# Integrated Transportation System Design Optimization

by

**Christine Taylor**

B.S. Cornell University (2001)

S.M. Massachusetts Institute of Technology (2003)

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of

**Doctor of Philosophy**

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2007



Author .....

Department of Aeronautics and Astronautics

January 29, 2007

Certified by .....

Olivier L. de Weck

Assistant Professor of Aeronautics and Astronautics & Engineering Systems

Thesis Supervisor

Certified by .....

Karen Willcox

Associate Professor of Aeronautics and Astronautics

Committee Member

Certified by .....

Jeffrey A. Hoffman

Professor of the Practice of Aeronautics and Astronautics

Committee Member

Certified by .....

Diego Klabjan

Associate Professor Civil and Environmental Engineering (UIUC)

Committee Member

Accepted by .....

Professor Jaime Peraire

Chairman, Department Graduate Committee

# Integrated Transportation System Design Optimization

by

**Christine Taylor**

Submitted to the Department of Aeronautics and Astronautics
on January 29, 2007, in partial fulfillment of the
requirements for the degree of
**Doctor of Philosophy**

## Abstract

Traditionally, the design of a transportation system has focused on either the vehicle design or the network flow, assuming the other as given. However, to define a system level architecture for a transportation system, it is advantageous to expand the system boundary during the design process to include the network definition, the vehicle specifications, and the operations, which couple the vehicle(s) and the network. The integrated transportation system formulation developed in this thesis examines these fundamental components by classifying the decisions required to define them and concurrently optimizing the entire design problem, resulting in a more efficient transportation architecture.

The integrated transportation system design models are developed for an air and a space transportation system and an example problem is implemented for each. The integrated air transportation system example of an overnight package delivery network quantifies at least a ten percent improvement in cost over traditional optimization approaches. The formulation for a space transportation system first requires the definition of a space network which is constructed by extending time expanded networks to account for astrodynamic relationships. An Earth-Moon logistics supply example quantifies a 19 percent improvement in total mass in Low Earth Orbit as compared to traditional optimization methods. The improvements in system objective values obtained can be attributed to the reduction in operational inefficiencies for the transportation system.

The concurrent optimization of the integrated transportation system design problem employs a new methodology, embedded optimization, to obtain solutions. Embedded optimization allows Simulated Annealing to effectively find good solutions to highly constrained problems by embedding deterministic solvers, such as linear or mixed integer programs, into the perturbation step. Comparing the solutions and computational performance of SA with and without embedded optimization reveals that embedded optimization performs significantly better, with 95 percent confidence.

Olivier L. de Weck
Title: Associate Professor of Aeronautics and Astronautics & Engineering Systems

Thesis Supervisor

Karen Willcox
Title: Associate Professor of Aeronautics and Astronautics
Committee Member

Jeffrey A. Hoffman
Title: Professor of the Practice of Aeronautics and Astronautics
Committee Member

Diego Klabjan
Title: Associate Professor Civil and Environmental Engineering (UIUC)
Committee Member

# Acknowledgments

I would like to thank my advisor, Prof. Oli de Weck, for his guidance and support throughout this entire process. His knowledge and insight have forever changed my research perspective. I would also like to thank Prof. Diego Klabjan who essentially became another advisor to me in the last year. I enjoyed all the time we spent working and joking together and thanks for all the postcards. I would like to thank Prof. Karen Willcox for her guidance on my research and career and Prof. Jeffrey Hoffman for providing me with the knowledge of his experience. Additionally, I would like to thank Julie Finn for your assistance in all administrative logistics matters, as she has made this whole process easier with her diligence and continual smile. I want to thank Prof. Anil Rao, my master's thesis advisor, whose encouragement and expertise helped me find my path. I would like to thank Prof. Earl Murman who believed in me and my pursuit of this doctorate. Finally, I would like to thank all of the exceptional teachers that I have had throughout my *many* years of education as they inspired my desire for learning.

I would also like to thank all of the people I have met at MIT that have provided me with both technical assistance and emotional support. Specifically I would like to thank Ryan Boas for being a study buddy and providing much needed procrastination conversations, Bill Simmons for all the latex support, Wilfired Hofstetter for the astrodynamics lessons and Jaemyung Ahn for always being willing to help. I would like to thank Matt Silver, Sarah Shull, Gergana Buonova and Erica Gralla for being my traveling buddies this summer. Europe would not have been the same without you. I would like to thank Dr. Heidi Davidz for all of her support and for showing me that it can be done. I would also like to thank Dr. David Benson for keeping my pool skills sharp and JK Srinivasan for providing me with Salsa lessons.

I would also like to thank all of my friends who have provided me with the emotional support that I have needed to reach this goal. Specifically, I would like to thank Jen for being my kick-boxing buddy and for providing the latest joy to my life, Matt Schaffer (a.k.a the peanut). I want to thank Margie for being my in-house support

system, feeding me delicious vegetarian food, and keeping my nails well manicured. I also want to thank my former Royce Road buddies, Sierra and Macaela for always being willing to listen. I would like to thank my friends back home, specifically Rob and Siobhan, for providing me an escape from Boston. To Erica and Allison, thanks for always supporting me in every step, encouraging me to succeed, and keeping my spirits light when I am down. And to Sarah, my fellow grad student in crime out there on the west coast, I really don't know if I would have made it through this without you. Thanks for providing me with a commiseration buddy and a great spring break destination.

Last, but most importantly, I would like to thank my parents, Theresa and Richard Taylor. Their love and support is epic. They have believed in me unfailingly throughout my whole life and it is only because of this that I am here. Thank you and I love you.

This research was supported by the NASA Advanced Planning and Integration Office under contract No. 6895818 and under the supervision of Mr. Gary Martin and the NASA Interplanetary Supply Chain Management and Logistics Project under contract number 6897715 under the supervision of Mr. Martin Steele. Financial support to the author was further provided in the form of an Amelia Earhart Fellowship from Zonta International.

This thesis is dedicated to the memory of my grandmother Jennie Ebert (1916-2001).

# Nomenclature

### Abbreviations

| | |
|---|---|
| CPLEX | Mixed-Integer Programming Software |
| EML1 | $1^{st}$ Earth Moon Lagrangian Point |
| LELO | Low Equatorial Lunar Orbit |
| LEO | Low Earth Orbit |
| LES | Lunar Equatorial Surface |
| LP | Linear Programming Software |
| LPLO | Low Polar Lunar Orbit |
| LPS | Lunar Polar Surface |
| MATLAB | Matrix Laboratory |
| SA | Simulated Annealing |

### Symbols

| | |
|---|---|
| $\mathcal{A}$ | set of all arcs in time expanded network |
| $\mathcal{AS}$ | set of all static arcs |
| $\mathcal{G}$ | time expanded network graph definition |
| $\mathcal{GS}$ | static network graph definition |
| $\mathcal{J}$ | objective function |
| $\mathcal{N}$ | set of all nodes in time expanded network |
| $\mathcal{NS}$ | set of all static nodes |
| $D$ | air network distance matrix |
| $E$ | element design type |
| $I_{sp}$ | specific impulse |

| | |
|---|---|
| $N$ | number of nodes in aircraft network |
| $P$ | aircraft demand matrix |
| $T$ | number of time periods in time expanded network |
| $W_0$ | aircraft take-off weight |
| $W_p$ | aircraft payload weight |
| $a$ | reference to a specific arc |
| $c$ | total aircraft cost per day |
| $c$ | element capacity |
| $e$ | element instance e |
| $f$ | fixed aircraft cost |
| $f$ | element fuel selection |
| $f_f$ | aircraft fuel fraction |
| $g$ | number of aircraft engines |
| $g$ | general non-linear function |
| $g_0$ | Earth's sea-level gravitation |
| $h$ | general linear function |
| $l$ | aircraft wing loading |
| $m$ | variable aircraft cost |
| $n$ | number of nodes in general network |
| $n$ | number of aircraft |
| $m$ | element fuel mass |
| $p$ | path |
| $q$ | sub-path corresponding to a burn sequence |
| $r$ | aircraft range |
| $s_f$ | aircraft structural fraction |
| $t$ | aircraft wing loading |
| $t$ | element thrust |
| $w$ | aircraft capacity |
| $v$ | aircraft cruise velocity |
| $x$ | cargo path variable |

| | |
|---|---|
| $y$ | element allocation variable |
| $\Delta V$ | velocity change for orbit transfer |
| $\delta t$ | time period length |
| $\phi$ | element fuel fraction |
| $\rho_0$ | sea-level density |

**Subscripts and Superscripts**

| | |
|---|---|
| $(\cdot)^A$ | aircraft of type $A$ |
| $(\cdot)^E$ | element of type $E$ |

## Definitions

**Path**: A path or route is a set of nodes or arcs in a network that defines the sequence of locations that a vehicle travels.

**Cargo**: Cargo is the general term describing goods or shipments.

**Commodity**: A commodity is a discrete unit of cargo that is distinguished by corresponding supply and demand information as well as its physical properties.

**Package**: A package is a commodity that can be represented by a continual flow instead of by discrete units.

**Transport**: Transportation refers to the movement between two physical locations. Thus, when transporting cargo, the vehicle that contains the cargo changes physical locations within the network.

**Transfer**; A transfer refers to a cargo containment change. Thus, when transferring cargo, the vehicle that the cargo is contained within changes, however the physical

location of the cargo does not.

**Optimal Solution**: An optimal solution, as referred to in this research, is a vector of *feasible* design values that obtains the best value of the system objective, as compared to all other feasible design vectors examined. Therefore, in this research, reference to an optimal solution does not imply any strict adherence to a mathematically defined optimum.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

## 1.1 Motivation

The system-of-systems philosophy [4, 5] has expanded the system boundary to encompass an integrated view of a system during the design process. Systems-of-systems are collections of systems that can operate independently, but deliver more value when designed and operated as a synchronized ensemble. Traditional approaches to system design focus on a narrow system boundary (e.g. a single vehicle) and analyze the best design given performance targets. The objective of this type of analysis is to minimize the resources required to design and operate the system. For system-of-systems, however the complexity lies within the interaction of the systems and therefore the true objective is not available when analyzing a single system in isolation.

As we expand the definition of the system under consideration, we effectively enlarge the design control volume, which defines the boundary of inputs and outputs of the system. The interior of the control volume is the design space under consideration, where the designer can manipulate the components to achieve desired outputs, given the inherent physical constraints and the external constraints across the system boundary. As the control volume expands, greater flexibility in decisions is achieved, but with this flexibility comes an increase in problem size and complexity.

In order to analyze a system in a holistic fashion we must consider where to construct the control volume. Traditionally, even in systems engineering the design

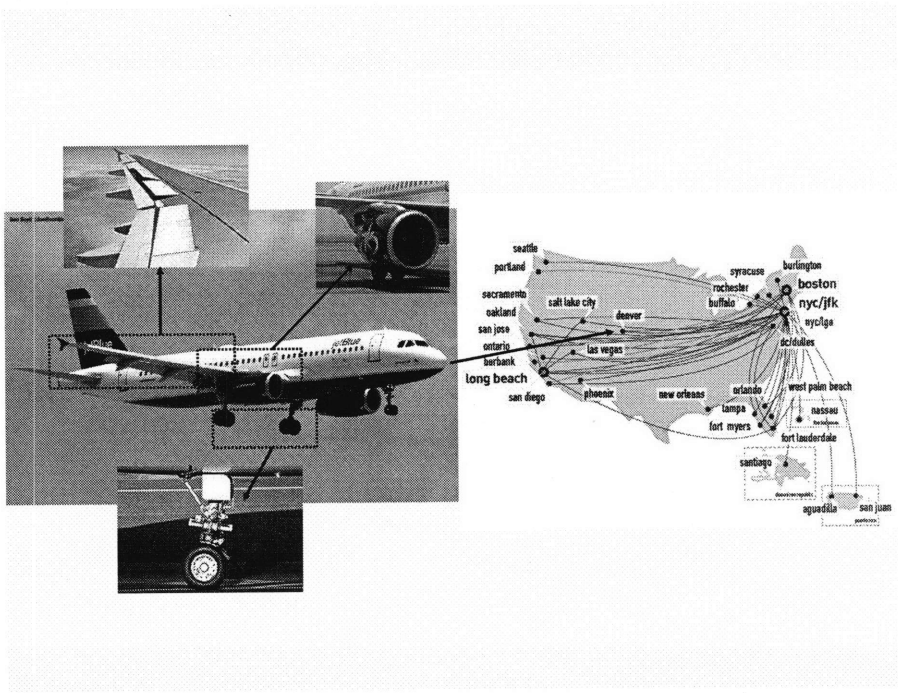


Figure 1-1: Left: Airbus 320[1] with specific sub-systems defined in greater detail as inserts. Right: Jet Blue[2] air transportation network

boundary has been limited to the vehicle design[6]. However for transportation systems, it is not simply the design of a single vehicle, but the interaction of multiple vehicles to achieve the desired mission objectives. By expanding the system definition to include not only the vehicle design but the network the vehicles travel through and the operations they perform, we can obtain a true system perspective of a transportation network.

Figure 1-1 depicts both an aircraft with sub-system components and an air transportation network that uses this aircraft. The control volume for vehicle design can be limited to any single sub-system, a limited interaction of sub-systems, or the entire vehicle design. Similarly, network optimization theory limits the control volume to encompass only the transportation network, with the vehicle designs as given inputs to the problem. Aircraft designers assume that network demand and routing are given and produce a vehicle design that satisfies the operational requirements, namely the *range and capacity*[6]. Operations researchers on the other hand often assume that vehicle specifications are known and held constant and seek to determine the best

allocation of the fleet[7]. In reality any vehicle is always a compromise design for its intended operations. This research therefore focuses on the concurrent optimization of the vehicle design and network flow, referred to as the integrated transportation system design problem, effectively enlarging the control volume to include all aspects of Figure 1-1.

The encompassing design space of the integrated transportation system design problem allows for the exploitation of the coupling inherent between the vehicle design and network flow. By increasing the decision space and explicitly modeling the coupling between the vehicle design and the interaction of a fleet of vehicles, efficient transportation architectures can be found that better satisfy the requirements of transportation systems.

### 1.1.1 Implications for Space Exploration

The integrated transportation system design problem is of particular interest when considering the opportunities available for the design of space transportation systems. With the announcement of the Vision for Space Exploration[8], we are given a new set of challenges for designing a new space transportation system. The directive given by President Bush dictates the design of a sustainable space exploration system for the Moon, Mars, and 'beyond'. Inherent to the problem of transporting people to destinations in space is sustaining the people and the operations while in transit and at the respective destinations. Especially for long-term missions, the amount of consumables required becomes a significant issue in terms of mass required in low Earth orbit (LEO). In order to develop a sustainable space transportation architecture it is critical that interplanetary supply chain logistics be considered[9].

To utilize the methods of terrestrial logistics in space mission design, it is necessary to shift the paradigm from single-point mission designs to an interplanetary network analysis. Realizing that the goal is no longer to develop an optimal system for a point design mission (such as Apollo) but to design a transportation architecture for a series of missions results in a reformulation of the design objectives. Viewing the set of missions together as multiple demands in a space network and analyzing the

effects of theses demands on the transportation architecture will potentially enable the production of a sustainable space exploration system that fulfills the given set of directives.

Unlike terrestrial logistics, space exploration logistics transportation systems have yet to be developed, which creates a unique opportunity to reduce operational inefficiencies in the transportation system by including the operational definition into the design process. By utilizing an integrated transportation system design approach, a significant advantage can be obtained in the design of the transportation vehicles that perform the logistics required for sustainable exploration. The increase in efficiency of the interplanetary logistics architecture may allow the goal of sustainable space exploration to be fulfilled.

## 1.2 Literature Review

The research into designing efficient transportation systems is ongoing from both the vehicle design and operations research prospective. While the literature in each of the two areas taken separately is voluminous, previous work associated with the *intersection* between vehicle design and network operations is surprisingly sparse. In this section a review of the relevant research in each of these areas is presented.

### 1.2.1 Vehicle Design

Aerospace vehicle design is a complex process, requiring the understanding of the physics involved with flight, and the numerous sub-systems required to obtain a feasible vehicle design. In *Raymer (1999)*[6] and *Anderson (1999)* [10] a guide to the conceptual design of an aircraft is presented. In both of these works, a detailed analysis of the mechanics of flight and the necessary sub-system design considerations are provided. For example, when defining the design of the wing, the structural requirements and aerodynamic considerations are detailed to determine the effect on aircraft performance. These works present the conceptual design process in the context of fulfilling a prescribed operations scenario which requires an iterative design process.

Traditionally, aircraft design research has focused on the design of an aircraft, given prescribed performance levels, such as required range and capacity. In *Aronstein and Schueler, (2005)*[11] the conceptual designs of two supersonic business aircraft are introduced. The aircraft design concepts are developed using a prescribed performance level and the results are compared. In *Neufeld and Chung (2005)*[12] the conceptual design of an unmanned aerial vehicle is explored using a combination of genetic algorithms and data mining. Again, the design parameters of the vehicle are optimized with respect to given operational performance parameters.

Recently, investigations into the design of an aircraft to fulfill multiple operations have been considered to understand the impact of these requirements on the vehicle design characteristics. *Frommer and Crossley, (2006)*,[13] compare the designs of fixed geometry and morphing geometry aircraft for satisfying multiple pre-defined operational scenarios. Additionally, the study compares the two aircraft designs when operating as a fleet for service in the operational scenarios, and thereby analyzes the effects of the fixed or morphing geometry on the system, not just the single vehicle design.

The effect of aircraft design on a fleet is further examined in *Crossley, et al. (2004)*[5] and *Crossley and Mane (2005)*[14] where the design of a passenger aircraft for a transportation system is considered. The network is pre-defined and two pre-existing aircraft types are available to satisfy passenger demand in a hub-spoke network of three cities. The objective is to optimize the design of a third new aircraft type and to allocate the newly composed fleet such that the demand is satisfied at the lowest total direct operating cost (DOC) for the system. Viewing the vehicle design within the context of a defined network creates a system of systems framework and allows the new vehicle design and routing to be optimized for the set of operations considered. This work is expanded in *Mane, et al. (2007)*[15] and *Mane (2005)*[16] where a 31-city problem with an existing fleet of eight different vehicle types are examined. Here, the results show not only a decrease in DOC by considering the design of a new vehicle within the fleet context but the scalability of the method for larger problems.

Research into the design of efficient space transportation vehicles is as numerous, since the cost of space travel is high and design budgets are tight. *Spacecraft Systems Engineering (1995)*[17] and *Space Mission Analysis and Design*[18] present complementary works that describe the conceptual design process for spacecraft. In both of these works, the design process is presented within the context of a mission analysis, which highlights the importance of understanding the operational requirements in order to produce a well defined design. These works begin with a background in astrodynamics in order to appropriately define the mission prior to vehicle design. Following a mission definition, a conceptual design process is presented that covers each step in the spacecraft vehicle design process, including all of the sub-system requirements that need to be examined.

Traditionally, the first step in defining a space mission is to define the trajectory to be examined. There are numerous works on trajectory analysis, such as *Battin (1999)*[19], and *Chobotov (1991)*[20], which discuss in detail the astrodynamics of space flight. Trajectory optimization presents a difficult optimization problem. In *Betts (1998), (1999)*[21][22] a discussion of potential optimization tools and analysis methods for trajectory optimization is presented. Given the mission details and the vehicle design parameters, the actual trajectory is then optimized to minimize the resources required to perform the mission or maximize the mission reliability.

*Walberg (1993)*[23] presents a classic spacecraft design analysis for a Mars mission. Different propulsion systems are considered and the resulting vehicle architectures are evaluated by comparing the total system mass in LEO required to fulfill the mission objectives. Every mission assumes a direct flight to Mars, and results in an architecture that promotes a high ratio of propulsive capability to propulsive mass requirements for the provided operational scenario.

In *Rowell, et al. (1999)*[24] the concurrent optimization of a launch vehicle and launch trajectory are discussed. By concurrently optimizing the control of the launch vehicle with the launch vehicle sizing and performance parameters, a better launch architecture can be developed. The recognition of the coupling inherent between the vehicle design and the trajectory allows for a more efficient launch system to be

developed.

As with aircraft design, recent investigations into the design of spacecraft to fulfill multiple missions has been considered. In *Cassady (1999)*[25], an overview of in-space propulsion technologies was conducted to determine the important technology areas and mission descriptions for each. Several teams were formed of various experts in the fields of reusable technologies, integral technologies, interplanetary transfer technologies and planetary capture and ascent/descent technologies. The outcome of this collaboration was a list of technologies that can be incorporated into spacecraft designs that would benefit a variety of mission scenarios, not just a single mission.

In *Meissinger and Collins (1999)* [26], an innovative method for designing an in-space propulsion system is discussed. The purpose of the orbit transfer vehicle (OTV) is to serve many small in-space missions. The paper proposes a set of missions and analyzes the design requirements, which defines the common elements for the basic spacecraft and the necessary attachments for the specific missions. By understanding the requirements of the future missions for the OTV, flexibility is built into the initial design, allowing for changes in mission requirements as time progresses.

In *Gonzalez, et al. (1998)*[27], the design of a spacecraft to fulfill multiple missions is formalized by utilizing a platform-variant decomposition. The relevant vehicle design variables are categorized into variables that are common to each mission, or platform variables, and variables that are unique to each mission, or variant variables. If a specific feature is categorized as a platform variable, the value can not be changed by a single designer, but must be re-negotiated by all members. Variant variables can be changed subsequently without consulting other missions, since these features are unique to the mission. Thus, the optimization is performed on two levels: the product family level, and the variant level. Although the method presented relies on a human interaction between iterations, it does provide a basis to adequately compare multiple designs and achieve results that are satisfactory to the program as a whole.

The new space exploration initiative has spawned a large amount of literature referring to the design of the space exploration system that will travel to both the Moon and Mars. *Wooster, et al. (2005)*[28] discusses a design approach for a crew

exploration vehicle (CEV), which will replace the Space Shuttle in the near term and function as a main component in space exploration. Using a decision tool known as object process networks (OPN) a large number of vehicle architectures can be evaluated to determine the best set of both physical vehicle parameters and associated operational architectures. *Stanley, et al. (2006)*[29] describes the process of designing the transportation architecture for the exploration missions. By defining a set of design reference missions, architectures are developed and evaluated against performance metrics such as cost and reliability. Again, the recognition that sustainable exploration requires extensible transportation systems that can be utilized to perform multiple missions is inherent in this analysis.

### 1.2.2 Network Optimization

The operations research community has produced a large amount of research focusing on efficient transportation system operations. Transportation systems are modeled using an underlying transportation network as the framework for analysis. The network model allows for a variety of transportation problems to be analyzed and solved. The classic network optimization problem formulation is the traveling salesman problem (TSP). *Lawler (1985)*[30] provides an extensive reference on the history of the TSP, multiple formulations of specific instances of the problem and the computational issues surrounding this combinatorial optimization problem. The TSP belongs to the class of combinatorial optimization problems that are defined as $\mathcal{NP}$-hard[30]. This definition states that for the general case, the computational effort grows exponentially with the number of cities to be visited.

Many practical applications that investigate the sequence of decisions have been modeled using the TSP. In *Alfriend and Lee (2002)*[31], the optimal servicing of geosynchronous satellites is considered. For small inclinations, an approximation to the distance between satellites is derived based on the angular momentum projection on the equatorial plane. Further approximations showed that the amount of $\Delta V$ required to perform the plane change is proportional to the distance between the angular momentum projections. Thus, when the objective of minimum fuel con-

sumption is considered, the problem effectively becomes the solution to the minimum distance problem, which can then be formulated as the TSP.

The dynamic TSP incorporates the element of time into the formulation. In *Takahashi (1998)*[32] a formulation of the dynamic TSP is developed to show the application of adaptive networks to dynamic optimization problems. In the dynamic TSP presented, the distance matrix between two cities is a function of time. The formulation of the problem then changes to impose additional constraints that relate the output at one time to the input at the next. In addition, the dynamics that determine how the distance matrix varies with time must be specified. Although some limiting constraints are placed on the problem, this paper provides a generalized formulation for the dynamic TSP and shows the usefulness of adaptive networks in solving them.

The TSP focuses on the routing of a single vehicle to meet the network demand. When more than one vehicle is considered, a transportation network routing problem is created. In transportation networks, the objective is to minimize the total cost of allocating vehicles in a fleet to routes in order to satisfy the network demand[33]. Additional constraints arise in order to track the fleet and ensure that the allocation is feasible given the limitations of the vehicle. *Toth and Vigo (2002)*[34] present a detailed examination of the vehicle routing problem (VRP), with corresponding models for many specific implementations with the associated constraints.

The vehicle routing problem is most often applied when examining logistics requirements. Logistics refers more broadly to the requirements associated with providing goods from manufacturers to customers, which includes efficient transportation of these goods[35]. Research into efficient transportation within a logistics network is often considered when examining the less than truckload (LTL) problem. Less than truckload refers to the size of a shipment that is smaller than the capacity of the transportation vehicle, thereby allowing multiple shipments to be carried by the same vehicle to reduce costs. *Chan, et al. (2002)*[36] examines the LTL problem an investigates efficient methods for obtaining solutions. *Simchi-Levi, et al. (1995)*[35] utilizes this analogy when examining the school bus routing problem for the New York City

public schools. In the school bus routing problem, the primary constraints focused on the timing restrictions inherent in school bus pick-ups and drop-offs, and the optimal routing and scheduling of the buses is determined. However, when considering the problem of optimizing the logistics for a transportation system, the capabilities and limitations of the vehicles available are defined as inputs to the problem.

Investigations into the optimization of the vehicle routing problem have been a prominent research topic for decades. In *Dantzig and Ramser (1959)*[37] initial investigations into the models required to develop the truck routing problem are developed. This work then discusses the solution of the resulting formulation by computational approaches. More recently, *Tan, et al. (2006)*[38] considers the trailer and truck routing problem. The objective of this problem is to determine the minimum cost routing of a fleet of trailer trucks that must pick-up a set of containers from multiple locations and return them to a warehouse. This paper investigates a hybrid multi-objective evolutionary algorithm to improve the computational efficiency of the optimization.

Transportation logistics is also necessary when defining aircraft routing. In *Barnhart, et al. (1998)*[7] the fleet assignment problem is solved in conjunction with the aircraft routing problem. By concurrently optimizing these two classes of problems, the optimal allocation of fleets to routes and the timing of each flight is determined to provide a more robust methodology for defining the flight scheduling for an airline. Alternatively, it may be desirable to define the flight paths and allocate aircraft to these flights to meet a prescribed demand at lowest cost. *Yang and Kornfeld (2003)*[39] examines the optimal allocation of a set of aircraft for an overnight package deliver system. Given a set of aircraft with different ranges, capacities and costs, the objective is to minimize the total network costs by choosing the vehicle routes through the network and allocating the appropriate vehicles to meet the given package demand. The resulting network is then compared to the traditional hub-spoke networks utilized for air transportation problems and shows a reduction in cost for the new aircraft routing.

## 1.3 Integrated Transportation System Design

The operations required for a transportation system to perform efficiently are dependent on the complex interactions of the vehicles in the fleet, which are significantly coupled to the vehicle design and performance parameters. As a pre-defined operations profile is often considered during the design process, the vehicle architecture is dependent on this profile. Similarly, the routing and allocation of a fleet of vehicles in a transportation system requires information about the vehicle's design limitations. However, since this process is performed sequentially, knowledge of the actual vehicle operations is not present during the design process, creating potential costly inefficiencies in the transportation architecture.

As shown in Section 1.2, a gap still exists between the vehicle design research and the network optimization research. Although recent research within the vehicle design community has expanded the traditional system design boundary to examine the interactions of a fleet in a network, the network definition is defined apriori.

Therefore, this research investigates the opportunities for efficient design when the operations of a fleet of vehicles are explicitly included in the vehicle design model. The integrated transportation system design problem removes the boundary between traditional vehicle design and traditional network flow optimization, creating an integrated model of a transportation system. The objective of this research is to create a methodology for examining transportation systems in a holistic framework and obtain more efficient designs. The key contributions of this work are as follows:

- Provide a formal definition and decomposition of the integrated transportation system design problem. Analyze the coupling relationships and mathematical structure of each component to provide insight into the complexities of modeling a transportation system.

- Clearly define a space transportation network by extending the definitions of a generic transportation network to account for differences in space transportation. Utilize the network modeling tool of time expanded networks to capture

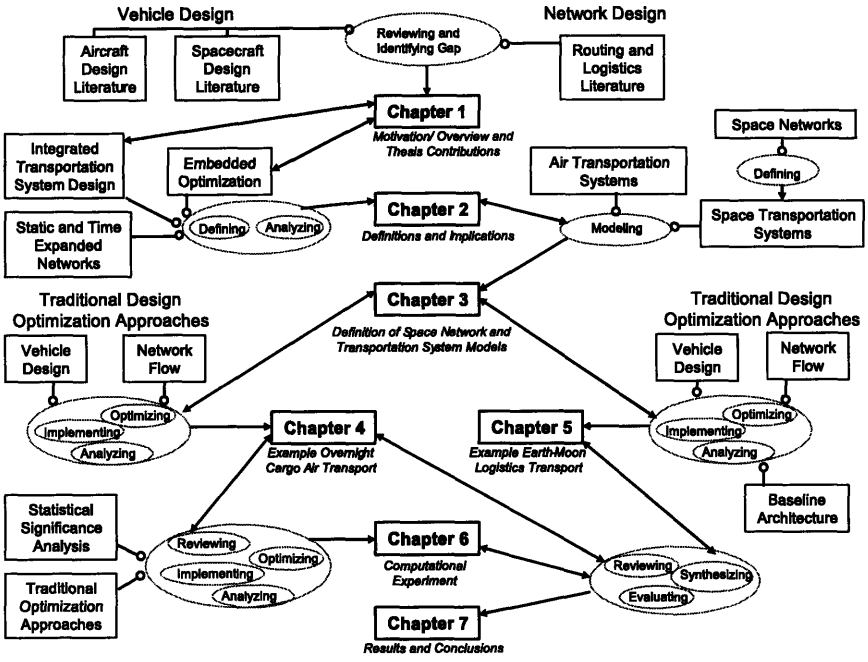


Figure 1-2: Object-Process Diagram Representation of this Dissertation (Thesis Roadmap)

the astrodynamic relationships of space transport and provide a mathematical description of the space network.

- Present two implementations of the integrated transportation system design problem. Utilize both an air transportation system and a space transportation system to show the benefits of the integrated transportation system design methodology as well as the applicability of the methodology to other domains.

- Develop an optimization methodology that effectively solves the integrated transportation system design problem. Validate the improvement in computational efficiency and scalability of this new method through a computational experiment comparing the new optimization methodology with traditional implementations.

### 1.3.1 Thesis Overview

Figure 1-2 provides a high-level overview of this dissertation using an Object-Process Diagram (OPD)[40]. In the first chapter, the motivations for defining the integrated transportation system design problem were explored with an emphasis on the potential impact for space exploration. A review of the research for both vehicle design and transportation networks was presented to identify the gap in the existing literature. Following this, the value of the integrated transportation system was then presented as the intersection of two often distinct views of transportation system design and the resulting contributions of the research were presented.

In Chapter 2, a concrete definition of the integrated transportation system design problem is presented as well as the complexities associated with solving the resulting implementations. To form an understanding of how the fleet operations and vehicle design are coupled, a clear distinction between the boundaries of the vehicle design, the network routing, and the operations must be created. These boundaries will clearly delineate the control volume of each component in the transportation system and most importantly, the intersection of these components, where the opportunity to improve efficiency lies.

Having defined the integrated transportation system design problem, Chapter 2 continues by analyzing the computational issues associated with the resulting problem size from the network model and the varied mathematical structure of the different components of the integrated transportation system design problem. To handle the complexities of the integrated transportation system design problem a new optimization methodology referred to as embedded optimization is presented that exploits the structure of the transportation system model.

The third chapter presents the integrated transportation system models for an air transportation network and a space transportation network. The air transportation system design problem is formulated based on the previous work of *Crossley, et al. (2004)*[5] and *Yang and Kornfeld (2003)*[39] and in accordance with the decomposition outlined in Chapter 2. To formulate the space transportation system models,

the definition of the space transportation network is presented. Drawing on analogies from terrestrial networks and extending these methods to incorporate the astrodynamic relationships required to perform space transportation, a formal definition of a space network is presented. Following this, the space transportation system models are presented.

In Chapter 4, the integrated air transportation system design problem is implemented for two example networks to define concurrent transportation architectures. The concurrent architectures are compared with traditional vehicle-only and network-only optimization, to illustrate the quantitative improvement of the integrated transportation system design methodology over current practice.

In Chapter 5, the integrated space transportation system design problem is implemented for an example space network and the concurrently optimized transportation architecture is defined. Again, to measure the value of the integrated transportation system design problem against current practice, the concurrently optimized solution is compared with traditional vehicle-only and network-only optimization approaches.

Chapter 6 presents a computational study to measure the benefits of the embedded optimization methodology developed in Chapter 2 and utilized to solve the integrated transportation system design problems presented in Chapters 4 and 5. To quantify the improvements in performance of the embedded optimization methodology, a computational experiment is performed to compare the performance of the system level optimizer, Simulated Annealing, with and without embedded optimization.

The final chapter reviews the key ideas and contributions presented in this dissertation and a discussion of the potential benefits and current limitations of the integrated transportation system design methodology follows. This chapter concludes with suggestions for future work in this research area.

# Chapter 2

# Integrated Transportation System Design Framework

The integrated transportation system design problem examines the coupling between the vehicle design and the interaction of the fleet. The motivation for examining a transportation system in this manner is to promote an efficient vehicle design that more accurately captures the requirements of the transportation system network. The challenge of this analysis is how to effectively describe, model and finally optimize the resulting problem, which can become very large and mathematically challenging to optimize.

This chapter focuses on the fundamental ideas developed to define and solve the integrated transportation system design problem. To illustrate the concepts presented in this chapter, a generic transportation system example will be utilized. Figure 2-1 depicts the example transportation system and the details of this system will be developed within the chapter. In Section 2.1 the integrated transportation system design problem is decomposed into the fundamental components that describe the complex system interactions. In Section 2.2 a brief background and nomenclature for network modeling is presented, and the complexities that arise from modeling transportation networks are discussed. Section 2.3 examines the limitations of existing methods for complex design problems and presents the embedded optimization methodology developed to handle the complexities of the integrated transportation

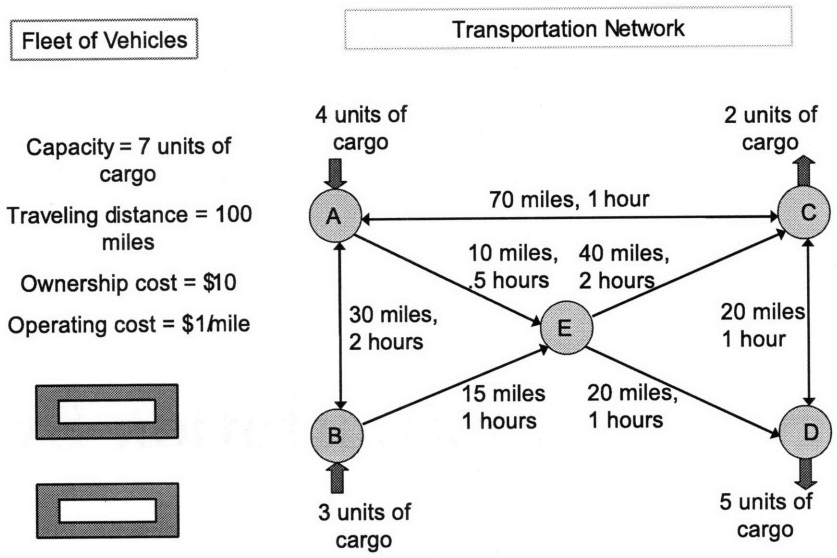


Figure 2-1: Generic Transportation System Example

system design problem.

## 2.1 Problem Decomposition

Transportation systems involve a complex interaction of a coordinated fleet of vehicles. Given a fleet of vehicles and a prescribed demand for service, each vehicle in the fleet can be assigned a route, assuming that the route is feasible given the vehicle's limitations. The integrated transportation system design problem provides insight into the interactions between the design of the vehicles within the fleet and the interaction of the fleet when performing the designated tasks.

The integrated transportation system design problem consists of four components: the transportation network flow, the vehicle design, the operations, and the system level objective. As shown in Figure 2-2, the vehicle design and the network flow are the sub-systems that determine the value of the transportation system as defined by the system-level objective, and the operations define the constraints that couple them. The following sub-sections provide a generalized definition of the models and

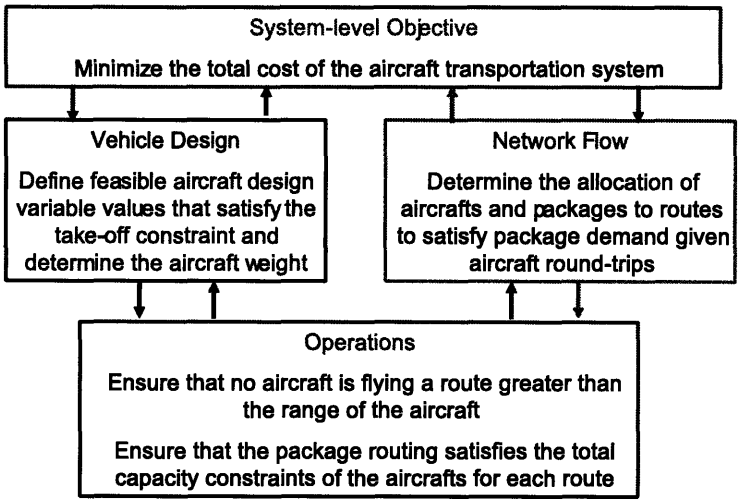


Figure 2-2: Diagram of the Integrated Transportation System Model

assumptions required to define each component of the integrated transportation system design architecture and utilize the transportation system defined in Figure 2-1 as an illustrative example.

### 2.1.1 Vehicle Design Models

The fundamental distinction of a transportation system from a network flow problem is the reliance on a set of transportation vehicles. The transportation vehicles have two functions. The first function of the vehicle is to hold cargo. In a transportation network, cargo must be contained within vehicles and the vehicles travel through the network. The second function of a vehicle in a transportation network is to provide the energy required to maneuver through the network. In order to travel within a transportation network, it is necessary to expend energy, generally in the form of propellant. Again, the vehicles and not the commodities, are responsible for performing these maneuvers.

Referring to Figure 2-1, the example fleet is composed of a single vehicle type and

two instances are available for use. The performance parameters for the vehicle design define the vehicle's ability to function as a cargo carrier and the vehicle's limitations, with respect to maneuvering through the network. In this example, the vehicle type defined has the ability to hold 7 units of cargo and has a maximum traveling distance of 100 miles.

Given the functions of a vehicle in a transportation system, the variables in the vehicle design sub-system define the vehicle design and performance parameters. Specifically, design variables that describe the physical characteristics, are required to define the level of functionality for carrying commodities. Design variables that describe the propulsive capability of the vehicle are necessary to define the vehicle's ability to maneuver within the network. The design variables defined in the vehicle design sub-system are denoted by $x_{veh}$.

The vehicle design vector can consist of variables that explicitly define the cargo capacity and propulsive capability, or indirectly define these quantities, by determining the size, shape, and propulsive force capabilities of the vehicle design, which can be related to the functional requirements through physical and dynamic relationships. In either case, however, when evaluating the operations of a fleet, the impact of the vehicle design on the transportation network is determined by the vehicle's ability to perform the two functions of carrying cargo and propulsive maneuverability.

The constraints within the vehicle design sub-system impose restrictions on feasible relationships between the vehicle design variables. The vehicle design constraints consist of both side constraints that define a permissable range for each of the vehicle design variables, and nonlinear constraints that govern multiple design variables ensuring a feasible vehicle architecture. This second set of constraints often poses the greatest difficulty for designing a feasible vehicle model as the complex interactions of the vehicle design variables can be difficult to satisfy.

In general, the vehicle design constraints can be represented as shown in Equations 2.1 and 2.2, where $x_{LB}$ and $x_{UB}$ represent the minimum and maximum permissable values of the vehicle design variables, and $g_{veh}()$ represents a general non-linear function of the vehicle design variables that expresses the vehicle design constraints

considered. Furthermore, although not explicitly depicted here, the vehicle design variables can be discrete, where the permissable range constitutes a set of feasible values.

$$x_{LB} \leq x_{veh} \leq x_{UB} \tag{2.1}$$

$$g_{veh}\left(x_{veh}\right) \leq 0 \tag{2.2}$$

### 2.1.2 Network Flow Model

The network provides the underlying architecture for the transportation system, which defines the information governing locations in the network and their connections. By defining a network, an inherent coupling of decisions is created, which results in the complexity of transportation systems. By examining Figure 2-1 we see that on the right hand side of the figure, a simple transportation network is defined. The arrows connecting the nodes define the allowable direction of transport between the nodes. The network flow model utilizes the network definition to determine the movement of both cargo and vehicles through the network.

In transportation systems, the goal is to determine the movement of cargo and vehicles within the network. Therefore, defining the subset of nodes visited or arcs traveled is insufficient. Instead, it is necessary to prescribe the *order* that the nodes are visited and the sequence of arcs traveled. For this reason, it is necessary to define paths or routes in the network to model the movement of cargo and vehicles in a transportation system.

The network flow model defines two sets of variables: cargo paths and vehicle routes. Cargo paths determine the sequence of locations that the cargo travels from origin to destination. Vehicle routes specify the assignment of each vehicle to travel a sequence of locations. In addition, the vehicle allocations, and potentially the cargo paths, are discrete units, generally requiring integrality of the network design variables. The cargo and vehicle variables in the network flow model can be denoted by $x_{net}$ and $y_{net}$, respectively.

Constraints in the network flow model govern feasibility of the cargo paths and vehicle routes. The cargo paths are required to satisfy the supply and demand constraints, as shown in Equation 2.3.

$$\begin{aligned} h_{supply}(x_{net}) &= s_i \ \forall i \\ h_{demand}(x_{net}) &= d_j \ \forall j \end{aligned} \tag{2.3}$$

Here, $h_{supply}()$ represents a generic linear function of the cargo path variables that expresses the total amount of cargo originating at node $i$ and $h_{demand}()$ is a generic linear function expressing the demand terminating at node $j$ and $s_i$ and $d_j$ define the supply and demand at nodes $i$ and $j$, respectively.

By examining Figure 2-1 we see that node A has a supply of 4 units, node B has a supply of 3 units, node C has a demand of 2 units and node D has a demand of 5 units. The supply and demand for all other nodes is zero. Formulating the constraints defined in Equation 2.3 for this example would ensure that the total amount of cargo starting and ending at each node satisfies the supply and demand of the network.

Vehicle allocation constraints ensure that the vehicle assignments are feasible. General vehicle allocation constraints limit the assignment of each vehicle to no more than one route in the network at a time (first equation in Equation 2.4); however additional constraints, such as limiting the total number of vehicles available for assignment, can be added as required by the specific transportation system implementation (second equation in Equation 2.4).

$$\begin{aligned} h_{route}(y_{net}) &\leq 1 \\ h_{max}(y_{net}) &\leq N_{max} \end{aligned} \tag{2.4}$$

Here, $h_{route}()$ represents a linear function of the vehicle route variables that expresses that each vehicle travel no more than one route. The first set of equations ensures that no more than one vehicle route is selected for every vehicle, and is imposed for every vehicle in the fleet. For the example presented in Figure 2-1, this equation yields two constraints, one for each vehicle. In the second equation, $h_{max}()$ expresses the total number of vehicles traveling within the network and limits this number to be less than the maximum number of vehicles available ($N_{max}$). For the

example presented in Figure 2-1, the maximum number of vehicles available for use is two.

### 2.1.3 Operations Model

The operations model defines the actual operations of each vehicle for each route, given the vehicle design parameters. Additional design variables specific to the operations can be defined and can potentially influence the objective function. These design variables, if they are defined, increase the flexibility of the transportation system by increasing the specificity of the architecture for the example considered. For example, the amount of propellant provided for each vehicle can be different, but must be within the limits provided by the vehicle design.

The operations constraints define the coupling between the vehicle design model and the network flow model. In this set of constraints, the actual operations of a vehicle are evaluated for the prescribed allocation. The vehicle has the associated physical and performance characteristics determined by the vehicle design and these limitations are compared to the requirements specified by the cargo paths and vehicle routes.

The operational constraints consist of two distinct sets of constraints: *capacity and capability.* Capacity constraints determine if the cargo can be contained within the vehicles, given each vehicle's capacity and the locations of vehicles as determined by the vehicle routes. For the integrated transportation system design problem, this set of constraints is non-linear, since both vehicle capacity and vehicle routes are design variables. Equation 2.5 specifies the set of generic capacity constraints that must be enforced everywhere within the network.

$$h_{mcap}\left(\alpha_{cargo}, x_{net}\right) \leq g_{mcap}\left(x_{veh}, y_{net}\right) \tag{2.5}$$

Again, $h_{mcap}()$ describes a linear function of the cargo path variables $x_{net}$ as well as relevant properties of the cargo ($\alpha_{cargo}$) to determine the actual amount of cargo that must be contained. The right hand side of these equations specifies a

non-linear function ($g_{mcap}()$) of the vehicle design variables and the routes of every vehicle instantiated. These constraints enforce that the total carrier functionality of all vehicles allocated is sufficient for holding the cargo everywhere within the network.

In the example presented in Figure 2-1, the total amount of cargo transported through the network is 7 units and the capacity of each vehicle in the fleet is 7 units. Therefore, for this example, the capacity constraints simply require that the cargo paths and vehicle routes coincide. However, if the capacity of the vehicles were to decrease or the demand in the network were to increase, the capacity constraints would be more difficult to satisfy.

The capability constraints ensure that every vehicle is capable of performing the assigned operations. In order to travel between two nodes in a transportation network, a vehicle must have the propulsive capability required to transport the arc. The propulsive capability can be influenced by the vehicle design, cargo paths and potentially other vehicle routes. Equation 2.6 specifies the set of constraints that must be enforced for every route and vehicle.

$$g_{cap}\left(\alpha_{cargo}, x_{net}, x_{veh}, y_{net}\right) \leq g_{vcap}\left(x_{veh}, y_{net}\right) \tag{2.6}$$

The left-hand side of Equation 2.6 is a non-linear function of the cargo path variables, vehicle design variables and vehicle route variables. The right-hand side of the equation is a non-linear function of the vehicle design and routing variables, which defines the total propulsive capability present for each vehicle and each route. Since the capability of a vehicle is often measured by the amount of propellant the vehicle contains, the ability of the vehicle to traverse the route is influenced by the potential to refuel at intermediate points along the route. If a vehicle can refuel at any node within the network, then the capability constraints examine the ability of the vehicle to traverse all arcs it transports across. If there exists refueling points along the arcs, as seen when modeling car and truck routes, the capability constraints become trivial. If however, there are no opportunities for refueling, as exists in space transportation systems, the capability constraints become extremely difficult to satisfy.

Examining Figure 2-1, the capability of each vehicle in the fleet is defined by the

traveling distance (100 miles). In this example, the definition of a fixed traveling distance decouples the capability of the vehicle from the cargo and vehicle paths. Thus, the capability of the vehicle is simply defined by the vehicle design variables. This simplifying assumption is reasonable if the other factors influencing the vehicle's capability, namely the cargo routes, do not change the capability significantly. With a traveling distance of 100 miles, the vehicle is capable of transporting across every arc within the network. Therefore, if the we assume that there exists refueling opportunities at every node within the network, it is not necessary to evaluate the capability constraints. If, however, refueling opportunities do not exist, this significantly impacts the vehicle's ability to transport cargo. For example, a single vehicle can not travel from node B to node A to node C to node D as the total distance of this route is 120 miles, which exceeds the traveling distance of the vehicle. Therefore, under the assumption of no re-fueling, either another route must be selected or multiple vehicles must be employed to transport the cargo.

### 2.1.4 Objective Function

For transportation systems, multiple configurations exist that satisfy all of the requirements specified by the sub-system constraints. Since resources are limited, the objective is to find the configuration that requires the least amount of resources to satisfy the demand requirements. Alternatively, for high-value delivery systems, the objective could be reformulated to maximize robustness to failure. Focusing on limited resource utilization, as we are considering the interaction of multiple vehicles within a network, this implies that the objective is to minimize the total system resource usage, not just the resources required to satisfy a single demand point. Equation 2.7 defines the generic objective function for the transportation system.

$$\mathbf{min}\ \mathcal{J} = g\left(x_{veh}, y_{veh}\right) \tag{2.7}$$

Equation 2.7 describes a non-linear objective function that combine the variables that describe the vehicle design and the vehicle routing variables. As described in

the vehicle design sub-section, the vehicle design variables are utilized to define an associated value corresponding to the vehicle design. Combined with the routing and allocation variables, the resource requirements for the transportation architecture can be evaluated within the system objective.

The objective of minimal resource utilization often translates into minimum system cost. Figure 2-3 provides a table listing some of the representative costs associated with transportation system designs. The first column denotes a title for the cost breakdown, the second column lists the associated costs for which the given category represents, and the third column describes how each cost is represented in the transportation architecture. As we can see by examining Figure 2-3, two costs associated with the vehicle design are defined. The fixed cost of the vehicle is a cost for assigning a vehicle to travel *any* route within the network. The variable cost of the vehicle is a cost for assigning a vehicle to a specific route, where the accumulation of the cost is dependent on the usage of the vehicle. The final row in this table examines the infrastructure costs associated with transportation networks. These costs are generally not captured when examining the design and allocation of a fleet to satisfy demand, but are commonly utilized to design the network (i.e. determine where to define new nodes and arcs).

Examining Figure 2-1 reveals both the fixed and variable cost assignments for the example problem. For each vehicle, a \$10 cost is incurred for utilizing a vehicle to transport the cargo and for each mile traveled, an additional cost of \$1 is imposed.

Cost can be difficult to represent accurately for a vehicle design, given the design parameters of the problem. In these cases, a representative cost metric can be utilized to evaluate the transportation architecture. The objective function consists of the vehicle design variables and the allocation variables, and therefore represents the resources required to operate the fleet. Costs associated with packages are generally ignored, since the total amount of cargo traveling through the network is independent of the transportation architecture, and the relative costs are small, as compared to the vehicle costs.

| Cost Breakdown | Representation | Modeling Parameter |
|---|---|---|
| **Acquisition Costs** | Design and Development Costs | Fixed Vehicle Design Costs |
| **Fixed Operating Costs** | Standing Army, Depreciation of Investment | Fixed Vehicle Design Costs |
| **Variable Operating Costs** | Fueling, Maintenance, Operating Personnel | Variable Vehicle Design Costs |
| **Infrastructure** | Warehousing, Maintenance Facilities, Pathways (ex. Railroad Tracks) | Additional Costs (Often not captured directly) |

Figure 2-3: Breakdown of Costs for Transportation System Modeling

### 2.1.5 Complexity of the Integrated Transportation System Design Problem

The goal for a transportation system is not simply to minimize the distance each vehicle travels. Instead, the goal is to minimize the resources required to transport all of the cargo from the supply nodes to the demand nodes, which requires analyzing the interactions of the fleet that transports them. For the example in Figure 2-1 there is a choice of utilizing a single vehicle to delivery all of the cargo to the demand locations (since the capacity of a single vehicle is large enough to hold all of the cargo) or utilizing two vehicles, creating separate pick-ups or deliveries. For this simple example, the options can be enumerated, as shown in Figure 2-4.

Figure 2-4 shows nine options for satisfying the cargo demand where the use of a single vehicle or two vehicles is considered. The path defines the sequence of nodes visited by the vehicle(s) and the accumulated travel distance and corresponding travel time are computed. The total distance refers to the distance traveled by all vehicles in the transportation system, however the total time refers to the maximum amount

| Routing Options | Path | Distance Traveled | Travel Time | Total Distance | Total Time | Total Cost |
|---|---|---|---|---|---|---|
| Option 1: 1 vehicle | B-A-C-D | 120 miles | 4 hours | 120 miles | 4 hours | $130 |
| Option 2: 1 vehicle | A-B-E-C-D | 105 miles | 6 hours | 105 miles | 6 hours | $115 |
| Option 3: 1 vehicle | A-B-E-D-C | 85 miles | 5 hours | 85 miles | 5 hours | $95 |
| Option 4: 1 vehicle | B-A-E-C-D | 100 miles | 5.5 hours | 100 miles | 5.5 hours | $110 |
| Option 5: 1 vehicle | B-A-E-D-C | 80 miles | 4.5 hours | 80 miles | 4.5 hours | $90 |
| Option 6: 2 vehicles | A-C-D | 90 miles | 2 hours | 125 miles | 2 hours | $145 |
| | B-E-D | 35 miles | 2 hours | | | |
| Option 7: 2 vehicles | A-E-C-D | 70 miles | 3.5 hours | 105 miles | 3.5 hours | $135 |
| | B-E-D | 35 miles | 2 hours | | | |
| Option 8: 2 vehicles | A-E-D | 30 miles | 1.5 hours | 105 miles | 4 hours | $135 |
| | B-E-C-D | 75 miles | 4 hours | | | |
| Option 9: 2 vehicles | A-E-D | 30 miles | 1.5 hours | 85 miles | 4 hours | $105 |
| | B-E-D-C | 55 miles | 4 hours | | | |

Figure 2-4: Table of Feasible Architectures for Example Transportation System

of time required to deliver the cargo, assuming both vehicles operate simultaneously. The total cost of each option is computed using the cost parameters defined in Figure 2-1.

Given that for this example, there are no constraints on delivery time implemented, the goal is to minimize the total system cost. Therefore Option 5, which uses a single vehicle to deliver all commodities would be selected with a total cost of $90. It is important to realize that this solution may vary for different vehicle architectures since a change in vehicle capacity, traveling distance, or the fixed or variable costs significantly influence the decision. It is this coupling that the integrated transportation system design problem, as presented in this thesis investigates.

The integrated transportation system design problem includes the vehicle design variables with the network flow variables to obtain a transportation architecture that efficiently sizes the transportation vehicles to meet the demand of the network at lowest cost. Improvements in efficiency are obtained by specifying the cargo capacity and propulsive capability to match the requirements of the routes selected. Changes in the vehicle design can alter both the fixed and variable costs of the vehicle which in

turn may promote a different route selection. As these decisions are highly coupled, the integrated transportation system design problem *concurrently* designs the vehicle and network flow variables to obtain efficient transportation architectures.

## 2.2 Network Theory and Development

In the previous section, the network sub-system describes the requirements of modeling the interactions within a transportation network. As the underlying network defined for the transportation system is fundamental in the decision process, a clear definition of a network, and the implications for transportation systems is desirable. Therefore, in this section, a brief development of the nomenclature necessary to describe the network is presented as well as a discussion of the implications that network modeling has on problem size. The network model described in Figure 2-1 will be utilized as the example network in this section.

### 2.2.1 Physical Networks

The physical network represents the interactions of points of interest within a transportation system. The locations of interest, or nodes, can be classified into three categories: supply nodes, demand nodes and transshipment nodes. Supply nodes designate nodes within a network where the cargo originates. Demand nodes designate nodes within the network where cargo is required. Transshipment nodes are way-points within the network, that posses neither a supply nor a demand for cargo, but can be utilized as intermediate points on route. Figure 2-1, diagrams a small transportation network with five nodes. In this network, nodes A and B are supply nodes and nodes C and D are demand nodes. Node E has neither supply or demand, and therefore is classified as a transshipment node.

Arcs define the connections between pairs of nodes and are represented as arrows to describe the direction the arc can be traveled. For example, in Figure 2-1 the arc from node B to E can only be traveled in that direction, however, the arc between nodes A and B can be traversed in either direction. The arcs in a transportation

system have an associated cost for traveling the arc, such as the distance between the two nodes. In Figure 2-1 there are ten arcs, each with an associated distance. For example, the arc between nodes A and B has a length of 30 miles. Arcs in a transportation network can also have an associated time, which corresponds to the time required to traverse the arc. The arc between nodes A and B in Figure 2-1 requires 2 hours to traverse.

When defining the network flow model in the previous section, the variables consisted of both cargo paths and vehicle routes. A route or a path through the network develops when a sequence of arcs is connected. The route or path specifies the order in which some or all of the nodes in the network are visited. As additional cost is incurred by traveling each arc, it is reasonable to assume that a path consists of a set of unique nodes and arcs, which defines an acyclic path.

As the location of the cargo supply and demand nodes is known, defining cargo paths corresponding to these locations simplifies the number of variables considered. Therefore, all cargo paths originate at a cargo supply node and terminate at a cargo demand node. For the example presented in Figure 2-1, the first node in every cargo path is either node A or node B, as both nodes have cargo supply, and the last node in every path is either node C or node D, as both nodes have cargo demand.

The number of vehicle paths that must be considered is potentially much larger. Depending on the problem, it may not be reasonable to restrict vehicle paths to originate at cargo supply nodes or terminate at cargo demand nodes. Furthermore, it may be necessary to require the vehicles to return to their initial locations at the end of the path. However, if the freedom to begin and end vehicle routes at any node in the network exists, it is then reasonable to assume that all vehicle paths begin and end at cargo supply and demand nodes, respectively. Examining Figure 2-1, we see that without additional restrictions on vehicle paths, it is wise to only define vehicle paths that originate at nodes A or B, where the cargo would be loaded, and terminate at nodes C or D, where the cargo would be un-loaded.

### 2.2.2 Time Expanded Networks

In network problems, where operational decisions are time-dependent, it is often useful to represent the decision framework using a time expanded network[41]. In transportation systems, arcs in the network often have an associated travel time, as well as an associated travel cost, as shown in Figure 2-1. In order to represent these networks in a framework convenient for analysis, a time expanded network is employed. Time expanded networks allow for the timing decisions to be explicitly expressed in the network definition, and thereby decouple timing from the network properties[33]. Thus, the time expanded network expresses the evolution of the static network in time.

The network shown in Figure 2-1 is classified as a *static* network, where the definition of the network is not related to time. Although the arcs in the network specify an associated travel time, time is only recorded, not utilized for decision making purposes. Some transportation system problems require operational decisions that are time dependent. For example, when considering the options for transporting cargo (Figure 2-4), each option assumes that all of the cargo a vehicle loads at the supply nodes is delivered by that vehicle at the demand nodes. Alternatively, each vehicle could pick-up cargo at a single supply node, and deliver cargo to a single demand node, however this option would require the vehicles to meet at a specific place and time and transfer some of the cargo. Coordination of the vehicles requires time to be explicitly stated in the transportation system definition, and therefore a time expanded network would be created to conveniently capture these interactions.

A time expanded network is generated from a static network as follows. The time horizon $\tau$ is discretized into $T$ time periods of length $\delta t$. Every node in the static network is replicated at each of the time periods. Thus for every node $i \in \mathcal{N}$, where $\mathcal{N}$ is the set of nodes, a node $(i,t)$ is created for every $t = 1 \ldots T$. For every transport arc $(i,j)$ in the static network, a transport arc $((i,t),(j,\bar{t}))$ is created where $\bar{t} - t = t_{ij}$ and $t_{ij}$ is the travel time of arc $(i,j)$ in the static network. The cost of each arc in the static network is replicated for each corresponding arc in the time expanded network.

The discretization of the time horizon is chosen to best balance the desire to accurately represent the dynamics of the transportation network with the desire to minimize the size of the resulting time expanded network for computation purposes. Therefore, in some cases, the travel time of an arc may not correspond to a multiple of the time period length. In these cases, the travel time for the arc is rounded up to the next time period, and the arc is replicated using this new transport time. For example, in Figure 2-1, the arc from node A to node E has a transport time of .5 hours. However, all other arcs in the transportation network have travel times in whole hour increments. Therefore, when defining the time horizon discretization, if a one hour time period length is selected ($\delta t = 1\text{hr}$), the transport time of the arc from node A to node E is subsequently rounded up to be one hour in order to fit within the time expanded network framework.

In addition to the transport arcs, waiting arcs are also included in the time expanded network. Waiting arcs specify the transport between the same static node at consecutive time periods. These arcs are generally excluded from the definition of the corresponding static network, but appear in the time expanded network to represent the operation of remaining at the same physical node for a finite duration of time. A waiting arc connects a node $(i, t)$ to the node $(i, t + \delta t))$ for every node $i$ where waiting can occur and every time period $t = 1 \ldots T$.

Using the rules to generate a time expanded network described above, the static network in Figure 2-1 is transformed into the time expanded network shown in Figure 2-5. Here, a time horizon of 4 is discretized into 4 time periods using a time period length of one hour. The horizontal arcs are the waiting arcs connecting the same physical node at consecutive time periods. The remaining arcs are the transport arcs. All arcs are defined from left to right, signifying forward movement in time. As such, the time expanded network is by definition acyclic.

### Time Expanded Network Size

The time expanded network can be considerably larger than the corresponding static network. For a static network with $n$ nodes, the corresponding time expanded net-

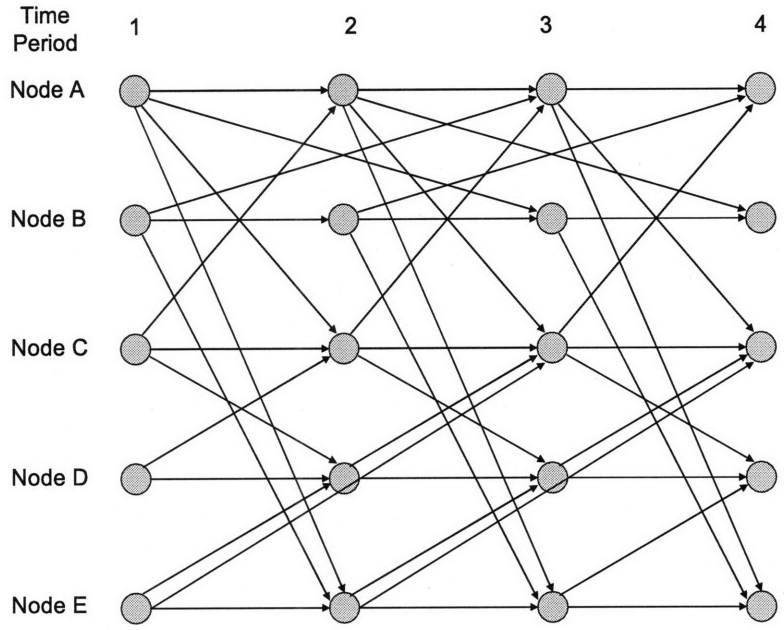


Figure 2-5: Visualization of Time Expanded Network

work has $nT$ nodes, where $T$ is the number of time periods. The number of arcs generated in the time expanded network is slightly more complicated to evaluate, as it is dependent on both the number of arcs in the static network, the number of time periods considered, *and the transport times for every arc.* The number of waiting arcs ($m_{wait}$) generated in the time expanded network is defined in Equation 2.8.

$$m_{wait} = n\,(T-1) \tag{2.8}$$

Equation 2.8 assumes that waiting arcs can be generated for every node in the static network. To compute the number of transport arcs in the time expanded network, it is first necessary to group the arcs in the static network by transport time. After rounding all static arc travel times to fit within the discretization, the number of arcs can be grouped by travel time. Defining $m_i$ as the number of arcs in the static network with transport time $i\delta t$ (for $i = 1 \ldots T-1$) the total number of transport arcs in the time expanded network is defined by 2.9.

$$m_{trans} = \sum_{i=1}^{T-1} m_i \left\lceil \frac{T-1}{i} \right\rceil \tag{2.9}$$

where

$$\sum_{i}^{T-1} m_i = m. \tag{2.10}$$

The total number of arcs a time expanded network ($m_{TEN}$) can therefore be expressed as

$$m_{TEN} = n\,(T-1) + \sum_{i=1}^{T-1} m_i \left\lceil \frac{T-1}{i} \right\rceil \tag{2.11}$$

The time expanded network shown in Figure 2-5 has 20 nodes and 42 arcs. Applying Equation 2.8, we determine that 15 of these arcs are waiting arcs. To demonstrate Equation 2.9, we classify 7 arcs in the static network (Figure 2-1 with a travel time of $\delta t$ (one hour), and therefore $m_1 = 7$. This includes the arc from node A to node E, as the travel time is rounded up to the next $\delta t$ multiple. The 3 remaining arcs have a travel time of $2\delta t$, and therefore $m_2 = 3$. Applying Equation 2.9, we confirm that 27 arcs in the time expanded network shown in Figure 2-5 are transportation arcs.

### Utility of Time Expanded Network for Transportation Systems

Time expanded networks are a valuable modeling tool when timing constraints enter the problem formulation. For example, if a restriction on the cargo delivery time is required, it may be necessary to eliminate some of the options presented in Figure 2-4. For example, if all cargo must be delivered within 3 hours, only one option (Option 6) is available for satisfying this additional constraint. However, since this constraint is not tight, there are a number of options available for routing the vehicles using the sequence of nodes defined by option 6. Figure 2-6 highlights the available paths in the time expanded network that deliver the cargo within the restricted time frame, and the corresponding paths are listed in Figure 2-7.

Examining Figure 2-7 reveals that three paths are available in the time expanded network for *each* vehicle route. As any combination is possible, this allows nine vehicle routing options to satisfy the demand. Therefore, even for small transportation networks, operational decisions dependent on timing introduce a significant amount of complexity into the formulation and solution.

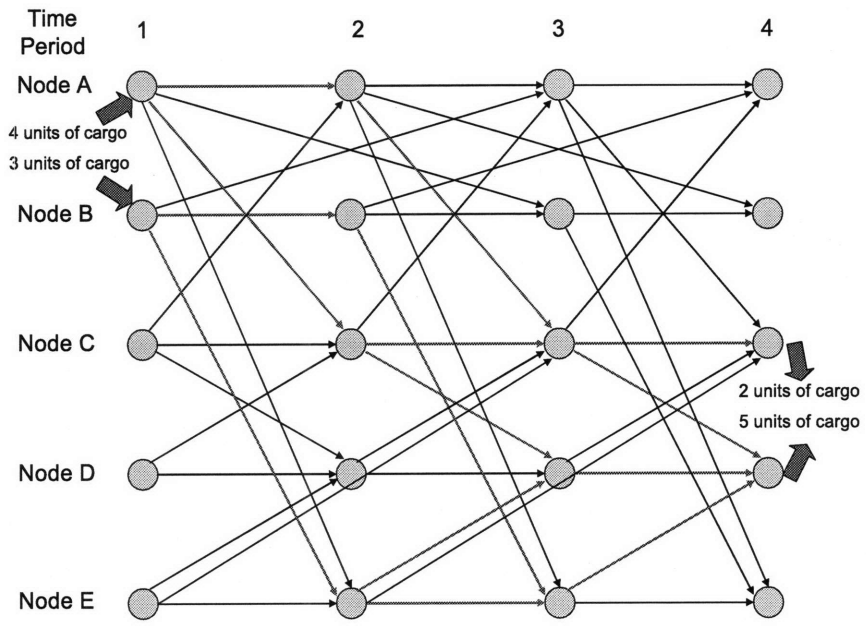


Figure 2-6: Time Expanded Network with a Sub-set of Paths Highlighted

| Routing Options | Path | Travel Time |
|---|---|---|
| 1st Vehicle | | |
| Option 1 | A,1 - C,2 - D,3 | 2 hours |
| Option 2 | A,1 - A,2 - C,3 - D,4 | 3 hours |
| Option 3 | A,1 - C,2 - C,3 - D,4 | 3 hours |
| 2nd Vehicle | | |
| Option 1 | B,1 - E,2 - D,3 | 2 hours |
| Option 2 | B,1 - B,2 - E,3 - D,4 | 3 hours |
| Option 3 | B,1 - E,2 - E,3 - D,4 | 3 hours |

Figure 2-7: Table of Available Paths for Example Transportation Network with Delivery Time Restrictions

### 2.2.3 Growth of Problem Size in Transportation Networks

In transportation systems, the goal is to determine the routes or paths traveled by the cargo and vehicles in the network. A path is a sequence of nodes that the vehicle or cargo travels. For cargo, the path originates at the supply node and terminates at the demand node. For vehicles however, a starting or ending node may not be specified. Therefore, every path in the network is a potential vehicle path. In order to understand how the problem size of a transportation network increases with the number of nodes, we consider a general network and count the number of paths that arise.

For a fully connected, directional network of $n$ nodes, the number of acyclic paths can be computed. A fully connected network is a network where an arc exists between every pair of nodes, yielding $m = n(n-1)$ arcs, where $n$ is the number of nodes in the network. An acyclic path implies that nodes are not revisited in a path. Given this assumption, the number of nodes in path $p$ is limited to $2 \leq |p| \leq n$. The number of resulting paths in the network is

$$\sum_{r=2}^{n} \frac{n!}{(n-r)!} \tag{2.12}$$

The assumption of full connectivity is important, as the reduction of only a few arcs can reduce the number of paths. A five node network with full connectivity has 20 arcs which results in 200 available paths. The network shown in Figure 2-1 has ten arcs, however the number of paths in this network is only 45. Therefore, the degree of connectivity of the network is important when defining a transportation network, as the number of paths considered is highly dependent on the number of connections in the network.

The number of paths in a time expanded network is much greater than the number of paths in the corresponding static network. Although every path can not be replicated at every time period, additional paths are created. The waiting arcs defined in the above section become additional arcs in the time expanded network. Although the time expanded network has a lower level of connectivity (due to only forward time arcs), the increase in the number of nodes creates a large number of paths for

consideration. Given a fully connected static network with the travel time of each arc defined as $\delta t$, and generating a time expanded network for $T$ time periods, the total number of resulting paths can be counted by applying Equation 2.13.

$$n(n-1)\left[(T-1)+(n+1)\sum_{r=3}^{T} n^{r-3}(T-r+1)\right] \tag{2.13}$$

Equation 2.13 computes all paths $p$, where $2 \leq |p| \leq T$, as there can be as many nodes in a path as there are time periods. In addition, this expression assumes that all paths involve at least one transport arc. Utilizing this expression, a fully connected static network with 5 static nodes that is expanded for four time periods has 1060 paths. Thus, even for small problems, the number of paths becomes large, resulting in many network flow variables. This analysis, however, only provides an upper bound on the number of paths in the time expanded network. Examining the time expanded network provided in Figure 2-5 reveals a total of 159 paths. Although this is considerably greater than the 45 paths defined in the corresponding static network, the lower level of connectivity in the static network and the longer travel times of the arcs provides a great reduction in the number of paths, and therefore the number of potential network flow variables.

## 2.3 Optimization Challenges

The integrated transportation system design problem enlarges the design space and creates new opportunities to optimize the entire system design and obtain better transportation system architectures. With the enlarged design space resulting from the integration of the different sub-systems, an inherent complexity arises in the formulation and optimization of the design problem.

The different sub-systems often have different *mathematical structures* which require specific optimization methods to efficiently obtain solutions. In these cases, combining all variables and constraints of the multi-disciplinary problem into a single aggregate design problem requires the use of the most general optimization algorithms which may require computation times that are impractical.

In order to handle the complexity and problem size of the integrated transportation system design problem, a new implementation of Simulated Annealing is proposed. The integrated transportation system design problem is decomposed by mathematical structure, instead of by function or sub-system definition. For each set of constraints, an appropriate optimizer is embedded within the system optimizer to find feasible solutions that allow the system optimizer to explore the design space. The result is a design space exploration tool that produces feasible solutions using deterministic software.

### 2.3.1 Motivation for Embedded Optimization

The integrated transportation system design problem, as described in Section 2.1 above, assimilates all of the variables and constraints into a single system level problem, which optimizes the vehicle design and the network routing concurrently. As such, the design problem can be classified as a mixed integer non-linear programming problem (MINLP), which is difficult to solve effectively [42],[43]. Typically, either simplifications are made to the constraints, or the problem is decomposed using methods such as Collaborative Optimization [44].

For system-of-systems design problems, Collaborative Optimization decomposes the problem into system and sub-system levels. The sub-systems are separated along disciplinary boundaries which allows for the utilization of optimization methods specific to the formulation of the particular sub-system. The sub-systems are related to the system by compatibility constraints that require the system level variables and the sub-system-level variables match. Thus, the problem fundamentally explores the design space by moving from infeasibility to feasibility or coherence of the design and consequently optimality. Although an effective method for sub-system design, the compatibility constraints at the system level can be difficult to satisfy and can present a challenge when optimizing complex, real-world systems.

In *Crossley, et al (2004)*[5], the design of a new aircraft within the context of a pre-defined fleet was examined for a small aircraft network where a hub-spoke network was assumed. Thus, although the allocation of the newly composed fleet was

considered concurrently with the design of a new aircraft, the routing of the fleet was pre-defined. In this paper, the Collaborative Optimization framework was utilized to solve the resulting MINLP as the design and allocation problems can be considered as two-different sub-systems. Utilizing the Collaborative Optimization framework produced solutions that were in most cases comparable to those obtained by other MINLP optimization algorithms but with less computational effort. However, for this decomposition approach to work, significant effort is required to appropriately decompose the problem and scaling issues can arise.

The integrated air transportation system design problem presented in Chapters 3 and 4 was originally implemented using the decomposition approach of Collaborative Optimization. As the system objective is a non-linear function of mixed-integer variables, Simulated Annealing was chosen as the system optimizer. The constraints at the system level were the compatibility constraints for the sub-system designs, and were formulated as penalties in the objective function, in accordance with the methodology of Collaborative Optimization. The vehicle design model was optimized using Simulated Annealing and CPLEX was utilized to solve the combined network flow and operations models. Although the optimization at the sub-system levels worked, the system level optimizer would not converge on a feasible architecture. Thus, when considering the design of larger mixed-integer programming problems, it is desirable to find good decomposition approaches that are robust to the problem definition.

For single-level system of systems problems, the variety of mathematical structure in the constraints can often cause inefficient or ineffective solution methodologies to be utilized because the most general optimizer is required to handle all of the constraints and variables. In the integrated transportation system design problem, both continuous and mixed-integer variables are defined to represent the vehicle design and network flow variables and the constraints include the linear network flow constraints and the non-linear vehicle design and operations constraints. *As the generality of the problem increases, the effectiveness of the solution methods available to optimize them decreases.* For the most general problems, heuristic methods, such as Simulated Annealing, have the capability of handling any problem structure.

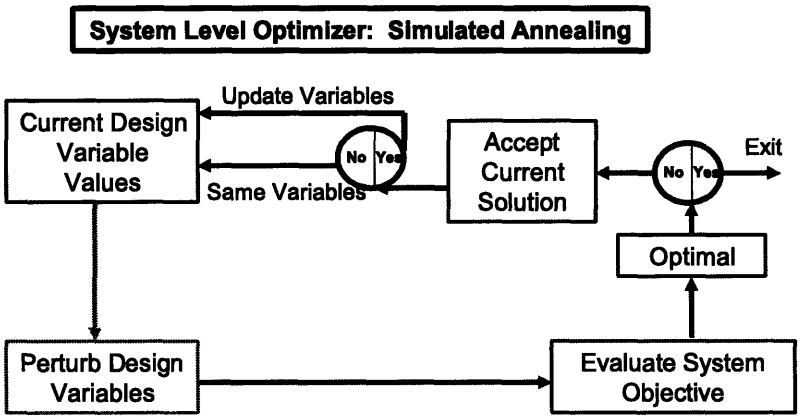


Figure 2-8: Flow Diagram of Simulated Annealing

Simulated Annealing (SA) was chosen as the system level optimizer because it can solve problems with mixed-integer variables and analysis functions can be implemented within the framework. Although this a common property of other heuristic algorithms, such as genetic algorithms (GA) and particle swarm optimization algorithms (PSO), Simulated Annealing analyzes the evolution of a *single* design vector, instead of a population. Furthermore, although a general design space exploration tool initially, the methodology for selecting new design points near the end of the optimization algorithm mimics the behavior of gradient based optimization methods where only lower cost architectures are selected. The general solution methodology of Simulated Annealing is to select a new set of design values by perturbing the original set and then evaluate the objective function. The new design variables are selected to replace the previous set if they lower the objective function, or if the raise it slightly in the early phases, since acceptance is based on a probability. Figure 2-8 describes the flow of information. A more detailed description of the principles and implementation of Simulated Annealing can be found in Appendix A or in *Kirkpatrick (1983)*[45].

The idea behind SA is to explore the design space, keeping track of 'good' solu-

tions found and as the probability of selecting a 'worse' solution decreases the system will converge to an 'optimal' solution. The constraints that determine feasibility of a particular set of design variables are generally placed as penalties in the objective function. However, with a large number of constraints penalty functions can undermine the effectiveness of the algorithm.

To utilize penalty functions, an appropriate value of the penalties must be determined with respect to each other and the actual system objective. If the penalties are too low, the optimizer will select an infeasible solution, but if the penalties are too high, the constraint violations will override the objective function and the optimizer will be unable to differentiate between two solutions. Thus, it is often necessary to restrict a design vector to be feasible before evaluation by the objective function.

To create a feasible set of perturbed design variables, the constraints are evaluated within the perturbation function, and the design variables are continually perturbed until a feasible set is obtained. However, for highly coupled systems with many constraints, it is difficult to randomly perturb variables and find a feasible set.

Simulated Annealing and other heuristic methods allow for problems of any structure to be analyzed, however these methods can be ineffective for problems with many constraints. Figures 2-9[46] and 2-10 present a notional graph and explanation chart, respectively. By examining Figures 2-9 and 2-10, we see that the goal of the embedded optimization methodology is to increase the effectiveness of Simulated Annealing without losing generality in the problem structures that it can optimize. However, the ability to maintain generality of the problem structure is hard to prove. As such, the exact location of embedded optimization in this design space is unknown, but the goal is to design an optimization methodology that is more effective than traditional heuristics while approaching the same level of generality in problem scope. Achieving this goal effectively raises the Pareto front of optimization methods, allowing more general problems to be analyzed and solved with greater computational effectiveness.

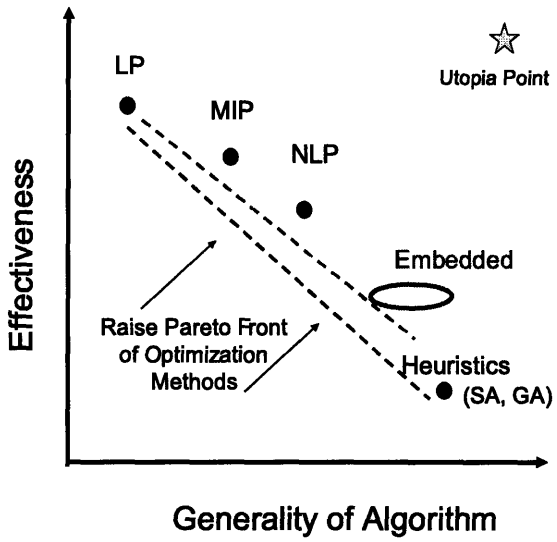


Figure 2-9: Notional Description of Effectiveness vs. Generality of Optimization Algorithms

| Methods | Variable Definition | Constraint Definition | Objective Function |
|---|---|---|---|
| LP | Continuous | Linear | Linear |
| MIP | Continuous and integer | Linear | Linear |
| NLP | Continuous | Linear and nonlinear (smooth) | Linear and nonlinear (smooth) |
| Heuristics | Continuous and integer | Linear, nonlinear and analysis function (smooth and non-smooth) | Linear, nonlinear and analysis function (smooth and non-smooth) |
| Embedded | Continuous and integer | Linear, nonlinear and analysis function (smooth and non-smooth) | Linear, nonlinear and analysis function (smooth and non-smooth) |

Figure 2-10: Comparison of Optimization Algorithms by Capability

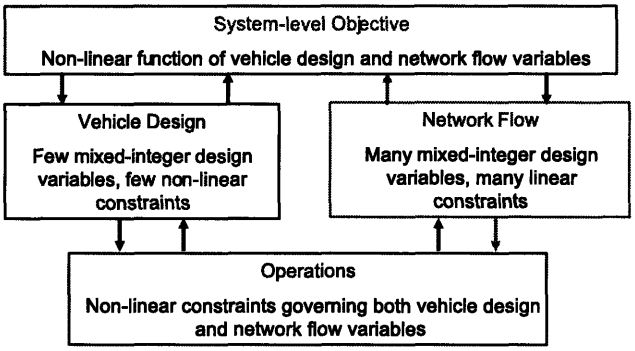


Figure 2-11: Mathematical Structure of Integrated Transportation System Design Problem

### 2.3.2 Embedded Optimization Methodology

It will be shown that the embedded optimization methodology provides an efficient approach for solving problems where large numbers of constraints are required to obtain feasible solutions. Using the decomposition of the integrated transportation system design problem presented in Section 2.1, an analysis of the mathematical structure of each component is shown in Figure 2-11.

Examining Figure 2-11 reveals that the vehicle design sub-system requires few variables and constraints to analyze a high-level conceptual design. Therefore, the majority of the constraints are supplied by the network flow (Equations 2.3 and 2.4) and operations sub-systems (Equations 2.5 and 2.6). The network flow constraints are generally linear constraints governing mixed-integer variables. The variables in the network-flow sub-system represent both the vehicle and cargo paths. The operations constraints are the non-linear constraints governing the vehicle design variables, vehicle routing variables and cargo path variables.

Examining the previous literature presented in Chapter 1.2 provides insight into the effect of problem size and mathematical structure for transportation system op-

| Previous Work | Research Focus | Variable Definition | Constraints | Objective Function | Optimization Methodology |
|---|---|---|---|---|---|
| Frommer and Crossley (2006) | Aircraft design(s) for fleet | 4 integer variables, 8 continuous variables | 4 non-linear constraints | Non-linear objective function | Enumeration of integer variables, optimization of continuous variables using SQP |
| Crossley, et al (2007) | Aircraft design and allocation for fleet | 249 integer variables, 3 continuous variables | 9 linear constraints, 32 non-linear constraints | Non-linear objective function | Decomposition and Modified Golden Search |
| Yang and Kornfeld (2003) | Routing of fleets | 147 integer variables, 343 continuous variables | 21 linear constraints | Linear objective function | CPLEX |
| Barnhart, et al (1998) | Concurrent fleet and aircraft assignment | 1million+ variables (majority integer) | 1781 linear constraints | Linear objective function | Column Generation and CPLEX |
| Taylor, et al (2006) | Concurrent aircraft design and fleet routing | 50 integer variables, 348 continuous variables | 21 linear constraints, 1 non-linear constraints | Non-linear objective function | Simulated Annealing with an LP embedded |

Figure 2-12: Comparison of Optimization Problem Size and Structure for Previous Aircraft Transportation System Design Problems

timization. Figure 2-12 defines the problem size and structure for the aircraft transportation system design problems detailed in previous work. The first two rows correspond to the aircraft design of a fleet with various operational scenarios [13],[15], and the next two rows correspond to the aircraft routing problems presented [39], [7], where the fleet design parameters were provided as inputs. The final row presents the integrated air transportation system design problem defined and analyzed in Chapters 3 and 4.

Analyzing Figure 2-12 reveals that the aircraft design problems utilize a small number of variables and constraints, however some of the variables are integer variables and the vehicle design constraints are non-linear. The variables represent parameters that specify the aircraft design and allocation of the fleet. The constraints impose restrictions on the aircraft design variables to ensure feasibility of the design for satisfying given operational tasks. Although these problems define only a small number of variables and constraints, the integrality of some of the variables in conjunction with the non-linear constraints and objective functions make these problems

difficult to solve[5]. As such, decomposition and heuristic optimization procedures are undertaken to simplify the problems and obtain solutions.

Alternatively, the aircraft routing problems examined define a larger number of variables. In *Yang and Kornfeld (2003)*[39] the problem size increases significantly, as compared to the vehicle design problems, however given the linear structure of the problem, the mixed-integer linear programming problem can be solved simply using CPLEX. In *Barnhart, et al (1998)*[7], over 1 million are defined, where the majority of these variables are discrete. In order to handle mixed-integer linear problems of this size, it is necessary to use column generation during the optimization process. Column generation utilizes information about the dual problem in order to select a sub-set of the variables for consideration[47].

The integrated transportation system design problems contain both the small number of vehicle design variables and non-linear constraints required to define the vehicle architectures, as well as the large number of mixed-integer variables and linear constraints governing the network flow. Therefore, the problems examined will be larger than the vehicle design problems analyzed, and potentially larger than the routing problem presented in *Yang and Kornfeld (2003)*, although will probably not approach the problem size presented in *Barnhart, et al (1998)*[7]. However, given the larger mixed-integer non-linear problem that results, it is necessary to analyze the structure of the integrated transportation system design problem to design a decomposition approach and define an effective optimization methodology.

By analyzing the decomposition shown in Figure 2-11, two interesting properties emerge. First, the network flow constraints governing the vehicle routing variables are completely de-coupled from the vehicle design variables and the cargo path variables. Thus, the vehicle routes can be defined independently of the vehicle design variables and cargo paths. Given the hierarchy of decisions described by the integrated transportation system decomposition, we see that the cargo paths variables are a secondary tactical decision to both the vehicle design and routing variables. The network flow sub-system only requires feasibility of the cargo paths with respect to demand. Therefore, the second observation is that if the vehicle design and routing

variables are defined, the operations constraints become linear constraints governing the cargo path variables and with the demand constraints, form a linear system.

Utilizing these properties to an advantage, embedded optimization regroups the decision order by defining feasible vehicle design values and feasible vehicle routes, as prescribed by only the vehicle design and network flow constraints. If there are many vehicle routing constraints (Equation 2.4), the system of linear constraints is *solved by embedding an appropriate optimizer into the perturbation function of Simulated Annealing.* Since, the constraints are grouped by structure, there is no clear objective. However, when using deterministic software to produce feasible design variable values, without an objective function, the first feasible set of values is returned. By employing an objective function consisting of randomly generated coefficients that are changed each time the embedded optimizer is called within SA, different feasible perturbed solutions are generated. Therefore the implementation of an objective function with randomly generated coefficients allows embedded optimization to mimic the methodology of Simulated Annealing as a design space exploration tool while improving the effectiveness of the optimization methodology for solving constrained problems.

Given the definition of the vehicle design and routes, the constraints governing the cargo paths, and any operations variables can then be defined. Again, using an appropriate optimizer, such as an LP for continuous variables, or CPLEX for integer variables, the demand constraints and operations constraints can be satisfied. Since the cargo paths do not affect the objective function, only a feasible distribution is required. Thus, embedded optimization decomposes the problem by both mathematical structure and hierarchy of decisions.

Figure 2-13 provides a visualization of the flow of SA with embedded optimization. Simulated Annealing begins by perturbing the initial set of design variables. The vehicle design variables are randomly perturbed as described in Appendix A. If there are few feasibility constraints governing the vehicle routing variables, then these variables can also be randomly perturbed by SA. If, however, there are a number of feasibility constraints (Equation 2.4), then these constraints can be solved by embedding CPLEX into the perturbation step in SA. Here, an objective function of

randomly generated coefficients is employed to allow for design space exploration. As the vehicle routing variables are required in both the system-level objective and the operations constraints, it is necessary to both find feasible values and to vary the values of these design variables to find good solutions. These steps are repeated until a feasible set of vehicle design variables and vehicle routing variables are defined, with respect to Equations 2.2 and 2.4.

Once a set of vehicle design and routing variables are found that satisfy the design and routing constraints, it is necessary to determine if this transportation architecture is feasible for the operations constraints and cargo demand. At this point, the constraints governing both cargo demand (Equation 2.3) and the operations constraints (Equations 2.5 and 2.6) are linear constraints governing the cargo variables. Thus, only feasibility of the cargo paths is required, as these variables are not included in the objective function. If the cargo variables are discrete, CPLEX is utilized to determine if a feasible cargo routing exists; however if the cargo variables can be considered as a continuous distribution, a linear programming algorithm can be utilized instead. If a feasible set of cargo paths can be determined, then the transportation architecture is evaluated by the system objective and then compared to previous solutions found. The new design vector for evaluation is selected as determined by the SA algorithm described in Appendix A and this process is continued until SA terminates.

## 2.4 Chapter Summary

This chapter began with a presentation of the integrated transportation system design problem. The four components of the transportation system, namely the vehicle design, network flow, operations and system objective, were defined for a general transportation system to understand how each component interacts to define a transportation system architecture. Next, a brief review of the relevant network theory involved with formulating a transportation system problem was presented. Issues concerning problem size resulting from both modeling requirements of paths and time expanded networks were discussed. Given the resulting size and complexity of the in-

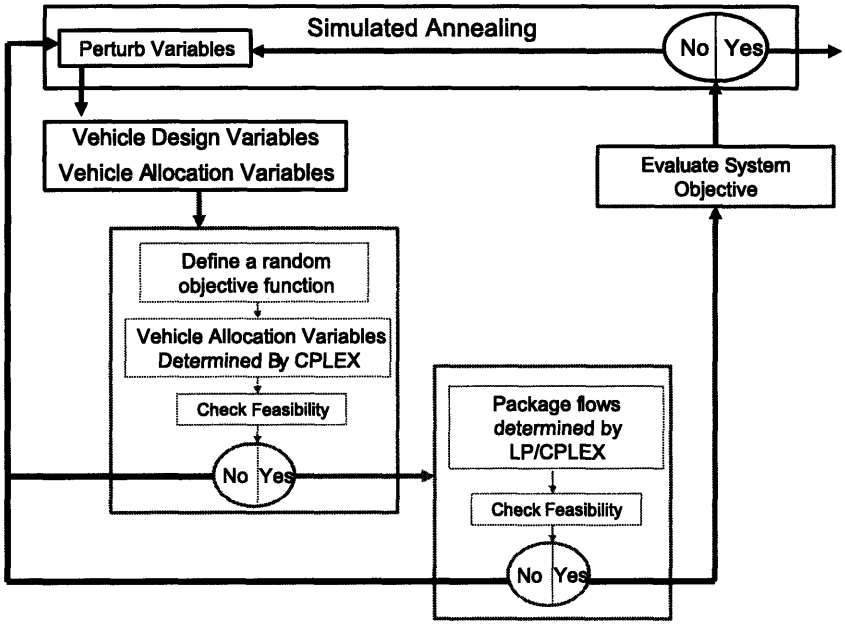


Figure 2-13: Generalized Flow Diagram of Embedded Optimization Within Simulated Annealing

tegrated transportation system design problem, a new optimization methodology was developed to alleviate some of the difficulties inherent with solving mixed-integer non-linear optimization problems. The motivation and outline of embedded optimization was presented.

# Chapter 3

# Transportation System Models

There are numerous examples of transportation systems on the ground, in the sea, in the air, and even in space. Automotive transportation systems include the truck transportation networks that transport supplies from factories to warehouses and to stores; from suppliers to delivery locations. The railway transportation network moves people and low-value bulk cargo across the land. Water transportation networks continue this flow across the oceans. Aircraft move cargo and people from their origin points to their respective destinations. And in space, the International Space Station (ISS) is both constructed, manned and re-supplied by shipments from both Kennedy Space Center (KSC) and Baikonur Cosmodrome.

In Section 2.1, the components of a generic transportation system were defined, specifically, the network model, the vehicle model, the operations constraints and the system objective. Although the specific details of each type of transportation system mentioned above may differ, every transportation system network is composed of these four sub-systems. This chapter will examine the specific model formulation for two types of transportation systems: air transportation systems and space transportation systems and differentiate between them.

This chapter defines the sub-system models for both the air transportation system and the space transportation system. In Section 3.1, the four sub-systems of the air transportation system model are presented. Section 3.2 defines a space network and contrasts the space transportation network to the air transportation network to bet-

ter understand the complexities of modeling space networks. Given the definition of a space network, Section 3.3 details the relevant models and assumptions required to formulate the space transportation system problem. The models defined for the vehicle design in both the air transportation and space transportation system problems are low-fidelity models consistent with the goals of this research and the 'pre-concept' exploration phase in design.

## 3.1 Air Transportation System Models

Air transportation systems arise from the need to satisfy the demand of both passengers and cargo that must be transported between locations within the network. The locations, or nodes, in the network are defined as the cities, or more specifically the airports under consideration, which are identified by a unique three letter code. The arcs in the network are the routes the aircraft fly between pairs of nodes. An air transportation system arises from the need to efficiently satisfy the demand by aggregating the shipments to and from multiple nodes into vehicles or aircraft for transportation.

In this section, we define the model for a hypothetical cargo airline network. The assumption of a cargo airline was made to limit additional considerations required for passenger transport, such as preference in routing. Additionally, we assume a fully connected network where all cities are connected and that the cargo or packages are small enough to fit within any prescribed volume. Finally, the model is generalized to allow multiple aircraft types to be designed for the fleet.

Figure 3-1 presents the decomposition for the air transportation system. As shown in Figure 3-1 the network and aircraft design are coupled through the operational constraints and the objective function which determines the cost of the transportation system. The remainder of this section provides the details of the model used to define the cargo air transportation system.

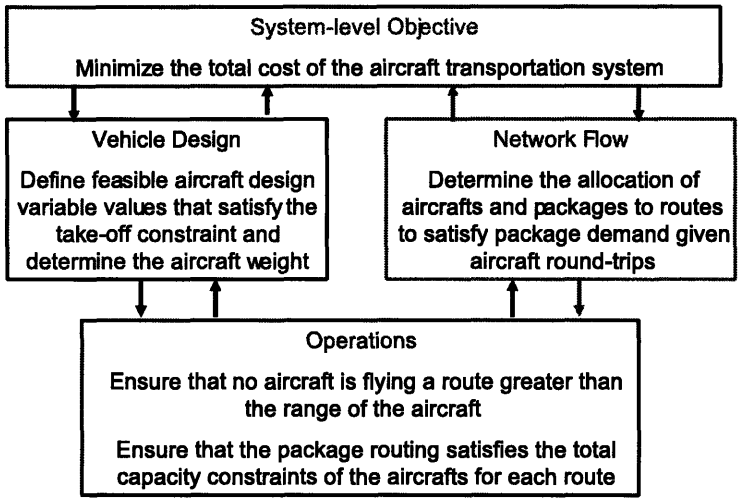


Figure 3-1: Air Transportation System Decomposition

### 3.1.1 Network Model Formulation

The network sub-system defines the allocation of vehicles and packages to routes through the network. It is assumed that an aircraft flies between two cities and performs the round trip flight once in a 24 hour period. The distance between each city pair is provided by $D_{ik}$. The number of aircraft of type $A$ flying from city $i$ to city $k$ and returning is defined as $n_{ik}^{A}$. Since only feasible routes are defined, the vehicle allocation constraints simply impose a limit on the number of aircraft of a given type flying a given route.

$$n_{ik}^{A} \leq 10 \quad \forall i, k = 1 \ldots N, \quad \forall A \tag{3.1}$$

The package flow constraints ensure that the demand of each city pair is fulfilled. Although aircraft can fly only round trips between two cities, we assume that packages may travel through an additional city towards the destination. By defining a route $(i, j, k)$ as starting at city $i$ traveling through city $k$ and terminating at city $j$, the number of packages traveling this route can be defined as $x_{ijk}$. The demand equality

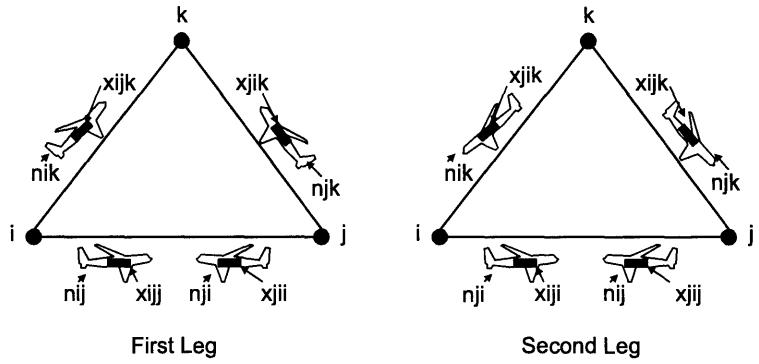


Figure 3-2: Description of Network Flow Variables

constraints that govern the feasibility of the package flow are supplied in Equation 3.2.

$$\sum_{k=1}^{N} x_{ijk} = P_{ij} \quad i, j = 1 \ldots N \tag{3.2}$$

Here, $P_{ij}$ is the package demand from city $i$ to city $j$, and $N$ is the total number of cities in the network.

Figure 3-2 defines both the aircraft and package variables in the context of a simple three city network example. On the left, the first leg shows the outbound flights and the first leg of the package routes. On the right, the second leg shows the return flights and the final leg of the package routes. As shown in Figure 3-2, packages originating at node $i$ and traveling to node $j$ can be delivered directly on the first leg ($x_{ijj}$), wait to be delivered on the second leg ($x_{iji}$), or travel both legs towards the destination ($x_{ijk}$).

### 3.1.2 Vehicle Model Formulation

The vehicle sub-system determines the architectural and performance characteristics of the aircraft design for each type of aircraft. The design of an aircraft of type $A$ is defined by the range ($r^A$), capacity ($w^A$), cruise velocity ($v^A$), wing loading ($l^A$), thrust-to-weight ratio ($t^A$), and number of engines ($g^A$)[5]. Equation 3.3 defines the range of feasible values for each of the design variables.

$$\begin{array}{lcccl} 1,000 \text{ nmi} & \leq & r^A & \leq & 5,000 \text{ nmi} \\ 5,000 \text{ lbs} & \leq & w^A & \leq & 250,000 \text{ lbs} \\ 250 \text{ kts} & \leq & v^A & \leq & 550 \text{ kts} \\ 95\frac{\text{lbs}}{\text{ft}^2} & \leq & l^A & \leq & 150\frac{\text{lbs}}{\text{ft}^2} \\ 0.3 & \leq & t^A & \leq & 0.4 \\ & & g^A & \in & \{1,2,3,4\} \end{array} \tag{3.3}$$

Additionally, a constraint on take-off length is included to ensure that the aircraft can fly out of any major airport.

$$d_{TO} = \frac{1.21 l^A}{g_0 \rho_0 C_L t^A} \leq 9000 \text{ ft} \quad \forall A \tag{3.4}$$

Here $g_0$ is the Earth's gravitational acceleration, $\rho_0$ is the density at sea-level, $C_L$ is the lift coefficient at take-off and the factor of 1.21 is a constant to account for differences in aircraft performance during take-off, as recommended in *Anderson (1999)*[10].

The take-off weight of the aircraft can be calculated from the design variables assuming a simple cruise profile, as shown in Figure 3-3. Although, the take-off weight is not constrained explicitly, it is a required input to the cost function described in the Section 3.1.4. Using a model provided in *Raymer (1999)* [6], a weight ratio is assigned to each segment of the cruise profile. The weight ratios for take-off, climb, and decent/landing are typical values provided by *Raymer (1999)* [6] and are listed in Table 3.1.

The weight ratios for the cruise ($W_C$) and loiter ($W_L$) segments are taken from the Breguet range and endurance equations, respectively, and are provided in Equation

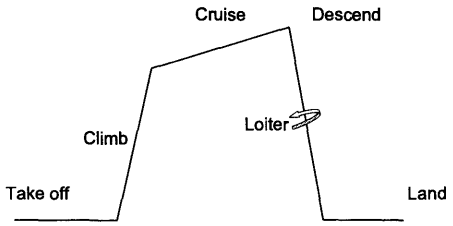


Figure 3-3: Diagram of a Simple Cruise Profile

Table 3.1: Defined Weight Ratios for Simple Cruise Profile Segments

| Segment | Weight Ratio |
|---|---|
| Take-off | 0.97 |
| Climb | 0.985 |
| Descent/Landing | 0.995 |

Table 3.2: Parameter Values for Aircraft Design

| Parameter | Value |
|---|---|
| $S_{FC}$ (1/sec) | .6 |
| $LoD$ | 17 |
| $t_L$ (min) | 30 |

3.5.

$$\begin{aligned} W_C^A &= \exp \frac{-r^A S_{FC}}{v^A LoD} \\ W_L^A &= \exp \frac{-t_L S_{FC}}{LoD} \end{aligned} \tag{3.5}$$

Here, $S_{FC}$ is the specific fuel consumption of the aircraft, $LoD$ is the lift to drag ratio, and $t_L$ is the time spent loitering before landing. The nominal values of these parameters are listed in Table 3.2. By multiplying the weight ratios together, the total weight ratio ($W_T^A$) for the entire flight profile of vehicle $A$ can be estimated. The fuel fraction ($f_f^A$) of aircraft $A$ is computed from the total weight ratio, as shown in Equation 3.6, where a six percent fuel reserve is assumed.

$$f_f^A = 1.06\left(1 - W_T^A\right) \tag{3.6}$$

The total take-off weight ($W_0^A$) is defined to be the sum of the cargo weight, the weight of the fuel for a fully loaded tank, and the structural weight of aircraft $A$. Rearranging this relationship, we can express the total take-off weight of the aircraft as shown in Equation 3.7.

$$W_0^A = \frac{W_p^A}{1 - f_f^A - s_f^A} \tag{3.7}$$

Here, the structural fraction ($s_f^A$) is the ratio of the structural mass to the total take-off mass. The payload weight ($W_p^A$) is the total cargo mass of aircraft-type $A$ plus the weight of two crew members, since these calculations are for cargo flights. The cargo mass is assumed equal to the aircraft capacity ($w^A$), which decouples the aircraft performance constraints from the package distribution. The structural or dry weight of the aircraft accounts for the total unloaded and un-fueled aircraft weight and

is estimated by an empirically derived formula for vehicle mass, taken from *Raymer (1999)* [6] and shown in Equation 3.8.

$$s_f^A = 1.02 W_0^{A^{-.06}} \tag{3.8}$$

The total aircraft weight and the weight of the fuel are determined by numerically solving the system of equations defined by Equations 3.7 and 3.8.

### 3.1.3 Operations Model Formulation

The operations of a transportation system determine how the vehicle performs on a given route and are defined by two sets of equations: capability and capacity constraints. The capability constraints, given in Equation 3.9, require that a given vehicle can not travel between two cities whose distance is greater than the range of the aircraft. Here, the range of the aircraft is defined for maximum loading, and therefore is a conservative value.

$$D_{ik} \leq r^A \quad \forall i, k = 1 \ldots N, \quad \forall A \tag{3.9}$$

To formulate the capacity constraints, we first define the capacity of route $(i, k)$ as $G_{ik}$, as in Equation 3.10, and then the capacity constraints can be formulated as shown in Equation 3.11.

$$G_{ik} = \sum_A n_{ik}^A w^A \tag{3.10}$$

$$\begin{aligned} \sum_{j=1}^{N} x_{ijk} &\leq G_{ik} \quad i, k = 1 \ldots N \\ \sum_{i=1}^{N} x_{ijk} &\leq G_{jk} \quad j, k = 1 \ldots N \end{aligned} \tag{3.11}$$

Since, we assume that a given vehicle travels only between two cities, the capacity of a route is the same on the return leg as it is on the outbound leg.

### 3.1.4 System Objective

For the air transportation system, the objective is to minimize the total system cost for a single day of operation. The aircraft has two associated cost values: a fixed cost

that is associated with an aircraft's allocation, and a variable cost that is associated with an aircraft's operation. The aircraft's performance parameters define both the fixed and variable costs for the design, which are taken from the DAPCA IV models provided in *Raymer (1999)*[6].

The cost model uses the structural weight of the aircraft ($W_s^A$), velocity ($v^A$), number of engines ($g^A$) and thrust per engine ($T_{eng}^A$) as inputs to compute the research, development, testing, and evaluation costs of aircraft-type $A$. These non-recurring costs are used to determine the depreciation of the aircraft. The fixed cost ($f^A$) of the aircraft is the cost per day of ownership of aircraft $A$, and is equivalent to the per-day depreciation of the aircraft. The variable costs ($m^A$) are the recurring costs associated with the usage of aircraft $A$, and can therefore be computed as the cost per hour of aircraft flight.

The total system operating costs are defined as

$$J = \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{A} c_{ik}^A n_{ik}^A \tag{3.12}$$

where $c_{ik}^A$ represents the cost of aircraft $A$ traveling on route $(i,k)$, as expressed in Equation 3.13.

$$c_{ik}^A = \begin{cases} f^A + m^A \dfrac{2D_{ik}}{v^A} & , \quad r^A \geq D_{ik}, \quad i \neq k \\ \infty & , \quad r^A < D_{ik}, \quad i \neq k \\ 0 & , \quad i = k \end{cases} \tag{3.13}$$

Equation 3.13 imposes a cost equal to the fixed cost plus twice the time required to travel a single leg of the trip (to account for the round-trip) multiplied by the variable cost per hour of flying the aircraft, if the aircraft can fly a given leg, as determined by the range requirement. If an aircraft does not have the range required to travel a given leg, a large cost is assigned to prohibit the selection. Finally, in the current model, storage at a given city is free, and therefore, same city 'transfers' have no cost.

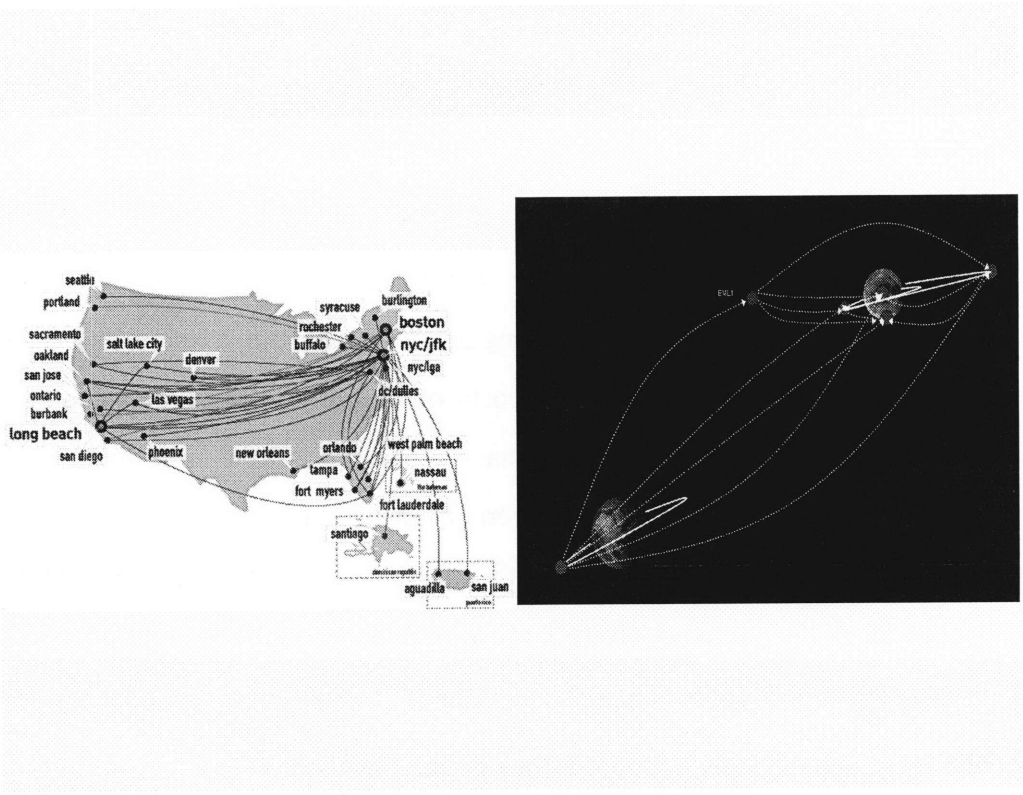


Figure 3-4: Representation of Air Transportation [2] and Space Transportation Networks

## 3.2 Space Transportation Networks

There exists many tools and methods in the operations research community for modeling and solving transportation network problems on Earth. [33, 48] However, before these methods can be applied to a space transportation system design problem, the differences in the assumptions required to adequately model the transportation network must be understood. In *Crossley, et al (2004)* [5], and *Yang and Kornfeld (2003)* [39], air transportation networks were analyzed for vehicle design and vehicle routing, respectively. However, since air transportation networks differ significantly from space transportation networks, as can be seen in Figure 3-4, it is desirable to compare the two types of transportation system so that the complexities of modeling a space transportation network can be observed.

Figure 3-5 lists the fundamental differences between modeling assumptions of air and space networks. The first difference arises in the definition of the underlying

network. In order to describe the underlying network for a transportation system, we need to define the nodes, with their associated supply and demand, and the arcs which represent how the nodes are connected. In air transportation networks, the nodes represent cities, or more accurately, airports, and each node may have a supply and demand of packages or people. The arcs represent the routes between cities, which are defined by the distances between the cities, and it is reasonable to assume a single route between two nodes which is time-invariant. In space transportation networks, the nodes can represent locations on the surface of bodies, such as locations on the Earth and the Moon, as well as locations in space that represent stable orbits, such as a LEO orbit, and it is reasonable to assume (for now) that all supply originates at Earth. The arcs represent trajectories between two nodes, where the trajectory can be defined by the $\Delta V$ required to traverse between the two nodes. However, since the nodes are in relative motion, the $\Delta V$ required to travel between two nodes may be time-varying, and in the extreme, the existence of the arc may be time varying as well.

The next important distinction between air and space transportation network modeling assumptions is in the definition of the vehicles traveling through the transportation network. In air transportation networks, the aircraft is considered to be a structure with constant physical parameters. In addition, although massive, an aircraft can carry substantial payloads compared to their wet mass, yielding high mass fractions. In space transportation networks, the spacecraft has a very small payload fraction. Therefore, in order to increase the payload fraction, segments of the vehicle may be staged, or discarded during transportation. This distinction implies that the configuration of the spacecraft may change while traveling through the network. Therefore, to accommodate changes in the spacecraft definition the transportation vehicles will be analyzed at the element level, where the definition of an element is described in greater detail in the next section.

Finally, an underlying assumption in air transportation networks is that the aircraft can refuel at any node in the network. This assumption does not hold in space transportation networks, since fuel is only available on Earth as in-situ resource uti-

| Network Definition | Terrestrial Transportation Networks | Space Transportation Networks |
|---|---|---|
| Graph Definition | Nodes represent cities<br>Arcs represent routes defined by distance | Nodes represent locations in space<br>Arcs represent trajectories defined by on V |
| | Travel between nodes is time-invariant | Travel between nodes is highly time dependent |
| Vehicles | Static vehicle definition | Dynamic vehicle definition |
| | Payload mass fraction is moderate | Payload mass fraction is small |
| Fuel | Fuel available at every node | Fuel only available at source node (Earth) |
| | Fuel consumption slightly effects capability | Fuel consumption significantly effects capability |

Figure 3-5: Comparison between Aircraft Transportation Networks and Space Transportation Networks

lization is still at the research stage. Thus, in order to reach a destination, all the fuel required by the space transportation system must be provided prior to launch. This assumption limits the capability of space transportation systems since the fuel accounts for the majority of the loaded mass of the spacecraft.

In order to capture the dynamic nature of space networks, the modeling concept of time expanded networks is employed. The following sections describe the definition of the underlying static network, or physical network that is defined first, as well as the transformation to the time expanded network.

### 3.2.1 Static Network

The physical network, or static network, represents the set of physical locations, or nodes, and the connections, or arcs, between them. The physical nodes, or static nodes, represent the different physical destinations in space, including the origin and destination of all the commodities, or supplies, as well as the possible locations for transshipment. Three types of nodes have been identified: Body nodes, Orbit nodes, and Lagrange point nodes. These classifications distinguish the type of information


Figure 3-6: Depiction of an Earth-Moon Static Network

required to define a node of each type. The physical arcs, or static arcs, represent the physical connections between two nodes, that is, an element can physically traverse between these two nodes. *Taylor, et al (2006)* [49] details the mathematical description of the network.

The mathematical description of the static network is given below.

- Define the static network as a graph $GS$, where $GS = (NS, AS)$.
- Define the set of nodes, $NS = \{s_1, \ldots, s_n\}$, in the static network.
- Define the set of arcs, $AS \subseteq NS \times NS$ in the static network.

An example of an Earth-Moon static network is provided in Figure 3-6. In this picture, we can see the connection of the Earth surface nodes to the Earth orbit node, representing launches and returns. Similarly, the lunar surface nodes are connected to the lunar orbit node, representing descent and ascent trajectories. In addition, the orbit nodes, as well as the Earth-Moon Lagrange 1 point are connected by in-space trajectories.

### 3.2.2 Time Expanded Networks

As described above, time is a critical component in defining space transportation networks. Moreover, in order to understand the dependency of multiple missions over an extended period of time, it is necessary to view the network in both space and time to capture the interaction of these missions.

In the time expanded network, the absolute time interval under consideration is discretized into $T$ time periods of length $\delta t$. A copy of each static node is made for each of the time points and the nodes are connected by arcs according to the following rules.

- The arc must exist in the static network.
- The arc must create a connection that moves forward in time.
- The arc must represent a feasible transport, with respect to the orbital dynamics.

The mathematical description of the time expanded network is given below.

- Define the time expanded network as a graph $\mathcal{G}$, where $\mathcal{G} = (\mathcal{N}, \mathcal{A})$.
- Define the set of nodes in the time expanded network as $\mathcal{N} = \{i = (s_i, t) \mid s_i \in NS,\ t = 1, \ldots, T\}$. To simplify the notation, for a given node $i \in \mathcal{N}$, let $s(i)$ and $t(i)$ denote the physical node and the time period corresponding to node $i$, i.e., if $i = (s_i, t)$ then $s(i) = s_i$ and $t(i) = t$.
- Define node $s$ as the general source that generates the supply of elements. This node is connected to every node in the network where an element can originate.
- Define the set of arcs in the time expanded network as $\mathcal{A} \subseteq \mathcal{N} \times \mathcal{N}$. An arc $a = (i, j) = ((s_i, t), (s_j, t + T^t_{s_i,s_j}))$ exists if and only if there exists an arc $(s_i, s_j)$ in the static network, and the transit time from static node $s_i$ to static node $s_j$ starting at time $t$ is $T^t_{s_i,s_j}$. Note that if $s_i = s_j$, then $T^t_{s_i,s_j} = 1$ for all $t$.

- Define path $p$ as a sequence of nodes. In particular, let $f(p)$ and $l(p)$ denote the first node and the last node of path $p$, respectively. If path $p$ originates at node $s$, $f(p) = s$ for all such $p$.

This definition creates two classes of arcs in the time expanded network: waiting arcs and transport arcs. Waiting arcs correspond to a transport in time between the same physical node, which represents an element or commodity waiting at a given node for a duration of time. Transport arcs correspond to the transport between two different static nodes and have an associated $\Delta V$ for each burn-arc. For convenience, transport arcs in a space transportation network with two burns are sub-divided into two burn arcs, representing the departure and arrival burns, respectively. The division of these transportation arcs is performed as follows. First, a new fictitious static node labeled $fic$ is introduced. Note that this node is not related to the static network. On every transport arc $(i, j), s(i) \neq s(j)$ requiring two burns we add a new auxiliary node $k = (fic, t)$ with two arcs; one connects $i$ to $k$ and the other one $k$ to $j$. The value of $t$ is irrelevant. In this new network, each arc $(i, j)$ with $s(i) \neq s(j)$ corresponds to a single burn. All such arcs are called *burn arcs* and we denote the set of all burn arcs as $\mathcal{A}_B$.

The fuel mass fraction, which represents the ratio of the fuel mass to the initial mass before a burn, for element $e$ to execute the burn corresponding to arc $a \in \mathcal{A}_B$ is defined as

$$\phi_a^e = 1 - \exp\left(\frac{-\Delta V_a}{I_{sp}^e g_0}\right) \tag{3.14}$$

which is taken from the rocket equation [19].

Using the static network depicted in Figure 3-6, we can create the time expanded network in Figure 3-7. Here, the time expanded network is notional as not all arcs are represented, but how the trajectories evolve in time can be readily seen.

## 3.3 Space Transportation System Models

Space transportation system design has become increasingly important as we begin to understand the challenges of sustainable space exploration. The goal of interplanetary

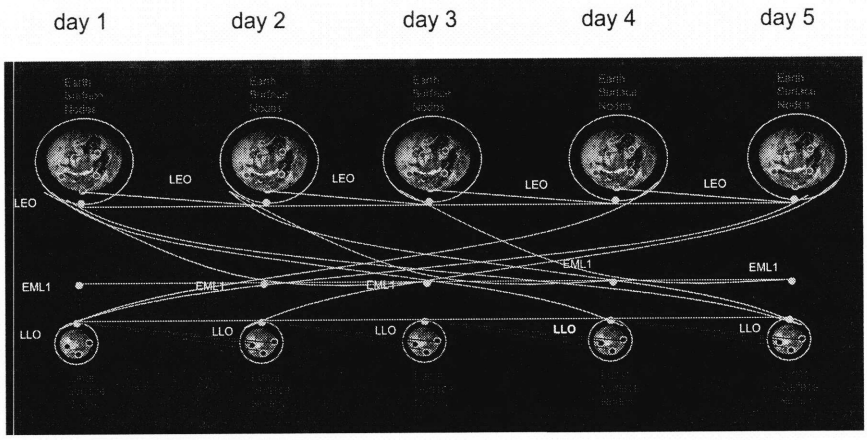


Figure 3-7: Depiction of an Earth-Moon Time Expanded Network

logistics is to determine feasible mission architectures to satisfy the demand generated by the needs of exploration[50]. The key concept of interplanetary logistics is that the demand of crew, consumables, equipment, and other exploration requirements at in-space locations drives the mission requirements.

Given the demand of the mission, it is necessary to determine how and when the supplies on Earth will be transported to the in-space locations. As missions become more complex and evolve over a period of time, a solution may become less obvious. Since the goal is to minimize the cost of any mission, it is desirable to optimize the timing and method of transport of the supplies to in-space locations. Therefore, it is necessary to define all pathways and structures used for transport, and allow a mission designer or optimizer to analyze the different architectures to select the best one.

This section describes the rationale for decomposing the interplanetary logistics transportation problem into multiple segments. Following this, the model for the in-space portion of the integrated transportation system design problem is presented, along with the assumptions necessary for both modeling and solving the problem.

### 3.3.1 Interplanetary Logistics Problem Decomposition

The execution of a space mission requires logistical decisions at every step. Logistics are required to accumulate all of the required commodities for space missions, as well as procure and assemble all elements at the launch site. However, since at the time of launch, all of the items required to perform a space mission are co-located at the launch pad, the terrestrial logistics can be decoupled from the interplanetary logistics model. Therefore, the interplanetary logistics model encompasses all of the logistical decisions required between the launch pad and the locations in-space.

There are numerous decisions made during space missions that can be modeled and optimized to create a better mission description. Although, from a system perspective, it would be desirable to make all of these decisions concurrently, due to computational limitations, this is not a reasonable approach. Instead, the interplanetary logistics model is decomposed into three fundamental components: *launch packing and scheduling, element packing, and in-space network optimization.*

Launch is a highly constrained transportation activity, where although traditional allocation and packing decisions are required, many additional constraints are necessary to model a feasible launch. Launching focuses on selecting the appropriate elements to perform the launch, satisfying the payload requirements for launch, and scheduling requirements for launch vehicles and launch sites. For this reason the launch problem is decoupled at low Earth orbit (LEO), creating a boundary between the launch allocation and the in-space network optimization. This assumption is only slightly restrictive, since for many mission architectures, there exists a delay at LEO before proceeding to in-space destinations and few upper stages are used for both launch and in-space propulsion.

Element packing is performed once all of the commodities and element routes have been determined. Given the assignment of commodities and elements to routes optimized in the in-space network optimization, commodities are assigned to elements. In this section, constraints focus on feasible assignments while minimizing transfers.

In-space network optimization examines the entire mission design space of routing

from LEO to all locations in-space. Due to the size of the time expanded network that is generated, this problem can become quite large, with millions of variables and thousands of constraints. The decision space of the in-space network optimization focuses on the routing of both commodities and elements to routes, and the assignment of elements to burns.

### 3.3.2 Assumptions

In order to define the mathematical model for the in-space network optimization, the modeling assumptions are first presented. The following assumptions about the behavior of elements are made to create a computationally tractable model.

**Consecutive Burns** When an element performs a burn, it is defined as an active element. An active element burns only on consecutive burn-arcs. Once an element becomes active, it stays active for a certain number of burns. As soon as it becomes passive, it can no longer revert to being active. Between two consecutive burns, an active element can be idle for an arbitrary length of time, by traveling on waiting arcs in the time expanded network. The number of consecutive burns is not constrained.

**Docking/Undocking** We assume that any two elements can be docked and undocked. In addition, if any cost is associated with these operations, it is not explicitly captured. If some elements cannot be docked together, then this must be captured in a post optimization analysis.

The first assumption eliminates the need to track the consumption of fuel during each burn by each element allocated within the network. Enforcing the second assumption eliminates the requirement of tracking the position of each element in the stack, as the stack can continually reconfigure.

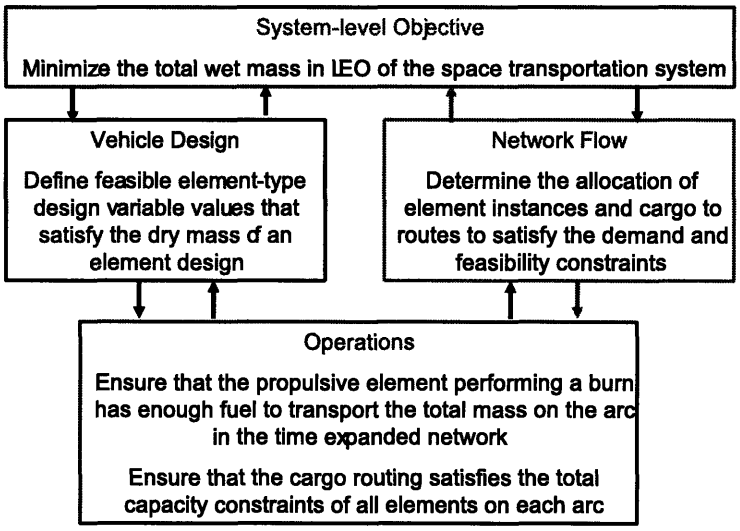


Figure 3-8: Decomposition of the Fundamental Components of a Transportation System

### 3.3.3 Space Transportation System Decomposition and Formulation

Figure 3-8 presents the decomposition for a generic space transportation system. As shown in Figure 3-8, the network and element designs are coupled through the operational constraints and are used to define the cost metric for the transportation architecture. The remainder of this section provides the details of the model used to define the space transportation system.

### 3.3.4 Network Model Formulation

The network sub-system defines the allocation of vehicles and commodities to routes through the network. A route or path through the network is defined as a sequence of connected nodes through the network. Since elements provide the propulsive capability that allows the paths to be traversed, it is also necessary to know which element is providing the propulsion for each propulsive maneuver. Therefore we define a variable

$y^e_{p,q}$ for every element instance such that

$$y^e_{p,q} = \begin{cases} 1 & \text{if element } e \text{ travels on path } p \\ & \text{and is active during sub-path } q \\ 0 & \text{otherwise,} \end{cases} \tag{3.15}$$

where $p$ is any feasible path in the time expanded network and $q$ is a sub-path of $p$. Furthermore, it is assumed that sub-path $q$ is defined as a sequence of consecutive burn-arcs and waiting arcs.

Constraints that govern the propulsive elements in the network model are imposed for feasibility. The first set of constraints restricts a given element instance from being assigned more than once.

$$0 \leq \sum_p \sum_q y^e_{p,q} \leq 1 \quad \forall e \tag{3.16}$$

The second set of constraints imposes the restriction that on any burn-arc only one instance can be active, or providing propulsion.

$$0 \leq \sum_e \sum_{p:a\in p} \sum_{q:a\in q} y^e_{p,q} \leq 1 \quad \forall a \in \mathcal{A}_b \tag{3.17}$$

The final set of constraints requires that for any element instance to travel on a burn-arc, a propulsive element must be assigned to perform the burn

$$\sum_e \sum_{p:a\in p} \sum_q y^e_{p,q} \leq M \sum_e \sum_{p:a\in p} \sum_{q:a\in q} y^e_{p,q} \quad \forall a \in \mathcal{A}_b \tag{3.18}$$

where $M$ is a large number.

The purpose of developing a model for a space transportation system is to determine how to meet the demand for the exploration missions. As such, we are investigating how to optimally ship multiple types of commodities. For the purpose of the logistics problem, a commodity will be defined as a high-level aggregate of a class of supply, such as crew provisions [51],[52]. Thus, we will define a set of $k = 1, \ldots, K$ commodities, each with the following parameters.

- Denote the demand of each commodity as $cd^k$.
- Denote the origin of each commodity as $so^k$.

- Define the destination of each commodity as $sd^k$.

- Define the availability interval of each commodity as $to^k = \left[sto^k, eto^k\right]$, where $sto^k$ is the starting time of the interval and $eto^k$ is the ending time of the interval.

- Define the delivery interval of each commodity as $td^k = \left[std^k, etd^k\right]$, where $std^k$ is the starting time of the interval and $etd^k$ is the ending time of the interval.

- Define the unit mass of each commodity as $um^k$ when it arrives at the destination.

For the purposes of this model, the transportation of the crew is ignored. This distinction is made to alleviate the extra constraints required to adequately model the complexities of crewed missions. However, this generic model can easily be extended to handle crewed missions by including additional constraints that restrict how the crew is transported from the origin to the destination locations and by narrowing the time windows for travel.

The flow of commodities through the network can then be defined as $x_p^k$, where $k$ is the commodity identification number and $p$ is the path, where

$$x_p^k = \begin{cases} u & \text{if } u \text{ units of commodity } k \text{ travels on path } p \\ 0 & \text{otherwise,} \end{cases} \tag{3.19}$$

For commodities, the path is defined as a series of nodes, where the first node is the availability node at a time within the availability interval and the last node is the destination node at a time within the delivery interval. The only constraints governing commodity flows in the network are the demand constraints

$$\sum_p x_p^k = cd^k \quad \forall k \tag{3.20}$$

where $cd^k$ is the demand of commodity $k$, which is provided in the commodity definition.

### 3.3.5 Element Model

In order to ship the commodities from the origin to the destination locations, we require 'containers' to both hold the commodities and provide propulsion to move the mass through space. These containers can be abstracted to a single definition of an element. An element is defined as a physical indivisible functional unit[49], and therefore when we consider the design of the spacecraft, we consider the design of the elements that compose the vehicle. The element design considered, is a simplified structure, as shown in Figure 3-9, and the parameters are defined as follows.

- The fuel type of element-type $E$ is denoted by $f^E$
- The fuel mass capacity of element-type $E$ is denoted by $m^E$.
- The structural mass of element-type $E$ is denoted by $s^E$.
- The mass capacity of element-type $E$ is denoted by $c^E$.
- The thrust of element-type $E$ is denoted by $t^E$.
- The engine mass of element-type $E$ is denoted by $g^E$
- The dry mass of element-type $E$ is denoted by $d^E$

The parameters of an element type can be described by three design variables that determine the architectural and performance characteristics. The first element design variable corresponds to a selection of a fuel for the element. The fuel of element type $E$ is defined as $f^E$ such that

$$f^E = \{1, 2, \ldots, F\} \tag{3.21}$$

where $F$ is the number of fuel types available. The fuel type is used as an input to the look-up table functions $I_{sp}\left(f^E\right)$ and $\alpha\left(f^E\right)$ to determine the specific impulse $(I_{sp}^E)$ and structural fraction $(\alpha^E)$ for the given fuel-type selection. The design models implemented here define three available fuel selections, as shown in Table 3.3.

For each fuel selection provided in Table 3.3, the specific impulse values are the average specific impulses for a set of both historical, current and proposed element

Table 3.3: Fuel Selection and Corresponding Look-Up Table Function Values

| Name | ID | $I_{sp}$ (sec) | $\alpha$ |
|---|---|---|---|
| LOX/kerosene | 1 | 330 | 0.045 |
| LOX/LH2 | 2 | 420 | 0.079 |
| N2O4/UDMH | 3 | 310 | 0.080 |

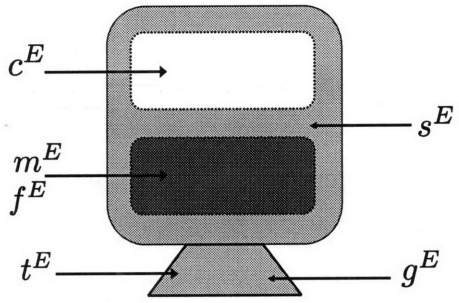


Figure 3-9: Representation of an Element

designs, whose parameters are defined in Table B.1. Furthermore, the structural fractions provided in Table 3.3 where defined by a least squares fit comparing the actual mass data to the calculated mass data for each element with the specified fuel selection.

The second element type design variable is the maximum mass of the fuel available for the design, $m^E$. The fuel mass for an element design is restricted to lie within the interval $0 \leq m^E \leq m_{UB}$ for every $E$. Finally, the mass capacity for payload is defined for each element type as $c^E$, and can be any value within the range $0 \leq c^E \leq c_{UB}$ for each element type.

The dry mass of an element-type is the sum of the engine mass and the structural mass. The dry mass of the element is required to define both the operations constraints in Section 3.3.6 and the system objective in Section 3.3.7 and is calculated as follows. First the thrust of the element type can be computed from the fuel mass

capacity $m^E$ and the fuel type $f^E$ as shown in Equation 3.22 [53]

$$t^E = \frac{m^E I_{sp}\left(f^E\right) g_0}{t_b} \tag{3.22}$$

where the specific impulse ($I_{sp}$) is determine by the fuel type selection, $g_0$ is the sea-level gravitational acceleration on Earth, and $t_b$ is the maximum burn time, which is set to 120 seconds to preserve the assumption of impulsive burns. The thrust of the engine can be related to the mass of the engine by an empirical relationship defined in *Hofstetter (2004)* [3] and shown in Equation 3.23.

$$g^E = \frac{.4189\left(t^E\right)^{.7764}}{g_0} \tag{3.23}$$

The structural mass of the element is determined by two components: the structural mass required to support the commodities and the structural mass required to support the fuel. The structural mass resulting from additional sub-systems is assumed to be small comparatively, and is therefore neglected in this analysis. For a given fuel selection $f$, the associated structural fraction, $\alpha$, is used as a guide to estimate the total structural mass of the element required to support propulsion[17]. The structural mass computation is modified to include a reduction in mass for large amounts of fuel of up to 20%. Therefore the total structural mass of element type $E$ is defined as

$$s^E = 2.3931c^E + \alpha\left(f^E\right) m^E \left(1 - \frac{.2m^E}{m_{UB}}\right) \tag{3.24}$$

where the coefficient of commodity mass availability was determined from a least squares fit to the empirical data provided in Table B.1. Here, $m_{UB}$ is the upper bound on the allowable fuel mass, which is set to 500,000 kg. The total dry mass of element type $E$ is defined as

$$d^E = s^E + g^E. \tag{3.25}$$

For each element type defined, multiple element *instances* are available for allocation in the transportation system. An element instance $e$ retains the physical properties of element type $E$ if $u(e) = E$, where $u$ is a look-up table function that maps the element instance $e$ to its corresponding element-type $E$.

### 3.3.6 Operations Model

The operational constraints in a transportation system couple the vehicle design and the network flow, and are defined by two sets of constraints: *capacity and capability.* The capacity constraints determine if the mass capacity of the elements allocated to each arc are sufficient to handle the total commodity mass on the arc. Equation 3.26 expresses this set of constraints.

$$\sum_{p:a\in p}\sum_{k} um^k x_p^k \leq \sum_{E}\sum_{e:u(e)=E}\sum_{p:a\in p}\sum_{q} c^E y_{p,q}^e \quad \forall a \tag{3.26}$$

The capability constraints govern the ability of the selected propulsive element to perform the burn required to traverse a given burn-arc. For space transportation networks, this set of constraints proves to be the most difficult to both formulate and satisfy, since the capability of the propulsive element is determined not only by the element design and arc assignment, but by the assignment of all other elements and commodities to this arc. To alleviate some of the burden caused by this constraint, we introduce an additional variable $l^e$ which represents the loaded fuel mass of instance $e$, where

$$0 \leq l^e \leq m^E \quad \forall e : u(e) = E, \quad \forall E \tag{3.27}$$

. Given the assumption that the burn sub-path $q$ consists of consecutive burn arcs, the capability constraints can be formulated as

$$\begin{aligned} l^e \sum_{p} y_{p,q}^e + M\left(1 - \sum_{p} y_{p,q}^e\right) \geq \\ \sum_{l=1}^{|q|} \Phi_{q,l}^e \times \Bigg[ \sum_{E'}\sum_{e':u(e')=E'}\sum_{p:a^l\in p}\sum_{q'} d^{E'} y_{p,q'}^{e'} \\ + \sum_{e',e'\neq e}\sum_{p}\sum_{q':a^l\in r(p,q')} l^{e'} y_{p,q'}^{e'} \\ + l^e + \sum_{k}\sum_{p:a^l\in p} um^k x_p^k \Bigg] \\ \forall e : u(e) = E, \quad E, \text{ path } q, \end{aligned} \tag{3.28}$$

where

$$\Phi^e_{q,l} = \phi^e_{a^l} \prod_{l'=l+1}^{|q|} \left(1 - \phi^e_{a^{l'}}\right) \tag{3.29}$$

and

$$\phi^e_{a^l} = 1 - exp\frac{-\Delta V_{a^l}}{I_{sp}\left(f^{u(e)}\right) g_0} \tag{3.30}$$

as defined by the rocket equation (Equation 3.14) [19]. Furthermore, the notation $r \in (p, q')$ refers to the path segment formed from the first node in $p$ up to and including the first node in the burn segment $q'$.

Equation 3.28 simply states that the total available fuel to traverse a burn sub-path $q$ is greater than the total fuel required. Therefore, the first term on the left-hand side of the equation determines the fuel available in the active element instance. The second term on the left-hand side ensures feasibility of the constraint *if the current arcs being examined are not utilized*, since the constraint is implemented for every path $q$ regardless of whether any commodities or elements transport across the arcs. In the first-term on the right hand side in brackets refers to the total dry mass of all elements traveling on each arc in the sub-path, the second term accounts for all fuel in the element instances and the final term accounts for the total commodity mass. All terms on the right hand side are multiplied by the mass fraction term defined in Equation 3.29 to determine the total amount of fuel required to traverse the burn-arc.

### 3.3.7 System Objective

The objective of the transportation system is to minimize the total system cost. For space transportation systems, cost is very difficult to adequately model, since few data points exist. Therefore, as a surrogate for cost, we seek to minimize the total wet mass of the system in LEO. The objective can therefore be written as

$$\mathcal{J} = \sum_{E} \sum_{e:u(e)=E} \sum_{p} \sum_{q} \left(d^E + l^e\right) y^e_{p,q} \tag{3.31}$$

## 3.4 Chapter Summary

In this chapter, the general integrated transportation system definition presented in Section 2.1 was implemented for two hypothetical transportation system problems: an air transportation system and a space transportation system. The air transportation network was defined in the first section, drawing on previous work performed by *Yang and Kornfeld (2003)*[39] and *Crossley, et al (2004)*[5]. However to develop a space transportation system model, a formal definition of a space transportation system network was first described and differentiated from air transportation networks. Given the space network, the space transportation system models were utilized to develop a model for interplanetary logistics transport.

# Chapter 4

# Air Transportation System Design

To evaluate the effectiveness of the integrated transportation system design methodology developed in this thesis against conventional practice, two examples of an air transportation system for an overnight package delivery network, that were presented earlier by *Yang and Kornfeld (2003)*[39], are considered. To show the benefits of the integrated transportation system design methodology over traditional design practice, each example is optimized using the traditional optimization methodologies of network-only (assuming the aircraft performance parameters are specified) and vehicle-only (assuming the network routing is specified) optimization as well as the integrated optimization methodology. In both examples, the integrated optimization methodology provides a total system cost that is at least 10% less than the best solution determined by either of the traditional optimization methods.

In Section 4.1 the details of the two examples are presented with the corresponding distance and demand information. In Section 4.2, the traditional network design optimization methodology is explained and the solutions for both examples are presented. Section 4.3 presents the optimization methodology and results for the vehicle-only optimization. Section 4.4 begins with a discussion of the specific implementation of the integrated transportation system design optimization methodology and then presents the results for both examples. In Section 4.5, the results for each case are analyzed to understand how and why the integrated optimization methodology provides a more efficient solution to the air transportation system problem.

Table 4.1: City to City Distances for First Seven City Network Example (nautical miles)

| | ABQ | ATL | BOS | CLT | ORD | CVG | CLE |
|---|---|---|---|---|---|---|---|
| ABQ | 0 | 1222 | 1933 | 1426 | 1160 | 1209 | 1393 |
| ATL | 1222 | 0 | 934 | 208 | 622 | 400 | 619 |
| BOS | 1933 | 934 | 0 | 731 | 882 | 755 | 563 |
| CLT | 1426 | 208 | 731 | 0 | 682 | 423 | 448 |
| ORD | 1160 | 622 | 882 | 682 | 0 | 260 | 309 |
| CVG | 1209 | 400 | 755 | 423 | 260 | 0 | 219 |
| CLE | 1393 | 619 | 563 | 448 | 309 | 219 | 0 |

## 4.1 Transportation System Design

The models defined in Section 3.1 are implemented for the two examples defined in *Yang and Kornfeld (2003)*[39]. The first example is a network of the first seven cities, alphabetically, consisting of Albuquerque (ABQ), Atlanta (ATL), Boston (BOS), Charlotte (CLT), Chicago (ORD), Cincinnati (CVG), and Cleveland (CLE). The distance and demand information is provided in Tables 4.1 and 4.2, respectively. The second example is a network of the largest seven cities, with respect to demand, consisting of Atlanta (ATL), Boston (BOS), Chicago (ORD), Dallas (DFW), Los Angeles (LAX), New York (JFK), and San Francisco (SFO). The distance and demand information is provided in Tables 4.3 and 4.4, respectively.

For each of the examples defined, traditional optimization approaches are employed to solve the problem in order to provide a basis for comparison for the integrated optimization methodology. The traditional optimization methodology embodies two separate views: network-flow optimization and vehicle design optimization. The remainder of this chapter details the methodology and results obtained for the three optimization cases as well as an analysis of all three approaches.

Table 4.2: Demand for First Seven City Network Example (lbs)

| | ABQ | ATL | BOS | CLT | ORD | CVG | CLE |
|---|---|---|---|---|---|---|---|
| ABQ | 0 | 2356 | 2051 | 673 | 4572 | 214 | 747 |
| ATL | 2356 | 0 | 14045 | 4610 | 31313 | 1465 | 5112 |
| BOS | 2051 | 14045 | 0 | 4014 | 27261 | 1276 | 4451 |
| CLT | 673 | 4610 | 4014 | 0 | 8948 | 419 | 1461 |
| ORD | 4572 | 31313 | 27261 | 8948 | 0 | 2844 | 9923 |
| CVG | 214 | 1465 | 1276 | 419 | 2844 | 0 | 464 |
| CLE | 747 | 5112 | 4451 | 1461 | 9923 | 464 | 0 |

Table 4.3: City to City Distances for Largest Seven City Network Example (nautical miles)

| | ATL | BOS | ORD | DFW | LAX | JFK | SFO |
|---|---|---|---|---|---|---|---|
| ATL | 0 | 934 | 622 | 688 | 1921 | 756 | 2179 |
| BOS | 934 | 0 | 882 | 1538 | 2629 | 183 | 2729 |
| ORD | 622 | 882 | 0 | 806 | 1767 | 713 | 1866 |
| DFW | 688 | 1538 | 806 | 0 | 1257 | 1360 | 1518 |
| LAX | 1921 | 2629 | 1767 | 1257 | 0 | 2454 | 330 |
| JFK | 756 | 183 | 713 | 1360 | 2454 | 0 | 2560 |
| SFO | 2179 | 2729 | 1866 | 1518 | 330 | 2560 | 0 |

Table 4.4: Demand for Largest Seven City Network Example (lbs)

| | ATL | BOS | ORD | DFW | LAX | JFK | SFO |
|---|---|---|---|---|---|---|---|
| ATL | 0 | 14045 | 31313 | 19984 | 34506 | 57949 | 37318 |
| BOS | 14045 | 0 | 27261 | 17398 | 30041 | 50451 | 32489 |
| ORD | 31313 | 27261 | 0 | 38788 | 66975 | 112479 | 72434 |
| DFW | 19984 | 17398 | 38788 | 0 | 42743 | 71784 | 46227 |
| LAX | 34506 | 30041 | 66975 | 42743 | 0 | 123948 | 79820 |
| JFK | 57949 | 50451 | 112479 | 71784 | 123948 | 0 | 134050 |
| SFO | 37318 | 32489 | 72434 | 46227 | 79820 | 134050 | 0 |

Table 4.5: Pre-defined Aircraft Type Specifications

| Parameter | Plane A | Plane B | Plane C |
|---|---|---|---|
| Capacity $w$ $(lbs)$ | 5,000 | 72,210 | 202,100 |
| Range $r$ $(nmi)$ | 1,063 | 3,000 | 3,950 |
| Velocity $v$ $(kts)$ | 252 | 465 | 526 |
| Fixed Cost $f$ $(\$/day)$ | 1,481 | 10,616 | 26,129 |
| Linear Cost $m$ $(\$/hr)$ | 758 | 3,116 | 7,194 |

## 4.2 Case 1: Network-Flow Optimization

In traditional network-flow optimization, a set of vehicles are pre-defined, each with an associated cost and capability. Using these pre-defined vehicles, an optimal allocation of vehicles to routes can be found to meet the demand of the network. In *Yang and Kornfeld (2003)*[39], three types of aircraft are chosen to provide a representative sample for a small (Plane A), medium (Plane B) and large (Plane C) airplane. Using the cost calculation described in Equation 3.13, the fixed and variable costs can be calculated from the vehicle characteristics and the relevant parameters of each aircraft are given in Table 4.5.

Using the parameters listed in Table 4.5, and the network and cost models described in Section 3.1, an optimal allocation of vehicles to routes can be determined by employing CPLEX, a mixed-integer linear optimization algorithm[54].

### 4.2.1 Example 1: First Seven City Network

For the example of the first seven city example defined in Tables 4.1 and 4.2, the minimum *total cost for one day of operations is $107,888*, and the corresponding allocation is depicted in Figure 4-1. By examining Figure 4-1, it is shown that only the small (Plane A) and medium (Plane B) aircraft are allocated due to the demand level and range requirements. The solution defines Atlanta as a hub and additionally routes direct flights between many other cities to lessen the package flow into and out of Atlanta. As such, most routes only require a single flight of the smallest

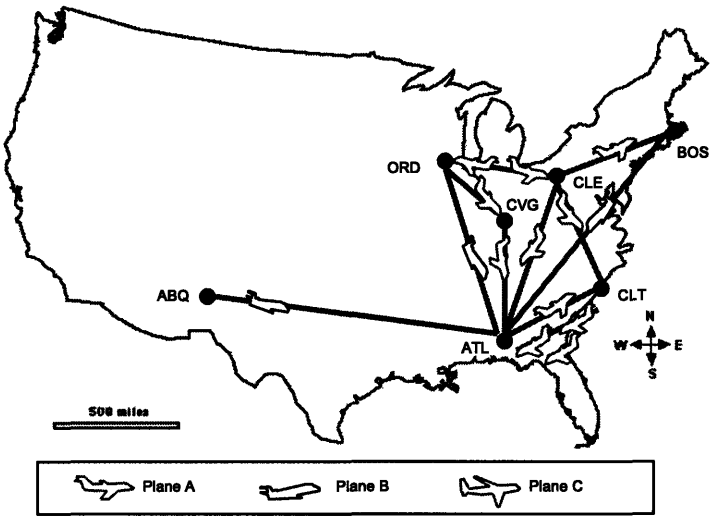


Figure 4-1: Optimal Configuration of First Seven City Example for Case 1

plane (Plane A). Although the demand at Albuquerque is low, the distance between Albuquerque and all other cities in the network exceeds the range of Plane A and therefore, a single Plane B is allocated. Since Plane B is oversized in terms of both range and capacity, for every route in the network for which it is allocated, there is a considerable amount of slack in this transportation architecture.

### 4.2.2 Example 2: Seven Largest City Network

For the example of the seven largest city example defined in Tables 4.3 and 4.4, the minimum *total cost for one day of operations is $517,030* and the corresponding allocation is depicted in Figure 4-2. By examining Figure 4-2, it is shown that only the medium (Plane B) and large (Plane C) are allocated due to both the range and capacity constraints. The solution defines Chicago (ORD) as a hub and additionally routes incoming flights from every city except San Francisco into Dallas. Since San Francisco has only two outgoing flights, it is necessary to utilize the largest capacity aircraft on both routes to accommodate the packages. Plane C is also allocated on

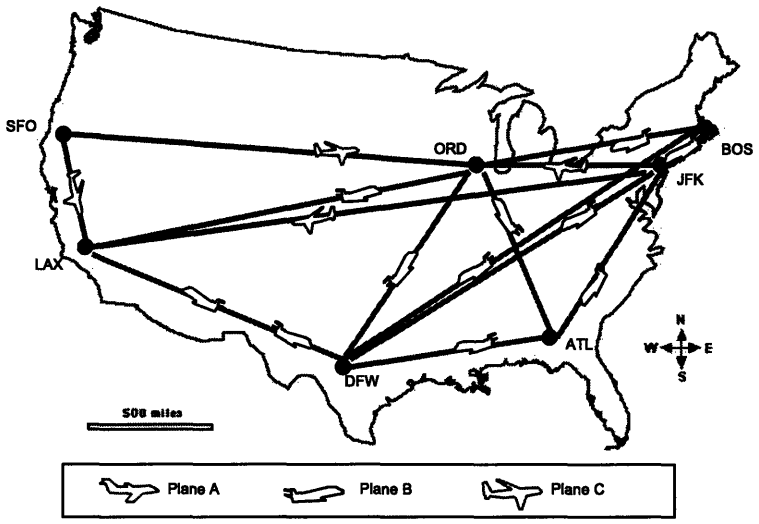


Figure 4-2: Optimal Configuration of Seven Largest City Example for Case 1

the New York to Los Angeles and New York to Chicago routes, to accommodate the large demand originating in New York.

## 4.3 Case 2: Vehicle Optimization

In traditional vehicle optimization the network flow is defined apriori and the vehicle design characteristics are optimized to produce the lowest system cost. For the traditional vehicle design optimization problem, a hub-spoke network configuration is assumed where a single city in the network is designated as the hub and all routes in the network connect to this city. More precisely, a hub is defined as a node with nodal degree $N-1$, where $N$ represents the number of nodes in the network. The optimal vehicle design characteristics defined are based on the *best compromise in performance* for the network configuration. The vehicle optimization requires an algorithm that can accommodate the mixed-integer variables and non-linear analysis functions required to define the vehicle design and allocation. As such, Simulated Annealing (*Kirkpatrick, et al. (1993)*[45]) is chosen as the optimization algorithm for

Table 4.6: Aircraft Specifications of First Seven City Example for Case 2

| Parameter | New Plane Design |
|---|---|
| Capacity $w$ $(lbs)$ | 17,995 |
| Range $r$ $(nmi)$ | 1,558 |
| Velocity $v$ $(kts)$ | 447 |
| Wing Loading $l$ $(lb/ft^2)$ | 105 |
| Thrust to Weight $t$ | .315 |
| Number of Engines $g$ | 2 |
| Fixed Cost $f$ $(\$/day)$ | 3,421 |
| Linear Cost $m$ $(\$/hr)$ | 1,251 |

this case, as it can handle an optimization problem with both mixed-integer variables and non-linear constraints.

### 4.3.1 Example 1: First Seven City Network

For the first seven city example, Atlanta is designated as the hub city, since it arises as a hub in Case 1 (Figure 4-1). The optimal *cost for this network is $94,264 per day* and the design parameters are provided in Table 4.6 for the corresponding network configuration shown in Figure 4-3. Examining the optimal aircraft design depicted in Table 4.6 and Figure 4-3 reveals that the range is defined to be higher than the range of Plane A in order to accommodate the distance requirements of the Albuquerque to Atlanta flight. The capacity is also set higher than that of Plane A to more accurately reflect the demand requirements of the network. Although there are still inefficiencies in the system design, the vehicle optimization produces a better solution for the given network, with $\approx$12% reduction in cost, as compared to Case 1.

### 4.3.2 Example 2: Largest Seven City Network

For the largest seven city example, Chicago is designated as the hub city, since it arises as a hub in Case 1 (Figure 4-2). The *optimal cost for this network is $570,720*

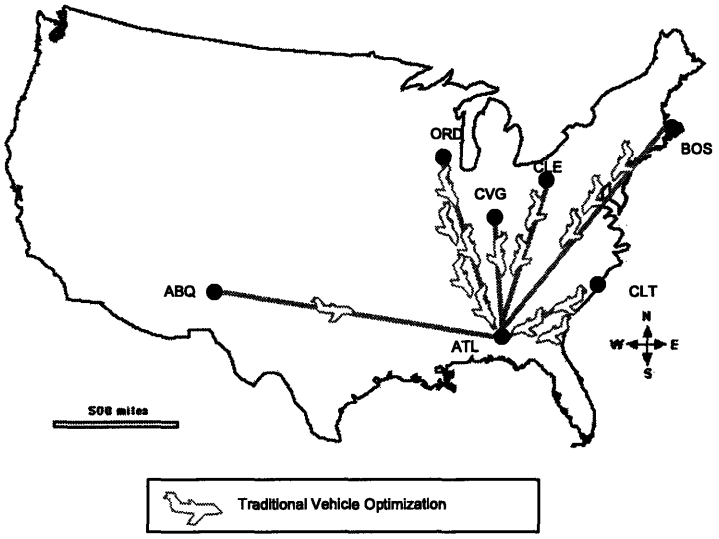


Figure 4-3: Optimal Configuration of First Seven City Example for Case 2

*per day* and the design parameters are provided in Table 4.7 for the corresponding network configuration shown in Figure 4-4. Examining Table 4.7 and Figure 4-4, shows that the range of the aircraft designed is between that of a Plane A design and a Plane B design. since a full $\approx$ 2500 nm transcontinental flight is not required due to the hub at ORD. The capacity of the aircraft designed by the vehicle optimization is between that of a Plane B and Plane C, and reflects the large demand requirements for direct flights into and out of Chicago. Although the vehicle is designed to reduce inefficiencies in the network, the requirement of only using direct flights (forcing a hub at ORD) and only allowing a single aircraft type to be used, actually increases the system cost by $\approx$10%, relative to Case 1.

Table 4.7: Aircraft Specifications of Largest Seven City Example for Case 2

| Parameter | New Plane Design |
|---|---|
| Capacity $w$ $(lbs)$ | 128,050 |
| Range $r$ $(nmi)$ | 1,920 |
| Velocity $v$ $(kts)$ | 540 |
| Wing Loading $l$ $(lb/ft^2)$ | 134 |
| Thrust to Weight $t$ | .315 |
| Number of Engines $N_{eng}$ | 2 |
| Fixed Cost $f$ $(\$/day)$ | 14,106 |
| Linear Cost $m$ $(\$/hr)$ | 4,083 |

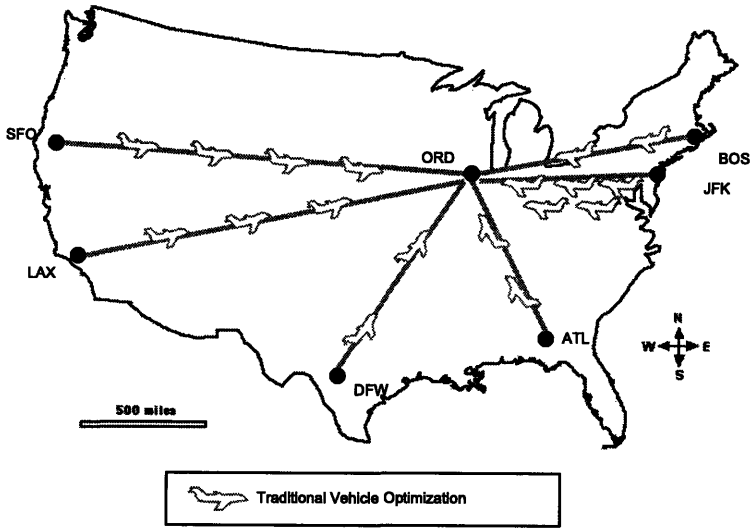


Figure 4-4: Optimal Configuration of Largest Seven City Example for Case 2

## 4.4 Case 3: Concurrent Vehicle and Network Optimization

For the integrated transportation network design, the vehicle, network and operations definition are concurrently optimized (Figure 3-1). The design vector includes variables that define both the vehicle design and network flow, and the system is subject to the constraints that govern the vehicle, network, and operations. The integrated transportation system design problem is solved using the following methodology.

### 4.4.1 Embedded Optimization for Integrated Air Transportation Design

The air transportation system design problem described in Section 3.1 assimilates all of the design variables and constraints into a single system level problem, resulting in a mixed-integer, non-linear programming problem. Therefore, the embedded optimization methodology outlined in Section 2.3 is utilized to solve the current vehicle design and network flow problem.

Figure 4-5 presents the optimization flow diagram for the integrated air transportation system design problem. The design vector consists of the aircraft design variables as well as the network allocation variables. By perturbing the values of these variables, the optimizer can evaluate the take-off constraint and determine if the aircraft design is feasible, given the vehicle design constraint. Due to the limited constraints governing the network allocation variables, simply perturbing these variables within a limited range is sufficient. Given a feasible aircraft design and vehicle allocation, the package variables can be determined by *embedding an LP solver to satisfy the demand and capacity constraints.* If there exists a feasible solution, the current design vector is evaluated to determine the total system cost; otherwise, the design variables are perturbed to define a new design vector. This process continues until the algorithm converges.

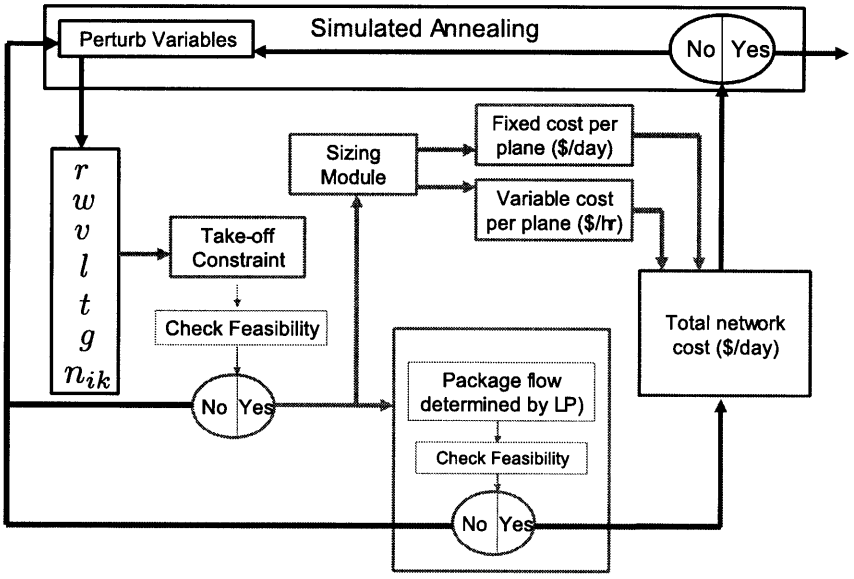


Figure 4-5: Integrated Transportation System Design Optimization with Simulated Annealing

### 4.4.2 Example 1: First Seven City Network

If we consider the design of a single vehicle and concurrently optimize the vehicle characteristics and the routes through the network for the first seven city example, *the optimal system design cost is $83,833, which is a reduction in cost of 22% over the traditional network optimization and a reduction of 11% over the traditional vehicle optimization.* The vehicle design parameters for the integrated optimization are provided in Table 4.8 and the optimal configuration is shown in Figure 4-6.

By analyzing the concurrently optimized design presented in Table 4.8 and Figure 4-6 we can see that a slightly larger aircraft, as compared to Plane A is designed to handle the distance requirements for both the Albuquerque-Atlanta and Albuquerque-Chicago flights and the demand requirements for more of the Chicago and Atlanta flights directly. However, since the concurrently optimized design is not constrained to fly only direct flights, the capacity of the aircraft is lower than that obtained by traditional vehicle optimization in Case 2.

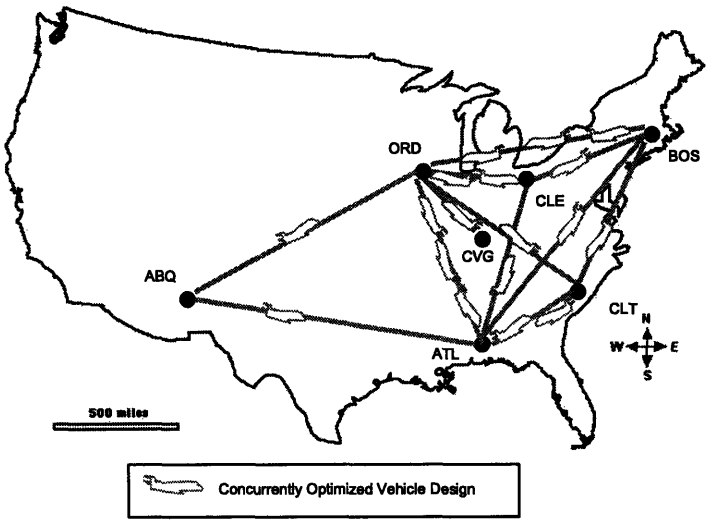


Figure 4-6: Optimal Configuration of First Seven City Example for Case 3

Table 4.8: Aircraft Specifications of First Seven City Example for Case 3

| Parameter | New Plane Design |
|---|---|
| Capacity $w$ $(lbs)$ | 9,850 |
| Range $r$ $(nmi)$ | 1,253 |
| Velocity $v$ $(kts)$ | 550 |
| Wing Loading $lS$ $(lb/ft^2)$ | 105 |
| Thrust to Weight $t$ | .302 |
| Number of Engines $g$ | 2 |
| Fixed Cost $f$ $(\$/day)$ | 2,320 |
| Linear Cost $m$ $(\$/hr)$ | 986 |

Table 4.9: Aircraft Specifications of Largest Seven City Example for Case 3

| Parameter | New Plane Design |
|---|---|
| Capacity $w$ $(lbs)$ | 69,884 |
| Range $r$ $(nmi)$ | 2,560 |
| Velocity $v$ $(kts)$ | 550 |
| Wing Loading $l$ $(lb/ft^2)$ | 106 |
| Thrust to Weight $t$ | .302 |
| Number of Engines $g$ | 2 |
| Fixed Cost $f$ $(\$/day)$ | 9,633 |
| Linear Cost $m$ $(\$/hr)$ | 2,807 |

### 4.4.3 Example 2: Largest Seven City Network

For the largest seven city example, *the optimal system cost is $463,723, which is a reduction in cost of 10% over the traditional network optimization and a reduction of 18% over the traditional vehicle optimization.* The optimal vehicle design parameters for the integrated transportation design optimization are listed in Table 4.9 and the optimal configuration is provided in Figure 4-7.

The concurrently optimized solution presented in Table 4.9 and Figure 4-7 is sized to be slightly smaller than Plane B. The reduction in range would no longer accommodate transcontinental flights from Boston, but would satisfy the distance requirements for the New York to Los Angeles and New York to San Francisco flights. By reducing the range and the capacity of the vehicle design slightly, a reduction in aircraft costs is obtained and it is cheaper to utilize more of these aircraft. Again, since a strict hub is not enforced, the capacity of the concurrently optimized solution is less than that of the vehicle optimization design for Case 2, but has a greater range.

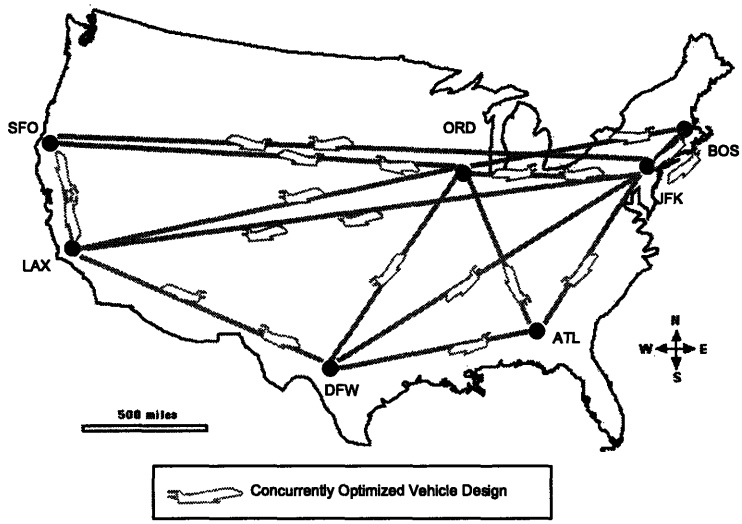


Figure 4-7: Optimal Configuration of Largest Seven City Example for Case 3

## 4.5 Comparative Analysis

The integrated transportation system design methodology exploits the coupling of the vehicle and network by defining a more efficient set of operations for the transportation system (Figures 4-8, and 4-9). This effect can best be explained and visualized by plotting the distance versus demand of each city in the network. In addition, vehicle design points are included by plotting the range versus capacity of the aircraft involved in Cases 1-3.

The relationship of the vehicle design specifications to the network requirements can be interpreted as follows. All demand points lying *within* the (dashed) bounding box of a vehicle design point can be fulfilled by a single direct flight of that vehicle. Any points to the right of the vehicle design point but below the upper bound of the box require at least one connection (stopover, hop), as the distance exceeds the aircraft's range. Additionally, any points above the vehicle design point but left of the right bound of the box require more than one flight as the demand exceeds the capacity of a single vehicle.

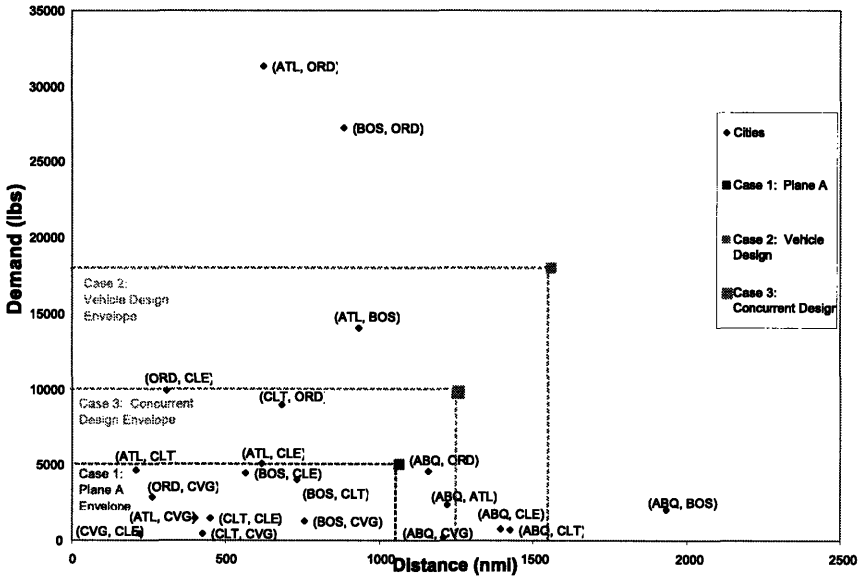


Figure 4-8: Distance versus Demand for First Seven City Example

### 4.5.1 Example 1: First Seven City Network

Figure 4-8 displays the distance and demand of the first seven city example and the vehicle design points from all three cases, as summarized in Table 4.10. It is important to note that Figure 4-8 only displays Plane A from Case 1 (Network-only optimization) as the other pre-defined aircraft design points far exceed the distance and demand of the network and are omitted for clarity. The integrated optimized aircraft design (Case 3) is only slightly right and above the Plane A design point; however, this difference allows the Albuquerque demand to be accommodated using a smaller and cheaper plane than Plane B. In addition, the Chicago to Cleveland and Chicago to Charlotte flights can be handled directly by a Case 3 design aircraft. By examining Figure 4-6, we see that only a single flight from Chicago to Charlotte is utilized. The Chicago to Cleveland route has a flight in each direction, however this is not a result of the package flow between Chicago and Cleveland, but flow from other cities into and out of Cleveland.

To provide a greater understanding of the transportation architecture efficiency

Table 4.10: Summary of Design Parameters for Three Optimization Cases of First Seven City Example

| Parameter | Case 1: Plane A | Case 1: Plane B | Case 1: Plane C | Case 2 | Case 3 |
|---|---|---|---|---|---|
| Capacity $w$ $(lbs)$ | 5,000 | 72,210 | 202,100 | 17,995 | 9,850 |
| Range $r$ $(nmi)$ | 1,063 | 3,000 | 3,950 | 1,558 | 1,253 |
| Fixed Cost $f$ $(\$/day)$ | 1,481 | 10,616 | 26,129 | 3,421 | 2,320 |
| Linear Cost $m$ $(\$/hr)$ | 758 | 3,116 | 7,194 | 1,251 | 986 |

Table 4.11: Percent-Utilization of Aircraft Capabilities for First Seven City Example

| Case | Carrier Capability Utilization | Propulsive Capability Utilization |
|---|---|---|
| Case 1 | 47% | 32% |
| Case 2 | 48 % | 40% |
| Case 3 | 72% | 52% |

for each case, Table 4.11 details the percentage of utilized functionality. The carrier capability utilization is defined as the ratio of the total package weight being transported through the network (256438 lbs) to the total capacity of each transportation architecture defined in each of the three optimization cases. The propulsive capability utilization is defined as the ratio of the total distance traveled in the network to the total range capability of the aircraft in the network.

Examining Table 4.11 reveals that the network-only optimization (Case 1) produces the lowest capacity and capability utilization which results in the highest total system cost. The concurrently optimized solution (Case 3) has both the highest carrier and propulsive capability utilization, which produces the lowest system cost for all three solutions.

Table 4.12: Summary of Design Parameters for Three Optimization Cases of Largest Seven City Example

| Parameter | Case 1: Plane A | Case 1: Plane B | Case 1: Plane C | Case 2 | Case 3 |
|---|---|---|---|---|---|
| Capacity $w$ $(lbs)$ | 5,000 | 72,210 | 202,100 | 128,050 | 69,884 |
| Range $r$ $(nmi)$ | 1,063 | 3,000 | 3,950 | 1,920 | 2,560 |
| Fixed Cost $f$ $(\$/day)$ | 1,481 | 10,616 | 26,129 | 14,106 | 9,633 |
| Linear Cost $m$ $(\$/hr)$ | 758 | 3,116 | 7,194 | 4,083 | 2,807 |

### 4.5.2 Example 2: Largest Seven City Network

Figure 4-9 displays the distance and demand of the largest seven city example and the vehicle design points for all three cases, as summarized in Table 4.12. By examining Figure 4-9 we see that the integrated optimized aircraft design (Case 3) has a range that can just handle the distance requirements of a New York to Los Angeles and New York to San Francisco flight (with a 6% fuel margin), but the demand between these cities is almost twice the aircraft's capacity. If we examine Figure 4-7 we see that there are two flights from Los Angeles to New York and a direct flight in each direction from New York to San Francisco, which can accommodate the New York to Los Angeles and New York to San Francisco demand, respectively. However, it is important to realize that some of the package flow between these city pairs may be handled by other connecting flights as there is a Boston to New York flight that may require some of the Boston to Los Angles and Boston to San Francisco packages be flown on the return flights from LA and San Francisco, respectively. Thus, the optimal solution is a hybrid between a hub-spoke and direct architecture.

Table 4.13 details the percentage of utilized functionality for the largest seven city example . The carrier capability utilization is defined as the ratio of the total package weight being transported through the network (2,284,006 lbs) to the total capacity of each transportation architecture defined by each of the three optimization cases for the largest seven city example. The propulsive capability utilization is defined as the

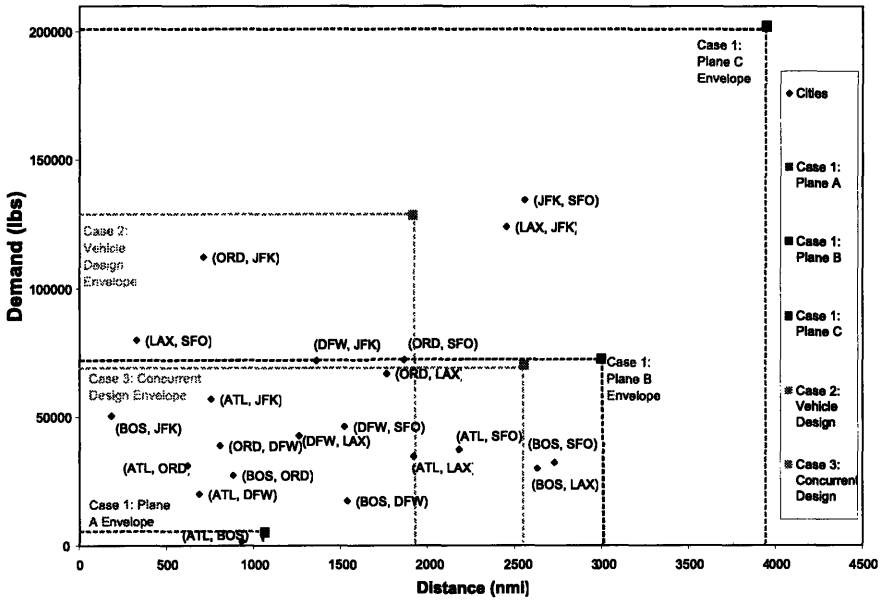


Figure 4-9: Distance versus Demand for Largest Seven City Example

ratio of the total distance traveled in the network to the total range capability of all aircraft traveling in the network.

Examining Table 4.13 reveals that the vehicle design-only solution (Case 2) has the highest capability utilization of all three cases, yet has the highest system cost. The concurrently optimized solution (Case 3) has the highest capacity utilization of all three solutions, and the lowest cost. Comparing this solution with Cases 1 and 2 shows that efficiency in carrier utilization is more important than efficiency in propulsive capability. This observation is supported by the dependence of design and

Table 4.13: Percent-Utilization of Aircraft Capabilities for Largest Seven City Example

| Case | Carrier Capability Utilization | Propulsive Capability Utilization |
|---|---|---|
| Case 1 | 75% | 40% |
| Case 2 | 50 % | 61% |
| Case 3 | 78% | 47% |

operating costs on aircraft size, and the assumption that the range of the aircraft is not significantly affected by the actual cargo loading.

## 4.6 Chapter Summary

In this chapter, the integrated air transportation system design problem presented in Section 2.1 was implemented for two examples of a cargo air transportation network. Utilizing the models defined in Section 3.1 and the distance and demand data presented in *Yang and Kornfeld (2003)*[39], computational results were obtained for three design methodologies. The traditional design approaches of network-only and vehicle-only optimization were defined, implemented and compared to the integrated optimization approach developed in this thesis. The integrated optimization approach showed a minimum of a 10% improvement in cost for the examples and the analysis of these solutions revealed that the concurrent design of the aircraft design and routing allowed for a more efficient transportation system to be defined.

# Chapter 5

# Space Transportation System Design

In this chapter, the integrated space transportation system design problem presented in Section 3.3 is implemented for an Earth-Moon example network to evaluate the effectiveness of this method against conventional practice. To quantify the benefits of employing optimization for space mission design, a baseline architecture is defined. Following the manual definition of the baseline architecture, the optimization methodologies of network flow optimization, vehicle design optimization and concurrent optimization are implemented and the results are compared to the baseline architecture. The example problem shows that the integrated optimization methodology provides a total system cost that is at least 19% less than the best solution determined by conventional practice and either of the traditional optimization methods.

In Section 5.1 the optimization methodology used to solve the space transportation system design problem is presented. In Section 5.2 the details of the example are presented with the corresponding astrodynamic and demand information. Section 5.3 develops the baseline architecture. In Section 5.4, the traditional network flow optimization results are presented. Section 5.5 presents the results for the vehicle design optimization. Section 5.6 discusses the integrated transportation system design optimization results for the lunar example. In Section 5.7, the optimization results are analyzed to understand how the integrated transportation design methodology

provides a more efficient solution to the space transportation system problem.

## 5.1 Optimization Methodology

The space transportation system design problem described in Section 3.3 assimilates all of the design variables and constraints into a single system level problem, resulting in a mixed-integer, non-linear programming problem. Therefore, the embedded optimization methodology outlined in Section 2.3 is utilized to solve the current vehicle design and network flow problem.

Figure 5-1 presents the optimization flow diagram for the integrated space transportation system design problem. The first step in the optimization procedure is to define the commodity paths under consideration for the space transportation system. Given the path definitions, the Simulated Annealing optimization algorithm is entered and the design vector is perturbed to a new design point. As the element path variables are highly constrained, CPLEX is employed to determine a feasible perturbation of these design variables as well as the staging variables. Once a feasible set of element design variables and element path variables are defined, CPLEX is utilized an additional time to determine if the transportation architecture defined is feasible with respect to the demand and operations constraints. If a feasible solution exists, the commodity routing and element fuel loading is defined and the transportation architecture is evaluated by the system objective. The remainder of this section describes in greater detail the components of this optimization process.

### 5.1.1 Commodity Path Algorithm

Prior to entering the SA algorithm, a shortest path algorithm is implemented to determine some of the feasible commodity paths. Due to the large number of paths that can potentially exist in a time expanded network framework (as defined in Sections 2.2 and 3.2), it is advantageous to select only a small number of the best paths. The shortest path algorithm proceeds as follows. A commodity is selected at random and an auxiliary network is constructed. The auxiliary network connects a single source

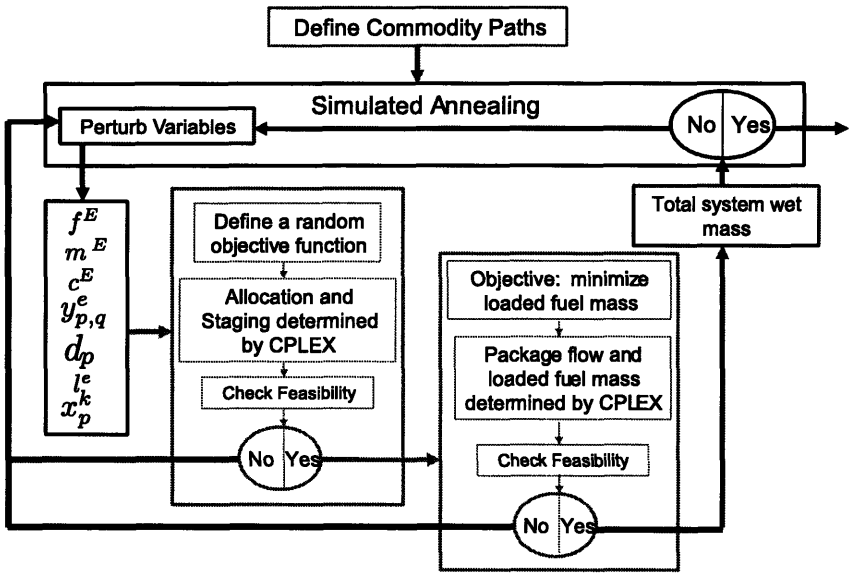


Figure 5-1: Optimization Implementation for Integrated Space Transportation System Design

node to the nodes in the availability window and a single sink node to the nodes in the delivery window. The shortest path from the source node to the sink node is then constructed, based on arc costs, and the path is returned without the source and sink nodes.

The arc costs in the auxiliary network initially correspond to the $\Delta V$ of the arcs. However, in order to encourage the shortest path algorithm to select a different path on the next iteration, the costs of the arcs are altered in the following manner. Given the current shortest path selected, the cost of each arc in the path is randomly perturbed by a pre-determined fraction, here, set to be 0.4. This perturbation can increase, decrease, or leave unchanged the cost of the arc, which therefore does not necessarily encourage or discourage common arcs between multiple paths. This process is repeated until the required number of paths have been defined for each commodity.

### 5.1.2 System Level Variables

For the integrated space transportation system design problem, the objective function presented in Equation 3.31 defines the cost of the system to be the total wet mass (kg) of all elements allocated. Therefore, the element design and allocation as well as the fuel loading, determines the objective value, not the package distribution. As such, the package flow variables are not considered by the system level optimizer.

If we examine Figure 5-1, we see that in addition to the design variables that contribute to the objective function evaluation, an additional set of decision variables are defined for each path. The decision variable $d_p$ determines if path $p$ is a potential path to be selected in the current iteration, where

$$d_p = \begin{cases} 1 & \text{if path } p \text{ can be utlized} \\ 0 & \text{otherwise} \end{cases} \tag{5.1}$$

The decision variable is then used to constrain the allowable paths of the instances such that

$$y^e_{p,q} \leq d_p \quad \forall e, q. \tag{5.2}$$

The inclusion of a decision variable within the optimization framework reduces some of the computation burden and promotes a feasible design by restricting the set of feasible paths in each iteration.

### 5.1.3 Routing and Allocation of Element Variables

The next step in the optimization flow for the integrated transportation system design problem is to embed CPLEX to solve the element allocation constraints. The objective function provided to optimize the set of element allocation variables consists of randomly generated coefficients and the product of the coefficients and the design variables are minimized. This objective function is chosen to aid SA in the design space exploration process, since an adequate objective is unknown at this stage, and feasibility of the design variables is the only concern. The objective function is defined to be a set of randomly generate integers between -1 and 9, which discourages too

many elements from being selected without minimizing the total number of elements selected.

Figure 5-1 also indicates that the staging of the elements is determined at this point in the optimization algorithm. The staging variable ($S_e$) determines if an element instance stages after it burns. Given that an element has two functions, to carry commodities and perform propulsive maneuvers, an element will either stage after its burn sequence is completed, or continue along the assigned path. By implementing the staging variable, a large reduction in the number of element paths is obtained, because only paths that correspond to feasible commodity paths are considered, instead of all intermediate paths in the network. The equations that govern the staging variables simply require that an element instance can not stage if the instance is not selected for a burn sequence.

$$S^e \leq \sum_p \sum_{q:q\neq\emptyset} y^e_{p,q} \forall e \tag{5.3}$$

### 5.1.4 Commodity Flows and Operations Constraints

Following the CPLEX computation, all system design variables have been defined, and assuming a feasible set of design values, the operations constraints are analyzed to determine if a feasible set of commodity flows exists for this architecture. The operations constraints are the only set of constraints that relate the commodity flows, element allocations and the element designs.

If we examine the capacity and capability constraints, Equations 3.26, and 3.28, we see that the interaction of the element design variables and element allocation variables creates a set of non-linear constraints. However, by decomposing the problem as we have described, the only design variables at this stage of the analysis are the commodity flows and the loaded fuel mass of each element instance, since all other variables have been defined. This decomposition significantly reduces the computational effort required to formulate the operational constraints, since all paths, and hence all arcs, under consideration have been selected by the element design variables, and therefore only a limited number of constraints must be implemented and

satisfied.

Again CPLEX is embedded into the perturbation stage of SA to allow for the efficient computation of the commodity flow variables and fuel loading variables subject to the demand constraints (Equation 3.20), the capacity constraints (Equation 3.26), and the capability constraints (Equation 3.28). However, the capacity and capability constraints can now be re-written since all terms that apply to the element design and assignments are known, and therefore are part of the right hand side of the equations. In addition, since any feasible flow is acceptable, the only objective is to minimize the total loaded fuel.

If CPLEX returns a feasible flow and fuel loading, the system design variables are passed to the system objective to be evaluated. If a feasible flow does not exist, the system design variables are re-perturbed and the process is repeated. SA continues to perturb the design variables until the system energy decreases and better design points can no longer be found.

## 5.2 Space Transportation System Example

The goal of the supply chain logistics problem is to adequately account for and optimize the transfer of supplies from Earth to locations in space. Although the consideration of the supplies is of high importance, the commodities themselves may be of low value on Earth. As such, it is desirable to find the cheapest way to transport these supplies. The Aquarius project investigated the delivery of 1000 kg "packages" of water from Earth to LEO using a cheap, low-reliability launch system[55]. Using low-priority transportation networks on Earth, such as rail and barge transportation systems, and the design of a single-stage to orbit launch vehicle, a significant theoretical reduction in launch costs was obtained.

The case study presented here begins where the Aquarius project left off, in low Earth orbit. In this example we divide the 1000 kg packages of water that the Aquarius project considers into ten 100 kg packages. The objective is to minimize the total system mass required to deliver a number of units of 100 kg packages from LEO to

multiple in-space destinations.

In this hypothetical example, the Earth-Moon network consists of six nodes: low Earth orbit (LEO, node 1), the first Earth-Moon Lagrange point (EML1, node 2), a low circular equatorial lunar orbit (LELO, node 3), a low circular polar lunar orbit (LPLO, node 4), the lunar equatorial surface (LES, node 5), and the lunar south polar surface (LPS, node 6). In this example, all commodities originate at LEO for transport to further in-space destinations, and therefore LEO has only outgoing arcs. In addition, the lunar surface nodes have only incoming arcs, which excludes lunar ascent trajectories and lunar surface transfers. All other in-space nodes are connected as astrodynamics permits. The network is depicted in Figure 5-2.

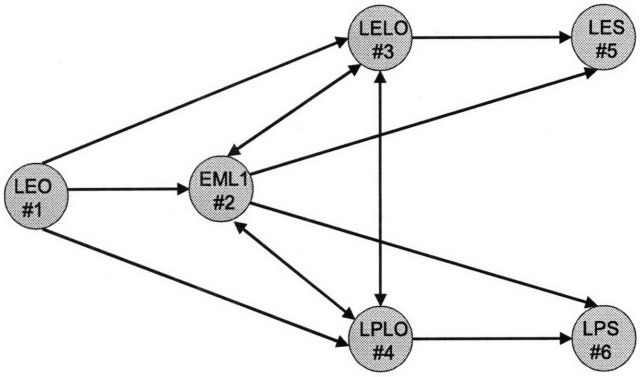


Figure 5-2: Static Earth-Moon Transportation Network

We assume an impulsive trajectory prior to analysis and compute the required $\Delta V$ and time of flight for each allowable transfer. It is important to note that due to the nature of the time expanded network definition, self-arcs in the static network (which become waiting arcs in the time expanded network) have a time of flight of one time period. The $\Delta V$ and transport time (tof) values are provided in Table 5.1.

Referring to Table 5.1, there are either one or two values of $\Delta V$ for the transport arcs. For the case of transport arcs with two $\Delta V$ values, the first burn allows the

Table 5.1: Transport $\Delta V$ (m/s) [3]

| NODES From To | EML1 | LELO | LPLO | LES | LPS |
|---|---|---|---|---|---|
| LEO burn 1 | 3085 | 3108 | 3108 | | |
| burn 2 | 740 | 850 | 850 | | |
| tof | 3 | 4 | 4 | | |
| EML1 burn 1 | 0 | 248 | 248 | 2746 | 2746 |
| burn 2 | 0 | 632 | 632 | 0 | 0 |
| tof | 1 | 2 | 2 | 2 | 2 |
| LELO burn 1 | 632 | 0 | 1500 | 2083 | |
| burn 2 | 248 | 0 | 0 | 0 | |
| tof | 2 | 1 | 1 | 1 | |
| LPLO burn 1 | 248 | 1500 | 0 | | 2083 |
| burn 2 | 632 | 0 | 0 | | 0 |
| tof | 2 | 1 | 1 | | 1 |

element to enter the arc and the second burn allows the element to exist the arc. For example, to travel from LEO to LELO, a trans-lunar injection (TLI) burn is performed with a $\Delta V$ of 3108 m/s and an orbit insertion burn is performed at lunar orbit with a $\Delta V$ of 850 m/s. These two-burn arcs are split into two arcs within the network, as described in Section 3.2.

An element can transfer directly from LEO to the Lagrangian point (EML1) and to the lunar orbits (LELO and LPLO), however to reach the surface nodes (LES and LPS), an element must transfer through another node. In this example, we restrict all elements and commodities to begin the transfer in LEO. Furthermore, both elements and commodities can travel through any feasible path in the network, however, elements are restricted to be a propulsive element on at most two consecutive burns.

Using the network defined in Figure 5-2 the problem is to determine the architectural and performance characteristics of the element types and the element path and burn assignments paths through the network to minimize the total system wet mass.

Table 5.2: Commodity Information

| Comm. # | Comm. Type | Demand (# Units) | Start Node | Availability Interval | End Node | Delivery Interval | Mass (kg/unit) |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 18 | LEO | 1, 15 | EML1 | 1, 15 | 100 |
| 2 | 1 | 9 | LEO | 1, 8 | LES | 8, 8 | 100 |
| 3 | 1 | 6 | LEO | 1, 15 | LPS | 9 15 | 100 |
| 4 | 1 | 2 | LEO | 1, 12 | LPS | 12, 12 | 100 |

The example considers the design of two element types, which allows for, although does not require, a specialization of carrier elements and propulsive elements, as is traditionally seen in space transportation systems. The demand scenario presented in Table 5.2 lists the information, supplied as an input to the problem, for each commodity. It is important to note that although the commodity type for each commodity is the same, corresponding to 100 kg of water, four separate listings are required to differentiate between the delivery locations and time intervals.

The commodity definitions in Table 5.2 were chosen to represent three mission scenarios. Commodity 1 represents the desire to 'warehouse' commodities at the Lagrange point. As such, the availability and delivery time windows are open. The second and fourth commodities represent the need to supply a mission to the lunar equatorial landing site and lunar polar landing site, respectively. As such, the delivery time window for these two commodities is set to only a single day. Commodity 3 represents a re-supply at the lunar equatorial surface, which is a mission situation that would become necessary if a lunar base were to be constructed. Thus, the four commodity definitions provide a variety of operational requirements that the transportation architecture must satisfy.

For the example space network defined, a manually defined architecture is developed to provide a baseline for comparison of the optimization results. The traditional optimization approaches are then employed to examine the benefits of optimization for space system design. The traditional optimization methodology embodies two views: network optimization with pre-defined element types and vehicle optimization

with pre-defined direct paths. The concurrent optimization, which expands the design space to *concurrently* define the vehicle and network flow, is then implemented and the results are compared to the baseline architecture and traditional optimization results. The following sections detail the results obtained for each of these approaches using the example Earth-Moon transportation network.

## 5.3 Baseline Solution

Traditionally, space mission design utilizes existing vehicle resources when designing space missions, to minimize cost. The mission requires a direct delivery, and the least costly operational architecture is defined. Therefore, for the baseline solution, a set of direct paths from LEO to the in-space commodity destinations are defined, corresponding to a separate flight for each delivery. Furthermore, the baseline architecture assumes that the missions performed will utilize existing vehicles. In order to select, useful elements to execute the mission objectives, it is necessary to understand the capabilities of existing elements.

Elements can be classified by their size and functionality. Figure 5-3 displays a plot of the pre-existing elements, defined in Appendix B, to asses the size and functionality of these elements. Examining Figure 5-3 shows a definite distinction between elements that are propulsive-only elements, carrier-only elements, and dual function elements. In addition, a range of sizes for each of these categories exists.

For each of the elements considered in Figure 5-3 the capacity, fuel-type, and fuel mass were used as inputs to the dry mass computation defined in Equation 3.25. The maximum $\Delta$ V achievable for each propulsive element was calculated assuming a fully loaded, fully fueled element assuming no other elements in the stack. For the network optimization, a single element-type is defined to represent the capabilities of elements in each category of functionality and size. Table 5.3 displays the categorization and representative element type assigned, with its corresponding physical properties.

For the baseline solution, the large propulsive-only element (Element 412) is selected to perform the trans-lunar injection burn from LEO for each of the delivery

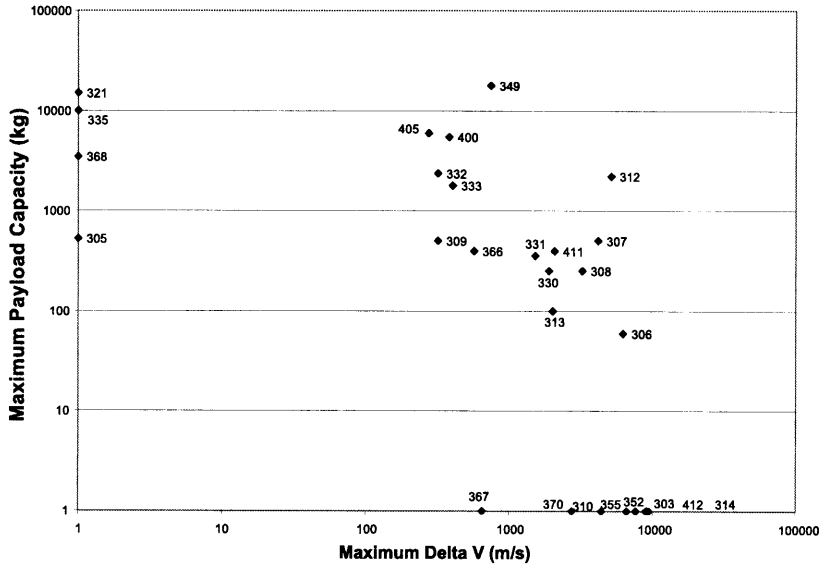


Figure 5-3: Maximum Delta V versus Capacity for Pre-existing Elements

Table 5.3: Pre-defined Space Transportation Element Specifications

| Classifications | Propulsive | Dual | Carrier |
|---|---|---|---|
| Small | Lunar CEV SM<br>Element 310<br>Com. Mass = 0 kg<br>Fuel Mass = 7222 kg<br>Dry Mass = 2143 kg | Apollo DS<br>Element 307<br>Com. Mass = 500 kg<br>Fuel Mass = 8804 kg<br>Dry Mass = 2509 kg | Apollo CM<br>Element 305<br>Com. Mass = 524 kg<br>Fuel Mass = 0 kg<br>Dry Mass = 1253 kg |
| Large | EDS<br>Element 412<br>Com. Mass = 0 kg<br>Fuel Mass = 129500 kg<br>Dry Mass = 15852 kg | LSAM DS<br>Element 312<br>Com. Mass = 2200 kg<br>Fuel Mass = 28932 kg<br>Dry Mass = 9449 kg | ISS CEV cargo<br>Element 368<br>Com. Mass = 3500 kg<br>Fuel Mass = 0 kg<br>Dry Mass = 8375 kg |

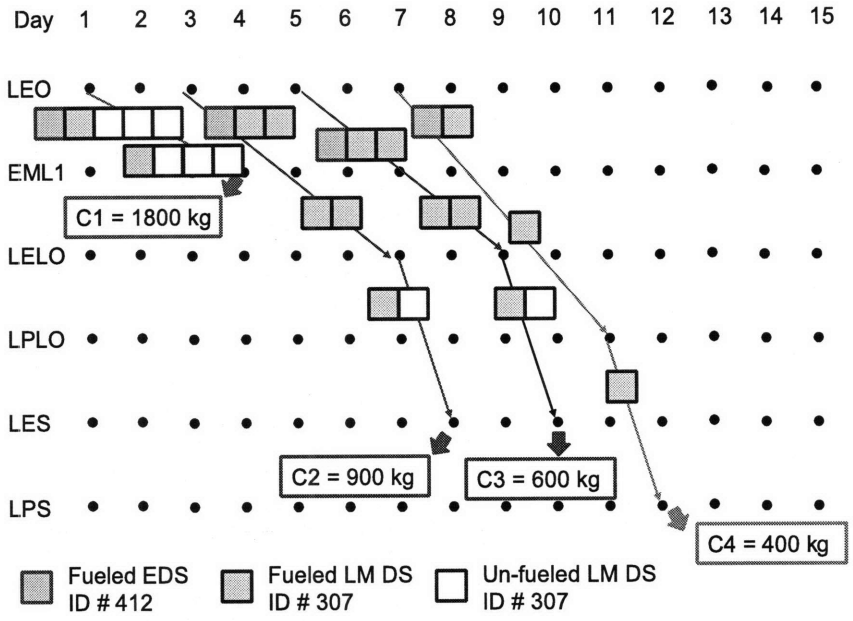


Figure 5-4: Manually Defined Baseline Architecture

flights. To provide carrier capability and some propulsive capability, the small dual function element (Element 307) is utilized to encapsulate the commodities and provide the remaining propulsion. The total system wet mass for the baseline architecture is 228,584 kg and the corresponding allocation is shown in Figure 5-4.

Examining Figure 5-4 reveals that the approach of using only direct missions creates inefficiencies in the transportation architecture. Since each mission must be performed separately, efficient utilization of the available carrier mass can not be exploited. Furthermore, the large propulsive capabilities of Element 412 are not always utilized, as the direct mission architecture cannot combine multiple flights leaving LEO. The direct mission architecture, and resulting element selection decisions, are analogous to a pure direct flight network in air transportation systems. As in the aircraft direct flight scenario, smaller vehicles with less propulsive capabilities are utilized when possible, as trans-shipments are not permitted.

## 5.4 Case 1: Network Optimization

In traditional network flow optimization a set of vehicles or elements are pre-defined, with their corresponding performance characteristics. Using the pre-defined element parameters, an optimal allocation of elements to paths and burns can be defined to meet the demand of the network. In this example two pre-defined element types will be optimally allocated to the network, at lowest system cost.

Network optimization determines the optimal routing of the transportation elements using the network, operations, and objective function models defined in Section 3.3. The optimization is performed by utilizing CPLEX, a mixed-integer linear optimization program[54]. Given the element design parameters as inputs to the problem, the model can be linearized by defining additional variables and constraints. The details of this linearized model are provided in Appendix C.

The network optimization was executed for each pair of elements in Table 5.3, where combinations of two propulsive-only elements and two carrier-only elements were excluded, since these combinations are automatically infeasible. Table 5.4 displays the results for the 13 combinations that were evaluated. Examining Table 5.4 we see that the *minimum total system wet mass for network only optimization is 196,501 kg*, which provides a 14% decrease in system wet mass, as compared to the baseline architecture. This cost is obtained by utilizing the large propulsive only element together with the small dual function element, thereby providing commodity carrying capability and both small and large propulsive capability. Furthermore, we see that only combinations that involve Element type 412 are feasible. Element type 412 (Earth Departure Stage, EDS) is the large propulsive element and therefore has the ability to maneuver large payloads through arcs with large $\Delta$ V's. The corresponding allocation is shown in Figure 5-5.

Examining Figure 5-5 shows that for the two element types defined, two flights are made from LEO to EML1. The demand for commodity 1 at EML1 can be satisfied by both flights since the delivery time for this commodity is open. The first transport flight at time one requires 5 instances of element type 307 to hold the commodities

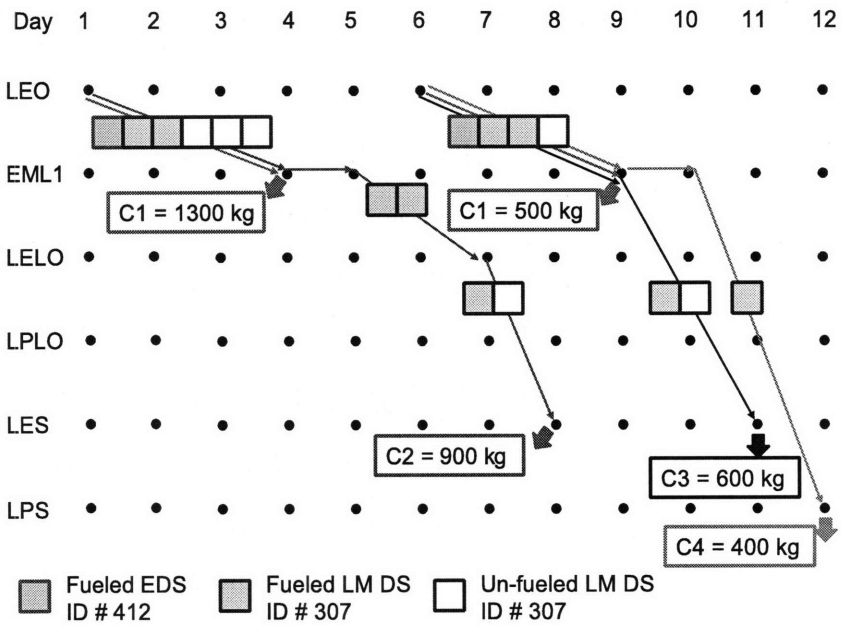


Figure 5-5: Optimal Configuration for Case 1

Table 5.4: Results from Optimization of Case 1

| Element ID | Element ID | Objective Value (kg) |
|---|---|---|
| SM #310 | LM DS #307 | Infeasible |
| SM #310 | LSAM DS #312 | Infeasible |
| SM #310 | CM #305 | Infeasible |
| SM #310 | ISS Cargo #368 | Infeasible |
| EDS #412 | LM DS #307 | 196,501 |
| EDS #412 | LSAM DS #312 | 624,489 |
| EDS #412 | CM #305 | 333,091 |
| EDS #412 | ISS Cargo #368 | 440,490 |
| LM DS #307 | LSAM DS #312 | Infeasible |
| LM DS #307 | CM #305 | Infeasible |
| LM DS #307 | ISS Cargo #368 | Infeasible |
| LSAM DS #312 | CM #305 | Infeasible |
| LSAM DS #312 | ISS Cargo #368 | Infeasible |

and a single element 412 to perform the burn from LEO to EML1. Upon arrival at EMl1, 1300 kg of payload are dropped off and the remaining elements travel to LELO and finally LES to satisfy the demand of 900 kg of payload at time 8. To perform the transfer, two fueled element type 307s perform the burns and carry the commodities to their final destination. The second flight from LEO travels to EMl1 and delivers the remaining 500 kg of payload before *splitting* to satisfy the demand for commodities 3 and 4, at LES and LPLO respectively. This architecture allocates 4 unfueled instances of element 307, four fueled instance of element 307 and 2 fueled instance of element 412 to satisfy the network flow and operational constraints at minimum total system wet mass.

This solution essentially establishes the Lagrangian point (EML1) as a 'hub' in the space transportation network. In contrast to the baseline solution, the network optimization, using the same pre-defined space elements chooses to incur the additional $\Delta V$ penalty of traveling through the EML1 point on route to the lunar surface, instead of utilizing the direct paths. However, by doing so, a 14% reduction in wet mass is obtained, which provides a quantitative basis for exploring EML1 as a potential 'warehouse' in space.

## 5.5 Case 2: Vehicle Optimization

In traditional vehicle optimization the network flow is defined apriori and the vehicle design characteristics are optimized to produce the lowest system wet mass. For the vehicle design optimization problem, direct routes are assumed, as is traditionally employed for space missions, and the element design characteristics are defined based on the best compromise in performance for the network configuration. The vehicle optimization uses the methodology defined in Figure 5-1, where a single direct path for each commodity is specified, however for this case, the optimization customizes the design of the elements, instead of selecting elements from Table 5.3. *The lowest system wet mass found for the space network with direct routes and optimized element designs is 203,260 kg* and the design parameters are provided in Table 5.5 for the

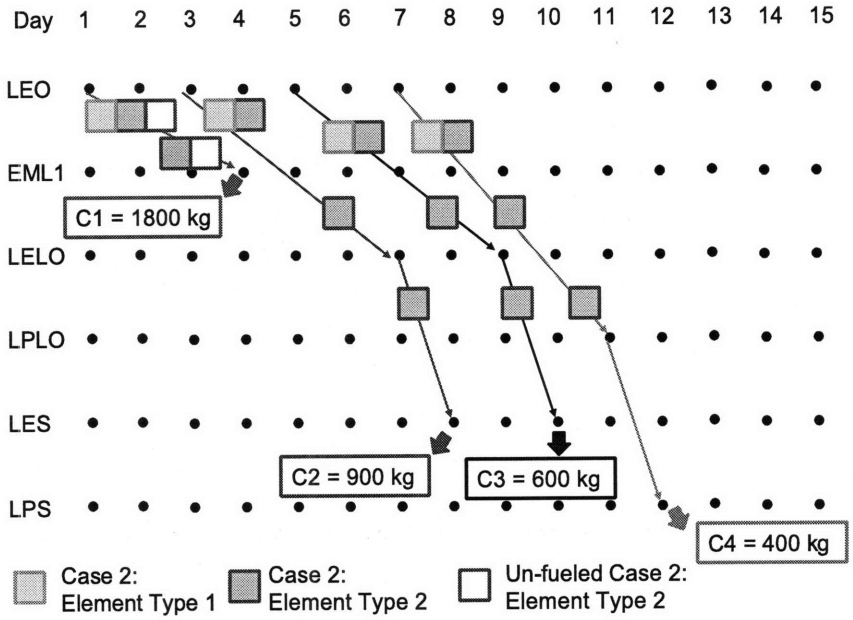


Figure 5-6: Optimal Configuration for Case 2

Table 5.5: Element Specification for Case 2

| Element Type | Fuel Type | Fuel Mass (kg) | Capacity (kg) | Dry Mass (kg) |
|---|---|---|---|---|
| 1 | 1 | 80646 | 0 | 10427 |
| 2 | 1 | 14191 | 1200 | 2222 |

corresponding network shown in Figure 5-6.

Examining Table 5.5 and Figure 5-6 reveals that the element design types continue the trend of selecting a large-propulsive only element and a smaller dual function element. Here, however, the dual function element is much larger than the dual function element selected in both the baseline and Case 1. By allowing the optimization to specify the vehicle size and operations, an 11% reduction in total system wet mass, as compared to the baseline architecture is obtained. The imposition of direct paths requires a large carrier size on some of the paths, and creates a large amount of slack on paths where the package flow is low. The direct path architecture does not allow for the efficient use of the element types by allowing an interaction of the commodities

and therefore the total system wet mass is an increase of 3% over Case 1.

## 5.6 Case 3: Concurrent Element and Network Optimization

For the integrated transportation system design problem, the element design, network flow, and operations definition are concurrently optimized. The design vector includes variables that define both the element design, and network flow, and the system is subject to the constraints that govern the elements, network and operations. The integrated transportation system design problem is solved using the methodology defined in Figure 5-1. However, due to the resulting problem size, it is necessary to utilize the information obtained from analyzing the results of Case 1 and Case 2 to make the computation of Case 3 more tractable.

Given the same set of 20 potential paths utilized in Case 1, the resulting problem defines 2960 element routing variables, which represent the path and burn allocations for 10 available instances of each of two design types. The large number of network flow variables within the integrated transportation system design framework requires too much computation time (over 3 weeks) for solution using the optimization methodology defined in Figure 5-1.

As the Case 1 solution provided a lower wet mass than the direct architecture in Case 2, and chose to exclusively route all commodities through the Lagrangian point, only these paths were selected as paths in this case. Thus, the concurrent element and network optimization examined here effectively seeks to improve the solution obtained in Case 1 by allowing the optimizer to *concurrently* define the element performance parameters and the element routing using a 'hub' at EML1.

*The minimum system wet mass obtained is 158,650 kg*, which is a reduction of 31% over the baseline solution, a reduction of 22% over traditional element design optimization (Case 2), and a reduction of 19% over traditional network optimization. The optimal vehicle design parameters for the concurrent design optimization are

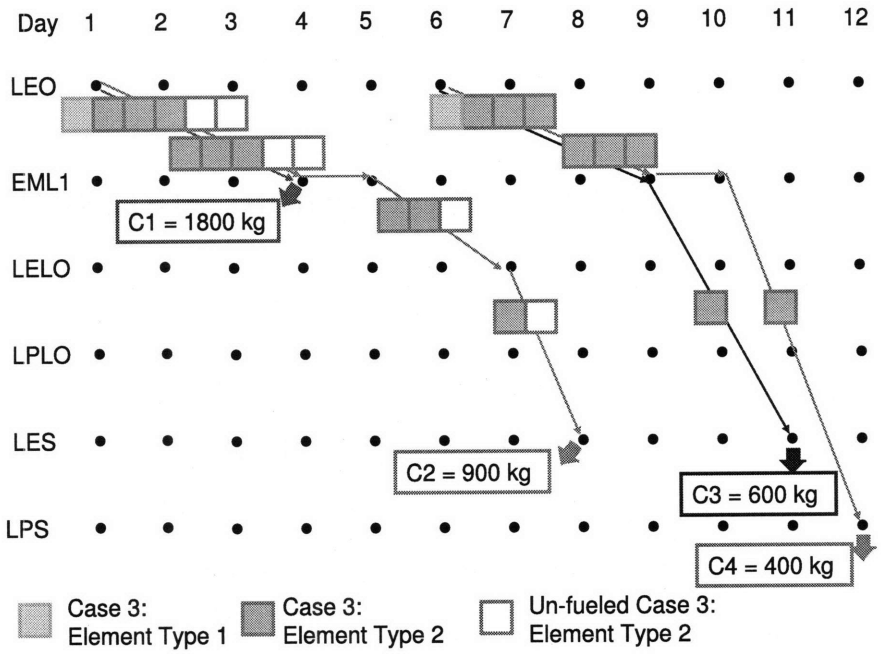


Figure 5-7: Optimal Configuration for Case 3

Table 5.6: Element Specification for Case 3

| Element Type | Fuel Type | Fuel Mass (kg) | Capacity (kg) | Dry Mass (kg) |
|---|---|---|---|---|
| 1 | 2 | 68581 | 0 | 9025 |
| 2 | 2 | 9519 | 600 | 2998 |

listed in Table 5.6 and the optimal configuration is provided in Figure 5-7.

Examining Table 5.6 and Figure 5-7 reveals that the element design types continue the trend of selecting a large-propulsive only element and a smaller dual function element. Here, however, the dual function element is only slightly larger than the dual function element selected in both the baseline and Case 1. The large propulsive-only element designed here (Element Type 1) is much smaller than the propulsive-only element in Case 1 (Element 412).

Comparing Figures 5-5 and 5-7, the two architectures seem remarkably similar upon an initial examination. Utilizing the same network paths and the same number of large propulsive-only elements and dual function elements, it is not initially apparent where the 19% reduction in total system wet mass is obtained. However,

| Classifications | Element Design Parameter | Baseline Solution & Network-Only (Case 1) | Vehicle-Only (Case 2) | Concurrent (Case 3) |
|---|---|---|---|---|
| Propulsive Only | Element ID | # 412 EDS | Element Type 1 | Element Type 1 |
| | Com. Mass | 0 kg | 0 kg | 0 kg |
| | Fuel Mass | 129500 kg | 80646 kg | 68561 kg |
| | Dry Mass | 15852 kg | 10427 kg | 9025 kg |
| Dual Function | Element ID | # 307 Apollo DS | Element Type 2 | Element Type 2 |
| | Com. Mass | 500 kg | 1200 kg | 600 kg |
| | Fuel Mass | 8804 kg | 14191 kg | 9519 kg |
| | Dry Mass | 2509 kg | 2222 kg | 2998 kg |

Figure 5-8: Summary of Design Parameters for Baseline and Three Optimization Cases of Space Transportation Example

the slight increase in both carrier capability and propulsive capability of the dual function element is enough to allow the large propulsive element to be much smaller in size. In Figure 5-5, Element 412 is utilized to perform both the burn from LEO and the burn into EML1; however in Case 3 (Figure 5-7), Element type 1 is only used for the burn out of LEO and the dual function element performs the burn into EML1 for both flights. The reduction in fuel mass for the large propulsive-only element design produced by Case 3 decreases the total system wet mass for the transportation architecture significantly.

## 5.7 Comparative Analysis

Utilizing the space representation in Figure 5-3, the elements utilized in the transportation architectures for Cases 1 through 3 are plotted in Figure 5-9. The maximum $\Delta V$ of each element is defined as the fully loaded fully fueled propulsive capability of the element when utilized in isolation. The maximum payload capacity defines each

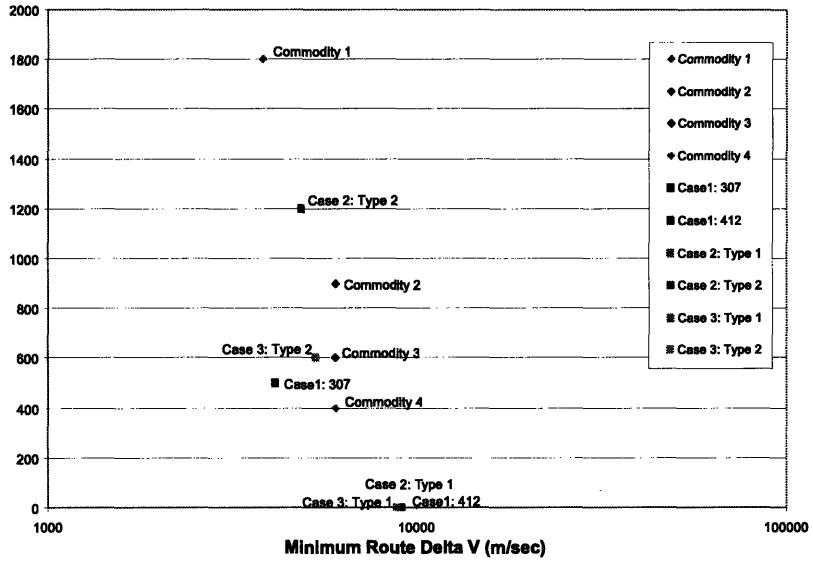


Figure 5-9: Delta V versus Demand for Space Transportation Network Example

elements carrier functionality.

In addition to the element design points, each commodity is plotted in this functional design space to obtain a perspective on the functional requirements of the network. The commodity design points represent the minimum cumulative $\Delta V$ required to transport each commodity from its supply node to its demand node and the total mass of each commodity. Thus, utilizing the commodity points as a guide, the efficiency of the elements can be examined.

Examining Figure 5-9 reveals that none of the elements designed are capable of delivering any of the commodities alone, and therefore a combination of elements must be selected to satisfy the demand. However, by forming stacks of elements, the total payload mass that any given element must transport is significantly increased, which decreases the total $\Delta V$ that the element can provide. As the effects of these combinations are non-linear, the space transportation system element design points can not be neatly decomposed, as defined in the air transportation system analysis provided in Section 4.5. However, analyzing Figure 5-9 provides insight into the efficiency of the elements selected and designed with respect to the network requirements.

Examining Figure 5-9 shows that the large propulsive elements designed in Case 2 and Case 3 have similar capabilities to that of Element 412. However, examining Table 5-8, reveals that the large propulsive-only elements designed in Cases 2 and 3 are smaller than Element 412 (Case 1). Thus, the additional fuel mass provided by Element 412 is balanced by the larger structural mass and therefore does not significantly impact the propulsive capability, as compared to the propulsive-only designs from Cases 2 and 3.

A greater differentiation exists when comparing the dual function elements from all three cases. As the fuel and structural masses of these elements are smaller than the propulsive-only elements, changes in the fuel and carrier capabilities produce a larger change in functionality. The pre-defined dual function element from Case 1 (Element 307) has the lowest propulsive and carrier capability. The dual function element designed by the vehicle-only optimization (Case 2) has significantly more carrier and propulsive capability than Element 307. The dual function element designed in the concurrent optimization has only slightly more carrier functionality than Element 307, however has significantly more propulsive capability than both dual function elements. By examining Figure 5-8 we see that the dual function element from Case 3 has only a slightly greater fuel mass capacity than Element 307 and significantly less fuel mass capacity, as compared to the dual function element in Case 2. However, as described above, the increased fuel mass capacity of the Case 2 dual function element results in an increased dry mass, which limits the capability of this element. Thus, the Case 3 dual function element seems to provide a good balance between carrier functionality and propulsive capability.

To provide a greater understanding of the transportation architecture efficiency for each case, Table 5.7 details the percentage of utilized functionality. The carrier capability utilization is defined as the ratio of the total commodity mass of the network (3700 kg) to the total carrier capacity of each transportation architecture. The propulsive capability utilization is defined as the ratio of the total fuel required to perform the transportation to the total fuel capacity of the elements within the network. The inclusion of propulsive elements that are not-fueled and are solely utilized

Table 5.7: Percent-Utilization of Total Element Architecture Capabilities for all Cases

| Case | Carrier Capability Utilization | Propulsive Capability Utilization (Fueled Elements Only) | Propulsive Capability Utilization (All Elements) |
|---|---|---|---|
| Baseline | 60% | 25% | 24% |
| Case 1 | 92% | 49% | 44% |
| Case 2 | 62% | 40% | 38% |
| Case 3 | 77% | 59% | 54% |

for their carrier functionality distinguish the last two columns.

Examining Table 5.7 reveals that the network-only optimization produces the highest carrier utilization, with a 92% efficiency. The highest propulsive capability utilization is obtained by the concurrently optimized element designs from Case 3. Comparing the transportation architecture costs for the baseline solution and the three optimization cases reveals that increased efficiency in propulsive utilization directly correlates with lower system wet mass and efficient carrier utilization is less important. For example, the vehicle-only optimization (Case 2) shows only a 2% improvement in carrier utilization, as compared to the baseline architecture, but a 15% improvement in propulsive capability, which can therefore be attributed to the 11% decrease in system wet mass.

## 5.8 Chapter Summary

In this chapter, the integrated transportation system design problem presented in Section 2.1 was implemented for a space transportation system example. Using the space transportation system definition presented in Section 3.2 and the models developed in Section 3.3, computational results were obtained for both traditional design methodologies and the integrated transportation system method. A baseline solution was first constructed, using a direct mission architecture and existing elements. The baseline architecture was compared to the traditional design methodologies of

network-only and vehicle-only optimization, which showed that by optimizing the transportation architecture by either of these methods produced a reduction in total system wet mass in LEO. Finally, using the results obtained from the traditional optimization approaches, a solution to the integrated transportation system design problem was obtained and showed a 19% improvement in total system wet mass, as compared to the best traditional solution found. Upon analyzing the results, the primary reduction in system wet mass was attributed to a higher efficiency in propulsive functionality.

# Chapter 6

# Computational Study of Embedded Optimization Methodology

The integrated transportation system design problems presented in previous chapters have shown improvements in operational efficiency over traditional design methods. The embedded optimization methodology presented for solving the integrated air and space transportation systems embeds a deterministic constraint solver, such as an LP or CPLEX, into the heuristic optimization algorithm Simulated Annealing (SA) to aid SA in solving the large number of constraints generated by the network flow and operations sub-systems. The motivation behind augmenting SA in this manner was to increase the effectiveness of SA for finding good, feasible solutions to highly constrained problems.

In this chapter, a computational study is presented to validate the use of embedded optimization over traditional implementations of Simulated Annealing, such as penalty parameters and smart perturbation, for solving constrained optimization problems. In Section 6.1 the formulations for each of the traditional implementations of Simulated Annealing, as well as the embedded optimization approach is presented for the integrated air transportation problem. Section 6.2 presents the example problem and a preliminary experiment used to determine a suitable set of algorithm parameters. Section 6.3, discusses the results of the computational experiment, and a statistical analysis of these results is presented in Section 6.4 to validate the efficiency

of embedded optimization over the traditional SA implementation.

## 6.1 Optimization Methodology Formulation

To measure the effectiveness of embedded optimization against the traditional Simulated Annealing implementations of penalty parameters and smart perturbation, all three methods are implemented for the integrated air transportation system design problem presented in Section 3.1. The remainder of this section details the specific implementations required to optimize the integrated air transportation network design problem using each methodology.

### 6.1.1 Formulation of SA with Penalty Parameters

Penalty parameters are a typical method for implementing constraints in optimization methods, and particularly in heuristics. Penalty parameters weigh constraint violations in the objective, allowing non-feasible solutions to be evaluated, but penalizing the objective proportionally to the violation. For this implementation, an increasing penalty is imposed as the algorithm progresses. Specifically, given the number of system evaluations ($n_{sys}$) the penalty parameter ($\lambda$) for each constraint is defined as

$$\lambda = \exp n_{sys}/1000 \tag{6.1}$$

where the constant of 1000 is imposed to provide the appropriate order of magnitude of the penalty parameter.

Figure 6-1 provides a description of the implementation of Simulated Annealing with penalty parameters. The design variables for this implementation include the vehicle design variables, the aircraft allocation variables, and the package flow variables. Each perturbed design vector is evaluated to determine the total system objective, which for this implementation, augments the system objective with the constraint violations as provided in Equation 6.2.

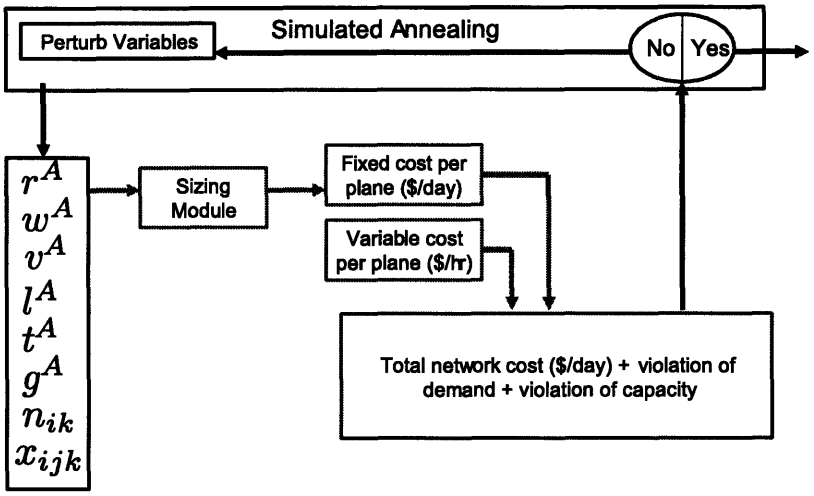


Figure 6-1: Flow Diagram of Simulated Annealing with Penalty Parameters

$$\begin{aligned}\mathcal{J} \;=\; & \sum_i \sum_k c_{ik} n_{ik} + \lambda \left[ \sum_i \sum_j \left| \sum_k x_{ijk} - P_{ij} \right| \right. \\ & + \sum_i \sum_k \left[ \left( \text{if} \sum_j x_{ijk} > G_{ik} \right) \left( \sum_j x_{ijk} - G_{ik} \right) \right] \\ & \left. + \sum_j \sum_k \left[ \left( \text{if} \sum_i x_{ijk} > G_{jk} \right) \left( \sum_i x_{ijk} - G_{jk} \right) \right] \right]\end{aligned} \tag{6.2}$$

The first term in Equation 6.2 is the system level objective, as defined in Equation 3.12, which represents the total operating costs of the air transportation system for a single day of operations. The second term represents the violation of the demand constraints. As the demand constraints are equality constraints, any deviation from equality produces increases the system objective. The third term and fourth terms represent violations in the capacity constraints, for the first and second legs of the trip, respectively. As the capacity constraint is a set of inequality constraints, the system objective is only penalized if the total package mass exceeds the total aircraft capacity available.

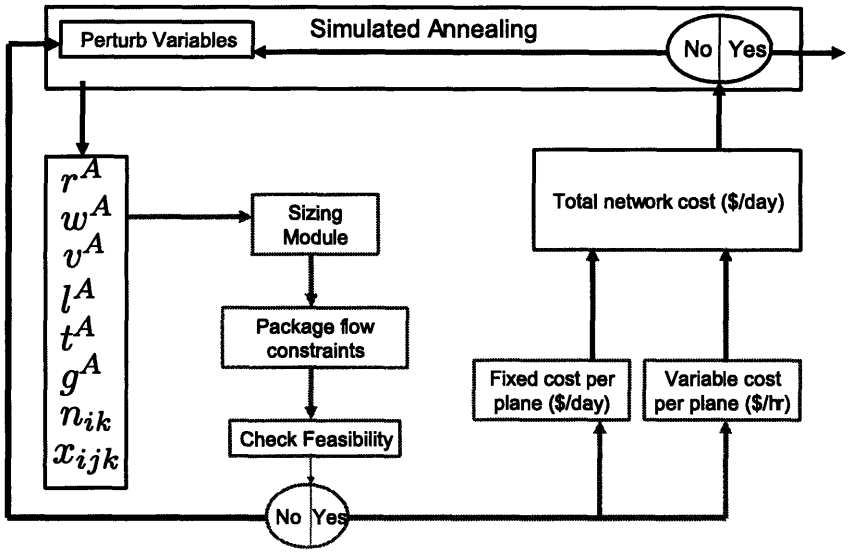


Figure 6-2: Flow Diagram of Simulated Annealing with Smart Perturbation

### 6.1.2 Formulation of SA with Smart Perturbation

For problems with many constraints, the use of penalty parameters may become ineffective. An alternative approach to satisfying constraints within Simulated Annealing is to implement constraints within the perturbation function to ensure that the perturbed solution is feasible, before evaluation by the objective function. However, for problems with equality constraints, randomly perturbing the design values is unlikely to produce a feasible solution. The idea of smart perturbation, as referred to here, is to perturb the variables in an intelligent manner that promotes constraint satisfaction, using a set of rules.

As can be seen in Figure 6-2 the design vector consists of the vehicle design variables, the aircraft allocation variables and the package flow variables. Since the demand constraints require equality of the package flow to the prescribed demand, randomly perturbing the package flow variables will not likely produce a feasible solution. Therefore, given an understanding of the problem structure a smart perturbation routine can be developed for perturbing the package variables.

The package variable perturbation proceeds as follows. First, a random package variable is selected. The package variable is denoted by $x_{i\acute{i}j\acute{k}}$ and is a continuous variable between $0 \leq x_{i\acute{i}j\acute{k}} \leq P_{i\acute{i}}$. Next, the perturbation value for this variable ($\delta x$) is selected such that $0 \leq \delta x \leq x_{i\acute{i}j\acute{k}}$, where $\delta x$ is rounded to the nearest integer. The amount $\delta x$ is subtracted from the given package variable and added to another package variable $x_{i\acute{i}j\bar{k}}$. This perturbation scheme produces a package flow that always satisfies the demand constraints, given an feasible initial solution. The remainder of the variables are perturbed as typically done in Simulated Annealing, and once a perturbed design vector satisfies all constraints, it is evaluated by the objective function.

### 6.1.3 Formulation of SA with Embedded Optimization

The embedded optimization methodology provides an efficient approach for solving problems where large numbers of constraints are required to obtain feasible solutions. The embedded optimization framework is similar to that of SA with smart perturbation in that the design variables are perturbed until a feasible design vector is found. However, unlike the smart perturbation methodology, embedded optimization uses a deterministic optimizer embedded into the perturbation step of Simulated Annealing to effectively solve the set of constraints governing the package variables.

By examining Figure 6-3, we see that the package variables need not be part of the design vector as they do not contribute to the objective function and are not perturbed by SA. Furthermore, by embedding a linear programming solver (LP) within the perturbation, a feasible set of package variables is determined, if one exists. If a feasible set of package variables does not exist, then the design vector is perturbed until a feasible architecture is constructed, and can be evaluated by the objective function.

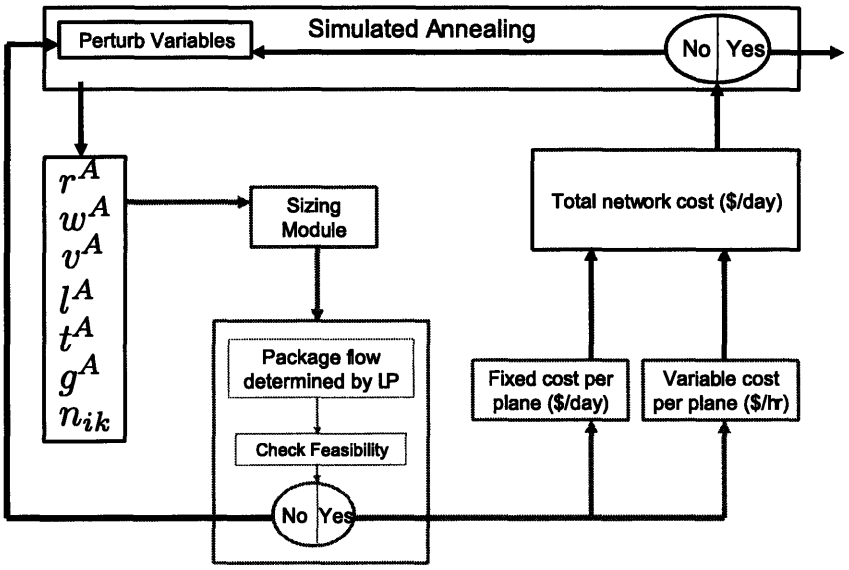


Figure 6-3: Flow Diagram of Simulated Annealing with Embedded Optimization

## 6.2 Computational Study Preliminaries

The computational study investigates the effect of increasing problem size on the computational requirements of each method described in Section 6.1. The integrated transportation system design problem can increase in size by including additional vehicle design variables or additional network flow variables. Increasing the vehicle design variables would consist of increasing the number of design variables required to define an aircraft design, or the number of designs considered. Increasing the network flow variables allows for additional cities to be considered in the network. Since the three optimization methodologies considered differ most significantly in constraint handling, an increase in the number of cities in the network would most drastically change the number of constraints. As such, the experiment conducted will measure the performance of the three optimization methodologies as the *number of cities in the network increases*, while keeping the number of vehicle design variables fixed.

Table 6.1: City to City Distances for Largest Seven City Network (nautical miles)

| | ATL | BOS | ORD | DFW | LAX | JFK | SFO |
|---|---|---|---|---|---|---|---|
| ATL | 0 | 934 | 622 | 688 | 1921 | 756 | 2179 |
| BOS | 934 | 0 | 882 | 1538 | 2629 | 183 | 2729 |
| ORD | 622 | 882 | 0 | 806 | 1767 | 713 | 1866 |
| DFW | 688 | 1538 | 806 | 0 | 1257 | 1360 | 1518 |
| LAX | 1921 | 2629 | 1767 | 1257 | 0 | 2454 | 330 |
| JFK | 756 | 183 | 713 | 1360 | 2454 | 0 | 2560 |
| SFO | 2179 | 2729 | 1866 | 1518 | 330 | 2560 | 0 |

Table 6.2: Demand for Largest Seven City Network (lbs)

| | ATL | BOS | ORD | DFW | LAX | JFK | SFO |
|---|---|---|---|---|---|---|---|
| ATL | 0 | 14045 | 31313 | 19984 | 34506 | 57949 | 37318 |
| BOS | 14045 | 0 | 27261 | 17398 | 30041 | 50451 | 32489 |
| ORD | 31313 | 27261 | 0 | 38788 | 66975 | 112479 | 72434 |
| DFW | 19984 | 17398 | 38788 | 0 | 42743 | 71784 | 46227 |
| LAX | 34506 | 30041 | 66975 | 42743 | 0 | 123948 | 79820 |
| JFK | 57949 | 50451 | 112479 | 71784 | 123948 | 0 | 134050 |
| SFO | 37318 | 32489 | 72434 | 46227 | 79820 | 134050 | 0 |

### 6.2.1 Example Problem

To measure the effectiveness of embedded optimization against the traditional Simulated Annealing implementations of penalty parameters and 'smart' perturbation, all three methods are implemented for the integrated air transportation system design problem presented in Section 3.1. The computational experiment will measure the performance of each algorithm as the number of cities in the network grows. To this end, the specific example considered is the N-largest city network presented in Section 4.1, and reproduced below for convenience.

The N-largest city network is defined to be the N-cities with the greatest demand.

Beginning with the largest two cities of JFK and SFO, the number of cities considered is expanded by including the next largest demand city, and therefore the network expands to include LAX, ORD, DFW, ATL, and finally BOS.

### 6.2.2 Determining Suitable SA Parameters

As a heuristic method, Simulated Annealing does not guarantee a bound on the optimality of the solution, but instead relies on parameters that govern convergence and stopping criteria. Three parameters were identified as potentially having a significant effect on performance: temperature step (dT), number of perturbations at equilibrium for a given temperature (neq), and the number of temperature steps without improvement before termination (nfz). The temperature step governs the temperature profile for Simulated Annealing which determines how likely a given design vector is to be accepted as the current solution. The number of points at equilibrium specifies the required number of evaluations that do not improve the objective value before the temperature can be lowered. The final parameter, nfz, specifies the required number of temperature steps to be taken without improvement in the solution before the algorithm can terminate. A more detailed description of these algorithm parameters and the methodology of Simulated Annealing can be found in Appendix A and in *Kirkpatrick (1983)*[45].

As with many heuristics, good values of the SA algorithm parameters identified are highly problem dependent. In addition, a good set of parameters may vary for each optimization method; however to compare the optimization algorithms fairly it is desirable to find a single set of parameter values to apply uniformly for all cases. Therefore, a small design of experiments (DOE) is performed on the three city case (consisting of JFK, LAX and SFO) to determine a good set of parameters for the computational experiment.

To perform a design of experiments, multiple values, or levels, are selected for each of the factors in question, which in this case corresponds to the SA algorithm parameters. For each SA parameter identified above, a high and low level are defined, as shown in Table 6.3. The experiment consists of eight trials, corresponding to a full

Table 6.3: SA Algorithm Parameter Levels

| | dT | neq | nfz |
|---|---|---|---|
| High | .95 | 40 | 15 |
| Low | .85 | 20 | 5 |

Table 6.4: Trial Parameters for Design of Experiments

| Trial | dT | neq | nfz |
|---|---|---|---|
| **Trial 1** | High | High | High |
| **Trial 2** | High | High | Low |
| **Trial 3** | High | Low | High |
| **Trial 4** | High | Low | Low |
| **Trial 5** | Low | High | High |
| **Trial 6** | Low | High | Low |
| **Trial 7** | Low | Low | High |
| **Trial 8** | Low | Low | Low |

factorial experiment of the SA algorithm parameter values, and each trial is performed once for each optimization method. For each trial, two performance criteria were selected, computational time and best objective function value found. Table 6.5 and Figure 6-4 provide the results of this experiment.

Figure 6-4 reveals that Trial 6 is non-dominated for each of the three optimization implementations and produces the lowest objective function value for each. In addition, Trial 6 produces these objective values within a reasonable time (approximately 9 minutes or less). Trial 8 is also non-dominated for all three optimization methodologies as well, but results in significantly higher objective function values with only a small decrease in computation time. Therefore, the SA algorithm parameters of $dT = .85$, $neq = 40$, $nfz = 5$, corresponding to Trial 6, will be used for the computational experiment.

Table 6.5: Experimental Results from SA Parameter Study

| Trial # | Penalty Cost ($) | Parameter Time (sec) | Smart Cost ($) | Perturbation Time (sec) | Embedded Cost ($) | Optimization Time (sec) |
|---|---|---|---|---|---|---|
| **Trial 1** | 266410 | 3632 | 233240 | 4477 | 158320 | 2261 |
| **Trial 2** | 266410 | 3328 | 236250 | 3265 | 179480 | 1908 |
| **Trial 3** | 265560 | 1227 | 236100 | 1273 | 160300 | 799 |
| **Trial 4** | 262890 | 651 | 222110 | 916 | 186640 | 531 |
| **Trial 5** | 261020 | 932 | 205130 | 1121 | 164090 | 604 |
| **Trial 6** | 242690 | 502 | 193550 | 543 | 153210 | 366 |
| **Trial 7** | 266410 | 276 | 240210 | 379 | 175330 | 229 |
| **Trial 8** | 266410 | 203 | 254000 | 240 | 175190 | 178 |

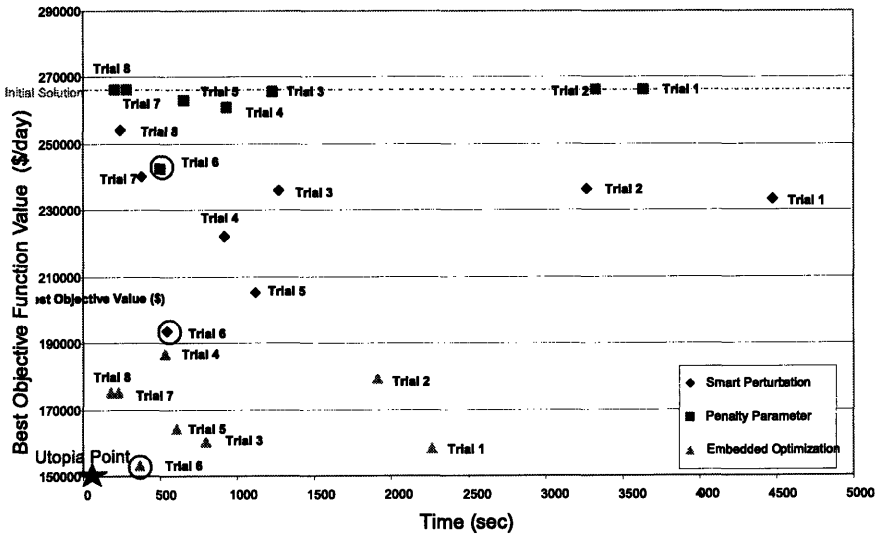


Figure 6-4: Best Objective Function Value vs. Computation Time for Varying SA Parameters

## 6.3 Computational Study

Utilizing the SA algorithm parameters obtained from the pre-experiment, each optimization method presented in Section 6.1 will be implemented for the N-City optimization problem. For each number of cities, each optimization methodology will be run five times to provide a reasonable sample size. The problem is implemented in Matlab[56] and the embedded LP is provided by the Matlab Optimization Toolbox. All results are run on a dedicated Dell Precision M65 laptop with a 2.33 GHz dual-core processor and 2 Gb DDR2 RAM.

To compare each optimization method, four performance metrics will be evaluated: best system objective found ($\mathcal{J}$), computational time ($t$), number of objective function evaluations ($nsys$), and number of perturbations ($npert$). The best system objective determines how effective the optimization methodology is at finding good design regions. The computational time measures how quickly good design regions can be found and how quickly the optimizer can evaluate the constraints. This performance metric is especially important when evaluating the embedded optimization methodology since it is important to determine how costly it is to embed an LP solver into a heuristic. The next two performance metrics determine how many evaluations SA requires in order to satisfy the termination criteria. The number of system evaluations determines how many design points are evaluated by the objective function. However, since both the smart perturbation and embedded optimization methodologies often require multiple iterations to obtain a feasible perturbed design point, it is important to measure the number of perturbed solutions as well to obtain a fair comparison with the penalty parameter implementation.

### 6.3.1 Computational Study Results

The results from the computational study described above are provided in this section. However, it is important to note two deviations from the plan described earlier. First, for the 2-city case of penalty optimization, it was necessary to run approximately 20 optimizations to obtain 5 feasible solutions. For this small case, the optimal

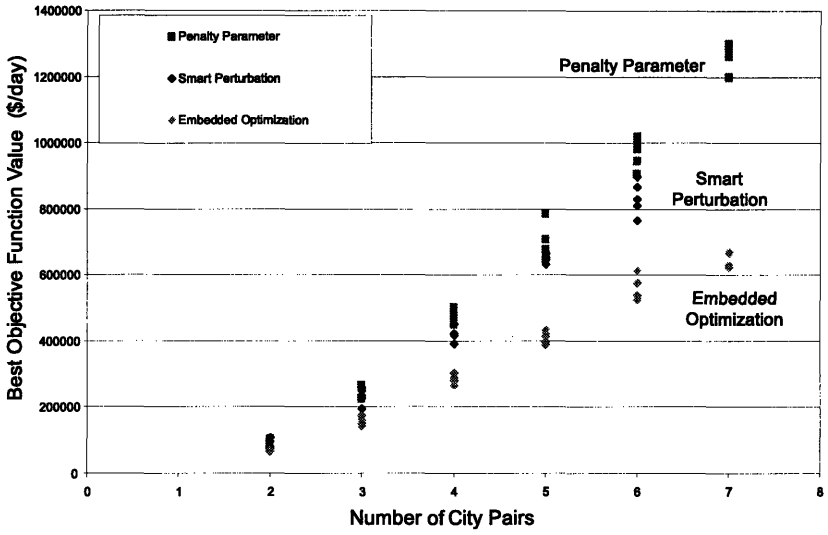


Figure 6-5: Best Objective Function Value verses Number of Cities

configuration returned often had violations in the constraints, and these solutions were discarded. Secondly, results are not available for the 7-city case with smart perturbation optimization. This case was neglected due to time constraints as the computation time significantly increased, as compared to the 6-city case and became prohibitive.

Figure 6-5 displays the results for the best objective function values ($\mathcal{J}$) versus number of cities ($N$) for each optimization method, and Table 6.6 provides the average values obtained by each optimization methodology for this performance metric. By examining Figure 6-5 and Table 6.6 we see that *embedded optimization has the lowest average objective function value for every N-city optimization.* In addition, the standard deviations in objective function values for embedded optimization are lower than the standard deviations for each of the traditional optimization methodologies for every N-city optimization, with the exception of the smart perturbation implementation for the 5-city network. Given that the objective function values obtained by embedded optimization are lower, by at least 20%, and generally more consistent than the traditional optimization implementations, implies that embedded optimiza-

Table 6.6: Average Objective Function Values for N-City Optimization

| # Cities | Penalty Parameter ($/day) | Smart Perturbation ($/day) | Embedded Optimization ($/day) | % Improvement Embedded vs. Penalty | % Improvement Embedded vs. Smart |
|---|---|---|---|---|---|
| 2 | 93886 | 87200.6 | 66022 | 30% | 23% |
| 3 | 257842 | 232502 | 160970 | 38% | 31% |
| 4 | 479898 | 418998 | 286348 | 40% | 32% |
| 5 | 721384 | 651866 | 410260 | 43 % | 37% |
| 6 | 973552 | 834556 | 555870 | 43% | 33% |
| 7 | 1263880 | N/A | 650862 | 49% | N/A |

tion can reliably return lower cost solutions for the integrated transportation system design problem.

Figure 6-6 displays the results for the computational time versus number of cities for each optimization method, and Table 6.7 provides the average values obtained by each optimization methodology for this metric. By examining Figure 6-7 and Table 6.8 we see that *embedded optimization also has the lowest average computation times and grows at the slowest rate with increasing problem size.* In addition, the standard deviations in computation time for embedded optimization are lower than the standard deviations for each of the traditional optimization methodologies for every N-city optimization, which implies that the performance of embedded optimization is more predictable than the other methods.

Performing an empirical fit to the equation $t = \alpha N^{\beta}$ and requiring a confidence interval of 95%, we obtain coefficient values of $\alpha = .88 \pm 1.99$ and $\beta = 4.56 \pm 1.19$ for penalty parameter optimization, $\alpha = 2.88E-5 \pm 6.11E-4$ and $\beta = 11.67 \pm 11.85$ for smart perturbation optimization, and $\alpha = 100.66 \pm 25.11$ and $\beta = 1.21 \pm .14$ for embedded optimization. Examining these coefficients reveals that the computation time for the penalty parameter implementation scales at four times the rate as embedded optimization. For the smart perturbation implementation, this ratio increases to approximately ten times that of embedded optimization. Although, the number of

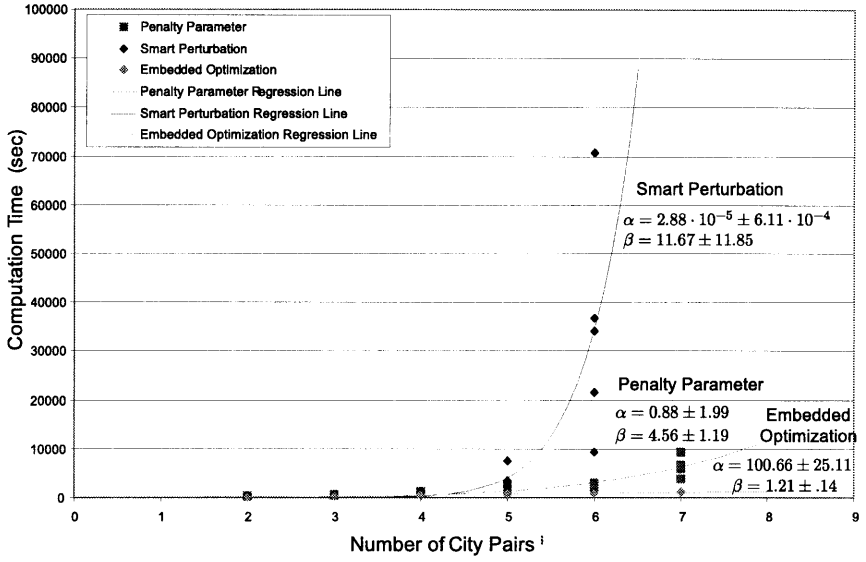


Figure 6-6: Computational Time versus Number of Cities

Table 6.7: Average Computational Time for N-City Optimization

| # Cities | Penalty Parameter (sec) | Smart Perturbation (sec) | Embedded Optimization (sec) | % Improvement Embedded vs. Penalty | % Improvement Embedded vs. Smart |
|---|---|---|---|---|---|
| 2 | 288.6 | 361.6 | 242.6 | 16% | 41% |
| 3 | 549.8 | 580.4 | 403.8 | 27% | 30% |
| 4 | 944.8 | 1121 | 551 | 42% | 51% |
| 5 | 1546.2 | 3967 | 707.6 | 54% | 82% |
| 6 | 2566 | 34462.6 | 789.8 | 69% | 98% |
| 7 | 6487.4 | N/A | 1129.2 | 83% | N/A |

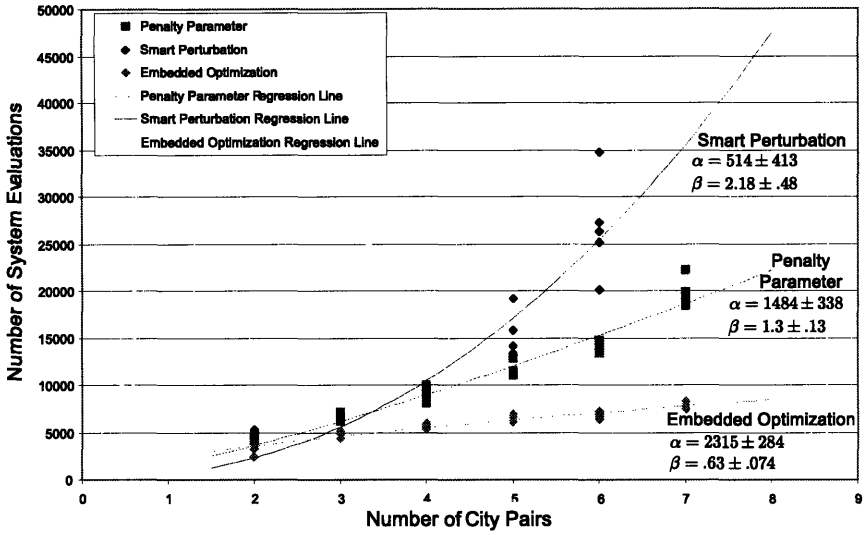


Figure 6-7: Number of System Evaluations versus Number of Cities

cities in the network ($N$) defines $N^2$ allocation variables and $N^3$ package variables, the computation time for embedded optimization increases $O\left(N^{1.21}\right)$, which is a considerable improvement, considering that embedded optimization also produces the best objective function values.

The ranges for the $\alpha$ coefficients are fairly large with respect to magnitude of the values, however, the ranges for the $\beta$ coefficients are tighter, which is important since the $\beta$ coefficients determine the increase in computation time for growing problem size. For smart perturbation the ranges for both coefficients ($\alpha$ and $\beta$) are so large that the numbers provided may be meaningless, which is due to the increased variation between data points for each run. The $\beta$ coefficient for embedded optimization has the tightest range which ensures that the actual growth in computation time for increasing problem size is close to the predicted value.

Figure 6-7 displays the results for the number of system evaluations versus the number of cities for each optimization methodology, and Table 6.8 provides the average values obtained by each optimization methodology for this performance metric. By examining Figure 6-7 and Table 6.8 we see that *embedded optimization has the*

Table 6.8: Average Number of System Iterations for N-City Optimization

| # Cities | Penalty Parameter | Smart Perturbation | Embedded Optimization | % Improvement Embedded vs. Penalty | % Improvement Embedded vs. Smart |
|---|---|---|---|---|---|
| 2 | 4311.2 | 5035.8 | 3200.8 | 26% | 43% |
| 3 | 6588.4 | 6677.6 | 4837.8 | 27% | 28% |
| 4 | 8965.6 | 9333.6 | 5741 | 36% | 38% |
| 5 | 11557.4 | 15125 | 6509 | 44% | 57% |
| 6 | 13987.6 | 26698.8 | 6752.2 | 52% | 75% |
| 7 | 19736.8 | N/A | 7843.4 | 60% | N/A |

*lowest average number of system evaluations and grows at the slowest rate with increasing problem size.* In addition, the standard deviations in the number of system evaluations for embedded optimization are lower than the standard deviations for each of the traditional optimization methodologies for every N-city optimization except the 2-city case, which implies that the performance of embedded optimization is more predictable than for the other methods.

Performing an empirical fit to the equation $t = \alpha N^{\beta}$ and requiring a confidence interval of 95%, we obtain coefficient values of $\alpha = 1484 \pm 338$ and $\beta = 1.3 \pm .13$ for penalty parameter optimization, $\alpha = 514 \pm 413$ and $\beta = 2.18 \pm .48$ for smart perturbation optimization, and $\alpha = 2315 \pm 284$ and $\beta = .63 \pm .074$ for embedded optimization. Examining these coefficients reveals that the number of system evaluations for the penalty parameter implementation scales at about twice the rate as embedded optimization. For the smart perturbation implementation, this ratio increases to approximately four times that of embedded optimization. Given the order of increase in problem size with number of cities ($N$), the resulting increase in system evaluations for embedded optimization, which is approximately $O\left(N^{.625}\right)$, is a significant improvement.

The coefficient ranges are tighter when evaluating the curve fits for number of system evaluations, providing a reliable estimate on the number of system evaluations.

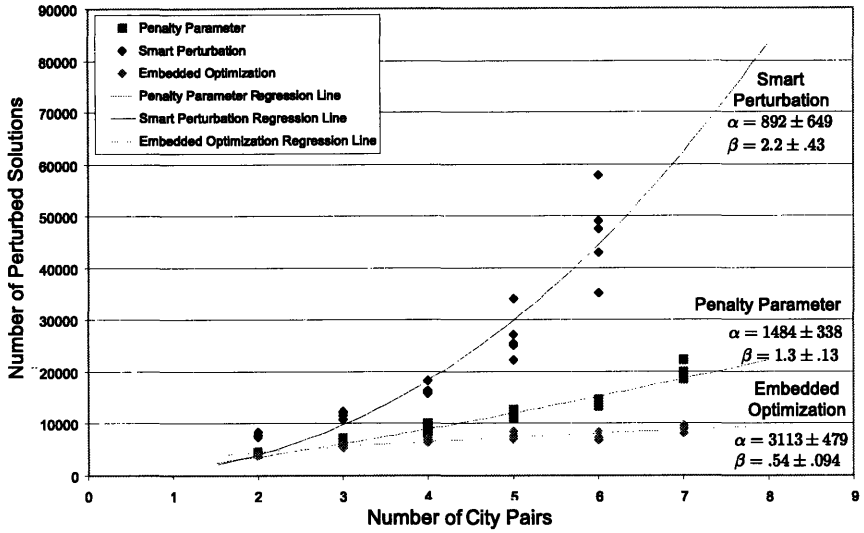


Figure 6-8: Number of Perturbed Solutions versus Number of Cities

The penalty parameter curve fit provides a reliable estimate for the $\beta$ coefficient which predicts a super-linear growth in number of system evaluations with growing problem size. For smart perturbation, there is still ambiguity in the $\alpha$ coefficient, however the $\beta$ coefficient shows that the number of system evaluations scales approximately quadratically with the number of city pairs. The $\beta$ coefficient for embedded optimization provides the tightest range and reveals that the growth in number of system evaluations for increasing problem size is sub-linear.

Figure 6-8 displays the results for the number of perturbed solutions versus number of cities for each optimization methodology, and Table 6.9 provides the average values obtained by each optimization methodology for this metric. By examining Figure 6-8 and Table 6.9 we see that *embedded optimization has the lowest average number of perturbed solutions and grows at the slowest rate with increasing problem size.* In addition, the standard deviations in number of perturbed solutions for embedded optimization are lower than the standard deviations for each of the traditional optimization methodologies for every N-city optimization except the 2-city case, which implies that the performance of embedded optimization is more predictable than the

Table 6.9: Average Number of Perturbed Solutions for N-City Optimization

| # Cities | Penalty Parameter | Smart Perturbation | Embedded Optimization | % Improvement Embedded vs. Penalty | % Improvement Embedded vs. Smart |
|---|---|---|---|---|---|
| 2 | 4311.2 | 7742.4 | 4010.8 | 7% | 87% |
| 3 | 6588.4 | 11856.6 | 5896.8 | 10% | 50% |
| 4 | 8965.6 | 16554.6 | 6976.4 | 22% | 58% |
| 5 | 11557.4 | 26812.4 | 7677.6 | 34% | 71% |
| 6 | 13987.6 | 46539.8 | 7471.8 | 47% | 84% |
| 7 | 19736.8 | N/A | 8954.4 | 55% | N/A |

other methodologies.

Performing an empirical fit to the equation $t = \alpha N^{\beta}$ and requiring a 95% confidence interval, we obtain coefficient values of $\alpha = 1484 \pm 388$ and $\beta = 1.3 \pm .13$ for penalty parameter optimization, $\alpha = 892 \pm 649$ and $\beta = 2.2 \pm .43$ for smart perturbation optimization, and $\alpha = 3113 \pm 479$ and $\beta = .54 \pm .094$ for embedded optimization. Examining these coefficients reveals that the number of perturbed solutions for the penalty parameter implementation scales at about twice the rate as embedded optimization. For the smart perturbation implementation, this ratio increases to approximately four times that of embedded optimization. The number of perturbed solutions required by embedded optimization increases at approximately $O(N^{.53})$, which is a marked improvement given the increasing dimensionality of the problem.

The coefficient ranges produced for the curve fits for number of perturbed solutions are fairly tight. For smart perturbation, there is still ambiguity in the $\alpha$ coefficient, however the $\beta$ coefficient shows that the number of perturbed solutions scales approximately quadratically with the number of city pairs. The $\beta$ coefficient for embedded optimization provides the tightest range and reveals that the growth in number of perturbed solutions for increasing problem size is sub-linear.

## 6.4 Statistical Analysis

The computational results presented in the previous section show an improvement in the computational performance of embedded optimization over traditional Simulated Annealing optimization implementations. However, it is important to quantify these improvements using a formal analysis. As such, a statistical test of the hypothesis that embedded optimization performs better than the other two methods.

Due to the small sample sizes, a t-test was used to compare two groups, namely embedded optimization and each of the traditional optimization methodologies. The t-test determines if a result is significant, given a required confidence interval for the results. In order to perform a t-test, the average, standard deviation, and sample size for each group is used to compute the t-value, which is then compared to a table of values to determine if the null hypothesis can be rejected. Details regarding the specifics of the t-test can be found in any basic statistics reference, such as *Schaum's Outlines*[57]. For each of the hypothesis tests performed, an unpaired two-tail t-test with a 95% confidence interval was evaluated to determine if the results were significant. The significance tests were performed for the four performance metrics, objective function value, computation time, number of system evaluations, and number of perturbed solutions, and the results are provided in Tables 6.10, 6.11, 6.12, and 6.13, respectively.

Table 6.10 presents the hypothesis test for a decrease in the objective function values obtained by embedded optimization as compared to optimization with penalty parameters ($\mathcal{J}_E < \mathcal{J}_P$), and smart perturbation ($\mathcal{J}_E < \mathcal{J}_S$). Examining Table 6.10, reveals that *embedded optimization produces significantly lower objective function values than both penalty parameter and smart perturbation with a confidence interval of 95%*. Furthermore, for the networks with three or more cities, the results are considered extremely statistically significant, with probabilities of the null hypothesis less than .001%.

Table 6.11 presents the hypothesis tests for a decrease in computation times obtained by embedded optimization as compared to optimization with penalty param-

Table 6.10: Hypothesis Test for Improvement of Objective Function Value using 95% Confidence Interval

| # of Cities | Hypothesis | t-value | Probability | Significance |
|---|---|---|---|---|
| 2 | $\mathcal{J}_E < \mathcal{J}_P$ | $t(8) = 4.99$ | $p =$ 1.065E-3 | Significant |
| 3 | $\mathcal{J}_E < \mathcal{J}_P$ | $t(8) = 9.29$ | $p =$ 1.463E-5 | Significant |
| 4 | $\mathcal{J}_E < \mathcal{J}_P$ | $t(8) = 16.30$ | $p =$ 2.024E-5 | Significant |
| 5 | $\mathcal{J}_E < \mathcal{J}_P$ | $t(8) = 10.72$ | $p =$ 5.026E-6 | Significant |
| 6 | $\mathcal{J}_E < \mathcal{J}_P$ | $t(8) = 15.60$ | $p =$ 2.846E-7 | Significant |
| 7 | $\mathcal{J}_E < \mathcal{J}_S$ | $t(8) = 30.22$ | $p =$ 1.559E-9 | Significant |
| 2 | $\mathcal{J}_E < \mathcal{J}_S$ | $t(8) = 3.28$ | $p =$ 1.127E-2 | Significant |
| 3 | $\mathcal{J}_E < \mathcal{J}_S$ | $t(8) = 5.64$ | $p =$ 4.877E-4 | Significant |
| 4 | $\mathcal{J}_E < \mathcal{J}_S$ | $t(8) = 11.71$ | $p =$ 2.579E-6 | Significant |
| 5 | $\mathcal{J}_E < \mathcal{J}_S$ | $t(8) = 23.37$ | $p =$ 1.196E-8 | Significant |
| 6 | $\mathcal{J}_E < \mathcal{J}_S$ | $t(8) = 9.75$ | $p =$ 1.029E-5 | Significant |

eters ($t_E < t_P$), and smart perturbation ($t_E < t_S$). Examining Table 6.11 reveals that *embedded optimization produces significantly lower computation times than both penalty parameter and smart perturbation with a confidence interval of 95%.* In addition, many of these tests produce results that are considered extremely statistically significant.

Table 6.12 presents the hypothesis tests for a decrease in the number of system evaluations obtained by embedded optimization as compared to optimization with penalty parameters ($nsys_E < nsys_P$), and smart perturbation ($nsys_E < nsys_S$). Examining Table 6.12 reveals that *embedded optimization produces significant decreases in number of system evaluations as compared to both penalty parameter and smart perturbation with a confidence interval of 95%.* Furthermore, for the networks with three or more cities, the results are considered extremely statistically significant, with probabilities of the null hypothesis less than .001%.

Table 6.13 presents the hypothesis tests for a decrease in the number of perturbations obtained by embedded optimization as compared to optimization with penalty

Table 6.11: Hypothesis Test for Improvement of Computation Time using 95% Confidence Interval

| # of Cities | Hypothesis | t-value | Probability | Significance |
|---|---|---|---|---|
| 2 | $t_E < t_P$ | $t(8) = 4.62$ | $p$ = 1.709E-2 | Significant |
| 3 | $t_E < t_P$ | $t(8) = 5.07$ | $p$ = 9.607E-4 | Significant |
| 4 | $t_E < t_P$ | $t(8) = 5.81$ | $p$ = 3.986E-4 | Significant |
| 5 | $t_E < t_P$ | $t(8) = 7.01$ | $p$ = 1.116E-4 | Significant |
| 6 | $t_E < t_P$ | $t(8) = 11.38$ | $p$ = 3.211E-6 | Significant |
| 7 | $t_E < t_S$ | $t(8) = 6.10$ | $p$ = 2.884E-4 | Significant |
| 2 | $t_E < t_S$ | $t(8) = 10.66$ | $p$ = 5.24E-6 | Significant |
| 3 | $t_E < t_S$ | $t(8) = 8.39$ | $p$ = 3.096E-5 | Significant |
| 4 | $t_E < t_S$ | $t(8) = 8.91$ | $p$ = 1.991E-5 | Significant |
| 5 | $t_E < t_S$ | $t(8) = 3.58$ | $p$ = 7.2156E-3 | Significant |
| 6 | $t_E < t_S$ | $t(8) = 3.27$ | $p$ = 1.133E-2 | Significant |

Table 6.12: Hypothesis Test for Improvement in Number of System Iterations using 95% Confidence Interval

| # of Cities | Hypothesis | t-value | Probability | Significance |
|---|---|---|---|---|
| 2 | $nsys_E < nsys_P$ | $t(8) = 4.81$ | $p$ = 1.340E-3 | Significant |
| 3 | $nsys_E < nsys_P$ | $t(8) = 8.17$ | $p$ = 3.747E-5 | Significant |
| 4 | $nsys_E < nsys_P$ | $t(8) = 9.11$ | $p$ = 1.696E-5 | Significant |
| 5 | $nsys_E < nsys_P$ | $t(8) = 14.52$ | $p$ = 4.942E-7 | Significant |
| 6 | $nsys_E < nsys_P$ | $t(8) = 26.48$ | $p$ = 4.443E-9 | Significant |
| 7 | $nsys_E < nsys_S$ | $t(8) = 17.58$ | $p$ = 1.119E-7 | Significant |
| 2 | $nsys_E < nsys_S$ | $t(8) = 7.91$ | $p$ = 4.769E-5 | Significant |
| 3 | $nsys_E < nsys_S$ | $t(8) = 10.40$ | $p$ = 6.317E-6 | Significant |
| 4 | $nsys_E < nsys_S$ | $t(8) = 18.60$ | $p$ = 7.2101E-8 | Significant |
| 5 | $nsys_E < nsys_S$ | $t(8) = 7.63$ | $p$ = 6.126E-5 | Significant |
| 6 | $nsys_E < nsys_S$ | $t(8) = 8.46$ | $p$ = 2.912E-5 | Significant |

Table 6.13: Hypothesis Test for Improvement in Number of Perturbations using 95% Confidence Interval

| # of Cities | Hypothesis | t-value | Probability | Significance |
|---|---|---|---|---|
| 2 | $npert_E < npert_P$ | $t(8) = 2.11$ | $p$ = 6.830E-2 | Not Significant |
| 3 | $npert_E < npert_P$ | $t(8) = 2.74$ | $p$ = 2.549E-2 | Significant |
| 4 | $npert_E < npert_P$ | $t(8) = 5.21$ | $p$ = 8.099E-4 | Significant |
| 5 | $npert_E < npert_P$ | $t(8) = 9.20$ | $p$ = 1.575E-5 | Significant |
| 6 | $npert_E < npert_P$ | $t(8) = 17.22$ | $p$ = 1.318E-7 | Significant |
| 7 | $npert_E < npert_S$ | $t(8) = 15.42$ | $p$ = 3.108E-7 | Significant |
| 2 | $npert_E < npert_S$ | $t(8) = 18.00$ | $p$ = 9.392E-8 | Significant |
| 3 | $npert_E < npert_S$ | $t(8) = 18.43$ | $p$ = 7.734E-8 | Significant |
| 4 | $npert_E < npert_S$ | $t(8) = 19.57$ | $p$ = 4.826E-8 | Significant |
| 5 | $npert_E < npert_S$ | $t(8) = 9.63$ | $p$ = 1.126E-5 | Significant |
| 6 | $npert_E < npert_S$ | $t(8) = 10.46$ | $p$ = 6.047E-6 | Significant |

parameters ($npert_E < npert_P$), and smart perturbation ($npert_E < npert_S$). Examining Table 6.13, reveals that *embedded optimization produces significant decreases in number of perturbed solutions as compared to penalty parameters for every N-city network except 2 cities and every network for smart perturbation with a confidence interval of 95%.* The comparison of embedded optimization to penalty parameters for the 2-city case is significant with a 90% confidence interval.

## 6.5 Chapter Summary

In this chapter, the embedded optimization methodology outlined in Section 2.3 and utilized for solving the air and space transportation systems in Chapters 4 and 5, respectively is compared to traditional SA implementations of penalty parameter and smart perturbation. The air transportation system example in Chapter 4 is utilized as a test problem and the three optimization methods are compared by best objective function value, computation time, number of system evaluations and number

of perturbations as the number of city-pairs in the network grows. The results showed a significant improvement in the performance of embedded optimization for each performance metric.

# Chapter 7

# Conclusions and Recommendations

System-of-systems design problems require the analysis of the system design and the resulting interaction of the systems. Transportation systems are classified as system-of-systems problems as the value of the system is not in a single vehicle but in the coordinated interaction of the fleet of vehicles. Traditional research into transportation system design has focused on either the coordination of the fleet or the vehicle design assuming the other as given, however for a true systems perspective these decisions must be made *concurrently*.

Integrated transportation system design expands the system boundary to include the interactions of a fleet into the design process of the vehicle. By explicitly modeling these interactions simultaneously with the vehicle design, the coupling between the vehicle and the fleet coordination can be exploited. Utilizing this flexibility to an advantage, the design space is enlarged and more efficient transportation architectures can be developed.

## 7.1 Thesis Summary and Conclusions

The integrated transportation system design problem arises from the need to develop efficient transportation architectures. Furthermore, the advent of a new era of space exploration presents an opportunity to design new vehicles that will enable sustainable space exploration. To promote sustainability in space exploration activities it

is necessary to provide the insight and efficiency currently available for terrestrial logistics activities to space transportation design problems.

Chapter 1 provides a brief review of the current state of the art in both aerospace vehicle design and network flow optimization. Recent trends, especially in the vehicle design sector, show a movement towards understanding the implications of multiple operational requirements within the design process. Specifically research into aircraft design problems [5] and spacecraft design problems [28] displays a new paradigm for design where multiple missions are considered and the impact of a fleet on the vehicle design is analyzed. However, there still exists a gap between designing a vehicle using multiple operational scenarios as a test measure and designing the vehicles *concurrently* with the operations.

The integrated transportation system design problem *bridges the gap between the vehicle design community and the operations research community* by expanding the control volume under consideration to include the models utilized by both communities into a single problem definition. In Chapter 2, a formal decomposition of the integrated transportation system design problem was presented to pinpoint the coupling between the vehicle design and network flow optimization. The decomposition distinguishes the variables and constraints that govern only the vehicle design problem, only the network routing and allocation problem, and the interactions of the fleet and vehicle design. The operations constraints, which define the coupling parameters are explored to understand the inherent difficulty that arises by allowing both the vehicle design and operations variables to be simultaneously defined.

Chapter 2 also examines the resulting complexity that arises from expanding the system boundary to include models that are generally separated. The resulting mixed-integer non-linear design problem creates difficulties when selecting an efficient optimization methodology. However, having analyzed the constraints for the fundamental structure that is created by the integrated optimization system design problem, an optimization methodology was developed to specifically handle the integrated transportation system design problem. Chapter 2 further details the embedded optimization methodology.

Given the definition of an integrated transportation system, two domain applications are developed in Chapter 3: air transportation system models and space transportation system models. The air transportation system models are developed by combining both the work of *Crossley, et al (2004)*[5] on aircraft vehicle design and *Yang and Kornfeld (2003)*[39] on aircraft routing. The space transportation system example however, required some new developments prior to defining the integrated transportation system models.

Unlike air transportation networks, space transportation networks have not been previously defined. Therefore, Chapter 3 details the definitions of a space network, drawing on analogies from terrestrial networks, extending modeling tools, and understanding where assumptions traditionally utilized in transportation network models are no longer valid for space networks. Given the space network definition, the space transportation system models are developed.

In Chapter 4, the air transportation system design problem is implemented for two examples of an overnight package delivery network. For each example, three cases were considered: network flow optimization, vehicle design optimization and the concurrent optimization presented in this research. For each of the examples, *concurrent optimization produced a solution that reduced the total system costs by at least a 10% over the traditional approaches.* By analyzing the design space of the network from the viewpoint of distance verses demand, it can be readily seen that the concurrently optimized solution more closely fits the requirements of the transportation network. Thus, by expanding the decision space more efficient transportation architectures were developed.

Chapter 5 examines an implementation of the space transportation system design problem. An Earth-Moon example network is considered, however, for space transportation systems, time becomes a variable as well. Therefore, the resulting time expanded network is much larger than the aircraft transportation network. Additionally, employing the knowledge of current space transportation systems, the vehicle model was defined at the element-level, and two types of elements are considered for allocation within the network. Again, the concurrent optimization of the space

transportation system is compared against a traditional network flow optimization and a vehicle design optimization. The results yielded a *decrease in total system wet mass of at least 19% as compared to the traditional optimization approaches.* The improvement in system wet mass can be attributed to the high percent utilization of the propulsive capability in the transportation architecture.

For each of the integrated transportation system design problems implemented, an optimization approach defined as embedded optimization was utilized to obtain solutions for the concurrently optimized designs. Chapter 6 provides a quantitative analysis of the benefits of embedded optimization over traditional implementations of Simulated Annealing, the system-level optimizer. The integrated air transportation system design problem was used as a test example, and four computational performance metrics were measured for increasing problem size. For the performance metrics of best objective function, computation time, and number of system evaluations, *embedded optimization provided a statistically significant improvement over traditional SA implementations for every problem size considered with a confidence interval of 95%* as determined by a t-test. Furthermore, for the final performance metric of number of perturbed solutions, *embedded optimization provided a statistically significant improvement with a confidence interval of 95%* for every network size except the smallest one, and in all cases this improvement was significant with a confidence interval of 90%.

## 7.2 Thesis Contributions

The goal of this thesis was to develop and validate a comprehensive methodology for defining, modeling, and solving integrated transportation system design problems, where both the network flow and vehicle design are considered together. The following specific thesis contributions can be identified.

- Further defined the *integrated transportation system design problem*
  - Decomposed the integrated problem into fundamental components and

analyzed the structure and coupling of the model

- Formulated a concrete definition of a *space network*
  - Systematically analyzed each component of a general transportation network and developed corresponding definitions for a space network
  - Extend the modeling tool of time expanded networks to incorporate astrodynamic relationships
  - Developed and validated a network generation tool that is currently utilized in SpaceNet[50],[9]
- Demonstrated the integrated transportation system design method by modeling and optimizing *air and space transportation networks*
  - Developed and validated toolbox suites for both integrated air transportation and space transportation system design
  - Quantified improvement of integrated air transportation system design over traditional design approaches ( a minimum of 10% cost reduction)
  - Quantified improvement of integrated space transportation system design over traditional design approaches (a minimum of 19% wet mass reduction)
  - Demonstrated future potential for modeling other transportation systems by analyzing two types of transportation systems
- Developed and validated the *embedded optimization* methodology for transportation system design problem
  - Demonstrated a statistically significant improvement ( with a 95% confidence interval) of embedded optimization over traditional Simulated Annealing using either penalty parameter or smart perturbation approaches

## 7.3 Limitations

The limitations of the integrated transportation system design problem primarily arise from two areas: model fidelity and computational requirements. In both the air and space transportation networks, the vehicle design models are rather limited in scope. The focus of this research was to demonstrate the gains in efficiency from expanding the system boundary in the design process; however to utilize this methodology for designing transportation systems, a greater level of detail in the vehicle design models is required to capture the complexity of vehicle design requirements. However, the formulation of both the model and software allows for a more detailed vehicle design to be included without significant changes.

For air transportation system design problems computation time is low, both in terms of real time and in comparison to the other optimization methodologies, as shown by the computational experiment in Chapter 6. However, the computational requirements for space transportation system design are much greater. Due to the size of the resulting time expanded network, and the removal of path length assumptions, the number of variables required to define the integrated space transportation system problem are very large, even for small instances. The result is a problem that is two orders of magnitude larger than the aircraft design problem, and thus much more difficult to solve and may present insurmountable difficulties when applied to real-world design problems, in the current state.

## 7.4 Recommendations for Future Work

The integrated transportation system design problem explored in this thesis only begins to investigate the complex interactions inherent in transportation systems. There exists a great deal of potential efficiency gains during the design process by employing the integrated optimization methodology in both model formulation and solution approaches. Some specific recommendations for future work include

- Expand the fidelity of the vehicle models for both air and space to provide a

greater level of detail and increased accuracy of the resulting designs

- Examine the effects of multiple vehicle type designs to understand the trade-off between specialization and cost. In the extreme, a custom vehicle could be developed for each route. Thus, an important application of this research would be a method for formally defining the optimal number of design types in a fleet; however for this analysis to be valid, the scope of the cost models would need to expand to capture costs associated with training and maintenance of multiple vehicle types.

- Examine the definition of a new vehicle within the context of an existing fleet

- Extend the integrated transportation system formulation to account for and optimize the transportation network within a stochastic demand framework. By modifying the objective function to define a low cost transportation architecture that is robust to changes in network demand would result in a stochastic decision tool for transportation system design

- Include non-chemical propulsion technologies into the design decisions for space transportation and analyze the effect these differing trajectories have on the definition of the time expanded network

- Decrease the computational requirements for the space transportation system optimization by employing formal selection strategies of the path variables, such as column generation, on each iteration. Conversely, investigations into the utility of parallel computing may alleviate some of the computational issues

- Expand the system boundary by including facility costs and other effects that influence the transportation architecture

- Understand the effect of increasing the problem size in other ways on the performance of embedded optimization. The computational study of embedded optimization presented in Chapter 6 examines the performance against increasing number of cities in the network; however including more vehicle designs

would increase the problem size as well, and possibly result in different algorithm performance

- Apply embedded optimization approach to other heuristic algorithms, such as genetic algorithms and particle swarm optimization, that can solve mixed-integer non-linear programming problems. Determine if a type of heuristic optimization algorithm is better suited for embedded optimization or if embedded optimization provides a universal improvement in solution effectiveness

## 7.5 Implications of Thesis Research

The research developed in this thesis can potentially impact many areas of the aerospace design community. The primary potential customers would be the aircraft design community. By expanding both the scope and fidelity of the aircraft design models, and incorporating the transportation networks for potential customers, the design decisions affecting both a new design type and potential variants could be analyzed in a holistic framework to produce competitive and efficient aircraft designs that better meet customer needs.

Similarly, this research is applicable to the space transportation design community as we embark on a new era of space exploration. As space transportation is an expensive industry, efficient research utilization is at a premium. By defining the logistics transport for space exploration within the context of a space network, a formal interplanetary logistics analysis can be performed, which can significantly impact the efficiency of the transportation architecture. Furthermore, by concurrently defining the design decisions within the transportation network framework, as shown by example in this research, even greater gains in efficiency can be obtained.

The embedded optimization methodology developed within this thesis can potentially impact the systems design community, as well. Although further investigations are required to define the extensibility of the method to larger problem sizes and other application domains, great promise is shown by the current performance of embedded optimization. As the system design community often faces optimization

problems with mathematical structures that are not effectively solved by traditional optimization approaches, expanding the scope and efficiency of embedded optimization could serve to provide an optimization tool that allows system design problems to be effectively optimized.

# Bibliography


[1] http://www.airliners.net/. Accessed 2/9/2007.

[2] http://www.jetblue.com/wherewefly/. Accessed 2/9/2007.

[3] Wilfried Hofstetter. Extensible modular landing systems for human moon and mars exploration. Master's thesis, Institute of Astronautics, TU Munich, 2004.

[4] Mark Maier. Architecting principles for system-of-systems. *Systems Engineering*, 1(4), 1998.

[5] William Crossley, Muharrem Mane, and Nusawardhana. Variable resource allocation using multidisciplinary optimization: Initial investigations for system of systems. In *10th AIAA-ISSMO Multidisciplinary Analysis and Optimization Conference*, number AIAA 2004-4605 in AIAA, 2004.

[6] Daniel P. Raymer. *Aircraft Design: A Conceptual Approach, 3rd edition.* AIAA Educational Series, 1999.

[7] Cynita Barnhart, Natashia Boland, Lloyd Clarke, Ellis Johnson, George Nemhauser, and Rajesh Shenoi. Flight string models for aircraft fleeting and routing. *Transportation Science*, 32(3), 1998.

[8] President George W. Bush. A renewed spirit of discovery: A president's vision for U.S. space exploration. Speech given on January 14, 2004.

[9] Christine Taylor, Miao Song, Diego Klabjan, Olivier deWeck, and David Simchi-Levi. A mathematical model for interplanetary logistics. *Logistics Spectrum*, 2007. (in press).

[10] John Anderson. *Aircraft Performance and Design.* McGraw-Hill, 1999.

[11] David Aronstein and Kurt Schueler. Two supersonic business aircraft conceptual designs, with and without sonic boom constraints. *Journal of Aircraft*, 42(3), 2005.

[12] Daniel Neufeld and Joon Chung. Unmanned aerial vehicle conceptual design using a genetic algorithm and data mining. In *Infotech@Aerospace*, number AIAA 2005-7051 in AIAA, 2005.

[13] Joshua Frommer and William Crossley. Building surrogate models for capability-based evaluation: Comparing morphing and fixed geometry aircraft in a fleet context. In *6th AIAA Aviation Technology, Integration and Operations Conference*, number AIAA 2006-7700 in AIAA, 2006.

[14] William Crossley and Muharrem Mane. System of systems inspired aircraft sizing applied to commercial aircraft/airline problems. In *5th Aviation, Technology and Operations Conference (ATIO)*, number AIAA 2005-7426 in AIAA, 2005.

[15] Muharrem Mane, William Crossley, and Nusawarhana. System of systems inspired aircraft sizing and airline resource allocation via decomposition. *Journal of Aircraft*, 2007. (in press).

[16] Muharrem Mane. System of systems inspired aircraft sizing and airline resource allocation via decomposition. Master's thesis, Purdue University, 2005.

[17] Peter Fortescue and John Stark. *Spacecraft Systems Engineering.* John Wiley and Sons, Inc., 2 edition, 1995.

[18] James R. Wertz and Wiley J. Larson. *Space Mission Analysis and Design.* Microcosm Press and Kluwer Academic Publishers, 3 edition, 1995.

[19] Richard H. Battin. *An Introduction to the Mathematics and Methods of Astrodynamics, Revised Edition.* AIAA Education Series, 1999.

[20] Vladimir A. Chobotov, editor. *Orbital Mechanics.* AIAA Education Series, 1991.

[21] John T. Betts. Survey of numerical methods for trajectory optimization. *Jornal of Guidance, Control and Dynamics*, 21(2), March-April 1998.

[22] John T. Betts. A direct approach to solving optimal control problems. *Computing in Science and Engineering*, 1(3), May-June 1999.

[23] Gerald Walberg. How shall we go to mars? a review of mission scenarios. *Journal of Spacecraft and Rockets*, 30(2), March-April 1993.

[24] Lawrence Rowell, Robert Braun, John Olds, and Resit Unal. Multidisciplinary conceptual design optimization of space transportation systems. *Journal of Aircraft*, 35(1), 1999.

[25] R. Joseph Cassady. In-space propulsion systems and technology. In *35th AIAA/ASME/SAE/ASEE Joinit Propulsion Conference and Exhibit*, number AIAA 99-2610 in AIAA, 1999.

[26] H.F. Meissinger and J. Collins. Mission design and system requirements for a multiple-function orbital transfer vehicle. In *AIAA Space Technology Conference*, number AIAA 99-42028, 1999.

[27] Javier P. Gonzalez-Zugasti, Kevin N. Otto, , and John D. Baker. A method for architecting product platforms with an application to interplanetary mission design. *ASME DETC98/DAC-5608*, 1998.

[28] Paul Wooster, Wilfried Hofstetter, and Edward Crawley. Crew exploration vehicle destination for human lunar exploration: The lunar surface. In *Space 2005*, number AIAA 2005-6626 in AIAA, 2005.

[29] Douglas Stanley, Stephen Cook, John Connolly, and Jeffrey Hanley. Exploration systems architecture study: Overview of architecture and mission operations approach. In *SpaceOps 2006*, AIAA.

[30] E. L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, and D. B. Shmoys. *The Traveling Salesman Problem.* John Wiley and Sons, 1985.

[31] Kyle T. Alfriend and Deok-Jin Lee. Optimal servicing of geosynchronous satellites. *AIAA*, (AIAA 2002-4905), 2002.

[32] Yoshikane Takahashi. A mathematical framework for solving dynamic optimization problems with adaptive networks. *IEEE Transaactions on Systems, Man and Cybernetics-Part C: Applications and Reviews*, 28(3), 98.

[33] Ravindra Ahuja, Thomas Magnanti, and James Orlin. *Network Flows: Theory, Algorithms and Applications.* Prentice Hall, 1993.

[34] Paolo Toth and Danielle Vigo. *Vehicle Routing Problem.* Society for Industrial and Applied Mathematics, 2002.

[35] David Simchi-Levi, Julien Bramel, and Xin Chen. *The Logic of Logistics: Theory, Algorithms, and Applications for Logistics and Supply Chain Management.* Springer, 2005.

[36] L. Chan, A. Muriel, and D. Simchi-Levi. Uncapacitated production/distribution planning problems with piece-wise linear concave costs. October 2002.

[37] G.B. Dantzig and J.H. Ramser. The truck dispatching problem. *Management Science*, 6(1), 1959.

[38] K.C. Tan, Y.H. Chew, and L.H. Lee. A hybrid multi-objective evolutionary algorithm for solving truck and trailer vehicle routing problems. *European Journal of Operations Research*, 172(3), 2006.

[39] Lee Yang and Richard Kornfeld. Examiniation of the hub-and-spoke network: A case example using overnight package delivery. In *41st Aerospace Sciences Meeting and Exhibit*, number AIAA 2003-1334 in AIAA, 2003.

[40] Dov Dori. *Object-Process Methodology.* McGraw-Hill, 2002.

[41] John Jarvis and Donald Ratliff. Some equivvalent objectives for dynamic flow problems. *Management Sciences*, 28(1):106–109, 1982.

[42] Dimitri Bertsekas. *Nonlinear Programming.* Athena Scientific, 1999.

[43] Dimitris Bertsimas and Robert Weismantel. *Optimization over Integers.* Dynamic Ideas, 2004.

[44] R.D. Braun and I.M.Kroo. Development and application of the collaborative optimization architecture in a multidisciplinary design environment. In M.Y. Hussaini Natalia Alexandrov, editor, *Multidisciplinary Design: State of the Art*, pages 98–116. SIAM, 1997.

[45] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220, 4598:671–680, 1983.

[46] S. Boyd. Personal communication.

[47] Diego Klabjan. *Column Generation*, chapter 6. Springer, 2005.

[48] Dimitris Bertsimas and John Tsitsiklis. *Introduction to Linear Optimization.* Athena Scientific, 1997.

[49] Christine Taylor, Miao Song, Diego Klabjan, Olivier deWeck, and David Simchi-Levi. Modeling interplanetary logistics: A mathematical model for mission planning. In *SpaceOps 2006*, number AIAA 2006-5735 in AIAA, 2006.

[50] A modeling framework for interplanetary supply chains. In *Space 2006.*

[51] O.L. de Weck and Simchi-Levi D. Technical report.

[52] S. Shull, E. Gralla, X. Li, and O. de Weck. The future of asset management for human space exploration: Supply classification and an interplanetary supply chain management database. In *AIAA Space 2006*, number AIAA 2006-7232 in AIAA, 2006.

[53] George P. Sutton and Oscar Biblarz. *Rocket Propulsion Elements.* John Wiley and Sons, Inc., 7 edition, 2001.

[54] http://tomopt.com/. Accessed (2/9/2007.

[55] Loral Space Systems. *Aquarius: Economic Value of a Consumables Launcher*, June 2002. Final Briefing on Competitive Space Grant C00-0200.

[56] http://www.mathworks.com/. Accessed 2/9/2007.

[57] Murray Spiegel, John Schiller, and R. Srinivasan, editors. *Schaum's Outlines: Probability and Statistics.* McGraw-Hill, 2000.

[58] Olivier de Weck and Il Yong Kim. Adaptive weighted sum method for bi-objective optimization. *AIAA*, 2004.

[59] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller. Equation of state calculations by fast computer machines. *Journal of Chemical Physics*, 21:1087–1092, June 1953.

[60] http://spacelogistics.mit.edu/.

# Appendix A

# System Optimization Using Simulated Annealing (SA)

The information contained in this appendix is taken from *de Weck (2004)*[58] and references the original work by *Kirkpatrick (1983)*[45]. As Simulated Annealing is the primary optimization algorithm utilized in this research, a more detailed description of the optimization methodology and algorithm is presented in this appendix.

## A.1 Background of Simulated Annealing: Statistical Mechanics

The origins of Simulated Annealing (SA) lie in the field of statistical mechanics. Statistical mechanics is the central discipline of condensed matter physics, a body of methods developed to analyze the behavior and aggregate properties of a large number of atoms, typically found in samples of liquid or solid matter. Oftentimes a cubic centimeter of liquid under Normal Temperature and Pressure (NTP) will contain on the order of $10^{23}$ atoms.

Statistical mechanics is particularly interested in the behavior of agglomerations of atoms as they are cooled from their high temperature state (from above melting temperature). There are many configurations that these atoms could take and as the

number of states (configurations) is often too large to count we resort to statistics and the concept of *ensemble* to describe the behavior of a system. An ensemble is a large set of configurations, separated in time or space, that, taken as a whole, can describe the system via its statistical properties. Thus, in statistical mechanics (atomic) systems are not described by a single configuration, but rather by their ensemble statistics.

Each configuration can be described by a set of atomic positions, $\{r_i\}$, where $i = 1, 2, \ldots, N$ is the atomic index and $E(\{r_i\})$ is the energy of the index. The probability that a particular configuration, $\{r_i\}$, will occur in a given sample of atoms is expressed by the Boltzmann probability factor,

$$P(\{r_i\}) = \exp\left(-\frac{E(\{r_i\})}{k_B \cdot T}\right) \tag{A.1}$$

where $k_B$ is generally set to one in Simulated Annealing. Examining Equation A.1 reveals that at the same temperature, $T$, lower energy configurations are more likely to occur than higher energy configurations.

In practice, only the most probable behavior of a system in thermal equilibrium at a given temperature, $T$, is observed in experiments. The equilibrium state at a certain temperature is characterized by an average behavior of the system as well as small fluctuations around that average state. Statistically, the macroscopic ensemble behavior is then obtained as the summation of each theoretically feasible configuration, $\{r_i\}$, weighted by its probability of occurrence, as determine by Equation A.1. This results in a so-called Boltzmann distribution.

The central question then becomes, how each of the possible configurations of the system is represented in the ensemble. At very large temperatures (above "melting") the probabilities of occurrence all tend to one, as can be ascertained by substituting $T = \infty$ in Equation A.1. At lower temperatures, however, the low energy states will be more likely to occur than the high energy states.

A fundamental question in statistical mechanics pertains to the behavior of the system in the limit of low temperature. Ground states and configurations close to them are extremely rare in all the configurations of a macroscopic body, yet they

dominate its properties at low temperatures because, as $T$ is lowered, the Boltzmann distribution collapses into the lowest energy state(s).

## A.2 Finding Low Energy States by Simulated Annealing

The ultimate goal of Simulated Annealing is to find the ground state(s), i.e. the minimum energy configuration(s), with a relatively small amount of computation. In the previous section we defined minimum energy states as those with a high likelihood of existence at low temperatures. Therefore, the system temperature is set to a low value in order to find the lowest energy configuration in the ensemble. However, for real physical systems and large scale design problems, the number of configurations, $N_R$, can be extremely large and therefore performing a full factorial experiment is prohibitive for finding "optimal" configurations.

Drawing on another analogy with statistical mechanics reveals that when cooling liquids, defect-free crystals are formed when the liquid is cooled very slowly from melting temperature, with a lot of time spent at temperatures near freezing in order to allow the atoms to find the lowest energy state at a given temperature. This process of slow cooling is called *annealing*. If this is not done properly, the substance is allowed to get out-of-equilibrium and the resulting crystal will have many defects. Cooling very quickly from melting temperature is called *quenching*. It has been shown that *quenching* corresponds to steepest gradient search in optimization. This is an effective procedure for purely convex problems. For non-convex and combinatorial situations, however, *quenching* is prone to getting stuck in local minima. Therefore it is desirable to define an algorithm that allows the system to be cooled slowly enough from n initial high-energy configuration to a low energy configuration such that the system can reach equilibrium at each progressively cooler temperature.

### A.2.1 Metropolis Algorithm

*Metropolis et al (1953)*[59] provided an algorithm for simulating a collection of atoms in equilibrium at a given temperature. In each step of the algorithm, an atom is given a small random displacement, $\Delta r_i$, and the resulting change in the energy of the system, $\Delta E$, is computed. If $\Delta E \leq 0$, the displacement is accepted, and the perturbed configuration is used as the starting point of the next step. The case $\Delta E > 0$ is treated probabilistically: The probability that the perturbed, higher energy configuration is accepted is $P(\Delta E) = \exp(-\Delta E/k_B T)$. This probability decreases with increasing (positive) energy increments and decreasing temperature. Random numbers, uniformly distributed in the interval [0,1] are used to implement this random choice. At each step, one such number is selected and compared with $P(\Delta E)$. If it less than $P(\Delta E)$, the new perturbed configuration is retained; if not, the original configuration is used to start the next step.

By repeating this basic step many times, one simulates the thermal motion of atoms in thermal contact with a heat bath at temperature, $T$. This choice of $P(\Delta E)$ causes the system to evolve according to a Boltzmann distribution.

## A.3 Simulated Annealing Algorithm

It is clearer now how to extend the Metropolis algorithm to *simulate annealing* by searching for the equilibrium conditions at successively colder temperatures. This process of successive cooling is repeated until the system appears to be *frozen*. A flow diagram of the basic *Simulated Annealing* algorithm is shown in Figure A-1.

The algorithm begins with an initial configuration, $R_o$ and initial temperature $T_o$. This configuration can be random or an initial best guess. The energy of the initial configuration, $E(R_o)$ is evaluated. Next, a perturbed configuration, $R_{i+1}$ is created by (slightly) modifying the current configuration, $R_i$. Next, the energy, $E(R_{i+1})$ and energy difference $\Delta E = E(R_{i+1}) - E(R_i)$ are computed. If $\Delta E < 0$, then the new perturbed configuration is "better" than the current configuration and it is automatically accepted as the new configuration. Otherwise, if, $\Delta E > 0$, a uniformly distributed

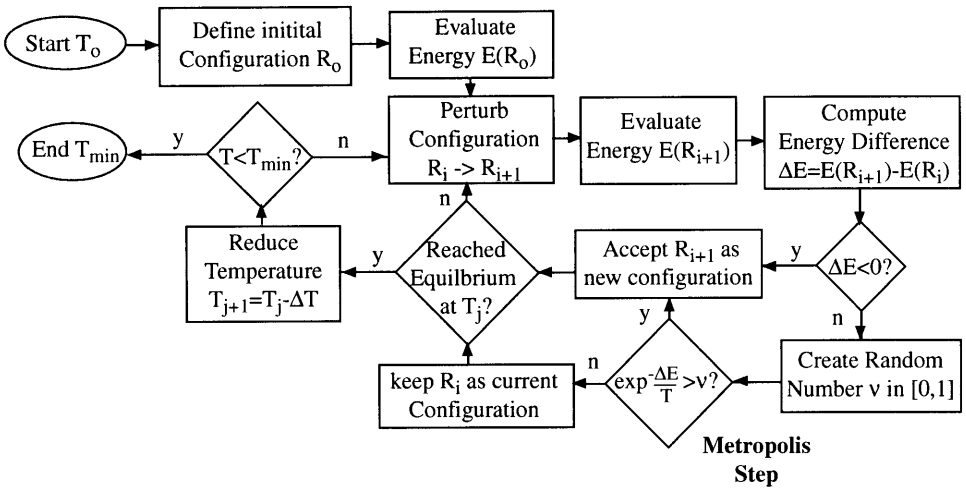


Figure A-1: *Simulated Annealing* Flow Diagram

random number $\nu \in [0, 1]$ is generated and it is compared with the Boltzmann probability $P(\Delta E) = \exp[-\Delta(E)/T]$. If $\nu$ is smaller than $P(\Delta E)$ the perturbed solution is accepted even though it is "worse", otherwise the unperturbed configuration, $R_i$, remains as the current configuration.

Following this analysis, the system is checked to determine if thermal equilibrium has been reached at temperature $T_j$. If thermal equilibrium has not been reached, the perturbation of the design vector continues at the current temperature. If thermal equilibrium has been reached the system temperature is reduced by some increment $\Delta T$ and the process of creating and evaluating configurations at the new, lower temperature $T_{j+1} = T_j - \Delta T$ is repeated. The algorithm terminates, once the system appears "frozen", which occurs when the system temperature falls below $T_{min}$ or no new configurations have been accepted in a large number of attempts. As there is no guarantee that the last configuration is the best, the lowest configuration found during *Simulated Annealing* is stored and returned upon termination of the algorithm. This Simulated Annealing process is quite generic and individual steps in this process may be implemented in various ways.

# Appendix B

# Existing Space Elements

The space transportation system models defined in Chapter 3.3 use empirically defined coefficients to determine the structural mass of the element designs. These coefficients provide the scaling criteria for the structural element mass based on both carrier capability and propulsive capability. In addition, the coefficients that scale the structural mass based on propulsive capability are unique for each fuel-type considered. Thus, the structural fraction $\alpha$ is defined in a look-up table function based on the fuel type. Finally a look-up table function is generated to define the specific impulse ($I_{sp}$) using the fuel type as an input.

This appendix provides the actual data utilized to define the empirically derived coefficients that determine the structural mass of the element designs. Table B.1 provides the list of pre-existing elements by ID number (as referenced in the thesis) with the corresponding names. In addition, the values corresponding to the element design variables and the actual dry masses are provided. The data presented represents a sub-set of the relevant information found in the SpaceNet database [60].

The look-up table function values for the structural fractions and the coefficient of the carrier mass capability are solved simultaneously using a least-squares analysis. The coefficients determined present the values that, in the least squares sense, minimize the difference between the calculated dry mass and the provided dry mass in Table B.1. The fuel ID numbers reference Table B.2 which is replicated below. This table includes the empirically derived structural fraction and the average spe-

Table B.1: List of Pre-existing Element Parameters

| Element ID | Element Name | Fuel ID | Fuel Mass (kg) | Commodity Mass (kg) | Structural Mass (kg) |
|---|---|---|---|---|---|
| 303 | S-IVB | 2 | 107725 | 0 | 12014 |
| 305 | CM | 3 | 0 | 524 | 4841 |
| 306 | SM | 3 | 18413 | 60 | 6053 |
| 307 | LM DS | 3 | 8804 | 500 | 2770 |
| 308 | LM AS | 3 | 2358 | 250 | 1719 |
| 309 | Lunar CEV CM | 6 | 363 | 500 | 8034 |
| 310 | Lunar CEV SM | 5 | 7222 | 0 | 3027 |
| 312 | LSAM DS | 2 | 28932 | 2200 | 6182 |
| 313 | LSAM AS | 4 | 5257 | 100 | 4964 |
| 314 | EDS | 2 | 226693 | 0 | 22500 |
| 321 | LSAM Cargo Carrier | 0 | 0 | 15000 | 1000 |
| 330 | Soyuz TM | 3 | 900 | 255 | 7250 |
| 331 | Soyuz TMA | 3 | 900 | 355 | 7220 |
| 332 | Progress M | 3 | 900 | 2350 | 7450 |
| 333 | Progress M1 | 3 | 900 | 1800 | 7150 |
| 335 | Service Module | 0 | 0 | 10000 | 20000 |
| 349 | STS - Stage 2 (Orbiter) | 2 | 12412 | 18000 | 78498 |
| 352 | Soyuz - Stage 2 (Upper) | 1 | 22845 | 0 | 2355 |
| 355 | Proton - Stage 2 | 3 | 46562 | 0 | 4115 |
| 366 | ISS CEV CM (3 Crew + Cargo) | 6 | 2000 | 400 | 8008 |
| 367 | ISS CEV CM (6 Crew) | 6 | 2000 | 0 | 8079 |
| 368 | ISS CEV (Pressurized Cargo) | 0 | 0 | 3500 | 7683 |
| 370 | ISS CEV SM | 4 | 2033 | 0 | 3997 |
| 400 | ATV | 5 | 2613 | 5500 | 10470 |
| 405 | HTV:H-II Transfer Vehicle | 5 | 2000 | 6000 | 10000 |
| 411 | ISS CEV CM Prop | 5 | 2000 | 400 | 8008 |
| 412 | EDS (75 mt) | 2 | 129500 | 0 | 19986 |

Table B.2: Fuel Selection and Corresponding Look-Up Table Function Values

| Name | ID | $I_{sp}$ (sec) | $\alpha$ |
|---|---|---|---|
| LOX/kerosene | 1 | 330 | 0.045 |
| LOX/LH2 | 2 | 420 | 0.079 |
| N2O4/UDMH | 3 | 310 | 0.080 |
| LCH4/LOX | 4 | 318 | .958 |
| MMH/N2O4 | 5 | 307 | .226 |
| GOX/Ethanol | 6 | 300 | 3.9353 |

cific impulse for each type of fuel considered, which provides the data to the look-up table functions of fuel properties as a function of fuel types.

The structural fraction ($\alpha$) for the last three fuel types is extremely high. However, the element types defined to have these fuel type IDs are not existing elements but rather elements in the conceptual design process, and as such, the data in Table B.1 may not be accurate. As these fuel types do not have specific impulses that differ greatly from the other fuel types, the high structural fractions will cause the elements designed to have significantly lower propulsive capability. Therefore, these fuel types will not be considered as options for the element designs.

In order to provide a fair comparison between the pre-existing elements considered and the new elements designed, the dry mass of these elements is computed using the models defined in Section 3.3 with the empirically derived coefficients. The relevant equations for the structural mass and engine mass calculations are replicated in Equations B.1 and B.2, for convenience. Table B.3 provides the computed dry mass values.

$$s^E = 2.3931c^E + \alpha\left(f^E\right)m^E\left(1 - \frac{.2m^E}{m_{UB}}\right) \tag{B.1}$$

$$g^E = \frac{.4189\left(t^E\right)^{.7764}}{g_0} \tag{B.2}$$

Given the values of the element design values and the computed dry mass, the

maximum $\Delta V$ that can be provided by this element is computed. The maximum $\Delta V$ assumes that the element is fueled to capacity and loaded to capacity. The maximum $\Delta V$ provides a metric of capability and allows for elements to be compared based on their individual capabilities.

Table B.3: List of Pre-existing Element Parameters and Calculated Values

| Element ID | Fuel ID | Fuel Mass (kg) | Commodity Mass (kg) | Structural Mass (kg) | Maximum $\Delta V$ (m/s) |
|---|---|---|---|---|---|
| 303 | 2 | 107725 | 0 | 107725 | 9040 |
| 305 | 3 | 0 | 524 | 524 | 0 |
| 306 | 3 | 18413 | 60 | 18473 | 6199 |
| 307 | 3 | 8804 | 500 | 9304 | 4154 |
| 308 | 3 | 2358 | 250 | 2608 | 3210 |
| 309 | 6 | 363 | 500 | 863 | 318 |
| 310 | 5 | 7222 | 0 | 7222 | 4436 |
| 312 | 2 | 28932 | 2200 | 31132 | 5137 |
| 313 | 4 | 5257 | 100 | 5357 | 2015 |
| 314 | 2 | 226693 | 0 | 226693 | 9391 |
| 321 | 0 | 0 | 15000 | 15000 | 0 |
| 330 | 3 | 900 | 255 | 1155 | 1892 |
| 331 | 3 | 900 | 355 | 1255 | 1525 |
| 332 | 3 | 900 | 2350 | 3250 | 318 |
| 333 | 3 | 900 | 1800 | 2700 | 406 |
| 335 | 0 | 0 | 10000 | 10000 | 0 |
| 349 | 2 | 12412 | 18000 | 30412 | 739 |
| 352 | 1 | 22845 | 0 | 22845 | 7665 |
| 355 | 3 | 46562 | 0 | 46562 | 6644 |
| 366 | 6 | 2000 | 400 | 2400 | 566 |
| 367 | 6 | 2000 | 0 | 2000 | 652 |
| 368 | 0 | 0 | 3500 | 3500 | 0 |
| 370 | 4 | 2033 | 0 | 2033 | 2744 |
| 400 | 5 | 2613 | 5500 | 8113 | 378 |
| 405 | 5 | 2000 | 6000 | 8000 | 273 |
| 411 | 5 | 2000 | 400 | 2400 | 2085 |
| 412 | 2 | 129500 | 0 | 129500 | 9120 |

# Appendix C

# Linear Space Network Model

The space transportation network model used to solve the Network-only optimization case in Section 5.4 utilizes the space transportation network flow model, operations constraints, and objective function that were presented in Section 3.3 and the additional model augmentations discussed in Section 5.1. In every transportation system model defined in this work, the commodities and network are provided as inputs to the problem. In addition, the element-type design variables are provided as inputs to the network-only optimization problem. In order to formulate this problem a mixed-integer linear programming problem, additional variables must be defined and additional constraints must be formulated to relate these newly defined variables. The linear model is presented below.

## C.1 Variable Definition

The definition of the element allocation variables and the package variables remains the same. The element allocation decision variable is denoted by $y^e_{pq}$ where

$$y^e_{p,q} = \begin{cases} 1 & \text{if element } e \text{ travels on path } p \\ & \quad \text{and is active during sub-path } q \\ 0 & \text{otherwise,} \end{cases} \tag{C.1}$$

and where $p$ is any feasible path in the time expanded network and $q$ is a sub-path of $p$.

The flow of commodities through the network is defined as $x_p^k$, where $k$ is the commodity identification number and $p$ is the path, where

$$x_p^k = \begin{cases} u & \text{if } u \text{ units of commodity } k \text{ travels on path } p \\ 0 & \text{otherwise,} \end{cases} \tag{C.2}$$

The additional variables defined for the linearized model are extensions of previously defined variables, namely the loaded fuel mass and staging variables. For the linearized model, each of these variables must be defined for every element allocation variable $y_{p,q}^e$. Therefore the loaded fuel variable becomes

$$l_{p,q}^e = \begin{cases} v & \text{if element } e \text{ travels on path } p \\ & \quad \text{and is active during sub-path } q \text{ with } v \text{ kg of fuel} \\ 0 & \text{otherwise,} \end{cases} \tag{C.3}$$

where $v$ is the amount of fuel in kg and $0 \leq v \leq mf^E \forall e : u(e) = E$. The staging variable is defined as

$$S_{p,q}^e = \begin{cases} 1 & \text{if element } e \text{ travels on path } p \\ & \quad \text{and is active during sub-path } q \text{ and is not staged after the burn} \\ 0 & \text{otherwise,} \end{cases} \tag{C.4}$$

It is important to note that an instance $e$ that is assigned to a path $p$ but is not assigned to a burn is automatically not staged.

## C.2 Linear Network Model

The linear network model can be formulated using these variables as shown below. Following each equation, a short description of the function of the constraint is provided.

The objective of the network model is to minimize the total system wet mass at

LEO.

$$\min \mathcal{J} = \sum_{E} \sum_{e:u(e)=E} \sum_{p} \sum_{q} d^E y^e_{p,q} + l^e_{p,q}$$

subject to

$$0 \leq \sum_{p} \sum_{q} y^e_{p,q} \leq 1 \quad \forall e \tag{C.5}$$

The first set of constraints ensures that an element instance is assigned no more than once within the network.

$$0 \leq \sum_{e} \sum_{p:a\in p} \sum_{q:a\in q} y^e_{p,q} \leq 1 \quad \forall a \in \mathcal{A}_b \tag{C.6}$$

The next set of constraints ensures that for every burn arc in the network, at most one element is assigned to perform the burn.

$$\sum_{e} \sum_{p:a\in p} \sum_{q} y^e_{p,q} \leq M \sum_{e} \sum_{p:a\in p} \sum_{q:a\in q} y^e_{p,q} \quad \forall a \in \mathcal{A}_b \tag{C.7}$$

The above set of constraints ensures that an element is assigned to perform a burn if any element is traveling that burn-arc.

The next set of constraints governs the additional variables required to linearize the network model.

$$l^e_{p,q} \leq m^E y^e_{p,q} \quad \forall e : u(e) = E, \quad E, p, q \tag{C.8}$$

The above set of constraints governs the fuel variable and ensures that an element is only fueled to its maximum fuel amount (as determined by the element type) *if* it is selected as a burn element.

$$S^e_{p,q} \leq y^E_{p,q} \quad \forall e : u(e) = E, \quad E, p, q \tag{C.9}$$

The above set of constraints governs the staging variable and ensures that staging can only occur for element instances selected to operate in the network.

The final two sets of constraints govern the capacity and capability requirements.

$$\begin{aligned} \sum_{p:a\in p} \sum_{k} u m^k x^k_p \leq & \sum_{E} \sum_{e:u(e)=E} \sum_{p:a\in p} \sum_{q:a\in r(p,q(L-1))} c^E y^e_{p,q} \\ & + \sum_{E} \sum_{e:u(e)=E} \sum_{p:a\in p} \sum_{q:a\in r(q(L),p)} c^E S^e_{p,q} \quad \forall a \end{aligned} \tag{C.10}$$

Equation C.10 represents the capacity constraints. The left hand side of the equation defines the total commodity mass on a given burn-arc. The right hand side of the equation defines the total carrier capacity on a given burn-arc. Note that $r\left(p, q\left(L-1\right)\right)$ refers to the sub-path defined by the path $p$ up to the last burn-arc of $q$ but not including it. This distinction accounts for the fact that if an element does stage after the burn, it is not available to traverse the arc it burns on, and therefore does not provide carrier capacity. Similarly, $r\left(q\left(L\right), p\right)$ denotes the sub-path defined by the path $p$ starting with the last burn-arc of $q$. If an element is not staged after its burn, it is available to provide carrier functionality on this arc.

$$\sum_{p} l^{e}_{p,q} + M\left(1 - \sum_{p} y^{e}_{p,q}\right) \geq \sum_{l=1}^{|q|} \Phi^{e}_{q,l} \times \left[\sum_{E'} \sum_{e':u(e')=E'} \sum_{p:a^l \in p} \sum_{q':a^l \in r(p,q'(L))} d^{E'} y^{e'}_{p,q'} + \sum_{E'} \sum_{e':u(e')=E'} \sum_{p:a^l \in p} \sum_{q':a^l \in r(q'(L+1),p)} d^{E'} S^{e'}_{p,q'} + \sum_{e'} \sum_{p} \sum_{q':a^l \in r(p,q(L))} l^{e'}_{p,q'} + \sum_{k} \sum_{p:a^l \in p} u m^{k} x^{k}_{p}\right]$$

$$\forall e : u\left(e\right) = E, \quad E, \text{ path } q, \tag{C.11}$$

The capability constraints require that the total amount of propulsive capability available is greater than that required. Thus, the left-hand side of the equation has two terms. The first term determines how much fuel is available and the second term ensures that the constraints are feasible if no elements are assigned to this burn sub-path. The first term on the right hand side represents the mass fraction corresponding to the $\Delta V$ of the arc, as shown in Equation C.12. The next two terms define the total element dry mass on every burn-arc in the sub-path. Again, a distinction is required on where the current arc is located in the path. Thus, $r\left(p, q'\left(L\right)\right)$ refers to the sub-path of $p$ defined by the sequence of arcs from the first arc of $p$ through the last arc in $q'$, and $r\left(q'\left(L+1\right), p\right)$ is defined as the sub-path from the arc following the last burn-arc of $q'$ until the last arc of $p$. The next term accounts for the fuel mass of all element instances on that arc that have yet to burn and therefore still

have fuel. The final term accounts for the total commodity mass on each arc in the burn sub-path $q$.

$$\Phi^e_{q,l} = \phi^e_{a^l} \prod_{l'=l+1}^{|q|} \left(1 - \phi^e_{a^{l'}}\right) \tag{C.12}$$

and

$$\phi^e_{a^l} = 1 - exp\frac{-\Delta V_{a^l}}{I_{sp}\left(f^{u(e)}\right) g_0} \tag{C.13}$$