector, $\mathbf{k}$, or for values in the design vector, $\mathbf{x}$. For the former, an example might be the constant value assumed for the instrument payload mass. This is not involved in any of the design decisions, yet its value may dramatically affect the performance of certain rocket designs. ANOVA is particularly well suited for identifying the design variables that are most important and should be considered in sensitivity studies. For a variable that effects large variation in the trade space, looking at the sensitivity to changes in that value can reveal how important it is to accurately determine the real world value of that variable.

Referring again to Figure 4-4, the ANOVA results show that the propellant choices have high relative influence in both the cost and altitude attributes. Comparing this to the trade space plots, it is evident that ORBUS 6E is a propellant choice that could have significant effects on both of the attributes. It is clear winner for the first stage choice of propellant, and one of the three best choices for the second stage propellant. The effects on the attributes due to variation in that choice is therefore of great interest to the designer. The approach then is to see how changes in the propellant properties affect the resulting trade space, particularly the Pareto front designs. The change in propellant properties is done by varying the propellant density and specific impulse independently. Since there is not an empirical uncertainty model for this case, the probability distribution is taken as a uniform distribution between 90% and 110% of the nominal property values. Because the non-dominated designs are the focus of the trade space, 100 Monte Carlo sensitivity simulations are run for each non-dominated design.

Figure 4-8 shows how the non-dominated designs are affected by variation in the propellant properties. The green points represent the mean value (for each attribute) for

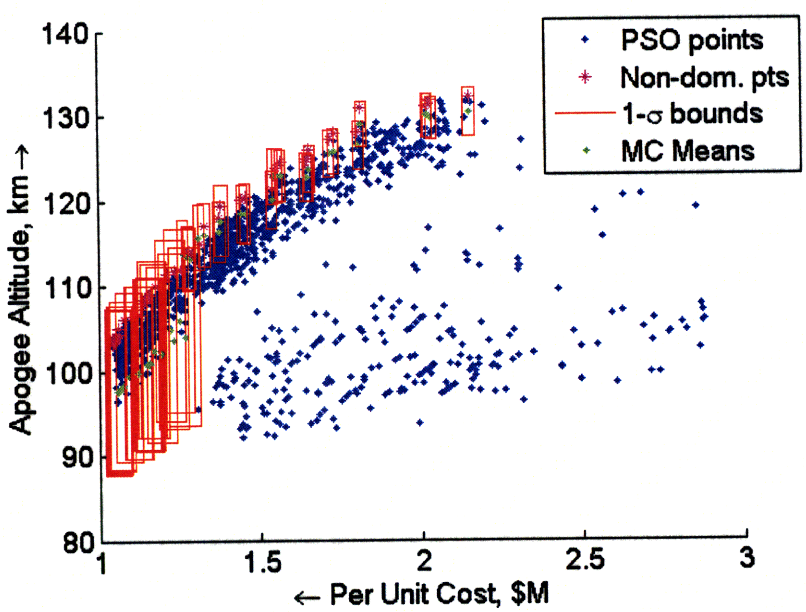

Figure 4-8: ALSR *Refinement* Monte Carlo sensitivity analysis conducted for designs along Pareto front under 10% variation in ORBUS 6E propellant properties. The boxes represent one standard deviation from the mean.

the 100 simulations. The red boxes represent one standard deviation from the mean in each attribute direction, which help visualize uncertainty in the trade space [86].

There are several insights the designer can gain from this sensitivity information. The first is to notice the shape of the boxes for different designs along the Pareto front. For instance, the boxes near the upper right portion of the Pareto front have smaller bounds than those near the lower left. This indicates that those near the lower left are more sensitive to changes in the propellant properties. Secondly, the sizes of the boxes tell the designer about the uncertainty tradeoffs. Depending on whether the box is narrow and tall or wide and short, the designer can gauge the relative effects on the attributes, helping make the decision whether a certain variation in cost is acceptable with respect to the expected variation in altitude. Finally, it is quite interesting to note how the mean values of the attributes move from the nominal Pareto set design points. The designs near the lower left, which have ORBUS 6E as both the first and second stage propellant have mean altitudes which drop substantially more than those near the upper right.

The sensitivity study helps direct the decisions of the designer. If the designer is looking to choose a design on or near the Pareto front that is least susceptible to variation, then the sensitivity information identifies these designs. Alternately, the sensitivity information can be used to direct further research into reducing the uncertainty for a parameter that results in large variation in the trade space.

4.1.3 ALSR Conclusions

The *Refinement* phase class results provide additional information about the trade space beyond that of the *Selection* phase class. Using the trade space insights and the sensitivity information for understanding the uncertainties, the designer can feel confident that a concept chosen for a further design development will be superior to a large number of other options.

Finally, a quick examination is made into the benefit gained by opting for an airborne launch. The trajectory flyout code is used to carry out a PSO run with the starting altitude at sea level. Figure 4-9 shows the difference between the two scenarios.

For an air-launched release at 45,000 ft. (13,716 m), the gain in apogee altitude is around 20–30 km. This 10–15 km net gain over the ground launch is an advantage, but perhaps smaller than expected. This may influence the initial decision to pursue an air launch over a ground launch. The knowledge of the trade space tells the designer that the diameter and propellant selections may be about as important as whether to pursue an air launch. In either launch decision, the primary goal of exploring the trade space has revealed many interesting insights into the propulsion system design.

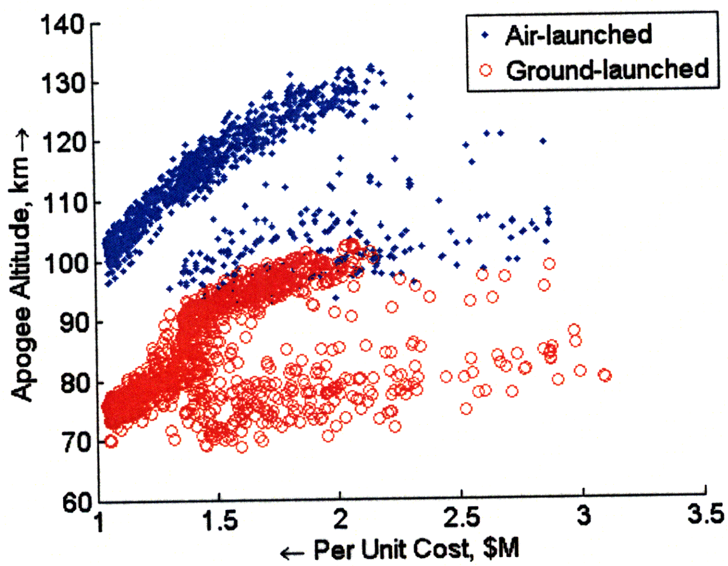

Figure 4-9: ALSR Comparison Between Ground-Launched and Air-Launched

4.2 Operationally Responsive System

The second design problem examined is a responsive disaster monitoring system. The problem is to design a system that provides high-quality, rapid coverage of an area of the world where observation capabilities do not exist or were not previously required. The system would gather information on a recent disaster site to help in the immediate recovery. Satellites are often utilized in these scenarios for imaging, and a responsive satellite is one option to include in the design space of solutions. To not limit the design space of options, however, this study will evaluate other approaches, as well as the combined effects of different vehicles working within the same system (aircraft and satellites, for example). The goal of the study is less about optimizing the design of a particular concept. Instead, the focus is on exploring the options for a system architecture incorporating several different concepts working together, called an Operationally Responsive System (ORS).

4.2.1 ORS Exploration

In contrast to the ALSR problem that started in the *Selection* phase class, the ORS problem starts in *Exploration*. The objectives are vague; the constraints are few; the problem itself is even unclear. This requires a pass through the Engineering Framework activities that is at a high level and of a qualitative nature.

ORS Exploration: Problem Characterization

In the *Problem Characterization* activity, the task is to determine what is important about the problem. There is only a small amount of qualitative information at this point, so a better characterization of the problem is necessary. The first consideration is to assess which disaster types and locations would be within the domain of a responsive monitoring system. A full list of many different disasters is generated to help identify patterns and characteristics of disaster types. This list is then narrowed down to those that only require rapid response (a drought, for instance, is a potential disaster that takes place over long time scales) and ones that affect human lives (i.e., they occur in populated areas). This compiled list of disasters

Table 4.2: Types of Disasters for an Operationally Responsive System

Forest fire	Hurricane
Thunderstorm	Tornado
Flood	Earthquake
Tsunami	Volcano
Riots	Oil spill
Asteroid (impact debris, etc.)	Avalanche/mudslide

is shown in Table 4.2.

This list of disasters is then used to identify people or organizations that would be interested in a responsive system. It is important to identify these stakeholders because the value that the system provides is based on who derives benefit and at what cost [13]. Overlooking who would be the important parties in either supporting or opposing the system could be more costly in the long run than a poor design. Compiling a list of stakeholders also helps the system designer figure out what capabilities the system needs to provide in a disaster situation. The list of stakeholders is shown in Table 4.3.

Table 4.3: Stakeholders for an Operationally Responsive System

Firefighters	Police
FEMA	Insurance Companies
Local volunteers	Property owners
Paramedics	Military/uniformed services
Red Cross	ORS owner

Another purpose of identifying possible stakeholders is to work toward determining what the requirements of the system should be. This is done by deriving the needs of the stakeholders and then the attributes of the system that meet those needs. The attributes are characteristics of the system that the stakeholders care about, which they use to judge the "goodness" of the system. From the stakeholder needs comes the derived set of attributes [68], shown in Table 4.4, which is used to measure and compare different concepts.

The procedure of identifying stakeholders and deriving attributes of the system can be seen as unimportant, but determining what value the system must deliver and to whom is crucial in designing a system that will meet the needs of the users/stakeholders.

This set of disasters, stakeholders and attributes is considered representative of what

Table 4.4: Attributes of an Operationally Responsive System

Attribute	Description
Acquisition Cost	Initial cost to acquire the system
Cost/Day	Operating cost of the system
Time to IOC	Lead time to initial operating capability (IOC)
Responsiveness	Time from first request to reach area of interest (AOI)
Max Coverage	Maximum coverage of an AOI, expressed as a fraction
Time to Max Coverage	Time it takes to reach maximum coverage
Time Between AOIs	Time it takes to retask the system to another AOI
Imaging Capability	How good the images or observations of the disaster are
Data latency	How long from user data request to acquisition

a responsive system must be designed for. With respect to the scope of this study, however, a particular disaster scenario is used for modeling and for obtaining the attributes of the system. The example disaster scenario is a hurricane similar to Hurricane Katrina that hit the city of New Orleans in August of 2005.

ORS Exploration: Alternative Generation

After determining what is important for the Operationally Responsive System, the next activity in the Engineering Framework is to generate alternatives. In the *Exploration* phase class, a wide range of alternatives is considered that could potentially meet the goals of the system. Table 4.5 shows this list of alternatives and gives a short description of each.

Table 4.5: Alternatives for an Operationally Responsive System

Alternative	Description
Satellite	single imaging satellite or satellite constellation
Crewed fixed wing	airplane with pilot in the aircraft
Crewed rotor	helicopter with pilot inside
Uncrewed fixed wing	drone airplane, unmanned air vehicle (UAV)
Uncrewed rotor	UAV helicopter
Airship	piloted dirigible or blimp
Distributed swarm	series of low-cost sensors/cameras launched over the AOI that parachute or glide down to the ground
Balloon	uncontrolled balloon, either tethered or untethered
Land vehicle	roving vehicle on the ground
Personal sensors	sensors given to personnel on the ground to make observations or measurements

ORS Exploration: Model Development and Evaluation

The next activity is to try to assess which of the concept alternatives are most promising and which are clearly inferior to the rest. At this stage the alternatives have very general descriptions, so the modeling and evaluation is qualitative and subjective. The technique most applicable for this case is Pugh analysis. As described in Section 3.2, Pugh analysis is a technique in which pair-wise comparisons are made between the different concept alternatives to determine which rank better in terms of the attributes.

Table 4.6 shows the Pugh matrix analysis with a satellite concept as the baseline. The other concepts are scored better (+1), worse (-1), or the same (0) for each of the attributes. The total sum is given in the bottom line. With the satellite as the baseline concept, the other concepts all had total scores above zero. For the given set of attributes, this subjective assessment indicates that there are probably better ways to get cost-effective, rapid, and large coverage data on a disaster than with a satellite.

To enhance the discrimination between the concepts, a multi-baseline Pugh analysis approach is used. After making pair-wise comparisons for the satellite baseline, three other

Table 4.6: Pugh Analysis for ORS: Satellite Baseline

Attribute	**Satellite**	Crewed fixed wing	Crewed rotor	Uncrewed fixed wing	Uncrewed rotor	Airship	Distributed swarm	Balloon	Land vehicle	Personal sensors
Acquisition Cost	0	1	1	1	1	1	1	1	1	1
Cost/day	0	-1	-1	-1	-1	0	1	1	0	1
Time to IOC	0	1	1	1	1	1	1	1	1	1
Responsiveness	0	0	0	0	0	0	1	1	0	1
Max Coverage	0	0	0	0	0	0	-1	-1	-1	-1
Time to Max Coverage	0	-1	-1	-1	-1	-1	0	-1	-1	-1
Time Between AOIs	0	1	1	1	1	1	1	1	1	-1
Imaging Capability	0	1	1	1	1	1	-1	0	1	-1
Data Latency	0	1	1	1	1	1	1	1	1	1
Total Score	0	3	3	3	3	4	4	4	3	1

Table 4.7: Pugh Analysis for ORS: Crewed Fixed Wing Baseline

Attribute	Satellite	**Crewed fixed wing**	Crewed rotor	Uncrewed fixed wing	Uncrewed rotor	Airship	Distributed swarm	Balloon	Land vehicle	Personal sensors
Acquisition Cost	-1	0	0	1	1	1	1	1	1	1
Cost/day	1	0	-1	1	0	1	1	1	1	1
Time to IOC	-1	0	0	0	0	0	0	0	0	0
Responsiveness	0	0	1	1	1	0	1	1	1	1
Max Coverage	0	0	0	0	0	0	-1	-1	-1	-1
Time to Max Coverage	1	0	-1	-1	-1	-1	1	-1	-1	-1
Time Between AOIs	-1	0	-1	-1	-1	-1	-1	-1	-1	-1
Imaging Capability	-1	0	0	0	0	-1	-1	-1	0	-1
Data Latency	-1	0	0	-1	-1	-1	-1	-1	-1	-1
Total Score	-3	0	-2	0	-1	-2	0	-2	-1	-2

baselines are chosen to better distinguish the advantages and disadvantages of the various concept alternatives. The crewed fixed wing baseline Pugh matrix is shown in Table 4.7, the uncrewed fixed wing baseline in Table 4.8, and the land vehicle baseline in Table 4.9.

ORS Exploration: Decision Analysis

The final activity in the *Exploration* phase class is to examine and learn about the results, and then make a decision on which concepts to carry forward to the *Selection* phase class of the Engineering Framework. To examine the Pugh results, the scores for each baseline are added to give the cumulative effect. Figure 4-10 shows the scores of each concept for each baseline, as well as the cumulative score across all the baselines. The height of each colored portion in the stacked bar chart represents the score for that baseline. If it is above the zero line on the vertical axis, it performed better than the baseline; if it is below it performed worse. The black diamond gives the final cumulative score summed from all of the baselines. This cumulative score is the best metric on which to evaluate the overall ranking of each concept. Both the crewed aircraft and the uncrewed fixed wing score well.

Table 4.8: Pugh Analysis for ORS: Uncrewed Fixed Wing Baseline

Attribute	Satellite	Crewed fixed wing	Crewed rotor	**Uncrewed fixed wing**	Uncrewed rotor	Airship	Distributed swarm	Balloon	Land vehicle	Personal sensors
Acquisition Cost	-1	-1	-1	0	-1	1	1	1	1	1
Cost/day	1	-1	-1	0	-1	1	1	1	1	1
Time to IOC	-1	0	0	0	0	0	0	0	0	0
Responsiveness	0	-1	-1	0	0	-1	1	1	-1	1
Max Coverage	0	0	0	0	0	0	-1	-1	-1	-1
Time to Max Coverage	1	1	0	0	-1	-1	1	-1	-1	-1
Time Between AOIs	-1	1	0	0	-1	-1	-1	-1	-1	-1
Imaging Capability	-1	0	1	0	0	-1	-1	-1	1	-1
Data Latency	-1	1	1	0	0	0	-1	-1	-1	-1
Total Score	-3	0	-1	0	-4	-2	0	-2	-2	-2

Table 4.9: Pugh Analysis for ORS: Land Vehicle Baseline

Attribute	Satellite	Crewed fixed wing	Crewed rotor	Uncrewed fixed wing	Uncrewed rotor	Airship	Distributed swarm	Balloon	**Land vehicle**	Personal sensors
Acquisition Cost	-1	-1	-1	-1	-1	1	1	1	0	1
Cost/day	0	-1	-1	-1	-1	0	1	1	0	1
Time to IOC	-1	0	0	0	0	0	0	0	0	0
Responsiveness	0	-1	1	1	1	-1	1	1	0	1
Max Coverage	1	1	1	1	1	1	0	0	0	-1
Time to Max Coverage	1	1	1	1	1	0	1	-1	0	-1
Time Between AOIs	-1	1	1	1	1	1	0	0	0	-1
Imaging Capability	-1	0	-1	-1	-1	-1	-1	-1	0	-1
Data Latency	-1	1	1	1	1	1	-1	0	0	-1
Total Score	-3	1	2	2	2	2	2	1	0	-2

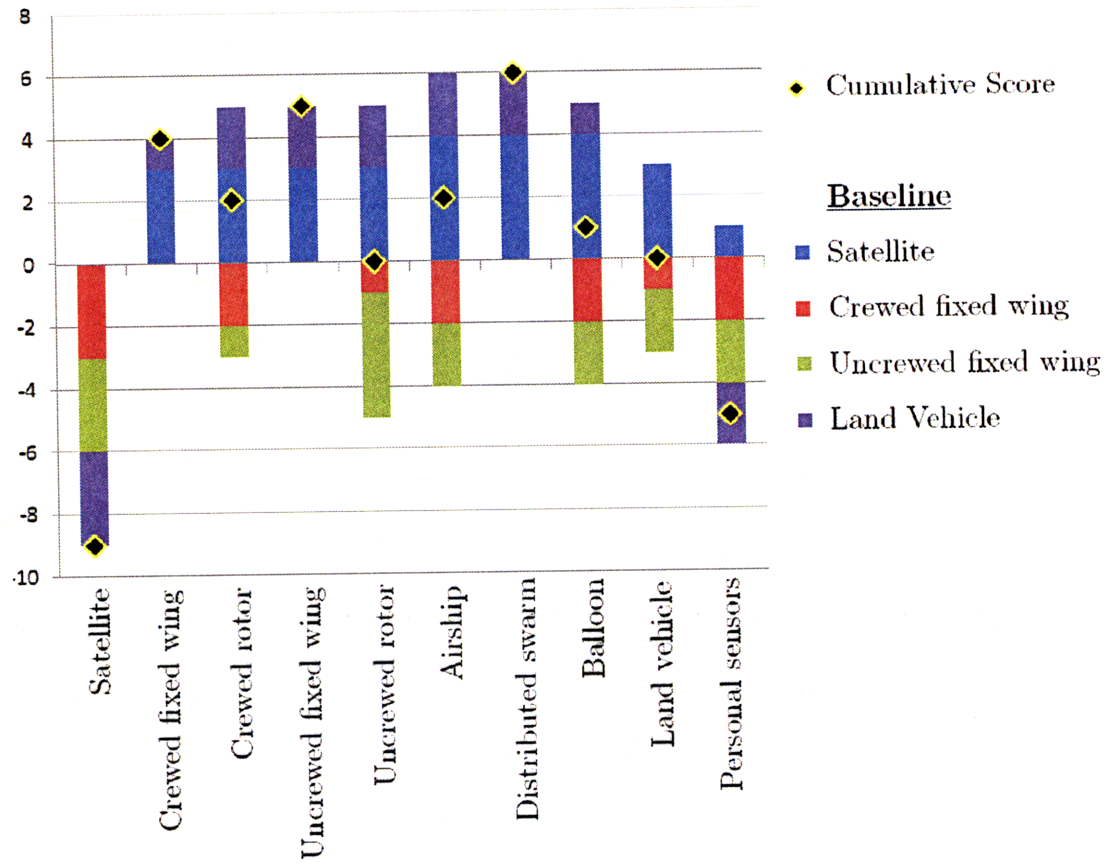

Figure 4-10: Cumulative baseline scores for ORS Pugh analysis

The scores for the uncrewed rotor, balloon, land vehicle, and personal sensor concepts are worse. The distributed swarm is the highest and the satellite concept is the lowest. This is a reflection primarily of their responsiveness and costs — expensive satellites may take a while for the orbit to align with the location on Earth where the disaster is, and a low-cost suite of cameras can be quickly dispersed over an area of interest. There are, however, hard-to-capture aspects of satellites that warrant their consideration in further investigations, such as their sustainability in orbit and potentially global coverage. The other concepts all need to be pre-positioned at specific sites. With these results as a basis for decision-making, the following concept alternatives are selected for deeper analysis in the *Selection* phase class of the Engineering Framework: satellites, crewed fixed wing, crewed rotor, uncrewed fixed wing, and distributed swarms.

4.2.2 ORS Selection

While the *Exploration* phase class of the Engineering Framework is focused on defining the problem and gauging initial limits on what can or cannot be used in the design, the *Selection* phase class aims to refine where in the trade space concept alternatives lie relative to one another. It includes more quantitative assessments with objective measures of performance.

ORS Selection: Problem Characterization

The ORS problem started with fairly vague, notional ideas of what the system should do, and no indication of what it should be. The problem became much better defined through the *Exploration* phase class. The *Selection* phase class *Problem Characterization* leverages those results heavily. For the purposes of modeling and attribute calculations, however, the attribute set is reduced slightly. The Cost/Day, Time to IOC, and Data Latency attributes are viewed as the less important attributes of the complete set, and are not considered in further investigations.

ORS Selection: Alternative Generation

The approach for *Alternative Generation* is to select existing systems whose roles are similar to what the Operationally Responsive System would be required to do. This includes crewed and uncrewed reconnaissance aircraft, imaging satellites, and distributed swarm examples. With this approach, the alternatives selected are shown in Table 4.10.

In addition to these alternatives, other design variables are included in the design space to increase the design freedom and measure their effects on the system performance. The number of individual assets considered at once is either 1, 2, or 3 vehicles for both satellites and aircraft. For the distributed swarms, a similar design variable is included, but these numbers are increased to 10, 20, and 30. For the aircraft alternatives (both crewed and uncrewed), different instrument payloads are considered, which have various options for aperture diameter and sensing wavelength.

Table 4.10: ORS *Selection* Baseline Alternatives

Imaging Satellites	***Crewed Fixed Wing***
IKONOS®	Proteus
QuickBird	Cessna 206
OrbView-3	P-3 Orion
SPOT-5	
Crewed Rotor	***Uncrewed Fixed Wing***
UH-60 Black Hawk	RQ-1 Predator
Robinson R44	ScanEagle
Distributed Swarm	Global Hawk
Camera Swarm	RQ-11 Raven
CanSats	RQ-2 Pioneer

ORS Selection: Model Development and Evaluation

The *Selection* phase class modeling incorporates an additional level of detail over that of the *Exploration* phase class. Concept evaluation via Pugh analysis is fairly subjective; more objective measures of the concepts are now implemented. The attributes of the concepts are determined by calculating how fast they can cover areas of interest, how good their imaging systems are, and how much they cost. Highly accurate calculations are not needed because the concept descriptions are still generally broad. Appropriate model fidelity among concepts is more useful at this point than highly precise data. The accuracy of detailed simulations may be completely overshadowed by the uncertainty in less precise models. The aim is for consistent measures of the attributes to provide objective comparisons of the concepts.

Low-order models are set up to evaluate the attributes of each of the satellite, aircraft and distributed swarm concepts. The baseline alternatives (Table 4.10) provide representative performance specifications for the concepts. This information, available in through websites in the public domain, includes satellite orbits, resolution, and revisit rates; aircraft cruise speeds, cruise altitudes, service ceilings, and endurances; and CanSat altitudes and payload capabilities.

The models incorporate these specifications to determine how each concept would perform in an ORS mission scenario. For instance, the time it takes for an aircraft to cover a given area is calculated based on the imaging footprint area and the speed of the aircraft.

The satellite models use the revisit rate to determine how long it takes to view the disaster site again, and the orbital speed to calculate coverage rates. The responsiveness is modeled as the time it takes the first asset to arrive at the disaster site. Satellites are assumed to be on orbit and their first viewing is based on the revisit time for the orbit. The aircraft responsiveness is modeled as the flight time from the airfield to the AOI. Since multiple assets for a specific type of aircraft take the same time to fly to the site, there is no variation in the time between 1, 2, and 3 assets.

The distributed swarm attributes are determined as follows. The camera swarm is considered to be a series of small cameras like those in camera phones. They are launched on Estes model rockets,with the altitude and coverage determined by the rocket performance and camera field-of-view, respectively. The CanSat concept is based on amateur rockets whose payloads fit in a cylindrical space about 3 inches in diameter and 12 inches in length [19]. These rockets fly higher, and the payload is larger, than those of the camera swarm.

The OPN model for all the different concept alternatives is shown in Figure 4-11. The ORS OPN model begins with initialization. After choosing the number of assets, each of the various alternative choices is enumerated, along with appropriate payload options for the aircraft concepts. The uncrewed aircraft are modeled as having smaller payload capabilities than the crewed aircraft, which is the reason for fewer options.

ORS Selection: Decision Analysis

Figures 4-12 and 4-13 show the trade spaces for all of the concept alternative combinations. The plot in Figure 4-12 shows the (peak) resolution-versus-cost trade space with the satellite, aircraft and swarm concepts identified by color. The aircraft concepts are modeled with different payloads, so the visual sensing payloads are differentiated from the infrared (IR) payloads. The lines connecting different design points represent the three levels of the number of assets for each concept.

It can be seen that the three main concepts (satellites, aircraft, and distributed swarms) occupy specific regions of the trade space, although there is some overlap between satellites and the expensive types of aircraft. The distributed swarms are on the lower end

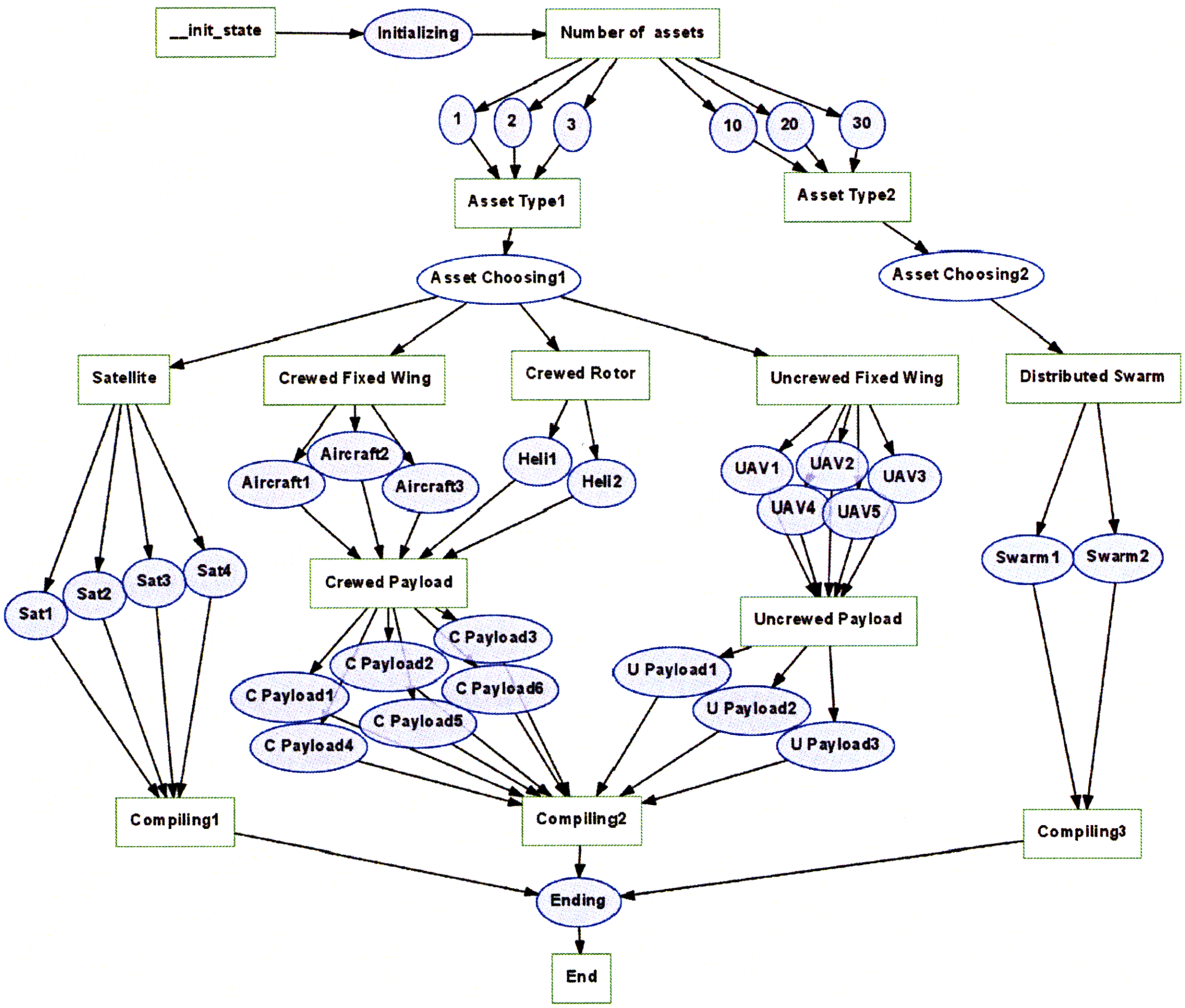

Figure 4-11: ORS *Selection* OPN model

of the cost axis, and its resolution is on par with the other concepts. The satellites are on the opposite end of the cost spectrum, with the best baseline alternatives achieving < 1 m peak ground resolution. The aircraft concepts are more widely dispersed through the trade space than the distributed swarm or satellite concepts, both because there are more of them and because they vary in their sizes and capabilities. For example, the smallest, the RQ-11 Raven, is a hand-launched UAV that flies low to the ground while the P-3 Orion and Global Hawk have high cruise speeds, high cruise altitudes, and high costs. The aircraft design point with the best (lowest) resolution is the Raven, and that with the highest cost is the Global Hawk. Finally, the difference in resolution between the visual and IR payloads is clear in the offset between the two in the trade space.

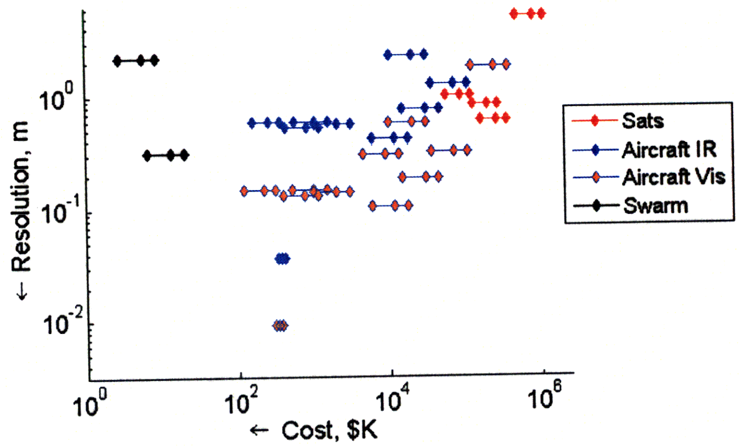

Figure 4-12: ORS *Selection* Resolution versus Cost Trade Space

Examination of the trade spaces for the other attributes reveals additional insights into the problem. The top four plots in Figure 4-13 have cost along the horizontal axis, while the bottom four have resolution. For the maximum coverage attribute, all of the concepts are modeled as achieving total coverage except for the distributed swarm concepts, which can only obtain data on about 0.3–10% of the 100,000 km^2 AOI. The coverage times associated with each concept are shown in the plot directly below the maximum coverage plot. The satellites and swarms both cover area the quickest. The variation in coverage for different aircraft concepts is based on the different speed and altitude characteristics of each.

An interesting trade space showing close tradeoffs between concepts is the lower right plot showing the trade space for time between AOIs versus resolution. Here even the satellite concepts and swarm concepts lie within close proximity, which is not the case when cost is considered. Yet it is still the case that several aircraft concepts perform better than the satellite and swarm concepts in these two attributes.

The large number of attributes and concepts makes it challenging to downselect to a smaller set of alternatives. The approach for selecting which alternatives to consider further uses the non-dominated designs from each trade space plot. For each two-attribute trade space (where the concepts are compared for two of the attributes), there are certain concepts that are non-dominated. The non-dominated designs can also be determined for a three-attribute trade space. The concepts that repeatedly show up as non-dominated across

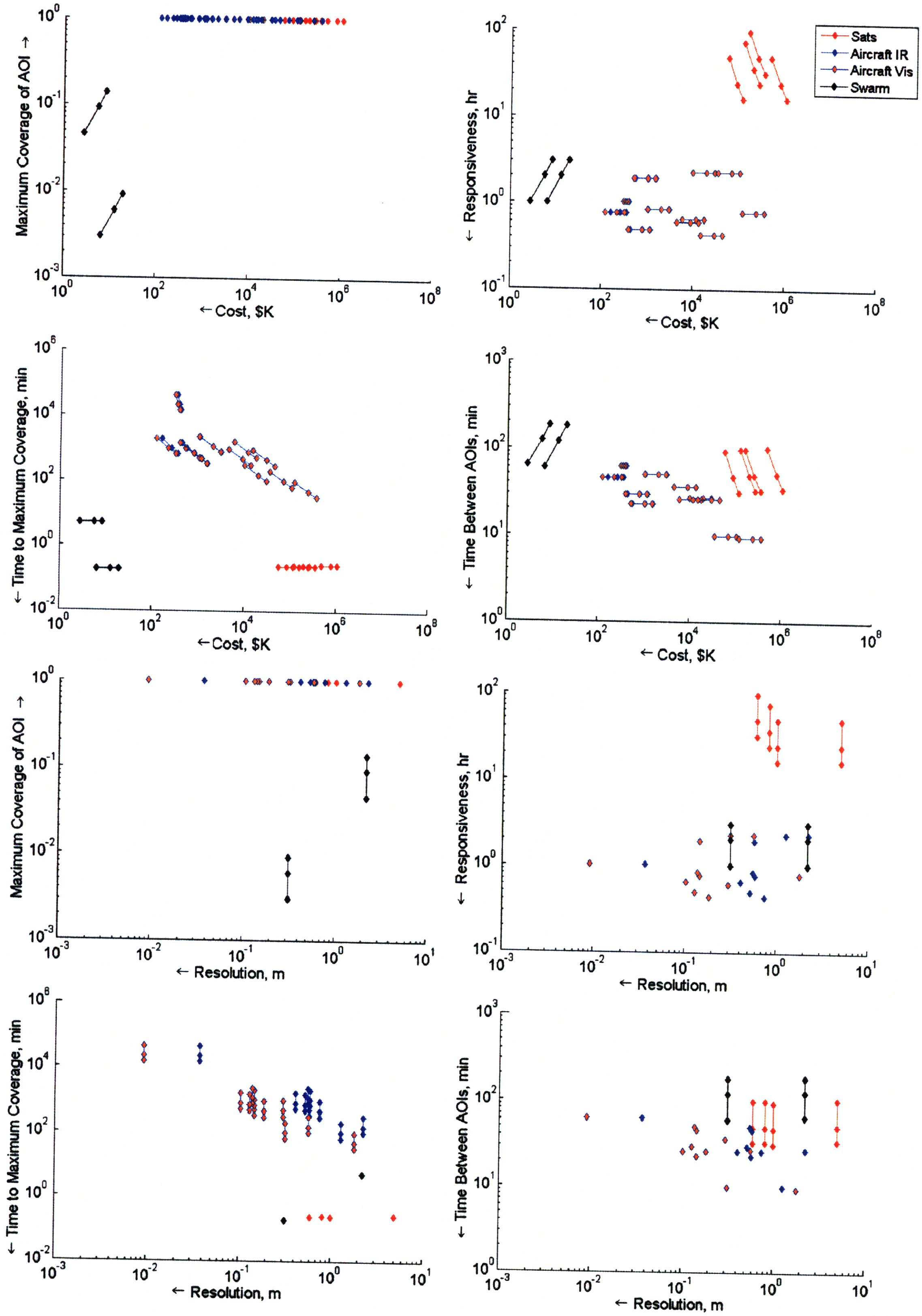

Figure 4-13: ORS *Selection* DoE Trade Spaces

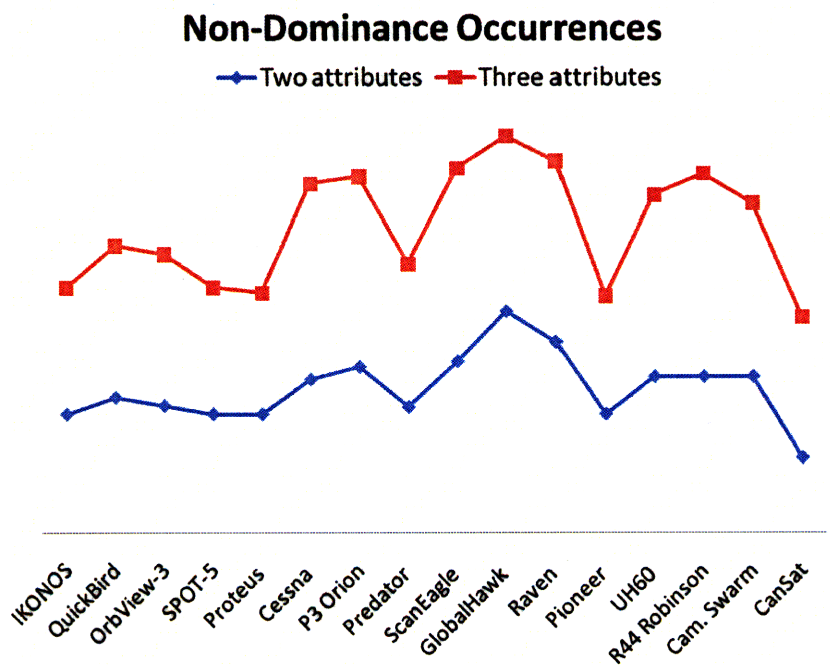

Figure 4-14: ORS *Selection* Non-Dominated Alternatives

different trade spaces are those that perform well with respect to the entire set of attributes. The downselection approach is then as follows. The non-dominated concepts are determined for each two-attribute trade space, and then for each three-attribute trade space. Every time a concept appears in a non-dominated set, a count is made for that concept. The better overall concepts are those with the highest non-dominated occurrence count. The results for these occurrences are shown in Figure 4-14.

From the information in the trade space plots as well as the count of non-dominated alternatives, four concepts are chosen for more detailed examination: an unmanned air vehicle similar in class to Global Hawk; a small aircraft similar to ScanEagle, which is a catapult-launched UAV; a single engine, propeller-driven aircraft similar to a Cessna 206; and a satellite based on the QuickBird system. The first two aircraft concepts are two of the best performing in terms of the non-dominance count. In addition, they are two aircraft concepts on opposite ends of the size spectrum. Global Hawk has a 116 ft wingspan and flies missions as long as 24 hours at time [62]. ScanEagle, on the other hand, has a wingspan of 10 ft and can be transported in the back of a truck [38]. The Cessna-class aircraft is sized between the two UAV concepts, is a crewed aircraft, and is also one of the best performing in the non-dominance count. There are advantages that orbiting the planet affords, and this provides the rationale for examining satellites in more detail. The QuickBird baseline alternative is the highest scoring satellite in the non-dominance count.

4.2.3 ORS Refinement

The final phase class of the Engineering Framework is *Refinement*. An additional level of detail is added in modeling the aircraft and satellite alternatives. Furthermore, a mission scenario timeline is developed that models how the combination of satellite and aircraft concepts together perform the task of monitoring a disaster situation.

ORS Refinement: Problem Characterization

In the refinement of the ORS problem, some updates are made to the attribute set. While all the attributes remain important, the three more important ones are selected to represent the fundamental features of the system. The first is the total ground area covered, measured in square kilometers. The second is the average ground resolution, measured in meters. The third and last attribute is acquisition cost, measured in thousands of dollars, and includes research, development, testing, and non-recurring engineering costs.

The attribute of responsiveness, whose absence from the set of three attributes may be questioned, is inherently related to the cost and coverage capabilities of the system. The amount a potential customer is willing to pay to acquire such a system largely determines how capable it is. For a system that includes a vast network of predeployed assets in space and at ground locations where they can respond quickly to possible disasters, the cost will be high and the response to any disaster will be very rapid. On the other hand, a more sparse system of assets will be less expensive but not capable of as fast a response. The three attributes of area covered, average resolution, and cost are therefore considered a representative attribute set for measuring the combined system performance of concepts in the *Refinement* phase class.

ORS Refinement: Alternative Generation

As mentioned at the end of Section 4.2.2, the four concepts carried forward from the *Selection* phase class are three classes of aircraft (one based on the large Global Hawk UAV, the second on the small ScanEagle UAV, and the third based on a Cessna 206 aircraft) and one satellite (based on the QuickBird imaging satellite). For each of these concepts, design variables are added to better differentiate the concepts. Table 4.11 shows the design variables.

Table 4.11: ORS *Refinement* Design Variables and Alternatives

Concept	Design Variable	Alternatives		
Aircraft	Class	Large UAV	Small UAV	Single Prop
	Payload Aperture Diameter, cm	2	8	
	Payload Sensing Wavlength, μm	0.4	2	
	Takeoff Gross Weight	low	high	
Satellites	Number of Orbital Planes	1	2	3
	Satellites per Orbital Plane	1	2	3
	Altitude, km	300	900	
	Payload Aperture Diameter, m	0.5	2	
	Payload Sensing Wavlength, μm	0.4	3	

The aircraft takeoff gross weight is the weight at takeoff, including the aircraft, payload, and fuel. It represents the size of the aircraft within each class and is varied to see how the endurance changes. The satellite orbit options describe different types of Walker constellations with circular orbits [47]. The orbital planes are equally spaced, as are the satellites in each plane.

ORS Refinement: Model Development and Evaluation

The *Model Development and Evaluation* activity has three main components for the ORS *Refinement* phase class: aircraft modeling, satellite modeling, and a mission timeline model. These are described individually.

Aircraft In order to carry out the concept design of aircraft for a disaster response mission, aircraft sizing is conducted based on the ORS mission scenario. The primary equations for this are the Bruguet range and endurance equations. Other aircraft specifications are based on historical trends from existing aircraft data [67, 53]. The Bruguet range and endurance equations are shown in Eqs. 4.3 and 4.4,

$$R = \frac{V}{C}\frac{L}{D}\ln\frac{W_{i-1}}{W_i} \tag{4.3}$$

$$E = \frac{1}{C}\frac{L}{D}\ln\frac{W_{i-1}}{W_i} \tag{4.4}$$

where V is the velocity, C is the specific fuel consumption (SFC), L/D is the lift to drag ratio, and W_{i-1}/W_i is the mission segment weight fraction. For propeller-driven aircraft the specific fuel consumption is expressed as a ratio of the propeller shaft power SFC, C_{power}, and is a function of the velocity and the propeller efficiency, η_{prop}, as shown in Eq. 4.5 [67].

$$C_{prop} = C_{power}\frac{V}{\eta_{prop}} \tag{4.5}$$

As explained further in the mission timeline description, the aircraft mission includes a takeoff portion, cruise segment to get to the disaster area of interest, loiter segment in which the data is gathered, and a cruise segment to return to the airfield. The mission segment weight fractions shown in Table 4.12 (from Ref. [67]) are used for the portions of the mission not considered cruise or loiter.

Table 4.12: Mission Segment Weight Fractions

Mission Segment	Weight Fraction, W_i/W_{i-1}
Warmup and Takeoff	0.970
Climb	0.985
Landing	0.995

In addition, an allowance for fuel reserve is estimated from historical values to be 6% [67]. The total fuel fraction is therefore

$$\frac{W_f}{W_g} = 1.06\left(1 - \frac{W_x}{W_g}\right) \tag{4.6}$$

where W_f is the fuel weight, W_g is the takeoff gross weight and W_x/W_0 is the total mission weight fraction.

The empty weight fractions for the different classes of aircraft can be estimated statistically from historical trends. For the three aircraft classes of Large UAV, Small UAV and Single Prop, the empty weight fractions, W_e/W_g, are as shown in Eqs. 4.7–4.9 [67].

$$\text{Large UAV:}\quad \frac{W_e}{W_g} = 1.02W_g^{-0.06} \tag{4.7}$$

$$\text{Small UAV:}\quad \frac{W_e}{W_g} = 0.397W_g^{0.084} \tag{4.8}$$

$$\text{Single Prop:}\quad \frac{W_e}{W_g} = 2.36W_g^{-0.18} \tag{4.9}$$

The takeoff gross weight, which is the sum of the the payload weight (W_p), fuel weight, and empty weight, is related to the mission fraction as shown in Eq. 4.10.

$$\begin{aligned} W_g &= W_p + W_f + W_e \\ 1 &= \frac{W_p}{W_g} + \frac{W_f}{W_g} + \frac{W_e}{W_g} \\ 1 &= \frac{W_p}{W_g} + 1.06\left(1 - \frac{W_x}{W_g}\right) + \frac{W_e}{W_g} \\ \frac{W_x}{W_g} &= \frac{1}{1.06}\left(0.06 + \frac{W_p}{W_g} + \frac{W_e}{W_g}\right) = k_1 \end{aligned} \tag{4.10}$$

Then the total endurance is determined from the Bruguet equations (Eq. 4.11),

$$\begin{aligned} \frac{W_x}{W_g} &= k_2 \exp\left(-\frac{1}{L/D}\sum_{i=1}^{2}\frac{R_iC_i}{V_i}\right)_{cruise} \exp\left(-\frac{C}{L/D}E\right)_{loiter} \\ E &= \left(\frac{L/D}{C}\right)_{loiter}\left[\ln\left(\frac{k_2}{k_1}\right) - \left(\frac{1}{L/D}\sum_{i=1}^{2}\frac{R_iC_i}{V_i}\right)_{cruise}\right] \end{aligned} \tag{4.11}$$

where $k_1 = \frac{W_x}{W_g} = \frac{1}{1.06}\left(0.06 + \frac{W_p}{W_g} + \frac{W_e}{W_g}\right)$ and $k_2 = (0.970)(0.985)(0.995)$.

For the cost attribute, the research, development, engineering, and testing (RDT&E) costs are taken from Ref. [67]. For the area coverage attribute, the cruise speeds and altitudes are interpolations based on the specifications from the baseline concepts in the *Selection* phase class.

Satellites For satellites, the important design elements are the orbit parameters and the imaging capabilities from those orbits. Therefore an orbit propagation model is developed that simulates various satellites in low-Earth orbit (LEO). Given any arbitrary target location around the world, the model determines the coverage characteristics of the satellite or constellation of satellites.

Assuming a spherical Earth, the satellite orbits are propagated using Kepler's Equation (Eq. 4.12). To determine the position and velocity of the satellite over time, the initial-value problem is solved using Lagrange coefficients, F and G (Eqs. 4.13 and 4.14):

$$M - M_0 = \sqrt{\frac{\mu}{a^3}}(t - t_0) = (E - e\sin E) - (E_0 - e\sin E_0) \tag{4.12}$$

$$\mathbf{r} = F\mathbf{r_0} + G\mathbf{v_0} \tag{4.13}$$

$$\mathbf{v} = \dot{F}\mathbf{r_0} + \dot{G}\mathbf{v_0} \tag{4.14}$$

where
$$F = 1 - \tfrac{r}{p}(1 - \cos\theta) \qquad G = \tfrac{rr_0}{\sqrt{\mu p}}\sin\theta$$
$$\dot{F} = \tfrac{\sqrt{\mu}}{r_0 p}\left[\sigma_0(1 - \cos\theta - \sqrt{p}\sin\theta\right] \qquad \dot{G} = 1 - \tfrac{r_0}{p}(1 - \cos\theta)$$

and M is the mean anomaly, $\mu = GM$ is the gravitational parameter, a is the semi-major axis, $t - t_0$ is the time, E is the eccentric anomaly, e is the eccentricity, $\mathbf{r}$ is the position vector, $\mathbf{v}$ is the velocity vector, p is the semi-latus rectum, $\theta = f - f_0$ is the difference in true anomalies, and $\sigma_0 = \dfrac{\mathbf{r_0} \cdot \mathbf{v_0}}{\sqrt{\mu}}$ [3, 2].

To calculate the coverage statistics, the geometry of Figure 4-15 is used to derive Eqs. 4.15–4.19,

$$\rho + \lambda_0 = \pi/2 \tag{4.15}$$

$$\sin\rho = \cos\lambda_0 = \frac{R_\oplus}{R_\oplus + H} \tag{4.16}$$

$$\tan\eta = \frac{\sin\rho\sin\lambda}{1 - \sin\rho\cos\lambda} \tag{4.17}$$

$$\eta + \lambda + \varepsilon = \pi/2 \tag{4.18}$$

$$R_s = R_\oplus\frac{\sin\lambda}{\sin\eta} \tag{4.19}$$

where ρ is the Earth angular radius, λ_0 is the Earth central angle, $R_\oplus$ is the radius of the

Earth, H is the altitude above Earth's surface, η is the nadir angle, ε is the elevation angle, and R_s is the slant range [47].

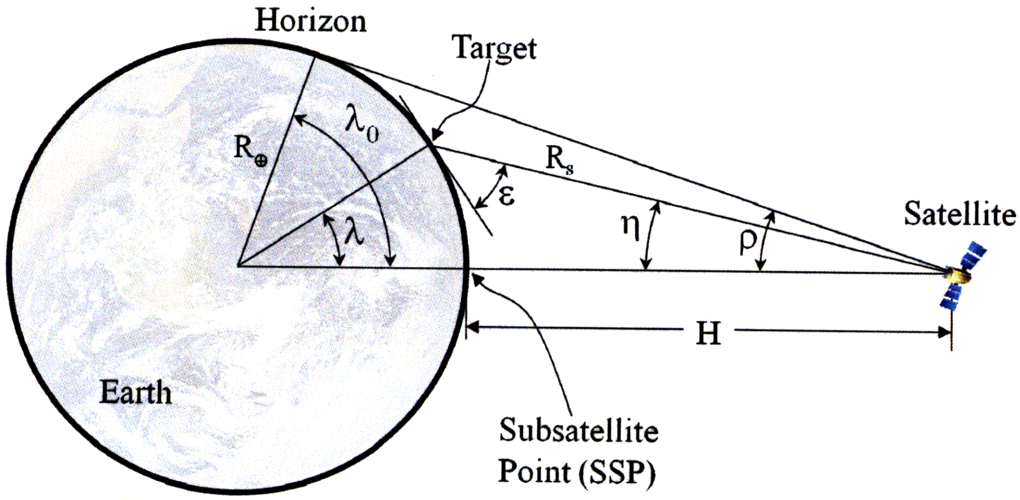

Figure 4-15: Geometry for Viewing of Earth from Space [47]

The ground resolution is calculated according to the diffraction limit of the optics payload, given in Eq. 4.20,

$$\Delta x = 2.44H\frac{\lambda_{obs}}{D}\frac{1}{\sin\varepsilon} \tag{4.20}$$

where λ_{obs} is the observing wavelength and D is the aperture diameter, and $1/\sin\varepsilon$ is the correction for off-nadir angles.

Figure 4-16 shows an example of a simulated orbit. The blue trace in the left graphic is the inertial orbit, and the pink dotted trace is the ground track. The graphic on the right shows the first two orbits on a 2-D projected ground track.

Mission Timeline Next an overall mission timeline is constructed in which both satellite and aircraft concepts perform in the mission scenario. This allows the designer to assess not only aircraft concepts and satellite concepts, but also their combined implementation. With growing interest in designing collaborative systems, sometimes referred to as systems of systems [50], this mission timeline model explores any synergistic behavior that emerges when two different concepts work independently toward a common goal.

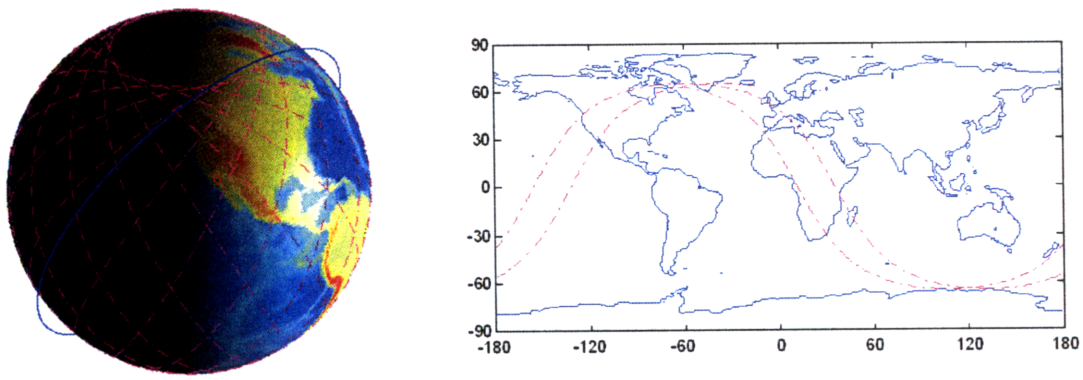

Figure 4-16: ORS Orbit Model Example

The mission timeline model combines the satellite orbit coverage information with the coverage performed by the aircraft. It can include any number of satellites and any number of aircraft. As it steps through time, it calculates the area covered by each satellite and aircraft. The satellite coverage is tracked whenever it meets a pre-defined ground resolution threshold. As discussed in the **Aircraft** modeling section, the aircraft mission consists of flying from the airfield to the disaster area, loitering at the area for a time determined by its endurance, and then returning to the airfield. It then repeats this sequence to continue covering the area of interest. A period of night is included in the mission timeline when neither satellites nor aircraft are able to obtain coverage data. This feature is also amenable to simulating poor visibility due to weather conditions.

To average out transient effects, the missions are all run for 24 hours of simulated time. The three attributes are calculated with the mission timeline model. The area covered adds all the area observed by each vehicle in the system. The resolution is averaged over time and across vehicles. The cost of the system is the sum of the costs of the vehicles comprising the system.

An example mission timeline is shown in Figure 4-17. (It is shown over 36 hours to better visualize the day/night effects.) Time is along the horizontal axis. The left vertical axis measures area coverage rate and the right shows the total area covered over time. The colored lines show the area coverage rates for the aircraft and satellite concepts.

The aircraft concepts have zero coverage at the start since they have not reached the

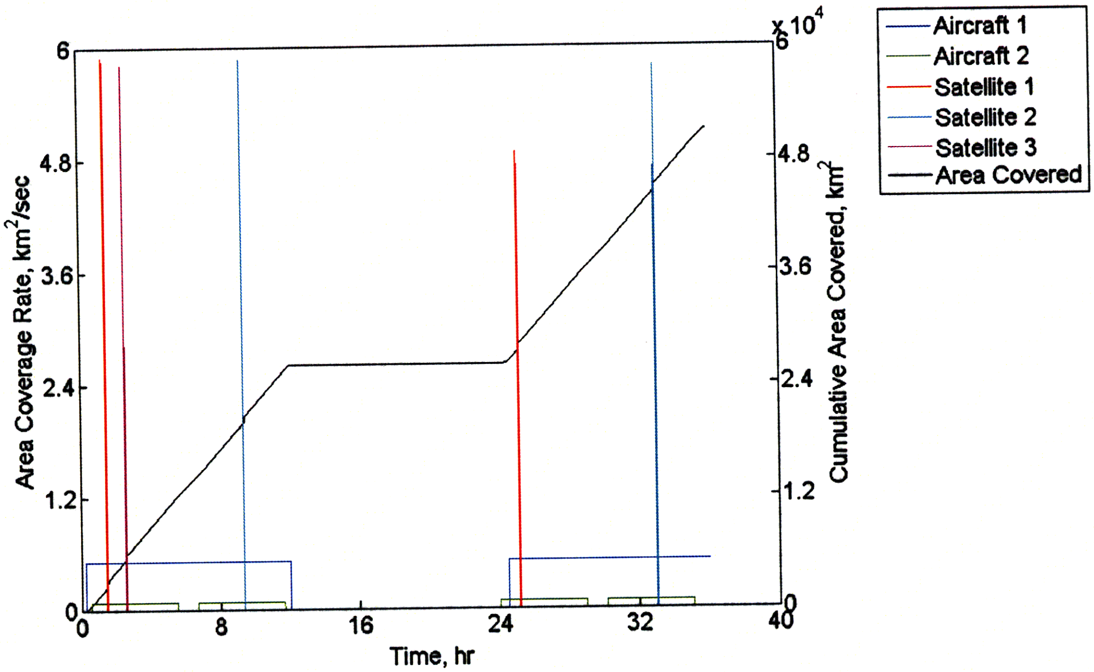

Figure 4-17: ORS *Refinement* Example Mission Timeline for Aircraft and Satellite Models

area of interest yet. Once they do, their coverage rate is constant since they are loitering over the area and collecting data. When they reach their endurance limit, they return to the airfield. This mission is then repeated. Each aircraft's coverage rate pattern therefore appears as a series of step functions. For the different aircraft designs, the coverage rates, endurance times, and response times vary. This is seen by the difference in the two aircraft lines in the example mission timeline.

The satellite coverage is much more intermittent than that of the aircraft, but their coverage rates are higher. The satellite coverage is zero until it views the AOI within the resolution threshold. When this threshold is initially met, the coverage rate is high because of the large, off-nadir footprint area. As the sub-satellite point nears the area of interest, the ground resolution improves and the area coverage rate decreases. In fact, what looks like a vertical line in the figure is actually a steep sinusoid curve that is flanked by two vertical lines. The coverage rate jumps from zero to its maximum, decreases as it nears its closest approach to the area of interest, increases again as it passes the AOI, and then drops back to zero when it no longer meets the resolution threshold.

The black line represents the total area covered over time. For the times when only the aircraft are gathering data on the area, the line has a constant slope. Whenever an aircraft arrives at or leaves the area of interest, the slope changes. The largest changes in slope occur when a satellite passes within range of the area. The flat portions of the black curve indicate where the night period occurs.

ORS Refinement: Decision Analysis

Because of the more complicated analysis in the *Refinement* phase class, two approaches are taken for analyzing the results. They incorporate both of the main design tools used in this thesis. First the design space is explored using a Design of Experiments full-factorial approach. Because the analysis is quite different from and more detailed than that of the *Selection* phase class, this examination of the trade space better informs how to focus the PSO analysis, as well as being able to verify the PSO results.

Design of Experiments Two full-factorial experiments are conducted, one for the aircraft designs and one for the satellite designs. The trade spaces for the aircraft designs are shown in Figure 4-18 Since there are three attributes, the results are displayed as three 2-dimensional trade spaces — area covered versus cost, average resolution versus cost, and average resolution versus area covered. The rows of plots show the trade spaces highlighted by different design alternatives.

In the area-versus-cost and resolution-versus-cost plots, there are distinct cost levels that correspond to the combinations of the three classes of aircraft. At the low end of the cost spectrum are the small UAV-class aircraft, and at the right are the combinations of two large UAV-class aircraft. The resolution-versus-cost trade space tends to favor the smallest designs, but the other two trade spaces have Pareto sets that include different concepts. In the plots with the payload aperture diameter highlighted, there is not much differentiation across the trade space. The same is true when looking at the wavelength choices. The instrument payload does not appear to be a significant driver in the design.

The trade spaces in Figure 4-18 have a cluster of designs near the origin in each

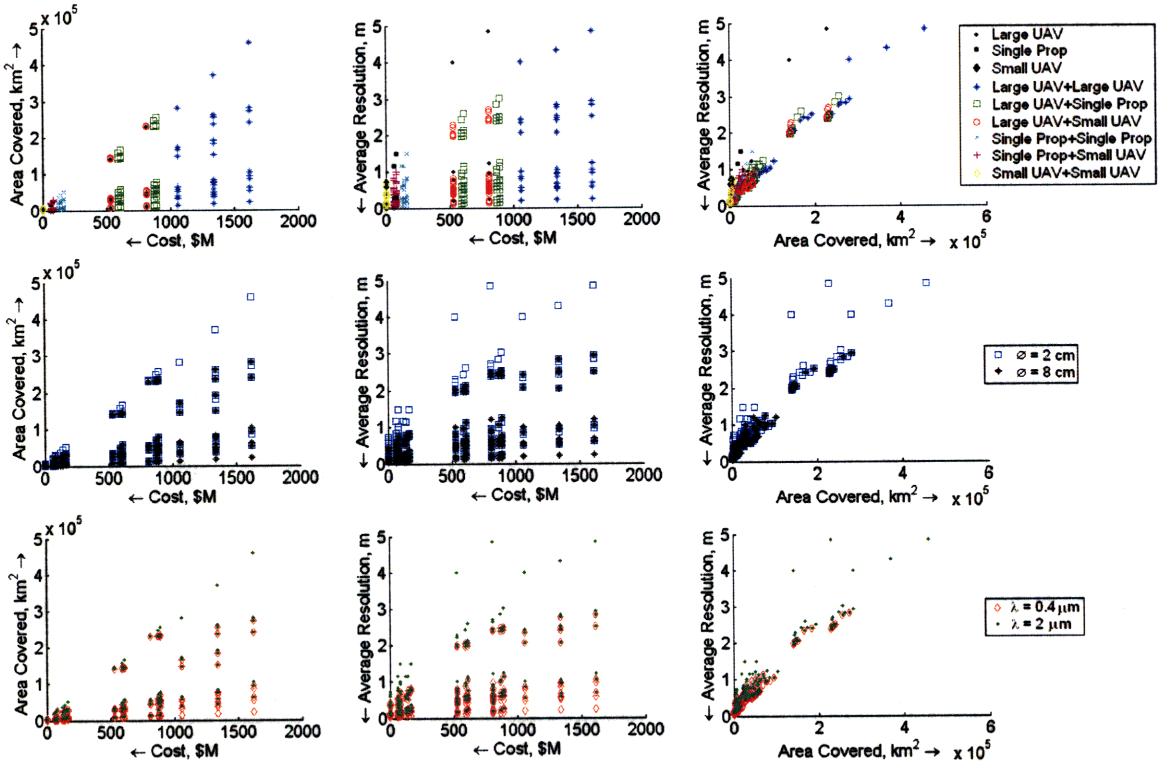

Figure 4-18: ORS *Refinement* Aircraft DoE Trade Spaces

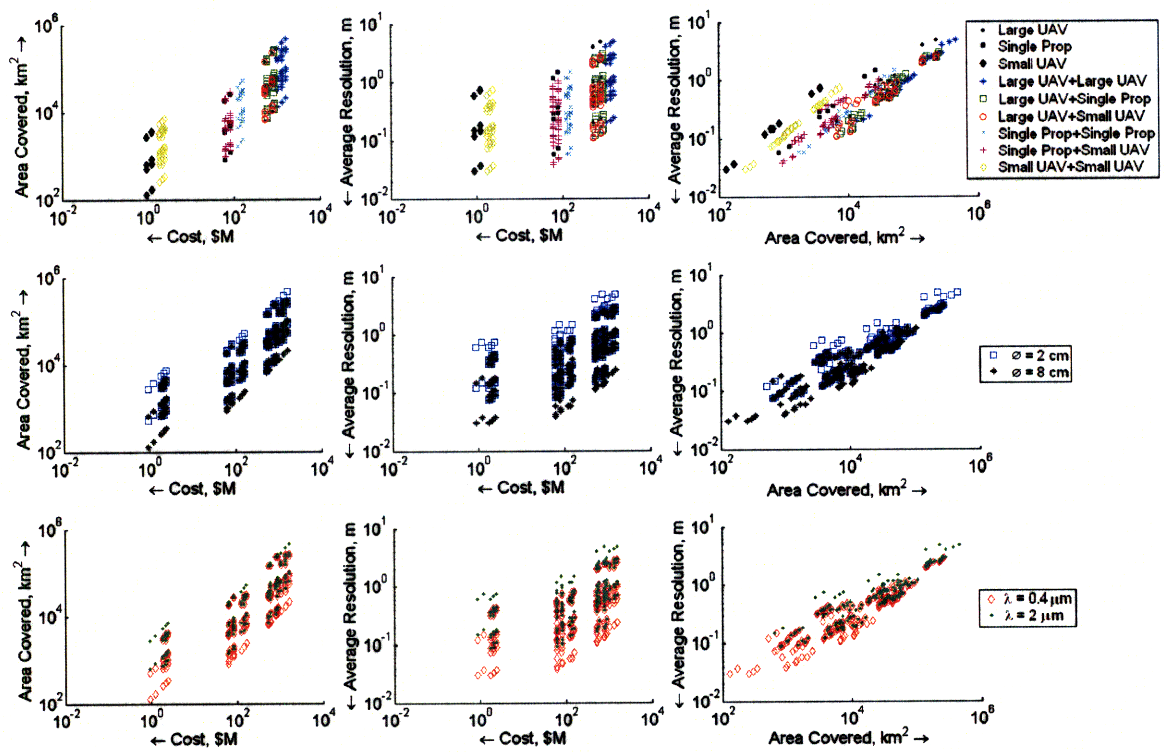

Figure 4-19: ORS *Refinement* Aircraft DoE Trade Spaces with Logarithmic Axes

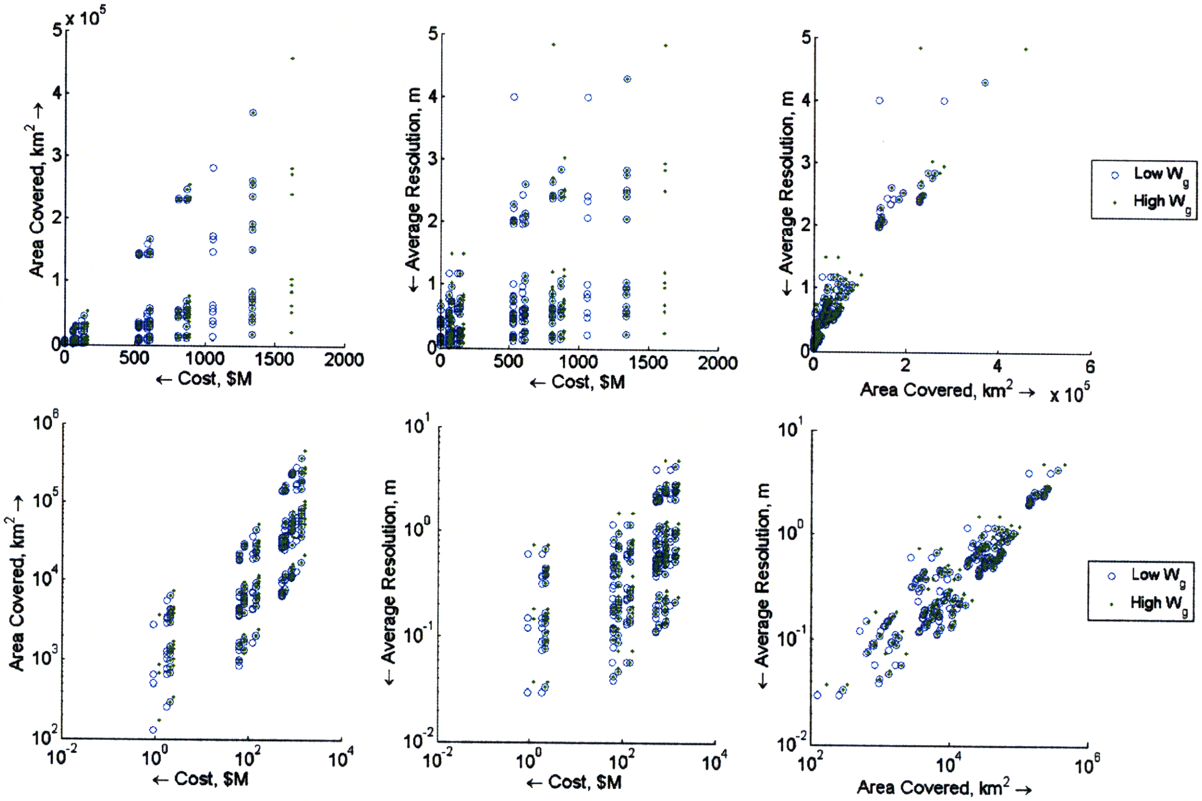

Figure 4-20: ORS *Refinement* Aircraft Gross Weight Trade Space

plot. Figure 4-19 shows the same trade spaces with logarithmic axes. This redistributes the points visually and clarifies the details for the smaller values of the data. It is much easier to discern the locations of certain designs in the trade spaces, especially for the different aircraft classes.

The trade spaces highlighted by takeoff gross weight, W_g, are shown in Figure 4-20 with both normal and logarithmic axes. There is a stratification that appears within each aircraft class combination; that is, the groupings of three sets of designs for each combination of aircraft classes (best seen as the horizontally separate colors in the area-versus-cost plot of Figure 4-18) are identified as resulting from the three different options in selecting W_g.

Next are the trade spaces for the satellites, shown in Figures 4-21 and 4-22, with the same plotting format for the attributes as used in the aircraft plots.

In the top row, where the designs are highlighted by the number of orbital planes and the number of satellites in each plane, there is a distribution of designs along the Pareto

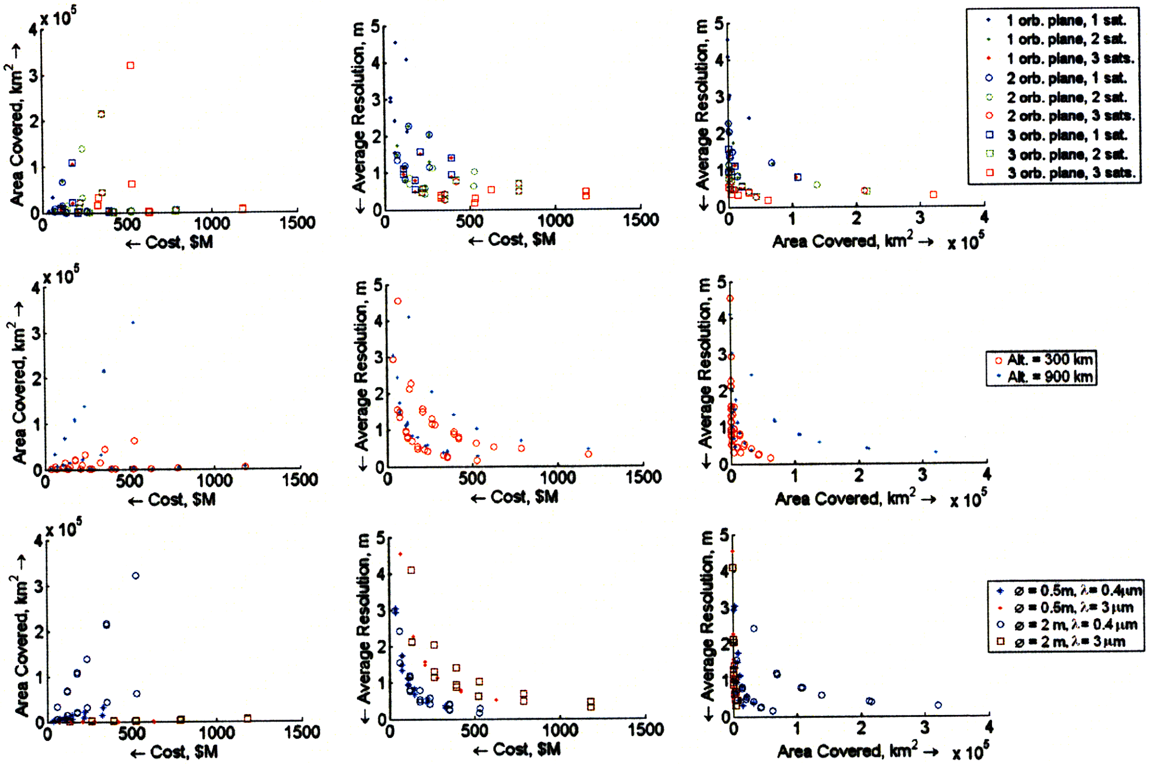

Figure 4-21: ORS *Refinement* Satellite DoE Trade Spaces

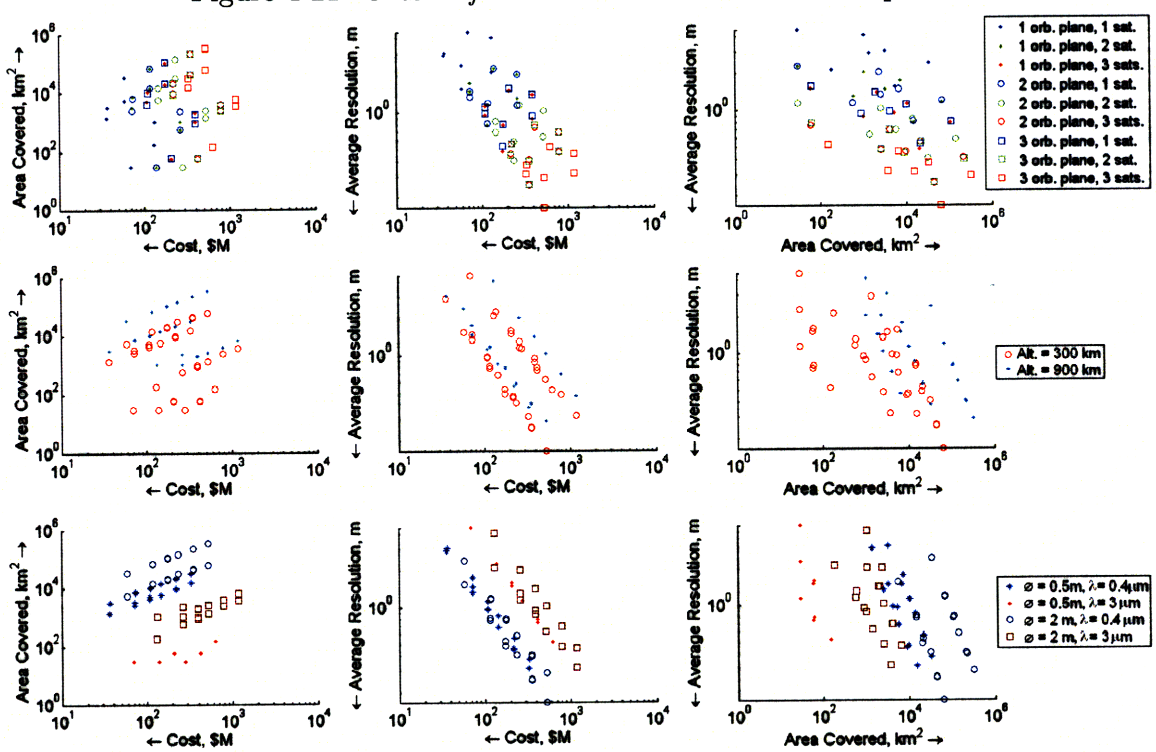

Figure 4-22: ORS *Refinement* Satellite DoE Trade Spaces with Logarithmic Axes

front that represent the tradeoffs associated with obtaining more satellite coverage. In the plot for average resolution versus cost, for instance, the single-satellite designs occupy the low-cost, poor resolution end of the Pareto front, and the three orbital plane, three satellites per plane designs cost more but obtain better average resolution. Highlighting by altitude indicates that the three attributes conflict in this design variable — the area-versus-cost plot favors high altitudes, the resolution-versus-cost favors lower altitudes, and the resolution-versus-cost is a tradeoff between the two. The same conflict in attributes occurs for the optics payload design variables of aperture diameter and sensing wavelength.

The log-log plots of Figure 4-22 help in clearly separating the trade spaces visually. Interpreting the data contained in these plots leads to an informed understanding of how the design variables interact with one another and with the attributes. An example is how the optics payload affects the resolution and area covered. One might initially think IR wavelengths could cover more area than visual wavelengths because the ground area corresponding to each pixel on the focal plane would be larger. However, having the resolution threshold in the model restricts the time that the IR designs can cover area, leading to less overall coverage. A similar effect occurs in the trade space for average resolution versus area, highlighted by altitude. The lower altitudes do not completely dominate the average resolution attribute because the higher satellite designs with good optics payloads spend more time over the area of interest, thereby lowering the average resolution to be on par with the lower altitude designs.

The next way to investigate the effects of the design variables on the trade spaces is by using ANOVA. Figure 4-23 shows the ANOVA results for both the aircraft and satellite Design of Experiments runs. The relative influences of the design variables are given for each of the attributes. Looking at the aircraft results, the decisions of aircraft class and takeoff gross weight show the most influence, meaning that these decisions are important in determining where in the trade space the design falls. Less important are the payload characteristics. For the satellite designs, there is more conflict among the attributes. This was mentioned in the previous discussion as well. For area covered, the optics and altitude have a high influence because they affect when the AOI is in view more than the number

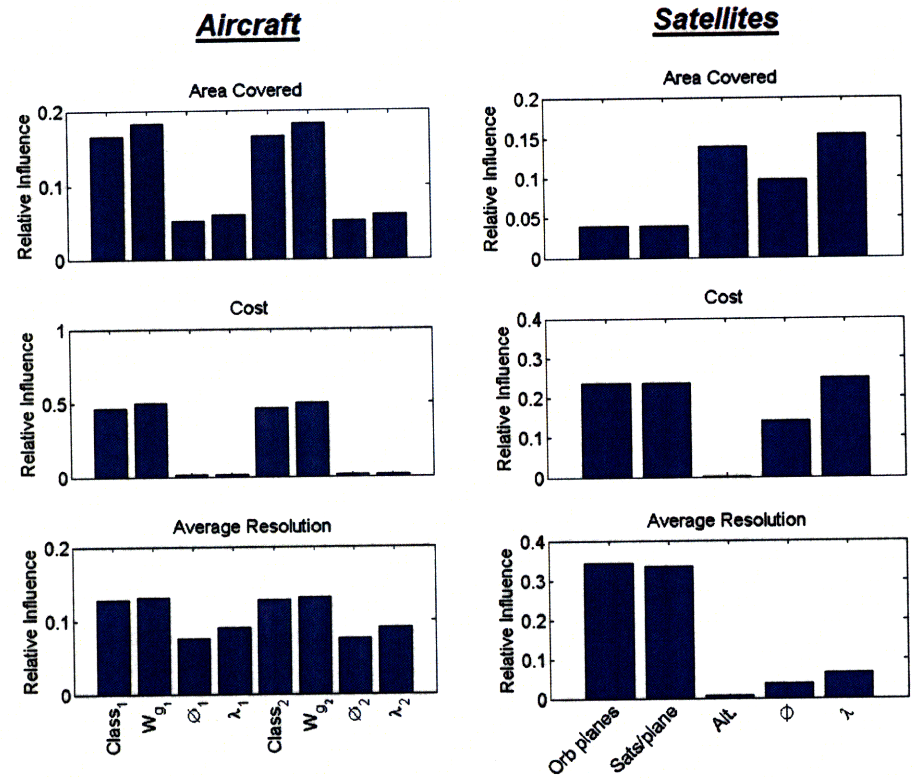

Figure 4-23: ORS *Refinement* ANOVA Results for Aircraft and Satellites

of satellites in orbit. The relative influence rankings for the cost attribute reveal the lack of importance in the altitude decision. For the average resolution results, the number of satellites takes a higher importance than the optics. This means that there is large variation in the resolution attribute for different numbers of satellites and is explained by the fact that having more satellites increases the likelihood that one or more of them will orbit more directly overhead than at a low elevation angle.

Particle Swarm Optimization Now with the DoE results as a reference for choosing a good PSO strategy, the salient aspect of this problem is ready to be tested. For the PSO implementation, the multi-concept mission timeline scenario is implemented. The design objective is to identify which combinations of aircraft and satellites together obtain the best attribute results.

The design vector for the PSO algorithm is shown in Table 4.13. Besides analyzing multiple concepts in the same mission timeline, there are two other differences in the design vector from the DoE analyses. Instead of having either one two aircraft designs, where the two could be the same, now there can be 0 or 1 of each class, making the total number range from zero to three, without duplicates allowed. In addition, the non-integer variables take on continuous values for PSO.

Table 4.13: ORS *Refinement* PSO Design Vector

Concept	Design Variable	Value
Large UAV Class	Number of Aircraft	0 or 1
	Payload Aperture Diameter, cm	0 or 1
	Payload Sensing Wavelength, μm	0.4 to 2
	Takeoff Gross Weight	low to high
Small UAV Class	Number of Aircraft	0 or 1
	Payload Aperture Diameter, cm	0 or 1
	Payload Sensing Wavelength, μm	0.4 to 2
	Takeoff Gross Weight	low to high
Single Prop Class	Number of Aircraft	0 or 1
	Payload Aperture Diameter, cm	0 or 1
	Payload Sensing Wavelength, μm	0.4 to 2
	Takeoff Gross Weight	low to high
Satellites	Number of Orbital Planes	1, 2 or 3
	Satellites per Orbital Plane	1, 2 or 3
	Altitude, km	300 to 900
	Payload Aperture Diameter, m	0.5 to 2
	Payload Sensing Wavelength, μm	0.4 to 3

Since there are three attributes to attempt to optimize with the PSO algorithm, the following approach is implemented: conduct three 2-objective particle swarm optimization runs, and one 3-objective optimization run. Each run uses a 30-particle population and searches through 20 time step iterations, for a total of 600 design points evaluated. Using a 2.61 GHz AMD Athlon dual core processor and 3 GB of RAM, each run took between 2.5 hours and 3 hours to complete.

The first results examined are the 2-attribute optimization runs of PSO. The plots in Figure 4-24 contain the information from three separate runs, one for each two-attribute run. There is good coverage of the space as well as concentration around the Pareto front for each run, more so in the resolution-versus-cost and resolution-versus-area plots. As a

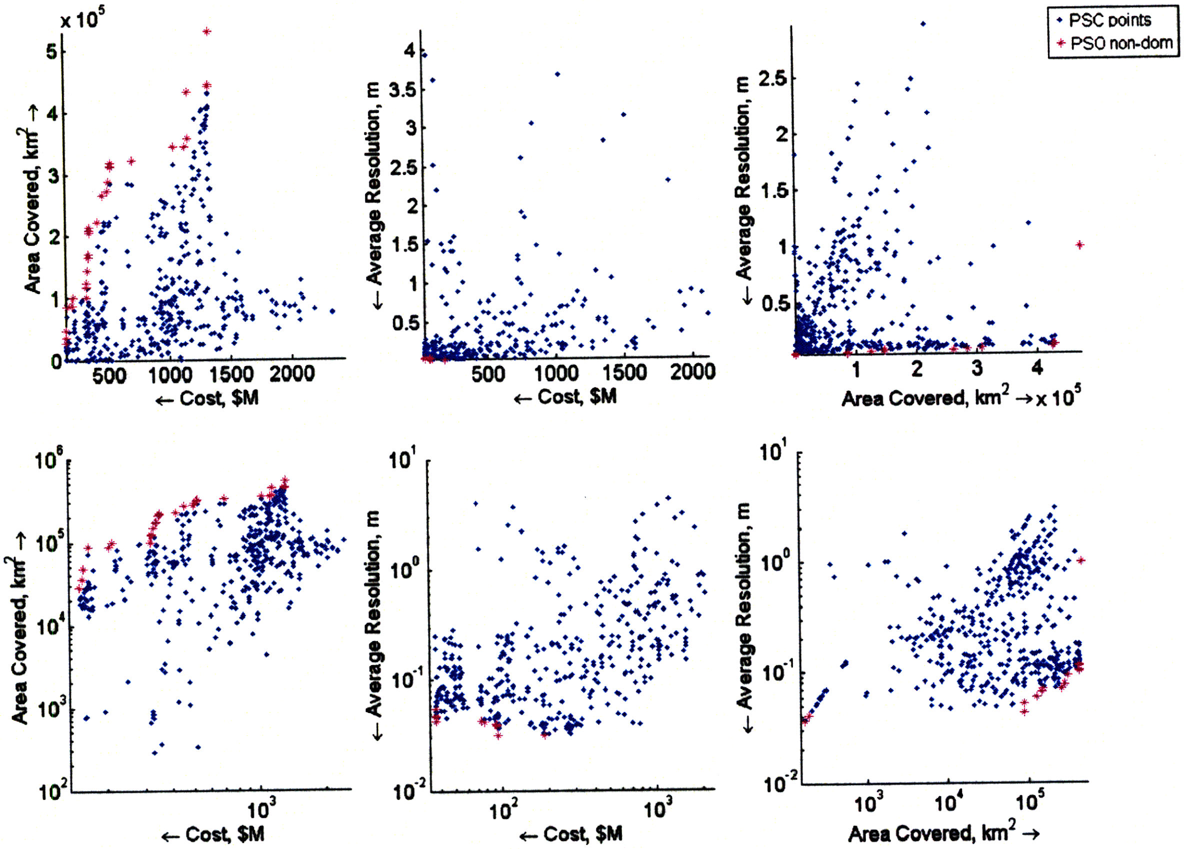

Figure 4-24: ORS *Refinement* 2-Objective PSO Results

comparison to the results from the Design of Experiments approach, Figure 4-25 shows the same PSO data plotted with the data from the DoE explorations. In the area-versus-cost plot, the non-dominated PSO designs are better than nearly all the aircraft designs from the DoE. Just seven of the DoE satellite designs are better than the non-dominated PSO designs. In the resolution-versus-cost plot, the PSO algorithm performed quite well. The particles started converging toward the Pareto front, with many reaching attribute values better than most of the DoE set. There are, however, some very low-cost designs that the PSO algorithm does not appear to have found in its search, best seen in the log-log plot. These low-cost designs are the small UAV-class aircraft. Because of the stochastic nature of the PSO algorithm, if a particle does not find this particular design choice, the optimizer will not know to investigate it further. This PSO run identified the small UAV-class aircraft as a good design, but also had in its design vector a satellite concept. Therefore, the PSO

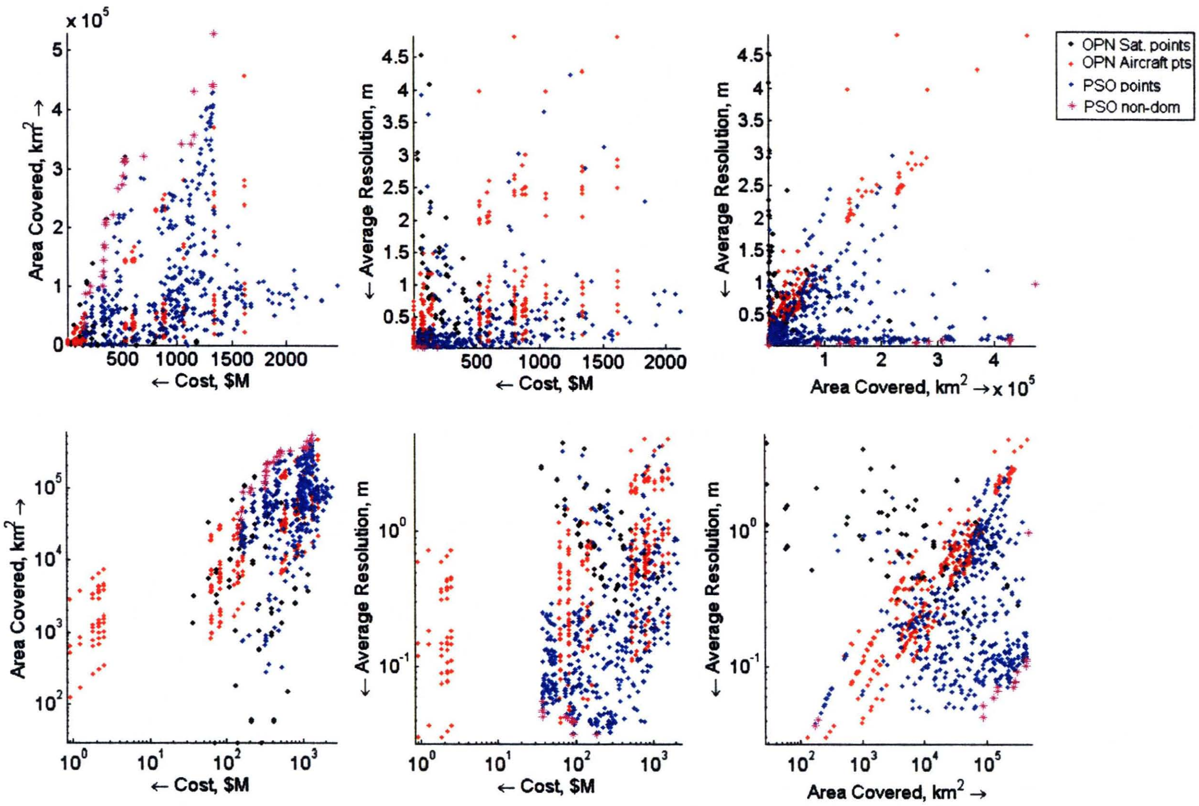

Figure 4-25: ORS *Refinement* 2-Objective PSO Results with Comparisons to DoE results

algorithm never found the design option of only one small UAV-class aircraft, which is why the DoE results have designs that are superior in the resolution-versus-cost plot. For the final two-attribute plot (resolution-versus-area), the PSO algorithm far outperformed the aircraft DoE concepts, and only a few satellite DoE designs are near the PSO Pareto front. This demonstrates the power of using the PSO algorithm to search through the design space.

Next the PSO trade spaces are highlighted by the design decisions. Figure 4-26 shows the trade space for the area-versus-cost PSO run. The left plot reveals which system designs include satellites and which include aircraft. Interestingly there are some satellite-only concepts but no aircraft-only concepts in this trade space. The middle plot differentiates the designs by aircraft alternatives. The small UAV-class aircraft covers much of the Pareto front, with the large UAV and small UAV-large UAV combination making up the rest. There are two clumped regions differentiated in the cost attribute. The collection of designs above the $1 billion point consist mostly of designs with a large UAV-class aircraft, and those below

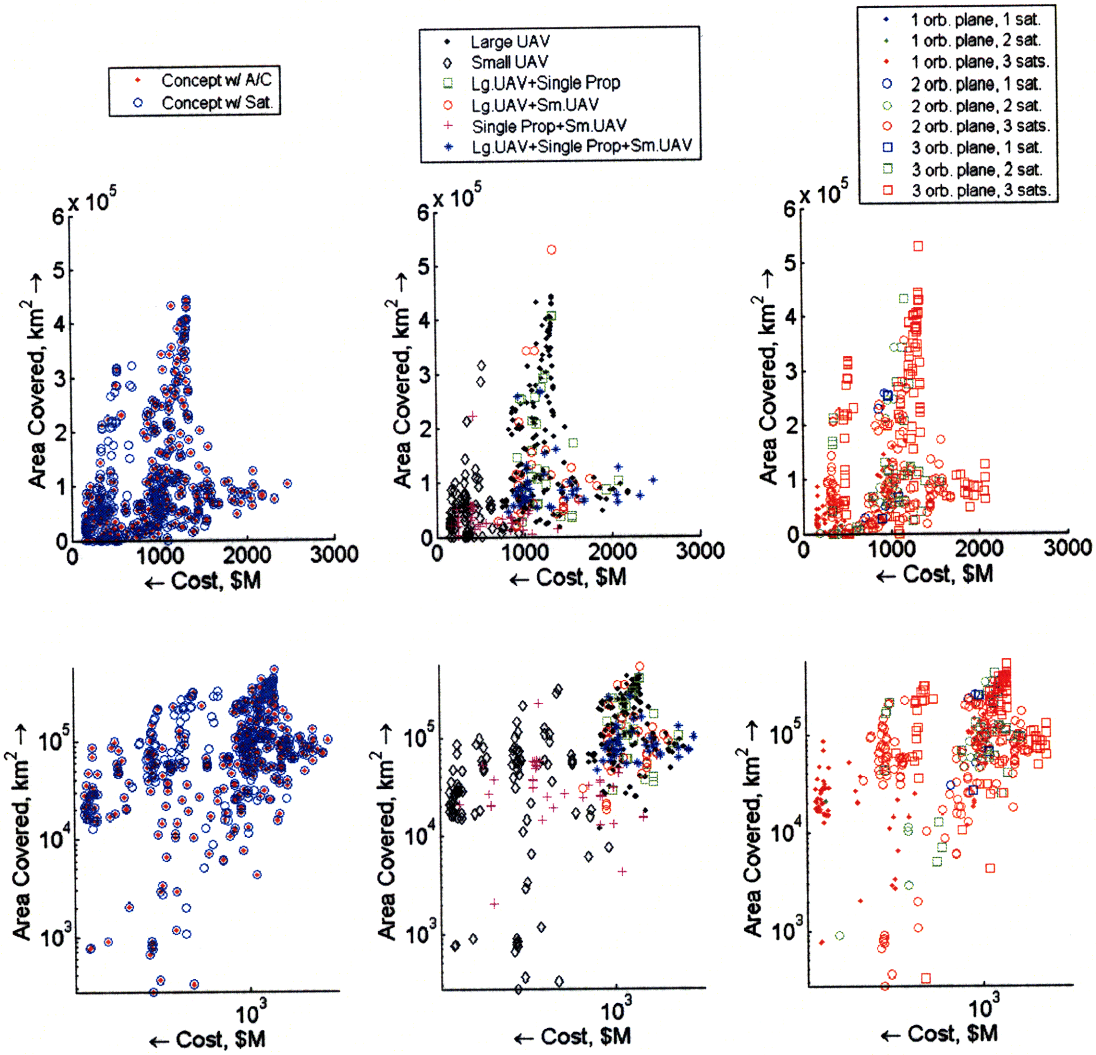

Figure 4-26: ORS *Refinement* 2-Objective PSO Area-versus-Cost Trade Space

do not. Finally, the plot on the right is highlighted by how many satellites are in the design concept. The three orbital plane, three satellites per plane design has the most points in the trade space, indicating the algorithm's preference for that design option.

The resolution-versus-cost trade space is shown in Figure 4-27. Again there are some satellite-only concepts, but not aircraft only. There are a lot of small UAV concepts pushing on the lower-cost/smaller-resolution Pareto front. A good distribution of satellite concepts is shown in the plot on the right, but the best for these attributes are the single orbital plane designs.

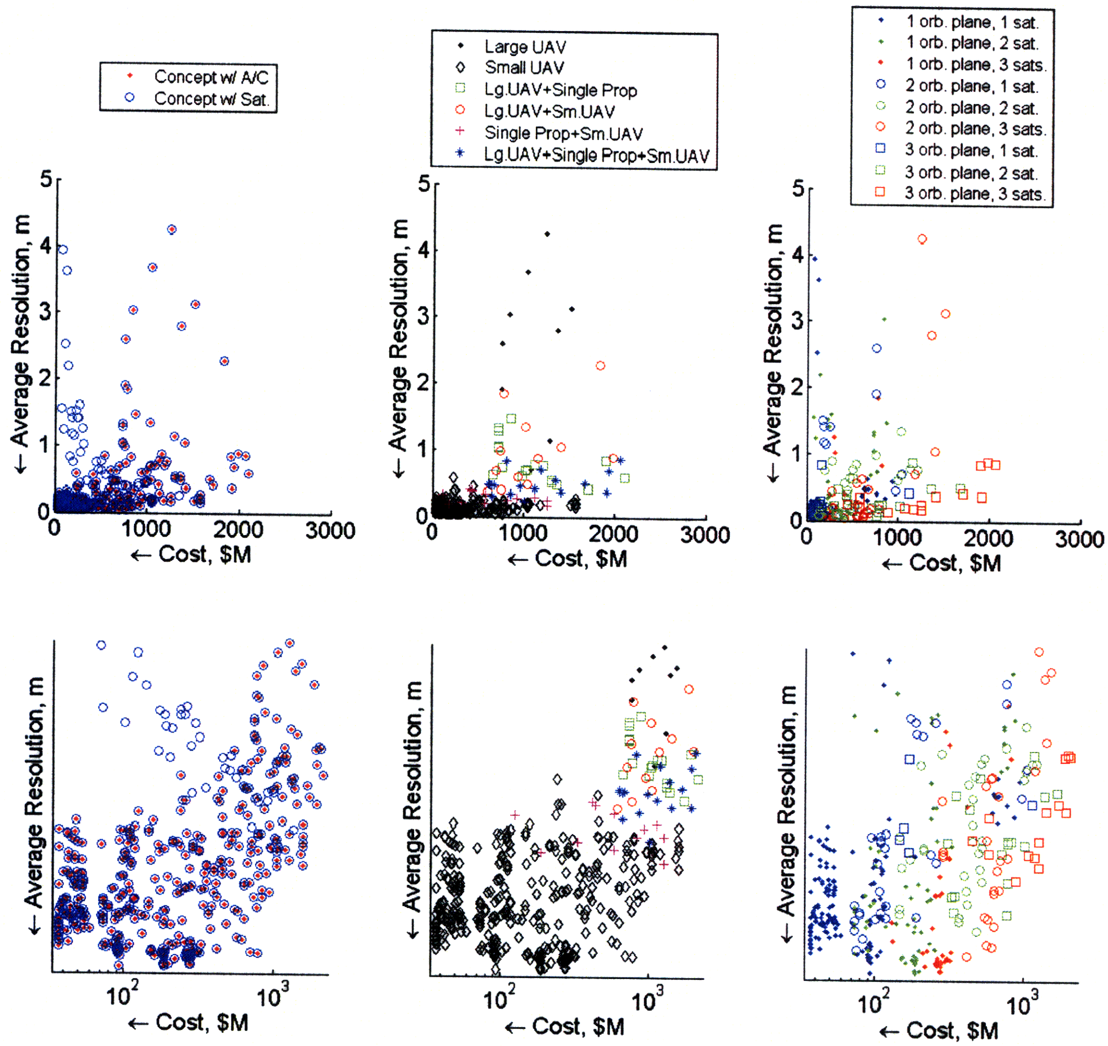

Figure 4-27: ORS *Refinement* 2-Objective PSO Resolution-versus-Cost Trade Space

The last two-attribute PSO run is for the resolution-versus-area trade space (Figure 4-28). In looking at the region around the Pareto front, small UAV-class aircraft and a large number of satellites (3 orbital planes, 3 satellites per plane) work well together to cover a lot of area with good resolution. In the DoE runs, this combination of the two concepts was not possible; the PSO formulation has revealed this interesting insight.

With the two-objective PSO runs as a means of comparison, the three-objective PSO algorithm is implemented. A three-sub-swarm version of the multiobjective particle swarm algorithm is used to optimize all three attributes. In this case, one-third of the population attempts to optimize the attribute of area covered, one-third of the particles search for low-

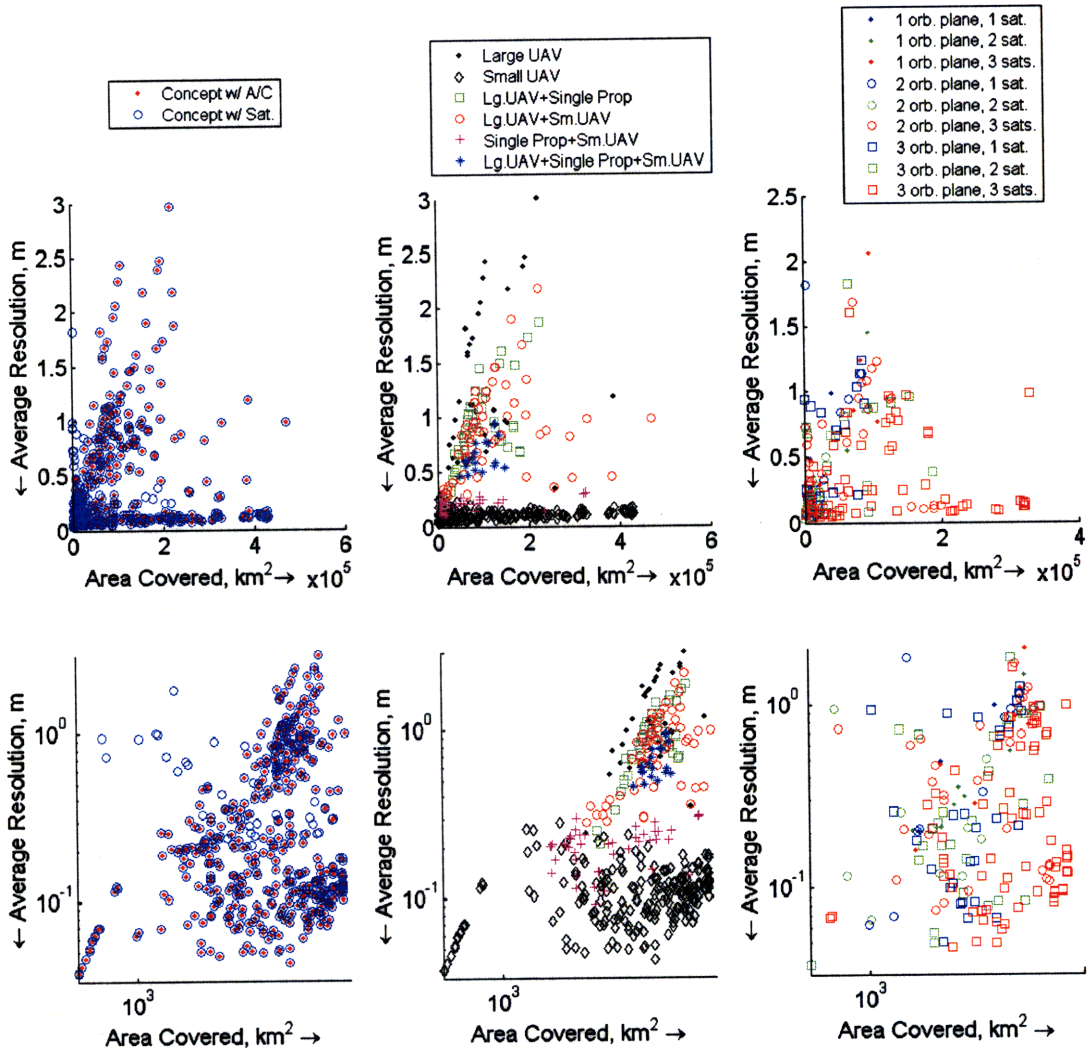

Figure 4-28: ORS *Refinement* 2-Objective PSO Resolution-versus-Area Trade Space

cost designs, and the remaining have the objective of finding the best average resolution. The results are shown in Figure 4-29. The system designs highlighted by a pink star represent the designs that are non-dominated in all three attributes. The two-attribute plots are thus projections from the three-attribute objective space. Each of the 2-D projections of the three-attribute Pareto front shows designs that are even better than what was achieved in the two-attribute PSO runs. This three-objective run has identified quality designs that perform well in all three attributes. In addition, the designs that are non-dominated in all three attributes closely follow the 2-D Pareto fronts as well, which indicates that these designs do not have characteristics that lead to poor performance in any particular attribute.

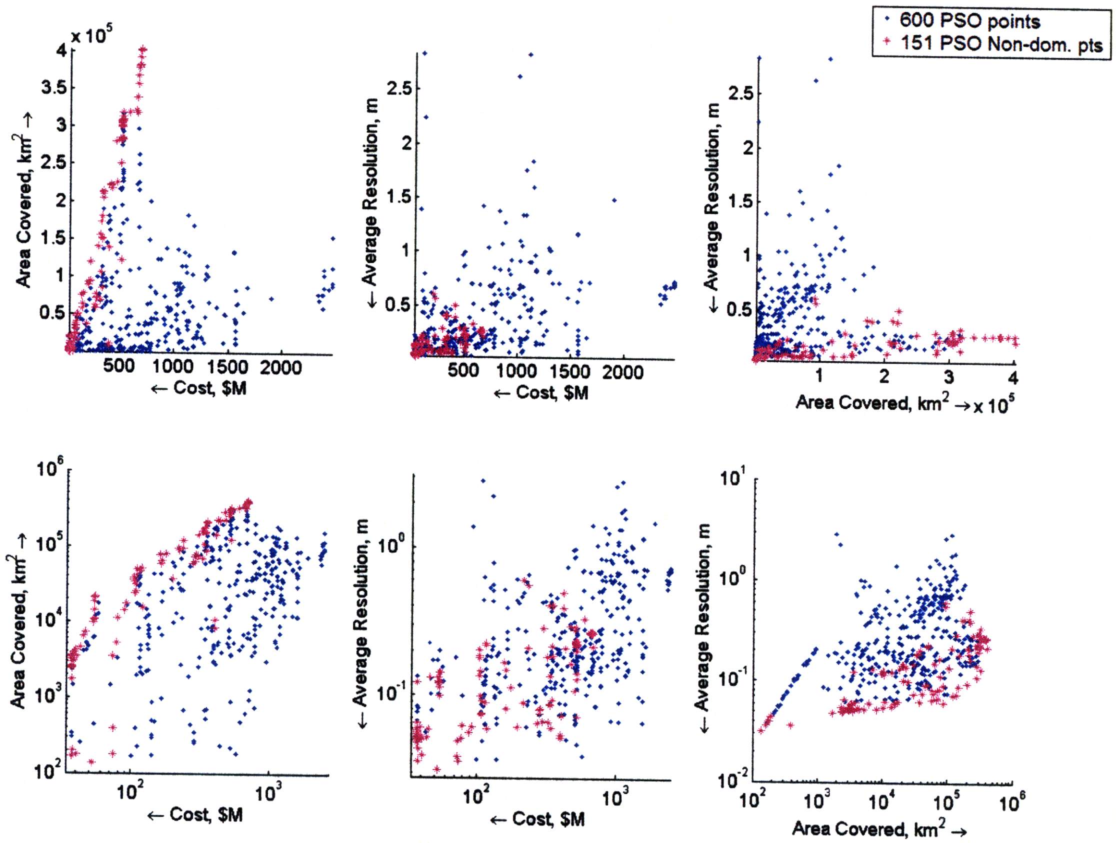

Figure 4-29: ORS *Refinement* 3-Objective PSO Results

The trade spaces highlighting the design decisions are examined next for the three-objective PSO data. Figure 4-30 shows the trade spaces with aircraft designs highlighted, and Figure 4-31 shows the satellite designs. In the aircraft trade spaces, the small UAV again appears to dominate the Pareto fronts, with good performance in all three trade spaces. There is one large UAV-class design that is non-dominated in the area-versus-cost trade space, and just two designs in the resolution-versus-area trade space that are not small UAV-class designs. The optimizer has identified the small UAV-class aircraft as a very good option. For the satellite trade spaces, there are more tradeoffs in the different trade spaces. The area-versus-cost trade space shows four different satellite constellation designs along the Pareto front. The good designs in the resolution-versus-cost trade space are mostly single satellite designs. The resolution-versus-area trade space Pareto front is comprised mostly of three orbital plane designs.

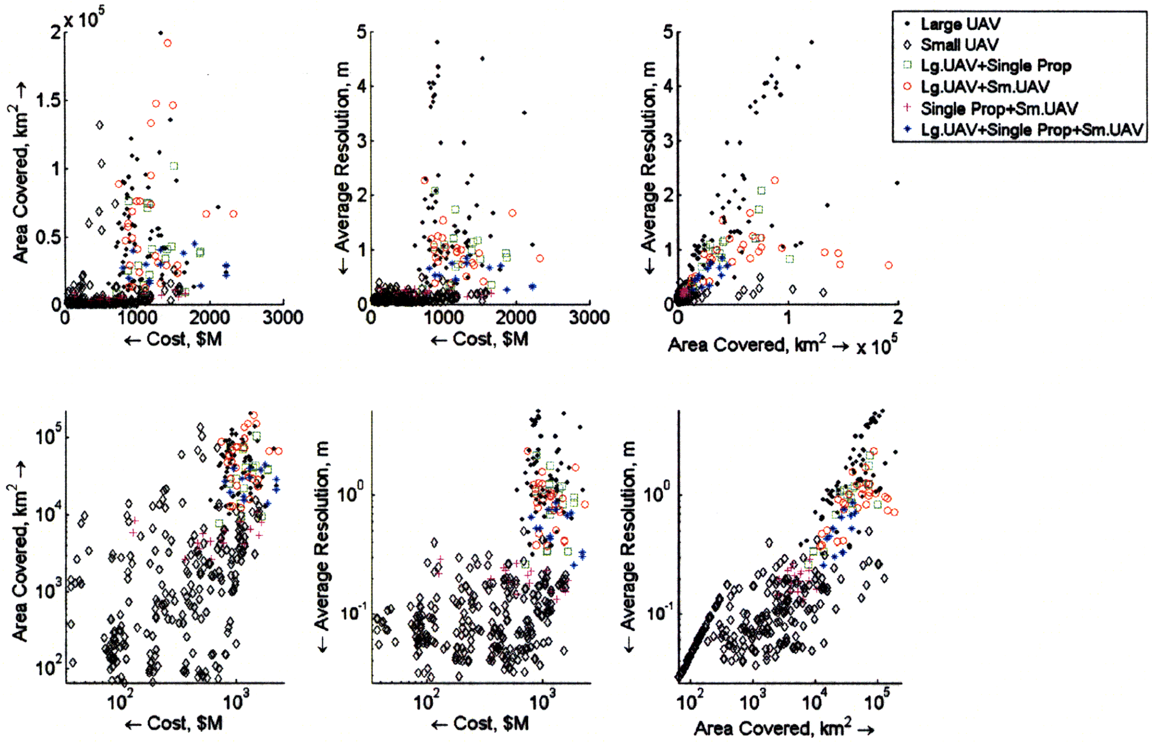

Figure 4-30: ORS *Refinement* 3-Objective PSO Aircraft Trade Spaces

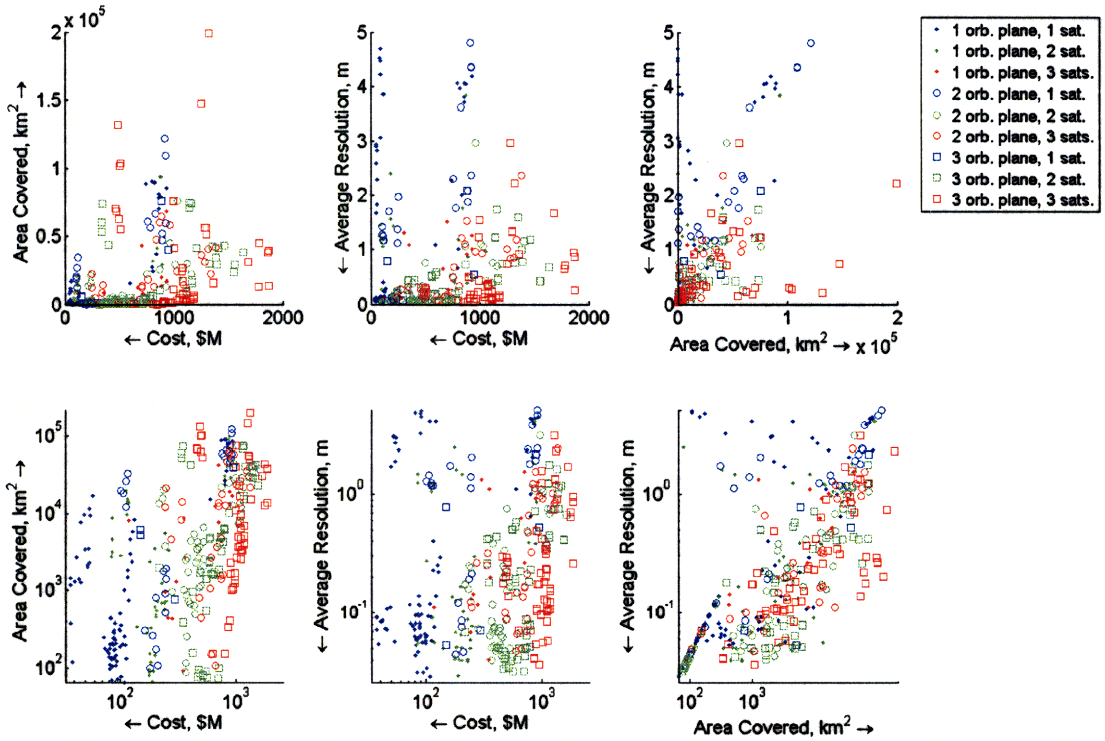

Figure 4-31: ORS *Refinement* 3-Objective PSO Satellite Trade Spaces

With this three-attribute data set, it is of interest to try to visualize the 3-dimensional trade space. Figure 4-32 shows the PSO designs as they are represented in the 3-D trade space. The Pareto front (here a surface) is illustrated at the extreme portion of the trade space where the cost is the lowest, the resolution is the lowest, and the coverage is the highest. The PSO designs in 3-D space are projected onto the three 2-D axis planes to help visualize their location in the trade space. These projections reproduce the two-dimensional trade spaces. The bottom plot is displayed with logarithmic axes.

Because it is useful during concept design to not only pay close attention to designs on the Pareto front, but also those nearby, the PSO points can be probed further in the 3-D attribute space. A way to get at this information is to look at the behavior of successive Pareto fronts. This is accomplished by the following iterative scheme. First find the original Pareto front. Remove all of those non-dominated designs from the data set. Again find the Pareto front from this smaller set. Repeating in this manner allows successive Pareto fronts to be identified. A visual portrayal of this is shown in Figure 4-33 with logarithmic axes for ease of visualization. It can be seen that the Pareto surfaces exhibit nesting, which is expected since the non-dominated designs are always better than or equal to the others for all of the objectives. There are some regions that protrude more than others. If this is the true behavior in that region of the trade space, that information can be used to choose designs that lie in the appropriate "crevice" of the Pareto front. It could, though, be an artifact of the sampling in that region, or an approximation of the true Pareto front. Analysis of additional designs in that region would be necessary to resolve this. The information gained by identifying the design variables that affect that region of the trade space can be used to explore that portion of the Pareto front in more depth.

The last piece of information that becomes especially important in the realm of uncertainty is the sensitivity of designs in the trade space. It was evident in examining the two- and three-attribute PSO trade spaces that the small UAV-class aircraft performs well. This concept is thus chosen for examination in the sensitivity study. After determining what design is of interest in the trade space, the next thing to assess is the important design parameters. From the ANOVA results from the aircraft DoE (Figure 4-23), one of the design

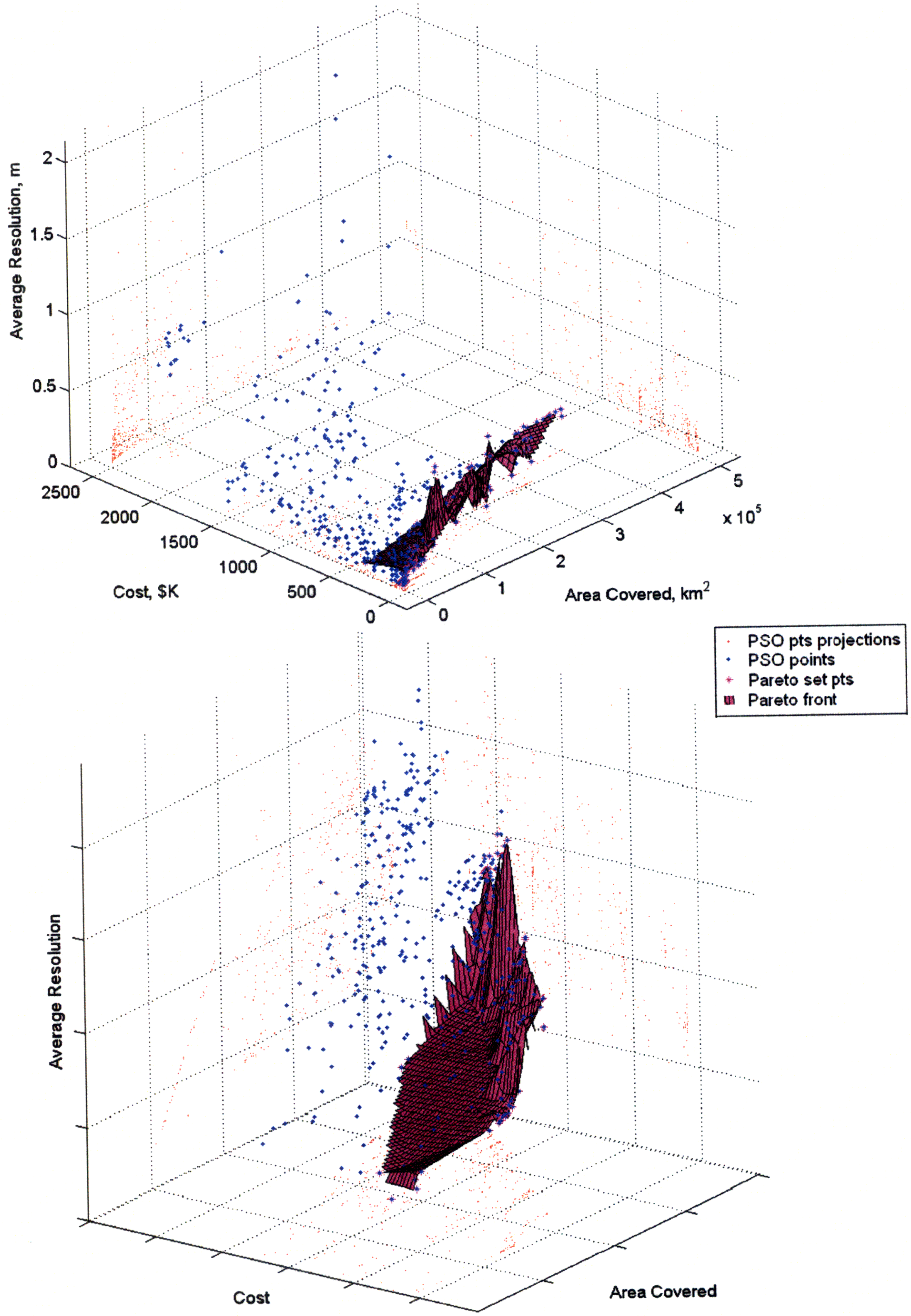

Figure 4-32: ORS *Refinement* 3-Objective PSO 3-D Attribute Space with Pareto surface and 2-D Trade Space Projections, with the Bottom Plot Displayed in Log-Log Space

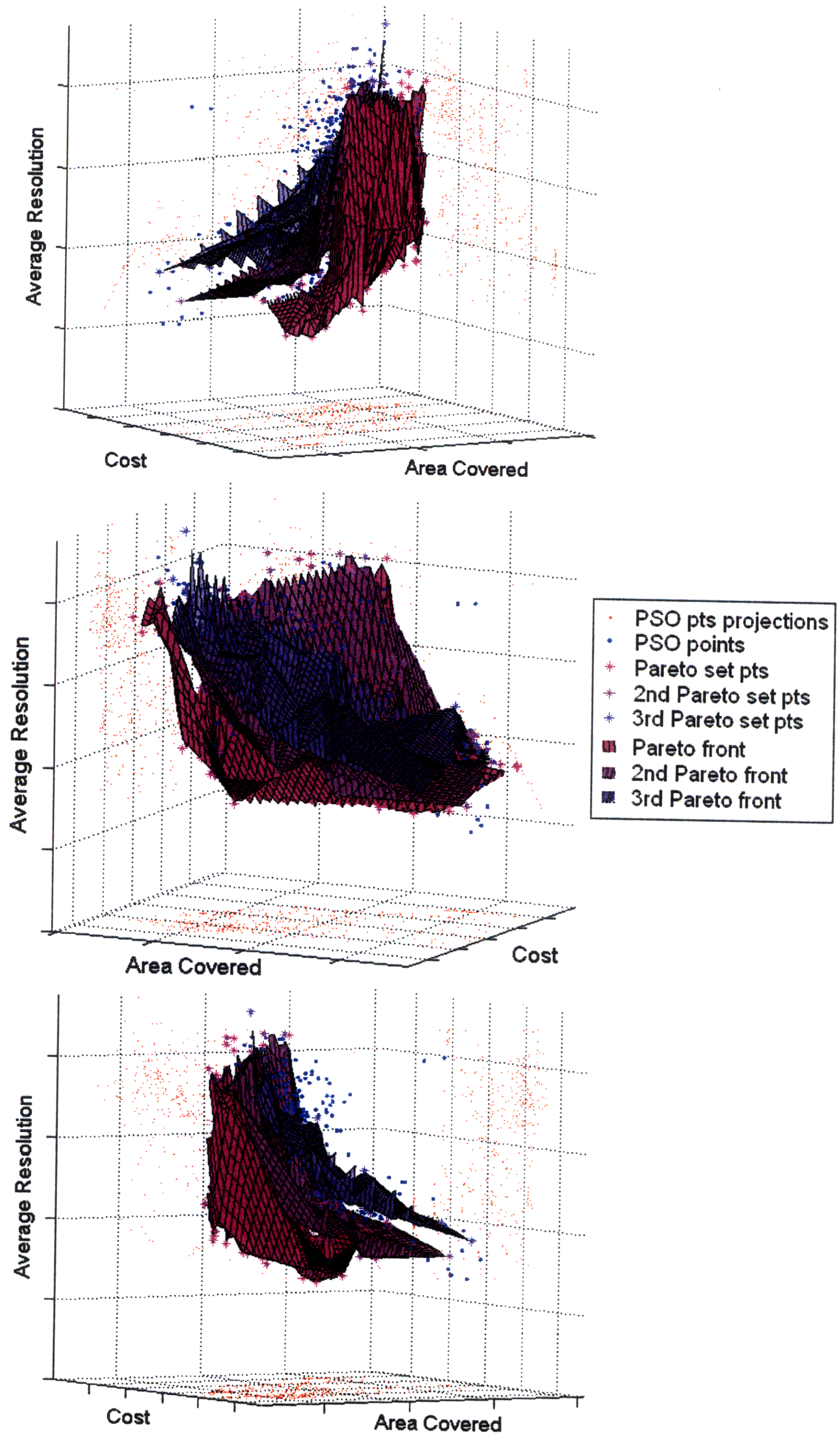

Figure 4-33: Three Views of ORS *Refinement* 3-Objective PSO 3-D Attribute Space with Pareto surface and 2-D Trade Space Projections with Logarithmic Axes

variables with the largest relative influence is the takeoff gross weight. Therefore, to see how the designs are influenced by changes in the design variables, a sensitivity study is conducted by varying the takeoff gross weight by 20%. Instead of using the Monte Carlo simulation method as in the ALSR problem, the ORS sensitivity study is done using a Design of Experiments approach. This is necessary because the run-time of the mission timeline and concept models prohibit a large number of random variations. Similar information can be obtained by using fewer runs that are organized in an experimental design. The sensitivity study is conducted on the designs that are non-dominated in all three attributes.

Figure 4-34 shows the sensitivity results, where the boxes show one standard deviation in each attribute direction. These results indicate that the small UAV-class aircraft are not greatly affected by the change in takeoff gross weight. The variation can be seen more clearly

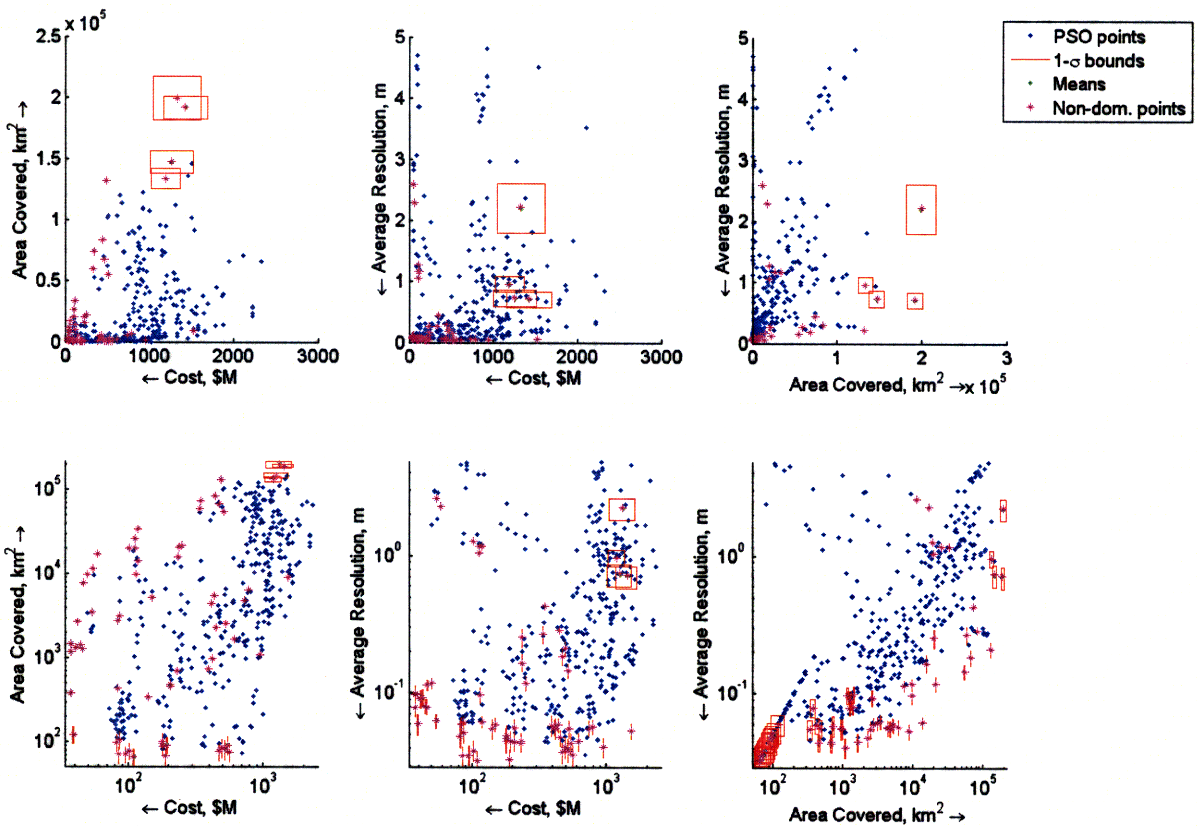

Figure 4-34: ORS *Refinement* DoE sensitivity analysis conducted for designs along Pareto front with a 20% variation in W_g. The boxes represent one standard deviation from the mean.

by the boxes in the log-log plots, but they are quite small in comparison to the larger boxes of the concepts toward the upper portions of the plots. This is due to the fact that the 20% change in the smaller aircraft designs has less effect than the same percentage change in larger designs. With this information the designer can feel confident that choosing the small UAV-class aircraft is likely to achieve good performance while not being extremely susceptible to changes in one of the more important design variables.

4.2.4 ORS Conclusions

The ORS case study example has demonstrated the complete scope of the Engineering Framework, all three phase classes and each of the four concept design tools. Not utilized in the ALSR example, the *Exploration* methods proved very useful for working through an ill-defined concept design problem. The method of stakeholder analysis provided justification for the system attributes, and Pugh analysis helped make logical comparisons to systematically reduce the number of concepts considered. The OPN analysis on this smaller set of concept alternatives constituted the *Selection* evaluation. Using existing baseline systems for performance benchmarking, low-order models were used to determine which aircraft and satellite concepts to examine in more detail. The last design iteration, *Refinement*, analyzed aircraft classes and satellite constellation designs in a disaster scenario mission timeline.

Many interesting insights were obtained through the trade space exploration and multiobjective optimization techniques. The aircraft class and size are found to be the most important design variables for aircraft, more so than the particular payload. For satellites, the optics payload plays a significant role in its ability to monitor the disaster site. Among the decisions for the number of satellites in the constellation and their altitude, the importance of each of these design variables varies depending on the particular attribute, and tradeoffs exist between each particular set of options. Combining both aircraft and satellite concepts in the mission timeline revealed an interesting combination to a system architecture-level design solution — a small UAV-class aircraft with a three-orbital-plane satellite constellation. The small UAV-class of aircraft performs well in all three attributes. Though more expensive, the three-orbital plane constellation with three satellites per orbital plane exhibits good performance for the attributes of area covered and average resolution. Clearly tradeoffs exist among the design choices, and other quality system designs exist. For the final choice of which candidate system design to choose, exploring and visualizing the trade space helps identify these tradeoffs and their relative magnitudes.

4.3 Case Study Conclusions

In this chapter, the Engineering Framework for Concept Development and the concept design tools were applied to two concept design problems. The first was the design of the propulsion system for an Air-Launched Sounding Rocket (ALSR). The *Selection* and *Refinement* phase classes were used to carry out the conceptual design. Using analytic models, OPN enumerated all of the feasible propulsion system designs, and ANOVA identified the diameter dimensions and propellant options as important design variables. After using the *Selection* results to eliminate one-stage options as dominated design choices, a more advanced 2-DoF flyout simulation code was implemented with the aid of particle swarm optimization. The algorithm efficiently identified the designs along Pareto front. The detailed simulation revealed the advantages of smaller diameter designs due to aerodynamic effects, something not present in the analytic evaluations. The PSO trade space exploration and the ANOVA results indicated the importance of ORBUS 6E as a propellant choice. The properties of this propellant option were then examined in a sensitivity study to see which designs were most senstive to variations. The results of the sensitivity study inform the designer about decisions among Pareto front designs.

The other design study presented was that of a responsive disaster monitoring system. This Operationally Responsive System (ORS) problem utilized all three phase classes of the Engineering Framework. Because the problem statement was not as well defined as the ALSR problem, additional work was required in the *Exploration* phase class. This included identifying types of possible disasters, stakeholders, and compiling a list of stakeholder-derived attributes of the responsive system. Pugh analysis was used to conduct comparisons between various dissimilar concept alternatives. This helped clarify each of the concept definitions and provided a ranking of the different alternatives. A smaller set of options was carried on to the *Selection* phase class, where OPN was used to evaluate low-order models. These models were based on representative existing systems used as performance baselines. Trade spaces of pairs of attributes were visualized to examine the performances of the satellite, aircraft, and distributed swarm concepts. A non-dominance occurrence count approach was implemented to select the aircraft classes and satellites to investigate further. Then the *Refinement* phase class explored more deeply the aircraft and satellite designs. This included more detailed aircraft sizing models, satellite orbit and coverage models, and a mission timeline model to simulate a disaster response scenario. Experimental designs were run on both the aircraft and satellite concepts, and PSO was implemented to search the space of system designs that incorporated both aircraft and satellite alternatives. These results indicate that an aircraft in the class of a catapult-launched unmanned air vehicle (UAV) used in conjunction with a two- or three-plane Walker constellation with three satellites per orbital plane are system designs that perform well with respect to the stakeholder-derived attributes of the system. From the ANOVA results, takeoff gross weight was identified as an important design variable for the aircraft concepts. A sensitivity study on takeoff gross weight for the Pareto front designs showed that the small UAV-class designs are relatively insensitive to variation in this variable.

In both case studies, the Engineering Framework provides a systematic and customizable approach to concept design problems. The concept design tools are also readily applicable to both problems presented here. Their use aids the designer in evaluating design options as well as interpreting the results to gain insight into the problem.

Chapter 5

Conclusions & Future Work

"Design is so simple; that's why it is so complicated."

— Paul Rand

As the complexity of engineering systems continues to increase, so do the challenges in their design. The decisions that are made early in the development of a concept are among the most important. They are made when little information is known on the design and its environmental context, as well as when the uncertainty in that information is high. These decisions significantly impact the remainder of the program. It is therefore desirable to build confidence that the design problem is well understood before large investments are made in the program. To aid in achieving that goal, this thesis set out to improve the process by which concept design is carried out.

The Engineering Framework for Concept Development was introduced as a guide to formulating a design process suitable to a diverse range of concept design problems. The Engineering Framework describes the stages of design in terms of *activities* and *phase classes.* The activities consist of *Problem Characterization*, *Alternative Generation*, *Model Development and Evaluation*, and *Decision Analysis.* Design iterations consist of repetitions of the four activities. The phase classes allow problems to be classified by the level of detail of the problem. Design problems characterized by qualitative and subjective assessments fall under the *Exploration* phase class. Problems that are described by quantitative measures that rank dissimilar concepts are categorized in the *Selection* phase class. The final phase class, *Refinement*, involves problems with quantified, parameterized data and detailed models. The *Refinement* analysis explores the ranges of the design parameters to identify a superior design concept.

Accompanying the Engineering Framework is a collection of design tools that aid the designer in his or her progression through the stages of design. The Engineering Framework allows the classification of the concept design tools based on their applicability to specific activities and phase classes. The design tools demonstrated in this thesis are Pugh analysis, Object-Process Network (OPN), particle swarm optimization (PSO), and analysis of variance (ANOVA).

Pugh analysis is useful when disparate concepts are subjectively compared in the *Exploration* phase class of the Engineering Framework. The technique consists of making pair-wise comparisons of concepts for each attribute of the design. Conducting a Pugh analysis helps to direct a discussion on the advantages and disadvantages of various concepts, identify the characteristics of good and bad concepts, stimulate creative associations that can lead to new concept ideas, and rank the concepts in order to choose among them.

The OPN tool, used in the *Selection* phase class, allows the designer to search through a large design space of alternatives. This approach of trade space exploration provides much more knowledge and insight into the design problem than single point design. OPN also allows visualization of the design space to aid understanding of the interactions of design choices. The mathematical functions in OPN are used to generate all of the feasible combinations of alternatives in the design space. User-defined metrics can be calculated to understand the drivers and tradeoffs among different designs.

As a designer moves through the phase classes of the Engineering Framework, increasing levels of model fidelity and computation time may prohibit evaluating a large number of designs. To help identify promising families of designs in an efficient manner, particle swarm optimization is implemented in the *Refinement* phase class to conduct a global search of the design space and rapidly focus on designs in the vicinity of the Pareto front.

One way to better understand the OPN and PSO results is by visualizing the impacts of different design decisions in the trade space. With trade space visualization techniques, the tradeoffs associated with decisions are clearly presented for the designer to evaluate. However, some design problems may include many design variables — the different design options the designer wishes to explore. Others may include many attributes — the characteristics of the design that the designer or decision-maker uses to assess how well the different designs perform. For problems with many design variables or many attributes, statistical techniques such as ANOVA can provide insight into the data that is difficult to obtain solely via visualization methods. ANOVA provides the designer with information on which design variables are most the important in terms of their effects on the attributes. This information can be used to conduct sensitivity studies on the most important design parameters in both the *Selection* and *Refinement* phase classes.

5.1 Demonstrated Applications

The Engineering Framework for Concept Development and the concept design tools were applied to two concept design problems. One was the design of the propulsion system for an Air-Launched Sounding Rocket (ALSR). Utilizing the *Selection* and *Refinement* phase classes of the Engineering Framework, this case study demonstrated the advantages of using OPN, PSO, and ANOVA in searching the space of feasible options and understanding the important aspects of the propulsion system design. The results from this study show that aerodynamic effects lead to smaller diameter options being favored, and the ORBUS 6E solid propellant is the preferred choice.

The other design study presented was a responsive disaster monitoring system. This Operationally Responsive System (ORS) problem utilized all three phase classes of the Engineering Framework. In the *Exploration* phase class, Pugh analysis was used to conduct early assessments of dissimilar concept alternatives. A subset of these options was carried on to the *Selection* phase class, where OPN was used to evaluate low-order models using representative existing systems as performance baselines. In the *Refinement* phase class, more sophisticated aircraft and satellite models were used in a simulated mission timeline scenario. This timeline scenario modeled a representative disaster mission and was used to compare multi-vehicle system architecture designs involving aircraft and satellite constellation alternatives. The particle swarm optimization algorithm was used to efficiently search through the space of design options, and it successfully found promising system designs in a three-attribute trade space. The results from this study indicate that an aircraft in the class of a catapult-launched unmanned air vehicle (UAV) used in conjunction with a two- or three-plane Walker constellation with three satellites per orbital plane are system designs that perform well with respect to the stakeholder-derived attributes of the system.

5.2 Future Work

This thesis has outlined a systematic guide for approaching concept design problems, as well as demonstrating the use of concept design tools as part of that Engineering Framework. There are ways in which each of these aspects to designing new systems can be improved or supplemented. The following list includes some of these possible areas for future work.

- The two main concerns with trade space exploration are how to gain the most information with the fewest number of design evaluations, and how to best understand the results.
  - This thesis demonstrated the full-factorial approach of OPN and the heuristic search method of PSO. Other exploration techniques can be examined, especially in the context of the latter two phase classes of design. For instance, how is the information from the *Selection* Design of Experiments best utilized in formulating the *Refinement* design variables, attributes, and objectives? Can the ANOVA and sensitivity results be incorporated into the search strategy for the optimization algorithm to better identify robust designs? Furthermore, certain combinations of tools may be more appropriate or more effective when used together. How should other DoE techniques such as orthogonal arrays, for example, best be implemented with statistical or optimization methods?

  - The other rich area for research related to trade space exploration is in understanding the resulting trade space. This includes gaining information on the design tradeoffs, the design drivers, and the sensitivities to uncertainty. The approach in this thesis was to use trade space visualization and analysis of variance to identify the important design variables. Then these results informed the focus of the sensitivity study, and boxes of uncertainty were drawn to visually depict sensitivity in the trade space. The challenge in visualizing the trade space data is that only a certain amount of information can be presented effectively in a single view. No more than three spatial dimensions can be used, and the use of a number of other properties (color, shape, shading, transparency) is necessarily limited by how much information someone can interpret simultaneously. Better ways to present trade space information would be very useful.
  - Another area for future work related to understanding the trade space is extracting the statistics of the design data. ANOVA has been shown to provide insight into the design space that might otherwise be missed when relying only on visualization methods. With large data sets, statistical approaches can possibly extract additional useful information. Mathematical correlations discovered via statistical evaluations can detect trends in the underlying data that can otherwise go undetected.

- An objective considered but never implemented by the author was to incorporate more design information into the sensitivity studies. In this thesis the sensitivity studies were conducted on the Pareto front designs using the design variable identified via the ANOVA results. Because of the uncertainty involved in concept design, there are other sensitive parameters that may be of interest. Furthermore, there are dominated designs that may prove to be non-dominated when more is known about the concept. Therefore, it is desirable to determine which parameters should be considered in a sensitivity study and on which designs to conduct the study. One approach might be to include designs around the Pareto front in the sensitivity analysis. This could be done in a fashion similar to the successive Pareto front approach utilized in the ORS three-attribute trade space visualization. Also, the results from a *Selection* sensitivity study could be used in *Refinement* to help identify robust designs.

- Pugh analysis was found to be a useful technique in this thesis for *Exploration* concept selection. Other approaches such as Quality Function Deployment have been popular in similar situations [22]. Future research can investigate these *Exploration* concept design tools further. Specifically, the work in this thesis found Pugh to be a logical and tractable method for assessing concepts with respect to their attributes. The QFD House of Quality involves different levels of performance scoring, relative preference weighting parameters, a correlation matrix, benchmarks, and engineering parameters. Plus the full QFD contains more than just the House of Quality. When the concepts of *Exploration* are necessarily vague and general, how much value is added by using the more involved QFD approach? Do those results provide additional insight than by using simple one-to-one comparisons? Are there problems better suited to QFD than Pugh, and vice versa?

- The use of OPN for concept design problems has had limited uses thus far. The tool has shown its value in modeling problems that have many interrelated options in the design space [77], and in visualizing a network [69], but additional work would help identify the types of problems for which it is most applicable. As a *Selection* phase class tool, it borders the region between qualitative and quantitative data. Should OPN be used more for subjective assessments and utilizing the visual aspects to facilitate dialogue on concept comparisons? Should its mathematical features be improved and harnessed to make it capable of more advanced evaluations?

- The goal of this thesis was not to fine-tune the PSO algorithm. Thus, there is room for improving the PSO algorithm to find the best families of designs during the conceptual stages. This includes not only the algorithm parameters, but also what to choose as the design variables and objective function.

The Engineering Framework for Concept Development described in this thesis is part of an ongoing effort at The C.S. Draper Laboratory to advance the methods for approaching hard design problems early in their development. Research into its capabilities, including the work presented here, is demonstrating the potential value that the Engineering Framework can add to an engineering program. Further recognition of the ideas set forth in this thesis by the larger systems engineering community will help to better organize the up-front stages of concept design. Improved concept exploration and increased knowledge of the design possibilities will not only increase the confidence of the designer and the decision-maker, it will also lead to more successful programs and better engineering systems.

Bibliography

[1] Richard Balling. Design by shopping: A new paradigm? In *Proceedings of the 3rd WCSMO ISSMO/UBCAD/UB/AIAA, Third World Congress of Structural and Multidisciplinary Optimization*, pages 295–297, Buffalo, New York, May 17–21 1999.

[2] Roger R. Bate, Donald D. Mueller, and Jerry E. White. *Fundamentals of Astrodynamics.* Dover Publications, Inc., New York, 1971.

[3] Richard H. Battin. *An Introduction to the Mathematics and Methods of Astrodynamics.* AIAA Education Series. American Institute of Aeronautics and Astronautics, Inc., Reston, Virginia, revised edition, 1999.

[4] Dimitris Bertsimas and John N. Tsitsiklis. *Introduction to Linear Optimization.* Optimization and Computation Series. Athena Scientific, 1997.

[5] Benjamin S. Blanchard and Wolter J. Fabrycky. *Systems Engineering and Analysis.* Prentice-Hall, Upper Saddle River, New Jersey, third edition, 1998.

[6] Michael A. Buonanno. *A Method for Aircraft Concept Exploration Using Multicriteria Interactive Genetic Algorithms.* Doctor of philosophy, Georgia Institute of Technology, December 2005.

[7] Carlos A. Coello Coello and Maximino Salazar Lechuga. MOPSO: A proposal for multiple objective particle swarm optimization. *IEEE*, 2002.

[8] Lucy E. Cohan, Elizabeth O. Jordan, and David W. Miller. Tradespace exploration for a parameterized modular optical space telescope (MOST). In *Proceedings of the*

48th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Honolulu, Hawaii, April 23–26 2007.

[9] AIAA Technical Committee. Current state of the art on multidisciplinary design optimization. White Paper, January 15, 1991.

[10] William R. Corliss. NASA sounding rockets, 1958–1968: A historical summary. NASA Special Publication SP-4401, 1971.

[11] Steven L. Cornford, Martin S. Feather, Vance A. Heron, and J. Steven Jenkins. Fusing quantitative requirements analysis with model-based systems engineering. In *Proceedings of the 14th IEEE International Requirements Engineering Conference*, Minneapolis, Minnesota, September 11–15 2006.

[12] Catherine Bly Cox and Charles Murray. *Apollo: The Race to the Moon.* Cox & Murray Inc., Burkittsville, MD, 2004.

[13] Edward Crawley. MIT System Architecture Class Notes ESD.34, January 2007.

[14] Olivier de Weck and Karen Willcox. MIT Multidiscplinary System Design Optimization Class Notes 16.888, Spring 2004.

[15] Russell C. Eberhart and James Kennedy. A new optimizer using particle swarm theory. In *In Proceedings of the Sixth International Symposium on Micromachine and Human Science*, pages 39–43, Nagoya, Japan, 1995.

[16] Russell C. Eberhart and Y. Shi. Comparing inertia weights and constriction factors in particle swarm optimization. *IEEE*, 2000.

[17] A.I. El-Gallad, M.E. El-Hawary, and A.A. Sallam. Swarming of intelligent particles for solving the nonlinear constrained optimization problem. *Engineering Intelligent Systems for Electrical Engineering and Communications*, 9(3):155–163, 2001.

[18] I.E. Fieldsend and S. Singh. A multi-objective algorithm based upon particle swarm optimisation, an efficient data structure and turbulence. In *Proceedings of the 2002 U.K. Workshop on Computational Intelligence*, pages 37–44, Birmingham, UK, 2002.

[19] Asia-Pacific Regional Space Agency Forum. International cansat workshop results. http://www.aprsaf.org, November, 21 2007. Accessed: 29 February 2008.

[20] Fred Glover. Tabu search — part I. *ORSA Journal on Computing*, 1(3):190–206, 1989.

[21] Fred Glover. Tabu search — part II. *ORSA Journal on Computing*, 2(1):4–32, 1990.

[22] B. E. Goldberg, K. Everhart, R. Stevens, N. Babbitt, III, P. Clemens, and L. Stout. System engineering toolbox for design-oriented engineers. *NASA STI/Recon Technical Report N*, 95:18932–+, December 1994.

[23] Daniel E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Reading, MA, 1989.

[24] Erica L. Gralla, William Nadir, Hamed Mamani, and Olivier de Weck. Optimal launch vehicle size determination for Moon-Mars transportation architectures. In *Proceedings of Space 2005*, Long Beach, California, August 30–September 1 2005.

[25] M. Grant and G. Mendeck. Mars science laboratory entry optimization using particle swarm methodology. In *Proceedings of AIAA Atmospheric Flight Mechanics Conference and Exhibit*, Hilton Head, South Carolina, August 20–23 2007.

[26] Matthias Gries. Methods for evaluating and covering the design space during early design development. *Integration, the VLSI Journal*, 38:131–183, 2004.

[27] INCOSE Systems Engineering Handbook Working Group. *Systems Engineering Handbook, Version 2a*. International Council on Systems Engineering, 2004.

[28] David Harris. Saving spaceship Orion — zero base vehicle task complete. http://www.nasaspaceflight.com/content/?cid=5242, October 1, 2007. Accessed: 30 April 2008.

[29] Rania Hassan, Babak Cohanim, Olivier de Weck, and Gerhard Venter. A comparison of particle swarm optimization and the genetic algorithm. In *Proceedings of the 46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics & Materials Conference*, Austin, Texas, April 18–21 2005.

[30] Daniel Hastings and Hugh McManus. A framework for understanding uncertainty and its mitigation and exploitation in complex systems, 2004. MIT Engineering Systems Symposium.

[31] T.A. Heppenheimer. The Space Shuttle decision: NASA's search for a reusable space vehicle. NASA Special Publication SP-4221, 1999.

[32] John H. Holland. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence.* The MIT Press, Bradford Books edition, 1992.

[33] Xiaohui Hu and Russell C. Eberhart. Multiobjective optimization using dynamic neighborhood particle swarm optimization. In *Proceedings of the IEEE Congress on Evolutionary Computation*, pages 1677–1681, Honolulu, Hawaii, 2002.

[34] Xiaohui Hu and Russell C. Eberhart. Solving constrained nonlinear optimization problems with particle swarm optimization. In *Proceedings of the Sixth World Multiconference on Systemics, Cybernetics and Informatics*, Orlando, Florida, 2002.

[35] Ronald W. Humble, Gary N. Henry, and Wiley J. Larson. *Space Propulsion Analysis and Design.* Space Technology Series. The McGraw-Hill Companies, revised edition, 1995.

[36] Kelly Humphries. NASA selects Orion Crew Exploration Vehicle prime contractor. http://www.nasa.gov/home/hqnews/2006/aug/HQ_06305_Orion_contract.html, August 31, 2006. Accessed: 30 April 2008.

[37] Jong-Bin Im, Soo-Yong Lee, and Jungsun Park. Hybrid simulated annealing with particle swarm optimization applied kriging meta model. In *Proceedings of the 48th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Honolulu, Hawaii, April 23–26 2007.

[38] Insitu. ScanEagle. http://www.insitu.com/scaneagle, 2008. Accessed: 15 March 2008.

[39] Cyrus Jilla. *A Multiobjective, Multidisciplinary Design Optimization Methodology for the Conceptual Design of Distributed Satellite Systems.* Doctor of philosophy, Massachusetts Institute of Technology, May 2002.

[40] Robert C. Seamans Jr. *Project Apollo: The Tough Decisions.* Monographs in Aerospace History, Number 37. National Aeronautics and Space Administration, Washington, D.C., 2005.

[41] Deepti Kanukolanu, Kemper E. Lewis, and Eliot H. Winer. A multidimensional visualization interface to aid in trade-off decisions during the solution of coupled subsystems under uncertainty. *Journal of Computing and Information Science in Engineering*, 6:288–299, September 2006.

[42] James Kennedy and Russell Eberhart. Particle swarm optimization. In *Proceedings of the IEEE International Conference on Neural Networks*, pages 1942–1945, Perth, Australia, 1995.

[43] S. Kirkpatrick, Jr. C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, May 13, 9183.

[44] Benjamin Koo. *A Meta-language for Systems Architecting.* Doctor of philosophy, Massachusetts Institute of Technology, January 2005.

[45] Elisabeth Lamassoure and Daniel E. Hastings. Generalized metrics for optimization of space systems cost-effectiveness. In *Proceedings of AIAA Space 2000 Conference & Exposition*, Long Beach, California, September 19–21 2007.

[46] Elisabeth Lamassoure, Stephen D. Wall, and Robert W. Easter. Model-based engineering design for trade space exploration throughout the design cycle. In *Proceedings of AIAA Space 2004 Conference & Exhibit*, San Diego, California, September 28–30 2004.

[47] Wiley J. Larson and James R. Wertz. *Space Mission Analysis and Design.* Space Technology Series. Microcosm Press and Kluwer Academic Publishers, El Segundo, California, 3rd edition, 1999.

[48] Kwanwoo Lee. A top-down approach to quality driven architectural engineering of software systems. *IEICE Transactions on Information and Systems*, E88-D(12):2757–2766, 2005.

[49] Xiyu Liu, Hong Liu, and Huichuan Duan. Particle swarm optimization based on dynamic niche technology with applications to conceptual design. In *Proceedings of the 10th International Conference on Computer Supported Cooperative Work in Design*, 2006.

[50] Mark W. Maier. Architecting principles for system of systems. *Systems Engineering*, 1(4):267–284, 1998.

[51] Mark W. Maier and Eberhardt Rechtin. *The Art of Systems Architecting.* CRC Press, Boca Raton, Florida, 2nd edition, 2002.

[52] Rebecca Masterson. *Dynamic Tailoring and Tuning for Space-Based Precision Optical Structures.* Doctor of philosophy, Massachusetts Institute of Technology, February 2005.

[53] Jack D. Mattingly, William H. Heiser, and David T. Pratt. *Aircraft Engine Design.* AIAA Education Series. American Institute of Aeronautics and Astronautics, Inc., Reston, Virginia, 2nd edition, 2002.

[54] Hugh L. McManus and Todd E. Schurman. Understanding the orbital transfer vehicle trade space. In *Proceedings of the AIAA Space 2003 Conference*, Long Beach, California, September 23–25 2003.

[55] Software Engineering Institute Carnegie Mellon. What is CMMI? http://www.sei.cmu.edu/cmmi/general/index.html, 2008. Accessed: 3 May 2008.

[56] Achille Messac. v. In *Proceedings of the 37th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference and Exhibit*, pages 962–974, Reston, Virginia, April 15–17, 1996.

[57] Achille Messac. From dubious construction of objective functions to the application of physical programming. *AIAA Journal*, 38(1):155–163, 2000.

[58] Douglas C. Montgomery. *Design and Analysis of Experiments.* John Wiley & Sons, New York, 4th edition, 1997.

[59] Todd Mosher. Applicability of selected multidisciplinary design optimization methods to conceptual spacecraft design. In *Proceedings of the 6th AIAA ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Bellevue, Washington, September 4–6 1996.

[60] Todd Mosher. Conceptual spacecraft design using a genetic algorithm trade selection process. *Journal of Aircraft*, 36(1):200–208, 1999.

[61] S. Mostaghim and J.R. Teich. Strategies for finding local guides in multi-objective particle swarm optimization (MOPSO). In *Proceedings of the IEEE Swarm Intelligence Symposium*, pages 26–33, Indianapolis, Indiana, 2003.

[62] Federation of American Scientists. Rq-4a global hawk. http://www.fas.org/irp/program/collect/global_hawk.htm, 2008. Accessed: 14 March 2008.

[63] Vilfredo Pareto. *Manuale di Economia Politica, Translated into English by A.S. Schwier.* Macmillan, New York, 1971.

[64] K.E. Parsopoulos and M.N. Vrahatis. Particle swarm optimization for constrained optimization problems. In *Proceedings of the Euro-International Symposium on Computational Intelligence*, 2002.

[65] Stuart Pugh. *Creating Innovative Products Using Total Design: The Living Legacy of Stuart Pugh.* Addison-Wesley, Reading, Massachusetts, 1996.

[66] Mateen-ud-Din Qazi and He Linshu. Rapid trajectory optimization using computational intelligence for guidance and conceptual design of multistage space launch vehicles. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, San Francisco, California, August 15–18 2005.

[67] Daniel P. Raymer. *Aircraft Design: A Conceptual Approach.* AIAA Education Series. American Institute of Aeronautics ans Astronautics, Inc., Reston, Virginia, 3rd edition, 1999.

[68] Adam Ross, Daniel E. Hastings, Joyce M. Warmkessel, and Nathan P. Diller. Multi-attribute tradespace exploration as front end for effective space system design. *Journal of Spacecraft and Rockets*, 41(1):20–28, 2004.

[69] Ziv Rozenblum. Object-Process Networks & Object-Process Diagrams: Implementation issues for oil exploration systems. Master's thesis, Massachusetts Institute of Technology, May 2007.

[70] Thomas L. Saaty. Priority setting in complex problems. *IEEE Transactions on Engineering Management*, 30(3):140–1555, 1983.

[71] Todd Schuman and Olivier L. de Weck. Integrated system-level optimization for concurrent engineering with parametric subsystem modeling. In *Proceedings of 46th AIAA/ASME/AHSI/ASC Structures, Structural Dynamics & Materials Conference*, Austin, Texas, April 18–21 2005.

[72] Yuhui Shi. Particle swarm optimization. *IEEE Connections*, 2(1):8–13, 2004.

[73] Yuhui Shi and Russell Eberhart. A modified particle swarm optimizer. In *Proceedings of the 1999 Congress on Evolutionary Computation*, volume 3, pages 69–73, Washington, D.C, July 6–9 1998.

[74] Yuhui Shi and Russell Eberhart. Empirical study of particle swarm optimization. In *Proceedings of the IEEE World Congress on Computational Intelligence*, pages 69–73, May 4–9 2004.

[75] Yuhui Shi and Russell C. Eberhart. Parameter selection in particle swarm optimization. In *Evolutionary Programming VII: Proceedings of the Seventh Annual Conference on Evolutionary Programming*, pages 591–600, 1998.

[76] Robert Shishko. *NASA Systems Engineering Handbook*. National Aeronautics and Space Administration, 1995.

[77] Willard L. Simmons, Benjamin H.Y. Koo, and Edward F. Crawley. Architecture generation for Moon-Mars exploration using an executable meta-language. In *Proceedings of AIAA Space 2005*, Long Beach, California, August 30 – September 1 2005.

[78] Willard L. Simmons, Benjamin H.Y. Koo, and Edward F. Crawley. Space systems architecting using meta–languages. *56th International Astronautical Congress*, 2005.

[79] Valdeon Sozo, Fernando Forcellini, and André Ogliari. Axiomatic approach application during the product conceptual design phase. In *Proceedings of International Conference "Mechanika 2001"*, pages 267–272, Kaunas, Lithuania, April 2001.

[80] NASA SP-4009. *The Apollo Spacecraft: A Chronology, vols. 1-4*. National Aeronautics and Space Administration, Washington, 1978.

[81] Donald V. Steward. Planning and managing the design of systems. In *Proceedings of IEEE Technology Management: the New International Language*, pages 189–193, October 27–31, 1991.

[82] Gary Stump, Timothy W. Simpson, Mike Yukish, and Lorri Bennett. Multidimensional visualization and its application to a design by shopping paradigm. In *Proceedings of 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Atlanta, Georgia, September 4–6 2002.

[83] Gary M. Stump, Timothy W. Simpson, Mike Yukish, and John J. O'Hara. Trade space exploration of satellite datasets using a design by shopping paradigm. In *Proceedings of the IEEEAerospace Conference*, pages 3885–3895, March 6–13 2004.

[84] Gary M. Stump, Mike Yukish, Timothy W. Simpson, and E. Nathan Harris. Design space visualization and its applications to a design by shopping paradigm. In *Proceedings of ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Chicago, Illinois, September 2–6 2003.

[85] Deborah L. Thurston. Real and misconceived limitations to decision based design with utility analysis. *ASME Journal of Mechanical Design*, 123:176–182, 2001.

[86] Scott Uebelhart. *Non-Deterministic Design and Analysis of Parameterized Optical Structures during Conceptual Design.* Doctor of philosophy, Massachusetts Institute of Technology, June 2006.

[87] Garret N. Vanderplaats. *Numerical Optimization Techniques for Engineering Design.* Vanderplaats Research & Development, Inc, third edition, 2001.

[88] Gerhard Venter, Gary C. Quinn, Vladimir Balabanov, and Garret N. Vanderplaats. What-if post-processing of general-purpose optimization results using VisualDOC. In *Proceedings of the 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Albany, New York, August 30–September 1 2004.

[89] Gerhard Venter and Jaroslaw Sobieszczanski-Sobieski. Particle swarm optimization. *AIAA*, 41(8), August 2003.

[90] Ivan N. Vuchkov and Lidia N. Boyadjieva. *Quality Improvement with Design of Experiments: A Response Surface Approach.* Kluwer Academic Publishers, 2001.

[91] Myles A. Walton and Daniel E. Hastings. Applications of uncertainty analysis to architecture selection of satellite systems. *Journal of Spacecraft and Rockets*, 41(1):75–84, 2004.

[92] Matthew O. Ward. A taxonomy of glyph placement strategies for multidimensional data visualization. *Information Visualization*, 1(3/4):194–210, 2002.

[93] Yuin Wu and Alan Wu. *Taguchi Methods for Robust Design.* The American Society of Mechanical Engineers, New York, 2000.

[94] Yong-Ling Zheng, Long-Hua Ma, Li-Yan Zhang, and Ji-Xin Qian. On the convergence analysis and parameter selection in particle swarm optimization. In *Proceedings of the Second International Conference on Machine Learning and Cybernetics*, Xi'an, China, November 2–5, 2003.