

Modeling and Optimizing Space Networks for Improved Communication Capacity

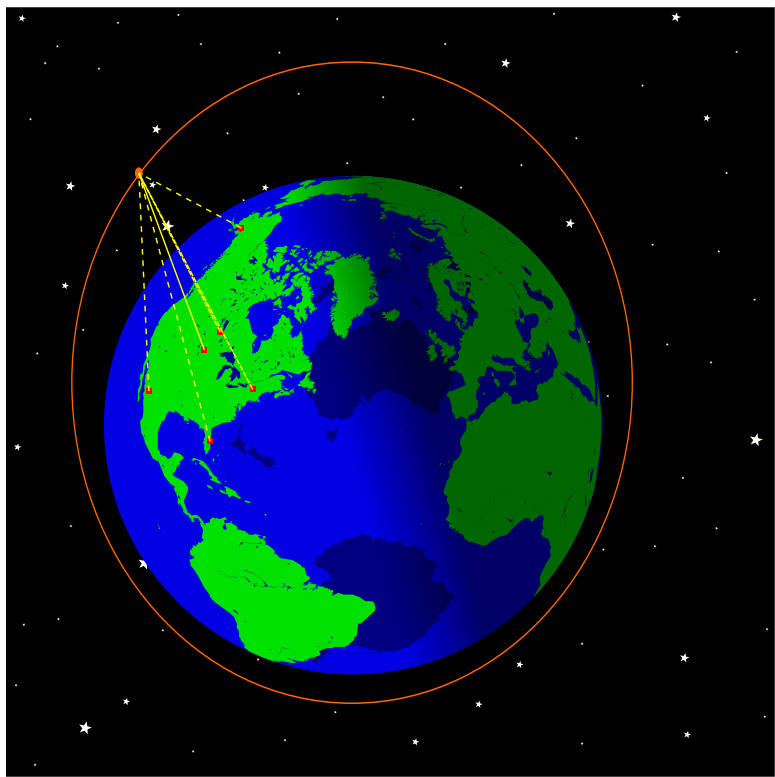
by

Sara C. Spangelo

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Aerospace Engineering)
in the University of Michigan
2013

Doctoral Committee:

Assistant Professor James W. Cutler, Chair
Associate Professor Ella M. Atkins
Professor Dennis S. Bernstein
Associate Professor Amy E. M. Cohn



©Sara C. Spangelo

2013

My family and friends.

TABLE OF CONTENTS

Dedication	ii
List of Figures	iv
List of Tables	v
List of Appendices	vi
List of Abbreviations	vii
Abstract	viii
Chapter	
1 Introduction	1
1.1 Emerging Trends in Satellite Communication	2
1.1.1 Small Satellites: Trends and Challenges	3
1.1.2 Federated Ground Station Networks	4
1.1.3 Protocol Modernization	6
1.1.4 Model-Based Systems Engineering	6
1.1.5 Summary of Emerging Trends	8
1.2 Literature Review and Areas for Extension	8
1.2.1 Space Communication Modeling and Simulation	8
1.2.2 Spacecraft Modeling and Simulation	10
1.2.3 Scheduling	12
1.2.4 Frameworks and Architectures for Space Operations	19
1.3 Thesis Contributions	23
1.4 Intellectual Innovations	25
1.4.1 Spacecraft Modeling Framework as Optimization Formulation	25
1.4.2 Coupling Operational Planning with Design	27
1.5 Thesis Outline	28
2 Model and Simulation Framework and Applications	29
2.1 Operational Modeling Framework	29
2.1.1 Model Elements	30
2.1.2 Framework Formulation	30
2.1.3 Block Diagram Representation	32
2.2 Framework Application to Communication-Focused Model	33

2.2.1	Definitions	33
2.2.2	Ground Station Network Model	36
2.2.3	Communication-Focused Spacecraft Model	41
2.3	Data Sets and Simulator	47
2.3.1	Satellite and Ground Station Data	47
2.3.2	Simulator Description	51
2.4	Application of Model and Simulator	52
2.5	Summary	53
3	Constraint-Based Capacity Assessment	58
3.1	Network Constraints	59
3.1.1	Orbit and Ground Station Coverage	59
3.1.2	Availability of Download Time	63
3.2	Energy Constraints	77
3.3	Comparison to Download Requirements	78
3.4	Summary	81
4	Deterministic Optimization: Formulation and Results	84
4.1	Problem Description and System Dynamics	85
4.1.1	Energy Dynamics	87
4.1.2	Data Dynamics	88
4.1.3	System Optimization	88
4.2	Problem Formulation	89
4.2.1	Notation	89
4.2.2	Under-Constrained Formulation (<i>UCF</i>)	91
4.3	A Special Case: Linear Dynamics	92
4.3.1	Real-World Computational Experiments	94
4.3.2	General Case Computational Experiments	98
4.3.3	Non-Integral Solutions Resulting in Branching	102
4.4	Applications to Non-Linear Dynamics	107
4.4.1	Algorithm for Solving Non-Linear <i>SMSPs</i>	110
4.4.2	Special Case: Piece-wise Linear Dynamics	111
4.5	Summary	119
5	Deterministic Optimization: Extensions and Applications	121
5.1	Generalized Under-Constrained Formulation (<i>GUCF</i>)	122
5.1.1	Notation	122
5.1.2	Formulation	124
5.2	Single Operational Satellite Problem (<i>SOSP</i>)	126
5.2.1	Problem Description	126
5.2.2	Problem Formulation	127
5.3	Diverse LEO CubeSat Missions Application	128
5.3.1	Optimal Results for Realistic CubeSat Missions	130
5.3.2	Sensitivity to Deterministic Problem Parameters	131
5.4	Interplanetary Mission Application	138

5.4.1	Mission Description and Proposed Communication Architectures	138
5.4.2	Mission Assessment	142
5.4.3	Optimization Results	145
5.5	Summary	150
6	Sensitivity to Stochasticity in Download Efficiency	155
6.1	Scheduling and Collecting Download Efficiency Data	157
6.2	Modeling Download Efficiency Data	160
6.3	Summary and Future Directions	165
7	Conclusions and Future Work	167
7.1	Conclusions	167
7.2	Future Work	169
7.2.1	Verification and Validation (V&V)	169
7.2.2	Operational Planning for Complex Spacecraft	170
7.2.3	Applications to Multi-Satellite Missions	171
7.2.4	Stochasticity in Operational Scheduling Problem	172
7.2.5	Coupled Vehicle and Operations Optimization	173
7.2.6	Applications to Interplanetary Missions	173
7.2.7	Summary	175
	Appendices	176
	Bibliography	184

LIST OF FIGURES

1.1	Small satellite launch trends demonstrating a growing number of launched and projected missions.	3
2.1	A generic representation of the subsystem function $\mathbf{Z}_{s,j} = g_{s,j}(Y_{s,j}, U, P_{s,j}, t)$ for $s = 1$ and $j = 1, 2, 3$. All values are time dependent.	32
2.2	Elements and dynamics of the system model represented with a conventional feedback control loop diagram. The non-italicized labels are the conventional elements of a control feedback loop. The italicized labels are the elements of the modeling framework.	34
2.3	Schematic with increasingly higher fidelity ground station models within smaller ellipses, where the ellipse area represents network capacity. Note this diagram is not to scale.	41
2.4	Global locations and projected visibility cones of stations in N2 and N3 assuming a satellite altitude of 500 km and an elevation mask of 0°	50
2.5	Time histories of on-board stored energy and total data downloaded for an instance of the RAX-2 mission with $p_{sol} = 5.5$ W and maximum eclipse duration of 35 % of orbit.	54
2.6	Time histories of on-board stored energy and total data downloaded for an instance of the RAX-2 mission with $p_{sol} = 3$ W and zero eclipse duration (all 97 minutes of orbital period are in sunlight).	55
2.7	Data downloaded for variable solar power collection values (p_{sol}) for an instance of the RAX-2 mission. Results are compared when there is maximum eclipse, with an eclipse fraction of 0.35 (34 minutes of a 97 minute orbital period), and zero eclipse (always in sunlight). Note data is not plotted for infeasible mission scenarios.	56
3.1	Percentage of satellite passes which have ground station coverage for different space node inclinations and ground node latitudes with a ground node minimum communication elevation of 0°	61
3.2	Percentage of satellite passes which have ground station coverage for different space node inclinations and ground node latitudes with a ground node minimum communication elevation of 10°	62
3.3	Earth coverage of latitude ranges for different numbers of ground stations (N_{gs}) and satellite inclinations assuming a circular orbit at an altitude of 500 km. See Figure 2.4 for the locations and footprints of the stations.	64

3.4	Average daily access time as a function of satellite inclination and ground station latitude for 650 km altitude circular orbits using SGP4 propagation method in STK.	66
3.5	Communication capacity as a function of diverse orbital properties and a range of ground network sizes for a one year simulation.	67
3.6	CubeSat survey of existing ground stations [1].	69
3.7	Effects of variation in AFSCN ground station latitudes from Table 3.2 on network capacity for the AeroCube 3 satellite with 99° inclination and 715 km altitude orbit.	70
3.8	2009 Minotaur-1 launched CubeSat group relative to Ann Arbor ground station ($42.27^\circ N$, $83.76^\circ W$) following epoch, the time the satellite emerges from the launch vehicle.	72
3.9	Total network capacity for 2009 Minotaur-1 launched CubeSat group relative to entire AFSCN with 15 antennas.	74
3.10	Effects of growing family of satellites on network utilization for a fixed ground station network. The CubeSats are launched into 400 to 800 km random orbits, and access time is computed relative to the 15 ground station antennas in the AFSCN. Excess access time is the time where available links are not utilized due to the capacity constraints of the ground station network.	76
3.11	Effects of growing family of satellites on excess access time for a fixed ground station network. The scenario is identical to the one shown in Figure 3.10. Excess access time is the time where available links are not utilized due to the capacity constraints of the ground station network. The solid line represents the excess time in hours and the dashed line represents the excess time in % of used time.	77
3.12	Eclipse fractions for circular orbits with variable altitudes and inclinations for a one year scenario.	79
3.13	Eclipse fraction for circular orbits with variable orbital inclinations, altitudes, and right ascension of the ascending node.	80
3.14	Communication capacity constraints compared to mission requirements and desired download for realistic satellites with diverse operational modes from Table 2.6. The simulation results are averaged for a two week period beginning.	82
4.1	Horizontal lines indicate when a ground station is in view of the spacecraft as a function of time. Vertical lines indicate the boundaries between intervals.	87
4.2	Ground stations in view of a single satellite for three intervals. The arrowed lines denotes the ground station is in view and available for communicating.	89
4.3	Single-interval instance of <i>SMSP</i> where a feasible solution to <i>UCF</i> results in infeasibilities when applied to the continuous-time dynamics.	93
4.4	Distributions of interval duration for predicted RAX orbits and three diverse networks for a one year planning horizon.	95
4.5	Distributions of number of download options for predicted RAX orbits and three diverse networks for a one year planning horizon.	96
4.6	Solve time as a function of planning horizon	97

4.7	Computational statistics for test cases in Table 4.3. The x represents the mean, and the upper and lower bounds show the maximum and minimum values. . . .	100
4.8	The effects of the number of intervals and average number of options per interval on solve time	101
4.9	Performance of worst-case instances of <i>UCF</i> under various optimality gaps. . . .	109
4.10	Example instance with PWL dynamics	112
4.11	Single interval instance of <i>SMSP</i> with piece-wise linear dynamics where a purely greedy download approach (downloading during the first segment) results in infeasibilities but an anticipative greedy approach (downloading during the third segment) is feasible.	113
4.12	Comparison of <i>NLSA</i> approaches when applied to solve PWL instances of <i>SMSP</i> (data sets are summarized in Table 4.8).	117
4.13	Effects of data characteristics on computational performance for <i>NLSA</i> with different check and split approaches. Results are shown for the data sets in Table 4.8.	118
5.1	Comparison of mission download requirements to data downloaded with optimized schedules for the satellite missions from Table 5.3 for a one day mission scenario assuming perfect download efficiencies ($\eta = 1$). Results are shown for instances of <i>SMSP</i> , which considers only the download decisions, and <i>SOSP</i> , which considers both payload and download decisions. Optimal data downloads are shown on a log scale.	132
5.2	The sensitivity of optimized schedules relative to variations in power generation and battery depth of discharge. The lines on the contour plot represent the amount of data downloaded per day in MBytes/ day.	134
5.3	Sensitivity of data downloaded from optimized schedules to variations in download data rates. Optimal results are shown for each spacecraft communicating to its TGN and the LGN for a one day scenario. Results are shown on a log-log scale.	135
5.4	Sensitivity of data downloaded from optimized schedules to variations in download data rates. Optimal results are shown for each spacecraft communicating to its TGN and the VLGN for a one day scenario. Results are shown on a log-log scale.	136
5.5	Sensitivity of data downloaded from optimized schedules to the number of ground stations in the LGN from Table 5.3.	139
5.6	Long-duration eclipse characteristics for Phobos lander. A Martian sol is 24.66 hours.	143
5.7	Short-duration access time characteristics for Phobos lander communicating to ExoMars TGO and DSN (consisting of three 34 m dishes). A Martian sol is 24.66 hours.	144
5.8	Path distances and feasible data rates for Phobos lander communicating to ExoMars TGO. Data rates are computed using the link budget in Table 5.6. . . .	146
5.9	Path distances and feasible data rates for Phobos lander communicating to DSN ground stations. Data rates are computed using the link budget in Table 5.6.	147

5.10	Properties of optimal solutions for dynamic and constant rate communication for Phobos lander to ExoMars TGO for a 48 hour planning horizon.	149
5.11	Data downloaded with different receive antenna gains for Phobos lander communicating to the DSN for a 48 hour planning horizon with optimized operational schedules.	150
5.12	Optimal downloaded data with variable collected power representative of different planets for communication between Phobos lander and the ExoMars TGO for a 48 hour planning horizon.	151
6.1	RAX-2 elevation profiles for representative high, mid, and low elevation passes over the primary RAX-2 ground station in Ann Arbor, MI.	157
6.2	RAX-2 efficiency as a function of elevation for download to representative ground stations over about a three month period. Each point represents averaged data over a single time interval that the elevation is within a 10° range. The horizontal lines show the mean for each elevation range.	161
6.3	RAX-2 efficiency data probability distribution functions (PDFs) as a function of elevation (e) for download to representative ground stations.	162
6.4	Cumulative distribution functions (CDFs) for RAX-2 download efficiency as a function of elevation (e) for download to representative ground stations over about a three month period.	163

LIST OF TABLES

1.1	Summary of <i>EOS</i> literature related to optimal deterministic scheduling of a single satellite. Problem characteristics are shown along the top row. The dots (●) indicate the characteristic is modeled in the scheduling formulation, UN indicate that it is unknown, and nothing indicates it was not considered.	15
2.1	Application of modeling framework to ground station and spacecraft models. All elements are time-dependent.	35
2.2	Parameters that are explicitly used in the formulation of the <i>communication-focused</i> model.	35
2.3	Ground station (GS) models for spacecraft communication.	42
2.4	Elements in the formulation of the <i>communication-focused</i> model.	43
2.5	Parameters that are explicitly used in the formulation of the <i>communication-focused</i> model.	43
2.6	Example missions representing the three operational modes.	48
2.7	Representative ground station networks from the Ground Station Survey . . .	49
2.8	Total data downloaded for different eclipse and power scenarios.	53
3.1	Satellite coverage for different satellite inclinations and number of ground stations (N_{gs}). These results assume the satellite spends equal time in the latitude range. The simulation is for two weeks using a J_4 orbital propagator.	68
3.2	Geographical locations of sample AFSCN ground stations	69
4.1	Parameters for sample ground station (GS) networks	95
4.2	Parameters that are constant for all test cases 0-15	98
4.3	Parameter distributions for test cases 0-15	99
4.4	Two-Option example where LP relaxation yields a fractional optimal solution	103
4.5	Feasible Solutions to the <i>MIP</i> and the optimal Linear Programming (LP) relaxation	103
4.6	Fractionality conditions for the single interval <i>UCF</i> with the conditions in Eq. 4.25.	106
4.7	Performance of worst-case instance of <i>UCF</i> under various optimality gaps. Numbers are rounded as relevant for our analysis.	108
4.8	Data sets with diverse irregularity and synchronization characteristics. Normalized values are the raw values divided by the maximum raw value of the sets. Data sets have 100 linear segments per interval.	119
5.1	Comparison of <i>SMSP</i> and <i>SOSP</i> and corresponding formulations	127

5.2	Relationship between commodities and functions in <i>OUCF</i>	127
5.3	Satellite mission parameters from Small Satellite Survey. The data collection strategies are: Type 1 is <i>targeted</i> , Type 2 is <i>repeating</i> , and Type 3 is <i>continuous</i>	129
5.4	Phobos lander mission parameters.	140
5.5	Short-duration access time statistics for Phobos lander access time for 14 days (as shown in Figure 5.7).	145
5.6	Link budget for Phobos lander communicating to the ExoMars TGO with the Electra Lite UHF transceivers [2] and to three 34 meter DSN ground stations [3,4]. with an X-Band transponder [5] and patch antenna [6].	153
5.7	Phobos lander solar radiation properties on different planets assuming surface area $A = 0.06m^2$	154

LIST OF APPENDICES

A Analytic Link Budget	176
B Derivation of Three Branching Cases	178
C Lemmas and Theorems	180
D Non-linear Data Descriptions	182

LIST OF ABBREVIATIONS

AFSCN	Air Force Satellite Control Network
AGACA	<i>Anticipative Greedy Assign and Check Algorithm</i>
AMMOS	Advanced Multi-Mission Operations System
ASPEN	Automated Planning/Scheduling Environment
CASPER	Continuous Activity Scheduling Planning Execution and Replanning
DSN	Deep Space Network
DSSA	Domain-Specific Software Architecture
ECEF	Earth Centered Earth Fixed
EOS	<i>Earth Observing Satellite</i>
FGSN	Federated Ground Station Network
GUCF	<i>Generalized Under-Constrained Formulation</i>
JPL	Jet Propulsion Laboratory
LEO	Low Earth Orbit
LGN	Large Ground Network
LP	Linear Programming
MAPGEN	Mars Exploration Rover Mission
MBSE	Model Based Systems Engineering
MDS	The Mission Data System
MIP	Mixed Integer Program
MOS 2.0	Mission Operations System
MUSE	Multi-User Scheduling Environment

MSL Mars Science Laboratory

NASA National Aeronautics and Space Administration

NEN NASA Near Earth Network

NLSA Non-Linear SMSP Algorithm

OUCF Operational Under-Constrained Formulation

RAX Radio Aurora Explorer

SAP Science Activity Planner

SGP4 Simplified General Perturbations Satellite Orbit Model 4

SMSP Single-Satellite Multiple-Ground Station Scheduling Problem

SOSP Single-Satellite Operational Scheduling Problem

SNR Signal-to-Noise Ratio

STK System Tool Kit

SysML Systems Modeling Language

TGN True Ground Network

TGO Trace Gas Orbiter

TLE Two-Line Element

UAV Unmanned Aerial Vehicle

UCF Under-Constrained Formulation

UHF Ultra-High Frequency

UML Unified Modeling Language

USN Universal Space Network

VLGN Very Large Ground Network

ABSTRACT

Modeling and Optimizing Space Networks for Improved Communication Capacity

by

Sara C. Spangelo

Chair: James W. Cutler

There are a growing number of individual and constellation small-satellite missions seeking to download large quantities of science, observation, and surveillance data. The existing ground station infrastructure to support these missions constrains the potential data throughput because the stations are low-cost, are not always available because they are independently owned and operated, and their ability to collect data is often inefficient. The constraints of the small satellite form factor (e.g. mass, size, power) coupled with the ground network limitations lead to significant operational and communication scheduling challenges. Faced with these challenges, our goal is to maximize capacity, defined as the amount of data that is successfully downloaded from space to ground communication nodes.

In this thesis, we develop models, tools, and optimization algorithms for spacecraft and ground network operations. First, we develop an analytical modeling framework and a high-fidelity simulation environment that capture the interaction of on-board satellite energy and data dynamics, ground stations, and the external space environment. Second, we perform capacity-based assessments to identify excess and deficient resources for comparison to mission-specific requirements. Third, we formulate and solve communication scheduling problems that maximize communication capacity for a satellite downloading to a network of globally and functionally heterogeneous ground stations. Numeric examples demonstrate the applicability of the models and tools to assess and optimize real-world existing and upcoming small satellite mission scenarios that communicate to global ground station networks as well as generic communication scheduling problem instances. We study properties of optimal satellite communication schedules and sensitivity of communication capacity to various deterministic and stochastic satellite vehicle and network parameters. The models, tools, and optimization techniques we develop lay the ground work for our larger goals: optimal satellite vehicle design and autonomous real-time operational scheduling of heterogeneous satellite missions and ground station networks.

CHAPTER 1

Introduction

Scientists and engineers worldwide are realizing the potential for small spacecraft to perform missions that have conventionally been accomplished with larger, highly capable, and expensive spacecraft [7]. As a result, there is an emerging trend towards using small spacecraft and constellations of small spacecraft to perform novel science, surveillance, and engineering demonstration missions [8]. Small satellite missions seek to download large amounts of data to accomplish their mission goals. However, small satellites cannot rely on over-design approaches used by conventional missions, where the design exceeds requirements to ensure mission robustness and a high probability of achieving mission requirements, such as large on-board buffers, highly-capable dedicated ground resources, and conservative download schedules (i.e. downloading only at short range distances or low data rates to ensure high signal strengths and resulting efficiencies). Small spacecraft are highly restricted by available mass, volume, and power, which limits their operational capabilities (e.g. the standardized CubeSat form factor has strict size and mass constraints [7]). Due to limited funding available for these missions, they rely on a low-cost ground infrastructure consisting of independently owned and operated stations [9]. Downloading to these types of stations is challenging as they are not centrally controlled, their availability may not be predictable, and they are often inefficient (i.e. do not collect all transmitted data). Combined, these vehicle and ground infrastructure restrictions limit the ability of small spacecraft to perform payload operations, process data, determine and control their position and attitude, prevent and recover from failures, and efficiently download payload and telemetry data to ground stations [10].

Our research is motivated by the ambitious goals of future small satellites coupled with their inherent operational challenges. In particular, we are motivated by the following operational questions:

- 1) *What is the optimal operational strategy to maximize data returns subject to on-board energy availability?*

- 2) *What is the optimal approach for globally-distributed ground resources to support satellite constellations?*
- 3) *What type of ground stations and satellite communication systems maximize data returns?*
- 4) *What is the best satellite vehicle design to overcome communication and operational challenges when there are multiple, simultaneous and conflicting mission constraints and objectives?*

To address the operational questions above in the context of operating small spacecraft, our research goal is to develop a modeling, simulation, and optimization framework that enables the analysis and design of operational schedules for spacecraft. To support current and future missions, our initial goal is to develop optimal scheduling algorithms that maximize communication capacity, defined as the amount of data successfully downloaded from satellites to ground networks over a specified planning horizon. We aim to develop a modeling and simulation approach to enable the design of future missions and networks. Our emphasis in this work is on satellite communication to Earth stations. Extensions of this work include space-to-space communication and interplanetary networks.

This introductory chapter is outlined as follows. We first motivate the work in this thesis by describing emerging trends and challenges of small satellite communication in Section 1.1. In the context of these emerging trends, we provide an overview of the literature related to this topic, drawing from the ground station and spacecraft operational communities, as well as the optimal scheduling and stochastic optimization communities in Section 1.2. Based on this review of the literature, we state the thesis contributions in Section 1.3 and describe how they address the key areas for extension identified relative to our research goals. Key intellectual insights are described in Section 1.4 and the full thesis is outlined in Section 1.5.

1.1 Emerging Trends in Satellite Communication

There are several emerging space system trends that motivate our work and are relevant to the modeling, analysis, and optimization of space network communication capacity. First, there is a growing number of small satellite missions that will place increased demand on ground networks. Second, complimenting the growth of small satellite missions is the development of loosely federated ground station networks; these are global ground stations accessible over the Internet providing satellite contact opportunities. Third, protocol

modernization is enhancing connectivity between space and ground nodes. Finally, model-based approaches to designing and engineering are emerging in the space systems domain. In the following section, we describe these trends and their relation to our work.

1.1.1 Small Satellites: Trends and Challenges

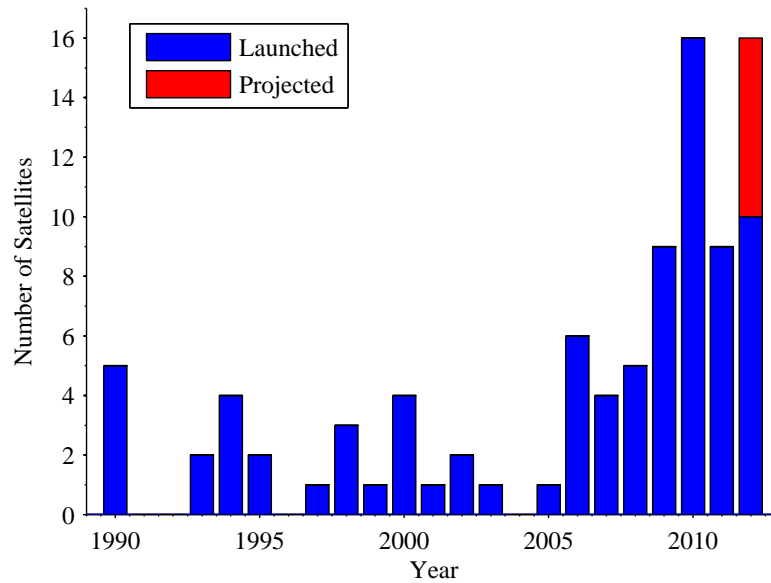


Figure 1.1: Small satellite launch trends demonstrating a growing number of launched and projected missions.

There is a growing number of innovative small satellite missions accomplishing novel science objectives and technology demonstrations, see Figure 1.1 [11, 12]. Small satellites offer several advantages relative to their larger counter-parts, including lower cost to design and manufacture, shorter development and build times, the potential to be used in educational programs, and low launch costs associated with the ability to be launched as secondary payloads [7, 12]. Small satellites are having a positive impact on the science community as satellite developers explore innovative space mission architectures and novel science missions [12]. In the past three years, the National Science Foundation (NSF) has sponsored multiple nanosatellite missions that harness recent advances in sensor and spacecraft technologies to explore space weather phenomena [13–19]. NASA is developing and launching a variety of satellites at their research centers [20–24] and is also providing launches to over thirty nanosatellite missions through the CubeSat Launch Initiative.

Beyond these single satellite missions, several globally distributed constellations have

been proposed using nanosatellites. The ARMADA mission proposes to study small scale plasma physics in the ionosphere/thermosphere system through deployment of over forty nanosatellites [25]. Each satellite will measure thermospheric and ionospheric composition, temperature, and winds. Related to ARMADA is the High-Latitude Dynamic E-Field (HiDEF) Explorer, a proposed network of 90 CubeSats that will perform measurement to study how energy is transferred from the magnetosphere into the auroral Lower-Thermosphere-Ionosphere region. Each satellite will have a deployed electric field (E-Field) sensor and the constellation will provide 10 km-scale sampling of the global electric field. Both ARMADA and HiDEF will provide significantly increased resolution in time and space of space weather related processes near Earth.

Despite the advantages of small satellites, current and future small satellite developers face communication challenges that restrict their mission capabilities. The challenges can be attributed to the resource constraints (cost, mass, volume, power) of small satellite missions that fundamentally restrict the capabilities of the communication system. Small satellites have limited power for transmission and constrained attitude control to properly point directional antennas. Thus, efforts made to improve communication network scheduling will have a positive impact on these future satellite missions beyond just increasing contact times, but also lay the ground work for optimizing downlink time, science collection, and energy efficiency.

In our work, we are motivated by the emergence of small satellites operating in Low Earth Orbit (LEO). Thus, we consider the operational challenges specific to these types of mission architectures, such as restricted on-board energy and data storage capacity, limited opportunities for download and energy collection, and realistic download efficiencies. That being said, our work is also applicable to a broader range of large spacecraft and interplanetary missions.

1.1.2 Federated Ground Station Networks

There is a growing community of untapped, global ground station resources that has the potential to benefit existing and future space missions. Many ground stations supporting current small satellite missions are built for a single mission or institution, used only when designated spacecraft pass overhead, and are idle a large majority of the time [26]. Therefore, there is often an underutilization of ground station capacity, and the stations could be used to support additional satellite contacts. The growing number of satellite users combined with the hundreds of existing and potentially thousands of future ground stations motivates the concept of loosely Federated Ground Station Networks (FGSNs).

FGSNs facilitate global communication coverage of satellites using Internet-enabled ground stations. The FGSN concept is a dynamic framework where the stations are *loosely federated* as they are independently owned and operated and may join or enter the network at will. The networks provide access to geographically and functionally diverse ground stations through Internet-accessible interfaces. These networks offer greater access to space science data at a potentially lower cost than the current infrastructure, where a single or very small existing network supports each mission [27]. Through interoperation of ground stations, networks can improve overall system efficiency and potentially support continuous satellite coverage [28]. These networks will benefit budget constrained satellite developers that may or may not have their own ground station capabilities by offering access to a global network, possibly for a financial cost or in exchange for tracking of their satellites. A FGSN provides downlinking opportunities to satellite users that would not otherwise be available, while allowing flexibility for individual institutions (i.e. they may choose if and when to track other satellites and when to reserve their station for their own testing or operations).

Ground station networks are not new and well established networks exist, such as the Air Force Satellite Control Network (AFSCN), Universal Space Network (USN), and the NASA Near Earth Network (NEN) [29, 30]. There are two groups currently exploring large scale, loose federations of distributed ground stations that support small satellite missions. Motivated to overcome the financial and engineering barriers and satisfy space operation trends, Ref. [31] introduced the concept of FGSNs and developed the Mercury Ground Station Network (MGSN), a prototype ground station control system to support advanced command and telemetry operation with spacecraft. Initially implemented with Orbiting Satellite Carrying Amateur Radio (OSCAR) class amateur radios [32], this system is comprised of university based ground stations, and satellite research groups including the Opal [33], Sapphire [34], and the university nanosatellite missions [31]. Advanced multi-mission support is enabled with virtual machine technologies combined with software defined radios [26, 31, 35]. Also, the European Space Agency (ESA) is leading an effort called the Global Educational Network for Satellite Operations (GENSO) project to develop a worldwide network of ground stations and spacecraft which interact through standard software [36].

In our work, we are motivated to consider the advantages of networked stations to enhance communication capacity. We model, analyze, and optimize mission scenarios with realistic existing and future networks. To ensure these networks are accurately captured, we collect operational, geographical, and efficiency data from existing ground station operators and planners.

1.1.3 Protocol Modernization

With the prevalence and success of terrestrial networks, satellite operators are exploring the concept of operating missions as nodes on the Internet [37, 38]. For example, the CHIPSat mission was designed to use end to end Internet Protocol (IP) techniques for command, control, and data downlinks [39]. Protocol work is enhancing IP-based packet performance over asymmetric communication links and large bandwidth-delay product systems [40–44]. Also, new standard protocols have been designed for space communications to meet a comprehensive set of file transfer requirements, such as the standards published by Consultative Committee for Space Data Systems (CCSDS) [45].

In our work, we consider the lowest layer of the Open Systems Interconnection (OSI) model, the physical layer [46]. We are concerned with capacity, the ability to move bits between ground and space. Our work is independent of high layer communication protocols, and thus applicable to diverse protocol architectures.

1.1.4 Model-Based Systems Engineering

The concept of Model Based Systems Engineering (MBSE) is introduced in this section as it is an underlying theme throughout the modeling, simulating, and optimization in this thesis. First we define and explain systems engineering and then describe MBSE in the context of our work.

Systems engineering is the “*robust approach to the design, creation, and operation of systems*” according to the NASA handbook [47]. It involves the methodologies, concepts, and structures used to engineer complex functional and/or physical systems. Systems engineering is an interdisciplinary approach that typically involves one or many of the following steps: identification of measures of performance and goals, design of alternative systems, performing design trades leading to the selection of designs, as well as verification of how well it will satisfy the goals and how well it performed once implemented [47]. The primary function of systems engineering is to develop systems model statements without disastrous over-simplifications or ambiguities [48]. Additional functions of systems engineering include resolving high-level problems into simpler problems and integrating the solutions to these simpler problems into systems solutions.

The International Council on Systems Engineering (INCOSE) defines Model-Based Systems Engineering (MBSE) as “*the formalized application of modeling to support system requirements, design, analysis, verification and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle phases*” [49]. MBSE reflects trends towards model-centric approaches for design, verification, and

optimization and a MBSE methodology is characterized as the collection of processes, methods, and tools for supporting systems engineering [50, 51]. MBSE is recognized for having two advantages relative to traditional document-centric (i.e. relying on documentation to communicate design decisions, requirements, trade studies, and designs) approaches: improved communication and knowledge-capture [52]. INCOSE projected that the state of the MBSE practice in 2020 will include additional modeling domains for complex predictions and the integration of engineering models with scientific, social, economic, and human behavioral models [49]. In particular, key elements of the future MBSE include domain-specific modeling languages, visualization, models with mathematical foundation to support high fidelity simulation, and reuse of libraries and design patterns.

Object-orientated modeling languages, like Unified Modeling Language (UML) and Systems Modeling Language (SysML), are often used to support MBSE approaches for system modeling, and provide additional advantages such as adoption of reuse concepts [52]. UML, is a technology that provides systems components with class properties, such as inheritance, dependency, association, aggregation, and cardinality [53]. UML offers a rich syntax with the ability to express relationships between modeling elements. SysML is an extension of a subset of UML which uses UML's profile mechanism.

MBSE has emerged in several projects and is continuing to gain popularity to model complex, multi-disciplinary systems. For example, the Jet Propulsion Laboratory (JPL) launched an initiative in 2004 to improve its approach to systems engineering, and identified MBSE as a key tool and technology to enable this change [54]. JPL has been involved in an INCOSE challenge to demonstrate the applicability of MBSE techniques to model space missions [55–57]. Ref. [51] described an approach for a strategic plan and roadmap for introduction of MBSE into an existing organization, which has been an ongoing effort at JPL over the past several years.

There are a few examples in the literature where MBSE has been used to model, verify, and design systems. For example, MBSE has been used for mechatronic design to assess the impact of design decisions and modification on the product's life phases [58]. In addition, MBSE is increasingly being applied to spacecraft design because of the challenges faced by complex, multi-disciplinary spacecraft projects. In the space systems domain, MBSE has been applied to model a telescope for the future European Extremely Large Telescope [59]. It has also been applied to small spacecraft, including the MOVE CubeSat [60], the design of a small artificial satellite named FireSat in the context of an educational space systems textbook [61], and the Radio Aurora Explorer (RAX) CubeSat mission (using the modeling framework developed in this thesis) [57].

In our work, we use a model-based approach to capture space operations and to enable

systems engineering in scheduling and design of future missions. Furthermore, we develop a framework that consists of templates that can be applied to analyze and optimize a variety of mission scenarios, congruent with the themes in the recent MBSE literature.

1.1.5 Summary of Emerging Trends

In summary, the work presented in this thesis is motivated by the growing number of small satellite missions with challenging communication constraints. Complimenting the growth of satellite missions is the rising number of network-enabled ground stations that can be leveraged to increase communication. Recent protocol advancements are applicable with our concepts of ground stations as we focus on the physical layer, the lowest layer present in all protocols. MBSE techniques are emerging to improve systems engineering in the space systems domain.

Although we are motivated by our representative examples highlight small satellite missions and FGSNs, our work can be applied to any type of network consisting of ground and space nodes. The models and tools we develop are applicable to space missions and ground networks of any scale and quantity.

1.2 Literature Review and Areas for Extension

This section summarizes the literature related to modeling, simulating, and optimizing space communication networks. The literature review draws from several disciplines, including literature on satellite communication and space networks, modeling and simulating spacecraft, and scheduling optimization. At the end of each subsection, we summarize the areas for extension towards our research goals stated at the beginning of this chapter.

Section 1.2.1 introduces literature related to modeling and assessing capacity of ground station networks. Section 1.2.2 describes literature on modeling and simulating spacecraft operations. Section 1.2.3 provides an overview of the literature on deterministic and stochastic spacecraft scheduling optimization. Section 1.2.4 discusses literature on existing architectures and frameworks for planning and operating space missions. This review leads to the statement of the thesis contributions in the next section.

1.2.1 Space Communication Modeling and Simulation

This section discusses literature on modeling and simulating communication links and coverage between ground networks and constellations of satellites.

There are precise analytical models and numerical solutions for determining ground visibility (also known as the footprint) between a single satellite and the ground [62]. In related work, many authors have optimized satellite constellations with the goal to achieve continuous global coverage [63–65]. Ref. [66] studied the problem of maximizing satellite observation coverage time while minimizing orbit transfer fuel costs. This work provides an analytic approach to orbital coverage and the sensitivity of coverage to orbital parameters. In a sense, this work is inverted from our effort as it is assessing and optimizing space to ground coverage, while we are interested in ground-to-space coverage; i.e. assessing and optimizing the amount of data the ground station network can support downloading from a single or collection of satellites.

Another approach for supporting satellite communication is via space-based networks such as National Aeronautics and Space Administration (NASA)’s Tracking and Data Relay Satellite System (TDRSS) [67]. Ref. [67] analyzes the link budget with representative hardware to assess the applicability of using TDRSS for small satellite communication. This work considers only a single point design and does not study overall system coverage or capacity. Space-based networks like TDRSS provide many advantages relative to ground-based networks, such as improved coverage relative to orbiting spacecraft; however they are prohibitively expensive for the type of low-cost small spacecraft operations considered in our work. Although this work focused on space-to-space networks, the approach for modeling and simulating communication links is applicable for modeling ground-to-space networks that support small satellite missions.

There are many simulation tools available for modeling and simulating aspects of space communication systems. However, the available tools are unable to simulate the capacity of dynamic networks and constellations. Capacity may be dynamic in the sense of daily or hourly access time or coverage, where we may be interested in average trends, or maximum and minimum capacity values. In addition, network and satellite availability or efficiency may be dynamic, impacting communication capacity. Various components of the System Tool Kit (STK)[®] developed by Analytical Graphics Incorporated (AGI) are useful for extracting high fidelity satellite coverage information, including coverage time over a target or ground station, number of access periods, revisit times, and gaps in visit times. However, STK itself is unable to calculate dynamic capacity between communication nodes. The Communications System Taxonomy (CommTax) toolkit uses STK and Scalable Network Technologys (SNT) QualNet network tool to model communication among multiple disparate nodes using Internet Protocols [68]. Simple ground station capacity tools were introduced in Refs. [69, 70] using STK, where the downlinking capacity of existing networks is assessed and the effect of satellite separation on downlink capacity is studied. In

similar work, communication capacity and utilization were modeled and traded with life cycle cost for a staged deployment of communications satellites constellation in LEO is addressed in Ref. [71]. This paper demonstrates the advantages of a staged deployment relative to conventional approaches based on extrapolations of current demand, which is particularly advantageous when demand is uncertain. A framework for flexible ground station networks has been proposed that virtualizes software and hardware [26]. Although this model does not model or assess capacity, it decomposes ground station functions into basic services, which is informative for building analytical models. This literature, and its approaches to measure, simulate, and optimize the ground coverage by satellites, is informative for our work.

In summary, prior work has not developed general analytical definitions and models for communication capacity or developed approaches for on assessing the capacity of dynamic, heterogeneous ground station networks and satellite constellations. Current software tools provide elements of the calculations needed for capacity model assessment, but no integrated system exists for full network and constellation studies. To assess network capacity, the existing work must be extended to include the time varying nature of the footprint over many orbits and include the relationships necessary to perform link budget calculations used in data rate estimates. Furthermore, the existing literature discusses modeling and approaches to assess capacity for *specific* mission applications only and fails to introduce a generalized approach with applicability to diverse network and mission scenarios.

1.2.2 Spacecraft Modeling and Simulation

There are multiple research fields studying various aspects of space mission design, simulation, operations, and scheduling. As a result, there are several diverse approaches towards modeling operational satellites, i.e. those that make operational decisions and trade-offs, in the literature. We summarize the modeling approaches into two representative categories: *high-fidelity, mission-specific* and *low-fidelity, application-focused*. Although both approaches are appropriate for certain applications, neither is satisfactory for a generic, extensible, analytical model and tool framework, as necessary to accomplish our research goals.

The *high-fidelity mission-specific* approach consists of a relatively high-fidelity (i.e. high level of accuracy in representing the realistic system) model focused on a single mission architecture. For example, McFadden et al. introduce a data handling and operations model and simulator for the Fast Auroral Snapshot Explorer (FAST) satellite. In this model, daily science and real-time commands are balanced to optimize daily science data collec-

tion and downlinking while satisfying a positive energy balance constraint [72]. The FAST simulator includes attitude control, ground communication, power management, and on-board data handling using a state-machine approach. The simulator is implemented using the MathWorks Simulink/Stateflow Toolbox. Analytical models and algorithms have also been introduced to optimize power allocation over a mobile satellite channel as a function of elevation angle [73] and to optimize energy utilization to maximize rewards subject to financial constraints for television broadcast-class missions [74, 75]. *High-fidelity mission-specific* approaches, such as those described above, lack flexibility and extensibility as they focus on a unique mission architecture and objective. Without a generalized analytical architecture, these types of models may not be applicable to broad classes of mission architectures with diverse subsystem interactions, mission constraints, and/or mission goals.

The *low-fidelity application-focused* approach utilizes simplified system models for a specific design, assessment, or optimization application (e.g. assessing financial cost, maximizing data returns). This approach has two advantages over the first *high-fidelity mission-specific* approach with respect to our modeling goals. First, the models are applicable to broader classes of missions because they are generalized and not customized for a specific mission. Second, these models generally capture subsystem interactions and thus enable system-level modeling. There is great diversity in the examples in the *low-fidelity, application-focused* approaches in the literature. Early satellite models focused on the financial trade-offs in the design of small satellites, and used models that captured the high-level relationships between the subsystems using this approach. Two specific examples are the Small Satellite Cost (SSCM) and Small Satellite Design Models (SSDM), developed by the Aerospace Corporation. The models were developed to achieve design-to-cost goals for satellites built with commercial off-the-shelf components and to minimize non-recurring development costs [76]. Another example of the *low-fidelity, application-focused* modeling approach is the Communications System Taxonomy (CommTax). This toolkit enables modeling of the interoperation of multiple communication nodes [68], however it does not model the other subsystems. Much of the theoretical literature towards designing, optimizing, and managing satellite schedules in the operations research community uses simplified models and fails to include on-board data storage, communication systems, and energy management subsystems [77–83]. In addition, many models in scheduling optimization assume there are no precedence (i.e. order of operations) or logistical constraints [80, 84]. Unfortunately, the *low-fidelity application-focused* approaches often neglect key elements and interactions required for end-to-end space system modeling. Furthermore, simplified models suffer from a lack of fidelity, and often are not extensible such that increasing model fidelity is difficult or impossible, and are unable to capture complex subsystem interactions.

Although these *low-fidelity, application-focused* models can be useful for initial high-level designs, they are often impractical for realistic applications.

The literature described above has several deficiencies with respect to the modeling goals stated at the beginning of this chapter. The *high-fidelity, mission-specific* and *low-fidelity, application-focused* approaches are not extensible because there is no generalized framework for adding model elements, subsystems, and states, and the interactions between these elements. Finally, the models and tools described above do not provide insight into the underlying subsystem interactions or allow for analytical approaches to optimization for general vehicle and network scheduling problems. Thus, it's not clear how they can be applied to network and vehicle assessment and design, such as evaluating the capacity of a ground station network or optimizing the size of the on-board batteries. The satellite scheduling problem is subject to uncertainty [77, 85] and requires dynamic task assignment [86], thus an extensible and analytical model is well suited for accommodating these challenges.

In summary, existing modeling approaches often neglect key elements required for end-to-end space system modeling, such as logical or temporal constraints. These simplifying assumptions reduce model fidelity and result in the model failing to capture complex interactions between the subsystems and the external environment. Thus, their applicability to real systems is severely limited, motivating the development of a new modeling approach, particularly a framework that is flexible and can be applied to a diverse class of missions.

1.2.3 Scheduling

We review literature on satellite scheduling optimization problems in Section 1.2.3.1, and stochastic scheduling optimization problems in Section 1.2.3.2. In this section we focus on optimization formulations and algorithms, while existing frameworks and architectures for planning and executing space missions are addressed in Section 1.2.4.

1.2.3.1 Scheduling Operational Satellites

This section addresses scheduling both single and constellations of spacecraft as well as ground networks that support these missions. First, we review the well-studied problem of scheduling imaging spacecraft, due to its extensive literature and similarities to the spacecraft communication scheduling problem. Next, we discuss approaches to solving downlinking optimization problems for single and constellations of spacecraft. Finally, we summarize the limitations of the existing work in the context of our research goals.

A very common scheduling problem addressed in the literature is the *Earth Observing Satellite (EOS)* scheduling problem [87]. In *EOS*, the goal is to take the maximum number of high-priority observations with on-board spacecraft sensors during a given time period. This problem is similar to the satellite scheduling problem in that they both consist of scheduling a set of complex tasks involving the exchange of limited resources between an orbiting spacecraft and Earth-based targets. Both problems are part of the larger class of *over-subscription scheduling problems*, where there are more requests for a resource than can be satisfied. Additionally, data and energy are collected and consumed in both problems, providing restrictions on when and how the desired tasks can be performed. The *EOS* problem is complicated due to numerous important constraints, including revisit limitations, the time to take the image, limited on-board data storage, power and thermal control, coordination of multiple satellites, cloud coverage, and pairs of observations of the same target [88]. *EOS* is often generalized to a more common problem structure, such as a knapsack problem [77, 89–91], a packing problem [80], a single-machine scheduling problem [86], or a network flow problem [78]. Constraint programming is used by others [77, 91–93]. One of the most common approaches for solving *EOS* is a greedy algorithm based on spacecraft priorities [77, 80, 82, 84, 94, 95]. Multi-stepped approaches consisting of allocating, identifying conflicts, and conflict resolution, and then a schedule generator with little guarantee of solution optimality are used by some authors, such as Refs. [83, 96]. Other common techniques include dynamic programming [77, 78], heuristic approaches [84, 85, 97], and genetic algorithms [80, 85, 86, 88, 89]. Look-ahead methods are used by Ref. [80], look-behind pre-emption methods by Ref. [81], repair-based iterative schemes by Refs. [83, 96], and particle swarm optimization by Ref. [92]. Other methods used to solve *EOS* include prune and search trees [79], branch and bound procedures [98], and tabu searches with intensification and diversification [90, 99]. Ref. [88] provides a comparison of several strategies for solving *EOS*, including a genetic algorithm, hill climbing, simulated annealing, squeaky wheel optimization, and iterated sampling implemented as permutation-based methods.

Table 1.1 summarizes the *EOS* literature, classifying the literature we’ve reviewed into a type. The key components relevant in the satellite scheduling problem are identified along the top row. We classify the literature into types based on which components they consider. The dots (●) indicate the characteristic is modeled in the scheduling formulation, the **UN** indicate that it is unknown, and no symbol indicates it was not considered. Refs. [78, 81, 96] do not include a formal analytic formulation, they simply describe the problem. Ref. [96] and does not include an optimization formulation, rather it uses a repair heuristic to deal with conflicts. Most of the literature does not consider all elements that are important in

the satellite scheduling problem. In particular, much of the literature neglects modeling the dynamics of the on-board resources (e.g. energy and data), which is important to capture the connection between decisions made at different times in the planning horizon. Although Ref. [93] considers many key elements, this work simply provides an algorithm for solving the problem without any discussion of its optimality or computational results to demonstrate its applicability to realistic problems.

Real-world applications of *EOS* optimization research include the Advanced Spaceborne Thermal Emission and Reflectance Radiometer (ASTER) [94], NASA's Landsat 7 mission [81], Space Imaging's IKONOS satellite [78], and the SPOTS satellite from the French Centre National d'Études Spatiales [91]. In the literature, many of the *EOS* formulations and solution strategies are demonstrated for a single or small set of example mission scenarios only. By contrast, Ref. [89] describes three classes of mission architectures: general, commercial, and tactical satellites, and demonstrates how they are each treated and optimized. Useful graphic user interfaces (GUIs) allows the user to see and modify the schedule results and make adjustments in [78, 89, 102], for example the user can input and modify the fitness function through the GUI in Ref. [89].

Scheduling multiple spacecraft for coordinated observations has been considered by Refs. [87, 103]. In particular, this literature address the problem of scheduling observations for a fleet of *EOS*s. These papers use a constraint-based approach to model the spacecraft constraints, including available resources, instrument details and duty-cycles, communication systems, and orbital information, and temporal constraints related to set-up and ordering of data events.

The problem of scheduling spacecraft downlinks has been considered by Refs. [85, 88, 97, 103–106]. While this problem is similar to the problem of scheduling payload-related mission events, such as *EOS*, it includes additional complexities, such as the opportunities to communicate with a ground station network, link requirements, and efficiency of communication. Ref. [104] addresses the problem of optimizing the download schedule for multiple satellites communicating to a single ground station. This problem is inverted from the one addressed in this thesis because we study optimizing the schedule for a single satellites relative to many ground stations. Polynomial time algorithms are used to solve several special cases, including a greedy algorithm and an approach based on exploiting a longest-path formulation in a directed acyclic graph. This paper provides useful fundamental clash-resolving strategies, but lacks additional operational requirements and constraints reflective of real-life problems. Ref. [104] introduces models and special cases that are polynomially solvable for scheduling two LEO satellites with overlapping opportunities to communicate to a single ground station. Ref. [97] studies a hierarchy of successively more

Table 1.1: Summary of *EOS* literature related to optimal deterministic scheduling of a single satellite. Problem characteristics are shown along the top row. The dots (●) indicate the characteristic is modeled in the scheduling formulation, **UN** indicate that it is unknown, and nothing indicates it was not considered.

Type	References	Priority	Energy Capacity	Energy Dynamics	Data Capacity	Data Dynamics	Downloads
1	Refs. [77, 78, 80, 94, 100] ^a	●					
2	Ref. [90]	●			●		
3	Refs. [81, 89, 92, 101]	●	●		●		
4	Ref. [96]	●	●	UN	●	UN	
5	Ref. [91]	●		●	●	●	
6	Ref. [97]	●		●	●	●	●
7	Ref. [93]		●	●	●	●	●

^aIn Refs. [77, 94], the energy and data capacity constraints are checked for feasibility after the problem is solved.

complex spacecraft scheduling problems and proposes a tight time-indexed formulation to solve them. A Lagrangian relaxation heuristic is implemented to solve the scheduling problem and results are shown for the GALILEO constellation; however, spacecraft energy collection, storage, and consumption are not considered in the analysis.

The problem of scheduling multiple space and/or ground nodes in coordination has been considered by Refs. [82, 83, 87, 99, 103, 103–105, 107]. This work generally focuses on high-level decision including assigning ground resources to track satellites based on prioritized task requests. Ref. [105] focuses on minimizing the communication time required to meet the download constraints of a system with multiple spacecraft and ground stations. They formulate and solve a non-linear constrained optimization problem and provide results for small network examples with a simplified set of communication parameters and constraints. Ref. [103] uses a two-phased search technique that includes a stochastic generate scheduler and constraint-based planning. Ref. [82] considers a greedy approach for resource allocation for scheduling satellites in LEO. Ref. [107] discusses an intelligent architecture for operations towards an automated planning, scheduling, execution, and analysis system of operating satellites that download to a ground network. Ref. [83] describes the Multi-satellite Scheduling System (MSS) for the ISRO Telemetry Tracking and Command Network (ISTRAC) and Ref. [99] introduces a tabu search heuristic to manage multi-satellite, multi-orbit, and multi-user system.

In summary, there is a large body of literature on scheduling spacecraft operations, however we have not encountered any other research explicitly studying the satellite scheduling problems addressed in this thesis. Despite the similarities between *EOS* and the satellite scheduling problem we address, there are noteworthy differences in the problem objectives, decisions, and constraints. For example, the satellite scheduling problem includes decisions on what option to use when performing a function (i.e. what data rate, energy utilization, and efficiency for download), which governs the relationship between the energy and data consumed during operations. Furthermore, the satellite scheduling problem must take into account the coupling between decisions related to performing payload operations and downloading, particularly due to the constrained resources required to support these operations. Most of the literature for scheduling downloading for single or constellation missions neglects on-board satellite data and energy collection, storage, and consumption [85, 97, 105]. Much of the theoretical literature towards designing, optimizing, and managing satellite schedules uses simplified models and fails to include logical constraints, on-board data storage, communication systems, and energy management systems [77, 78, 80]. Many models for scheduling optimization of satellite operations, such as the *EOS* problem, assume there are no precedence or logistical constraints [80, 84]. Thus

we must model and solve this problem in a new way. However, much of the *EOS* literature and download scheduling literature is informative in developing these models and algorithms.

1.2.3.2 Stochastic Scheduling

There is limited literature that addresses stochastic optimization for satellite scheduling. Thus, we first review some of the existing literature on the general class of stochastic optimization problems, including a discussion of its origins, analysis techniques, and approaches for modeling stochasticity. Second, we highlight literature addressing approaches for solving stochastic scheduling problems, emphasizing dynamic techniques. Third, we summarize the limited research that specifically addresses stochastic satellite scheduling problems.

Stochastic optimization originated in work by Dantzig [108] and Beale [109] in the 1950s, who introduced general formulations for stochastic problems. A good explanation of the formulations appears in [110]. In addition, Kutanoglu provides a brief overview of the literature related to scheduling in uncertain environments that dates back to its origins in the early 1980s [111].

Several useful techniques appear in the literature for analyzing stochastic problems. These techniques are independent of optimization and can provide information about the distribution of solutions and can be used as a baseline for the problem difficulty prior to generating solutions [85]. The first is *Random Sampling*, where schedules are generated by randomly perturbing the task requests and evaluating the resulting permutation of the performance metric. Note that stochastic linear programs may become large and complex when uncertainties are modeled. In response to this challenge, row and column aggregation has been proposed to obtain approximated *Bounds on Optimality* and on the error of the problem solution, see Refs. [112, 113].

Modeling stochasticity is a challenging task usually accomplished using probability distribution functions or sampling-based techniques. Several diverse probability distributions are proposed and used to model stochasticity in Refs. [111, 114–117]. Ref. [114] used a uniform distribution function to model the probability of breaks in the schedule and Ref. [115] used exponentially distributed constraint parameters. Ref. [116] used an exponential distribution to model the probability of failure. Ref. [111] demonstrated the advantages of using both the first and second moment information on their stochastic chance constraint. *Sampling methods* use Monte Carlo techniques to estimate the expected value of a stochastic process (such as an inventory level), which is an unbiased and consistent estimate of the true value. For example, *Sample Average Approximations (SAA)* is a heuristic approach

where samples are drawn from the scenario-distribution estimate to solve the two-stage optimization problem, which is widely used in practice [118, 119].

Next we review some examples of execution-time dynamic scheduling for problems with stochasticity. Dynamic scheduling uses real-time information to update the schedule at execution-time and has been used to address many stochastic scheduling problems. In particular, dynamics scheduling has been used for several aerospace applications such as telescopes, Mars rovers, and real-time avionics [120]. *Just-in-case* approaches build a schedule and then use a statistical model of action duration to predict possible breaks (potential errors that may occur when the schedule is executed) and builds buffer into the schedule to account for uncertainties and allow for modifications to the original schedule (such as extra time to perform a given task). Contingency plans and flexible schedules are generated using estimates of potential breaks to the schedule. Alternative schedules are pre-calculated and assessed, then they are accessed in real-time as needed based on the execution performance (to avoid the need of expensive real-time calculations). For example, Ref. [114] developed a *Just-in-case* robust algorithm for the telescope scheduling domain and Ref. [121] used this approach for enhanced robustness in the context of aircraft planning by aiming to reduce the impact of delays and creating opportunities for recovery. *Repair* or *Heuristical “match-up”* are related strategies that develop temporary schedules as a response to disruptions that aims to return to the previous pre-planned schedule in a finite amount of time [122]. Policies are often used in dynamic scheduling, for example, Ref. [123] developed *adaptive* policies and Ref. [115] developed *dynamic optimal* policies to minimize expected values. Recently, the concept of scheduling with *Planning and Dispatching stages* has been introduced, where the planning stage uses global information and the dispatching stage uses updated information on conditions and uncertainties [111]. In this approach, a schedule is generated *a priori*, and dynamic programming scheduling is used to update the schedule in real time. Ref. [124] discusses the challenges of dynamic scheduling and how it has not been effective for large stochastic optimization problems, in particular due to the curse of dimensionality (large or infinite sizes of formulations).

Many other approaches have been discussed in the literature to solve stochastic optimization problems. *Polynomial-time algorithms* have been proposed to solve two-stage recourse problems [125]; however many of these methods are appropriate only in the polynomial-scenario models (i.e. fail in the case of an exponential number of scenarios) [118]. *Perturbation methods* are used under certain conditions when the gradient of the stochastic process can be evaluated and use this for an infinitesimal perturbation analysis to help find solutions. *Likelihood Ratio* estimates derivatives when the probability distribution functions are discontinuous or depends on decision variables. *Heuristics* have been pro-

posed by several authors to study stochastic problems, including texture-based approaches (a method which relies on gathering information from the constraint graphs such as constraints, variables, sub-graphs), including slack-based heuristics [85] and greedy constructive approaches [126]. Ref. [127] discusses, compares, and shows applications for solving multiobjective stochastic linear programs, including stochastic approaches, multiobjective approaches, and hybrid approaches.

Next we describe the limited literature that addresses stochasticity for scheduling space systems. Ref. [120] emphasizes the importance of dynamic scheduling for imaging satellites in the presence of problem parameter uncertainty such as environmental events and task changes; however, this work does not develop models or algorithms to solve stochastic scheduling problems. Ref. [116] proposes two approaches for solving the Deep Space Network (DSN) communication scheduling problem with uncertainty. The first approach uses an objective function that maximizes robustness by optimizing a combination of the mean time to failure and mean time to recovery. The second approach requires that a certain objective is achieved with a required level of confidence, i.e. that the objective value be $\geq X$ with a $Y\%$ confidence. Ref. [117] studies the problem of scheduling image order for stochastic weather conditions using a stochastic integer programming formulation. In this work, a probability distribution is used to model the stochastic events and the authors develop a heuristic algorithm to solve the dual solution for a rolling horizon problem. While logistical constraints related to scheduling a single event and single set-up are considered, other constraints related to energy and data collection and storage are neglected.

In summary, there is limited literature that addresses satellite scheduling problems with stochasticity in the objective and/or constraints. These problems are challenging as they include dynamic on-board states, storage constraints, and coupled decisions on when and how to perform payload operations and download. Thus, there is a need to extend the models and algorithms in the literature work to solve stochastic satellite scheduling problems.

1.2.4 Frameworks and Architectures for Space Operations

There are several frameworks and architectures that have been developed for planning and operating both spacecraft missions and ground systems that support these missions. We discuss these frameworks and architectures separately from the spacecraft-specific models discussed previously as they focus on systems for operational planning and execution. The specific models and algorithms implicit in these systems are not typically described in detail in the literature, thus we discuss their characteristics at a high level. Several of these systems have been used or are currently in use for operational missions, thus we discuss

their performance and applicability towards our modeling and scheduling goals.

According to the Oxford dictionary, a framework is defined as “*an essential supporting structure of a building, vehicle, or object*”, and an architecture is defined as “*the complex or carefully designed structure of something*” [128]. In the literature, there are both software frameworks, which consist of basic building blocks and templates for building scheduling and planning systems, as well as several architectures that have been applied for diverse space missions. In this document we review these frameworks and architectures, listed below, and then discuss how they are related to our work.

- The Automated Planning/Scheduling Environment (ASPEN) is a modular, reconfigurable, ground-based batch planning and scheduling software framework that was first introduced in the 1990s. ASPEN encodes knowledge of hardware, science experiments, and operational rules and procedures and is designed for a wide range of spacecraft problems [96, 102]. The system is built using the ASPEN language, which includes activities, resources, states, temporal constraints and reservations [129]. ASPEN considers constraints associated with depletable (e.g. energy) and non-depletable (e.g. power) resources. ASPEN has a constraint and temporal reasoning management system and graphic user interface (GUI) for interactive problem solving. The ASPEN framework allows diverse scheduling algorithms to be implemented, including constructive and repair-based artificial intelligence (AI) scheduling algorithms.
- The Continuous Activity Scheduling Planning Execution and Replanning (CASPER) system is an embedded flight planner that modifies its schedule in real-time using a local search approach to ensure future constraints are satisfied [130]. Although the planning approach has many advantages for spacecraft operational planning, as a consequence of the local search method, solutions do not guarantee global optimality. CASPER can work with ASPEN to plan and execute space missions, and has been implemented for operation of the EO-1 small satellite mission [96, 130]
- Multi-User Scheduling Environment (MUSE) integrates existing domain and scheduling tools with multi-objective algorithms developed to solve the DSN scheduling problems [116, 131]. The DSN supports communication of planetary and interplanetary missions for both NASA and external users. Scheduling the DSN is a challenging problem that requires consideration of many complex trade-offs in the assignment of multiple antennas to support multiple missions. MUSE enables multiple participants to engage in optimization, and has been applied to schedule the Cassini science planning process and is planned to be used to schedule the James Webb Space

Telescope. Ref. [132] introduced a request-driven approach for DSN scheduling that differs from conventional activity-orientated approaches, resulting in several advantages including automated validation and traceability of requests.

- Ensemble is a multi-mission toolkit for building activity planning and sequencing systems developed by JPL and NASA Ames and deployed on the Phoenix Mars Lander and Mars Science Laboratory (MSL) missions [133]. The related Science Activity Planner (SAP) is the science operations software tool for the Mars Exploration Rover. SAP analyzes arriving data, constructs a plan of activities for the mission, and provides useful resource graphical displays to enable the user to test out “what-if” scenarios and how constraints are impacted. SAP addresses many realistic challenges for operating interplanetary missions, such as limited resources, irregular communication opportunities, and strict temporal constraints for communication [134]. SAP uses Mixed-Initiative Planning and Scheduling for the Mars Exploration Rover Mission (MAPGEN), and activity-planning tool that planners can use to generate feasible plans [135]. SAP has a public version called Maestro that engages the public by allowing them to track the progress of Spirit and Opportunity.
- The Mission Data System (MDS) is an advanced multi-mission architecture for deep-space missions initiated in the 1990s [136]. It was developed to enable collaboration, system and software design, and lower-cost design, test, and operation. The MDS framework is a collection of common problems and solutions that can be referenced and applied to a family of applications within a domain. MDS includes explicit use of models (e.g. tables, functions, rules, and state machines that are inspectable), goal-orientated operation, real-time resource management, and fault protection. The properties of MDS are inspired by Darpa’s Domain-Specific Software Architecture (DSSA) Project, which used a generic domain-specified software architecture with reusable components library for diverse applications [137, 138].
- The Advanced Multi-Mission Operations System (AMMOS) is a framework to operate and process the data returns of dozens of deep-space robotic missions [139]. It was designed and implemented by NASA in the 1980s to significantly reduce the cost of individual missions. AMMOS was developed such that individual missions could adapt the set of modular tools and services, including network capabilities and standard hardware and software, to support their needs. The Mission Operations System (MOS 2.0) is a major update to AMMOS that is currently in development at JPL [56, 140]. MOS 2.0 uses advances in technology, system architecting, and systems engineering, including MBSE approaches. MOS 2.0 is designed as a control

system to “provide multi-mission tools and services” to achieve mission goals and enable the concept of “develop with what you fly with,” which spans the mission life-cycle. MOS 2.0 provides several advantages relative to AMMOS, including improved stake-holder input, ability to focus on system-wide principles, use of design patterns and models, and is composed of reusable elements.

In summary, ASPEN, CASPER, MUSE, Ensemble, SAP, and MAPGEN are planning and execution tools that have been developed for ongoing and future missions such as EO-1 and the Mars missions. These frameworks and architectures are powerful and enable both ground and satellite operators to plan and execute schedules a priori and in real-time. However, they are not conventional systems engineering tools (i.e. to enable analysis, design, and optimization of spacecraft missions) and do not use foundational model-based approaches to enable state-based control, and do not support the design of future spacecraft missions or ground infrastructure. MDS introduced an architectural approach based on models for state analysis and control and was designed for system design and analysis. MDS was cancelled due to many complex factors, including political and financial issues. One technical challenge that contributed to its cancellation was that MDS was not able to provide a practical tool that could be implemented and tested at incremental stages throughout its development. Despite the challenges it faced, MDS has many desirable properties, including a model-based approach and use of standardized libraries, which should be incorporated into future architectural systems for space systems design and operation. MOS 2.0, currently under development, is a systems engineering tool and shares many key themes with the modeling approach presented in this thesis, including model-centric and state-based control. MOS 2.0 currently does not have planning and scheduling capability, but has the potential to interface with such tools.

To summarize, existing frameworks and architectures have a number of drawbacks relative to our research goals (modeling, assessing, and optimizing general space systems), listed below:

- The well-developed scheduling architectures and frameworks described above are generally designed exclusively for operations and not for mission analysis and design. Note that MDS supported software and hardware design and analysis; however faced some challenges and was not fully developed (despite having several key properties that should be incorporated into future systems). In addition, MOS 2.0 supports mission design; however is still in development and does not have scheduling capabilities. Thus, it is not clear how existing approaches can be applied to combined network and vehicle assessment and design, such as evaluating the capacity of

a ground station network or optimizing the size of the on-board batteries.

- The frameworks are well-described; however general, accessible, and analytic approaches for modeling and optimizing space systems do not appear in the literature.
- Most of the approaches for solving scheduling problems with existing systems use heuristic or repair-based approaches that do not guarantee optimality, for example ASPEN uses a repair-based approach and CASPER uses a local search method [96, 130]. Planning tools like SAP and MAPGEN require a human operator to develop feasible plans, and does not provide guarantees of optimality [133, 135, 141]. The approaches in the existing literature may not be globally robust to uncertainty. Although some of the approaches for spacecraft execution have on-board resource management and failure-safe policies to avoid overusing resources, they do not necessarily ensure resources are managed in an optimal way, particularly when there is uncertainty in the problem [96].

Despite their drawbacks, the frameworks and architectures described above have key themes, elements, and features that are consistent with our modeling and simulation goals. Thus, the approaches used to develop and the lessons learned from these systems are informative in developing our models, simulators, and optimization algorithms.

1.3 Thesis Contributions

Our primary objective is to develop a fundamental approach for modeling, assessing, and optimizing general space systems. Currently, there is no general, analytical, model-based framework that enables end-to-end system design, testing, and optimization, which is the focus of this thesis. The initial focus of our work is on communication systems for small spacecraft missions. The unique contributions of this thesis are listed below.

- *Develop a general, analytical model framework that can be applied to model space systems, such as satellites and ground networks.* We define a framework as a set of reusable elements and templates for describing dynamics, constraints, and goals. The framework analytically represents the dynamic interaction of states (such as position, energy, and data) and subsystem operations (such as communication and energy management) of an operational satellite. It captures mission constraints, which are often called requirements, that specify minimum performance levels. It also enables analytical expression of objectives, which are goals of the mission to be maximized. The

framework is generic and modular such that it is capable of supporting a variety of mission architectures and scenarios.

- *Perform constraint-based assessment of the communication capacity for representative problem instances.* We develop a flexible and extensible simulation environment that implements the modeling formulation. We use the toolkit to assess the sensitivity of communication capacity relative to diverse sets of constraints for representative small satellite missions and globally distributed ground station networks. This allows us to assess the constraints that limit mission potential for a variety of mission scenarios. This also enables comparison of constraint-based capacity to mission requirements, which can identify deficient and excess capabilities.
- *Formulate and solve deterministic operational scheduling optimization problems.* The operational scheduling optimization problem is to maximize the amount of data downloaded from a single spacecraft to a network of ground stations subject to realistic constraints. We develop algorithms to solve problem instances with both linear and non-linear dynamics. We demonstrate the applicability of these formulations and algorithms to diverse real-world and generic problem instances. We investigate theoretical conditions and computational results of these problem instances. We also demonstrate how the optimization framework can be used for optimal vehicle and network design for diverse classes of missions. We develop an approach to incorporate stochastic download efficiency into the model and perform sensitivity analysis to investigate the impact on solutions.

The framework has an analytical, modular, extensible structure, thus addresses the challenges described in Sections 1.2 in three ways. **First**, the analytical framework exposes all problem parameters to the modeler/ planner. Thus, the framework can be used for developing diverse operational models. The modeling approach can be used to gain insight into the underlying subsystem interactions and relax various constraints to enable constraint-based assessment to identify the parameters restrict mission performance. This allows us to perform sensitivity analysis relative to both deterministic and stochastic problem parameters, for example network size and stochastic download efficiency [77, 85]. In addition, the model can be built incrementally and scheduling and design approaches can be tested throughout the process to manage complexity and avoid the challenges faced in the development of MDS. **Second**, for problem instances with the appropriate problem structures (e.g. linear programs), the model can yield solutions to scheduling problems which are provably feasible and optimal, as demonstrated in Ref. [142], which are both highly desirable properties. **Third**, the framework is designed not only for operational planning and

execution, but for additional applications, in particular to explore the operational and vehicle design space. For example, it enables combined vehicle and operational planning, which is demonstrated in Refs. [143, 144].

The framework we propose shares key themes, elements, and features with the existing literature on modeling and optimizing spacecraft operations from Section 1.2, and is not meant to compete or replace the approaches described above, but rather compliment them. For example, the formulations for optimizing *EOS* and satellite downlink problems were informative for developing our models. In addition, the framework we propose shares the concept of states and temporal constraints with the ASPEN and MOS 2.0 frameworks [145, 146], uses a model-based approach like MOS 2.0 [136], and is extensible and modular like ASPEN, CASPER, Ensemble, MDS, and MOS 2.0. The algorithms we propose are also related to MUSE, however this system is based on a request-driven approach (i.e. where satellites request times for events), while our approach considers higher-fidelity modeling of on-board resources.

Throughout this thesis we employ general frameworks for modeling and simulating systems, sharing key themes with much of the literature on MBSE. In fact, our approach is used to develop a SysML model of RAX, including capturing the subsystems, their interactions, and operational scenarios in Ref. [57].

1.4 Intellectual Innovations

Next we describe intellectual contributions of this thesis work and key insights that emerged in the development of the thesis.

1.4.1 Spacecraft Modeling Framework as Optimization Formulation

The first intellectual innovation in this thesis work is the development of a spacecraft modeling framework that marries fundamental approaches for modeling systems from the operations research world with problems from the aerospace engineering domain. Small spacecraft mission designers and planners typically strive to satisfy mission requirements, and usually do not optimize their designs or schedules using a complete system-level model. Conventionally, back-of-the-envelope calculations and approximations are used when making design/operations decisions (e.g. sizing batteries according to average eclipse duration). Typically only a handful of analysis are performed, simplifying assumptions are used for subsystem modeling, and key state and subsystem interactions are neglected. By contrast, in the optimization research domain, typically only well-defined problems are formulated

and solved, i.e. those where all parameters are explicitly and uniquely described. In this thesis, we merge key concepts from these two domains by designing an optimization-based modeling framework template to support modeling, simulating, assessing, and optimizing spacecraft missions using an analytical foundation. Capturing the spacecraft problem using this type of framework enables designers/operators to capture the dynamic nature of all states, constraints, and objective(s) of the complete mission scenario, which are not typically considered simultaneously. The framework can be applied to develop specific models for well-defined problem instances and solvers and algorithms from the operations research domain can be used solve these problems.

The approach for mapping spacecraft problems to an optimization formulation is described next. Spacecraft requirements in the spacecraft domain are essentially the factors that constrain the problem solution space, thus they are mapped to constraints in the optimization domain. Example requirements/constraints of the problem are the maximum battery depth of discharge and minimum data download. An optimization problem requires an objective, which in our framework represents the mission goal. Conventional spacecraft mission descriptions may not have an obvious objective because they are typically requirements-driven. However; on closer investigation of the mission, the objective can usually be identified as maximizing the science return of the mission, which can often be interpreted as maximizing the science data collected or the amount of data downloaded, depending on the mission, its network, and its constraints. Other potential mission objective examples include minimizing the (average or maximum) depth of discharge of the battery to preserve lifetime or maximizing the pointing accuracy for instrument collection/data download.

Spacecraft operational problems are dynamic systems because the states evolve continuously. Although this can be modeled analytically, it can be more difficult to model in a way that can be solved in a computationally tractable way. Thus, for the problems in this thesis, where appropriate, we discretize the problem into a finite set of intervals, a conventional approach in both the aerospace and operations research domains. We then impose constraints on the state for every interval such that commercial optimization solvers can be employed to solve them in a reasonable solve time. Theoretical implications of the formulations and algorithms can also be investigated using fundamental concepts and theorems from the operations research domain. The work done modeling complex spacecraft systems using an optimization-based approach are also applicable to problems outside this domain, as discussed in Chapter 7.

Capturing a spacecraft problem using an optimization-based modeling framework results in several advantages relative to conventional approaches for modeling and optimizing

space systems, listed below:

- The ability to verify if requirements are satisfied by investigating the feasibility of a given mission scenario when it is formulated as an optimization problem.
- The ability to design vehicles and operational schedules to not only meet, but also exceed, mission requirements. This is because using an optimization-based modeling framework enables optimizing an objective (or several objectives, in some cases) instead of simply aiming to achieve the mission requirements.
- The ability to exploit existing optimization solvers to solve design/scheduling problems.
- The ability to make theoretical insights for problems that have well-known formulations in the operations research domain, e.g. linear programs, Mixed Integer Programs (MIPs).
- The ability to perform sensitivity analysis (or trade studies) to gain insight into the key deterministic/stochastic parameters that limit mission potential (by evaluating the feasibility/performance of solutions).
- The framework's extensibility and modularity and allows us to develop models to represent a diverse set of mission scenarios.

Furthermore, a foundational modeling framework allows many of these advantages to be achieved using a common model and simulation environment (i.e. the same models, code, and simulations).

1.4.2 Coupling Operational Planning with Design

Another innovation in this thesis is the coupling of operational planning and vehicle and network design studies. Conventional approaches for vehicle and network design use back-of-the-envelope approximations and simplified trade-studies, often neglecting dynamics and realistic constraints [61]. As these approaches do not use accurate models or optimization techniques, they often yield suboptimal operational vehicle and network design solutions. An integrated approach that couples vehicle and operational decisions has distinct advantages over conventional approaches because it verifies the feasibility and optimizes the schedule for each point design. For example, designing solar panels, batteries, or operational schedules with either the best-case, average-case, or worst-case annual eclipse

conditions would each yield infeasible or suboptimal solutions at some time throughout the year. Thus, an approach that considers the dynamic nature of the system is necessary.

Using our scheduling formulation and algorithms, most realistic spacecraft scheduling problems solve on the order of seconds (as demonstrated in our computational results in Chapters 4 and 5). Thus, they have the potential to be used for additional design problem applications. In particular, this enables solving a meta problem related to the vehicle or network, while optimizing each point design using these rapid scheduling techniques.

The key innovation was identifying the ability to exploit solving the satellite scheduling problem efficiently to use it as an internal problem solver for a larger design problem. Advantages of this approach include the ability to accurately compare competing designs because the operational schedules are actually optimized for each point design.

1.5 Thesis Outline

The remainder of the thesis is outlined as follows. Chapter 2 introduces a general modeling and simulation framework and demonstrates its applicability to operational space systems. Chapter 3 implements this model and simulation environment to perform capacity-based assessments useful for identifying operational limitations for diverse network and spacecraft. Motivated to optimally allocate constrained and excess resources, Chapter 4 applies the model framework to formulate a communication-focused optimization problem and demonstrates theoretical and computational results when applied real-world and generic problem instances. Chapter 5 extends the optimization formulation from Chapter 4 to include operational decisions and demonstrates its applicability to a broader range of both Earth-orbiting and interplanetary mission scenarios and networks. Chapter 6 discusses the sensitivity of the problem to stochasticity in download efficiency. The thesis is summarized and insights for future work are provided in Chapter 7.

CHAPTER 2

Model and Simulation Framework and Applications

In this chapter we introduce a framework for modeling operational space systems. The framework is modular and extensible in two ways. First, additional elements, for example states, subsystems, communication nodes, or components of the formulation such as constraints and objectives can be added. Second, model fidelity can be increased by improving the accuracy of the dynamics, subsystems, or interactions. The framework is generic such that it can be applied to model space system, for example a satellite, ground station, rover, Unmanned Aerial Vehicle (UAV), communication network, or system-of-systems. This modeling and simulation framework provides a foundation for assessing the communication potential and optimizing satellite and ground resources for existing and upcoming missions or the design of future vehicles and networks.

The framework is developed in Section 2.1, and applied to develop a *communication-focused* model in Section 2.2, which consists of an integrated satellite and ground network. We introduce realistic data sets and a simulation environment used for executing the model in Section 2.3 that are used throughout the remainder of the thesis. The applicability of the communication-focused model and simulation environment to capture a real-world mission scenario is demonstrated in Section 2.4 and the chapter is summarized in Section 2.5.

2.1 Operational Modeling Framework

The modeling framework is composed of elements, which are the building blocks of the model, and an analytical formulation that captures state dynamics, objectives, and constraints with a generalized template. The framework models a system that is functionally comprised of subsystems that operate on states and has specific mission goals and constraints.

2.1.1 Model Elements

The four main elements of the framework are parameters, states, subsystems, and the schedule. Elements are constant or time-dependent; however in this subsection, time notation is omitted for simplicity.

Parameters – A parameter, p , is a model input that provides numerical values to dynamically model system states and subsystem functions. Let P be the set of all model parameters, where $p \in P$. Examples parameters are orbital parameters, ground station locations, and T_f .

States – A system state is a model variable, and is defined as the information at some initial time that, combined with the input (parameters and the schedule) for all future time, uniquely determines the output for all future time [147]. Let $\mathbf{X} = [x_1, \dots, x_k, \dots, x_m]^T$ be the vector of all the system state variables, where there are m variables. Example states include on-board resources such as energy and payload data. Opportunities for mission operations such as payload operation and ground station availability are also system states. An opportunity is modeled as binary, $o \in \{0, 1\}$, where a value of one indicates an opportunity and zero indicates no opportunity.

Subsystems – A subsystem, s , performs functions on states. Let S be the set of all subsystems. A single function operates on state k and is denoted $f_{s,j,k} \in F$, where $j \in J_s$ is the function index, and J_s is the set of all function indices. $f_{s,j,k}$ is an element of the set of functions, F .

Schedule – The schedule, $U(t)$, is a series of time-dependent events that describes how and when the subsystem functions operate on the states. Events are scheduled when there are opportunities. For example, a data download event may occur when there is a line of sight between a ground station and satellite. The schedule is designed to achieve the mission objectives while satisfying the mission constraints. The schedule may be an output (e.g. when a solver is used to find an optimal schedule) or an input (e.g. when simulating a given schedule to test performance).

2.1.2 Framework Formulation

The model is formulated as a conventional optimization problem in Eqs. 2.1-2.4. Mission objectives, represented in Eq. 2.1, maximize the total transfer of a mission-specific system state, x^* , a component of \mathbf{X} , over the planning horizon. The decisions in the optimization problem are when and how the events occur, which are captured in the schedule, $U(t)$,

which is an output of the optimization problem as formulated here. The constraints in the formulation include state dynamics (Eq. 2.2), bounds on state values (Eq. 2.3), and mission requirements (Eq. 2.4).

$$\max_{U(t)} \{x^*(T_f)\} \quad (2.1)$$

s.t.

$$\mathbf{X}(t + \Delta t) = \mathbf{N}(\mathbf{X}, P, t) + \sum_{s \in S} \sum_{j \in J_s} \mathbf{F}_{s,j}(\mathbf{X}, U, P_{s,j}, t) \quad 0 \leq t \leq T_f \quad (2.2)$$

$$\mathbf{X}_{min} \leq \mathbf{X}(t) \leq \mathbf{X}_{max} \quad 0 \leq t \leq T_f \quad (2.3)$$

$$\Theta_{k,i} \leq \int_{t_i}^{t_{i+1}} \dot{x}_k(t) dt \quad \forall x_k \in \mathbf{X}, i \in I_k \quad (2.4)$$

States evolve over time due to nominal dynamics and subsystem functions (see Eq. 2.2). Nominal dynamics are independent of subsystem functions. The vector of nominal dynamics equations is defined in Eq. 2.5, where each element k represents the nominal dynamics of state x_k . Orbital motion and battery self-discharge are example nominal dynamics of the state variables position and on-board energy, respectively.

$$\mathbf{N}(\mathbf{X}, P, t) = [n_1(\mathbf{X}, P, t), \dots, n_k(\mathbf{X}, P, t), \dots]^T, \quad (2.5)$$

The vector of subsystem functions that operates on the state vector is expressed in Eq. 2.6. The inputs to each function $f_{s,j,k}$ include the states, parameters, schedule, and time. Note the vector in Eq. 2.6 contains zero entries when combined subsystems and functions do not operate on specific states.

$$\mathbf{F}_{s,j}(\mathbf{X}, U, P_{s,j}, t) = [f_{s,j,1}(\mathbf{X}, U, P_{s,j}, t), \dots, f_{s,j,k}(\mathbf{X}, U, P_{s,j}, t), \dots]^T \quad \forall s \in S, j \in J_s \quad (2.6)$$

The nominal and functional dynamics in Eqs. 2.5 and 2.6 may each be described by any type of function, for example they may be analytical or extracted from a simulation system.

The state vector, \mathbf{X} , is constrained by lower and upper bounds, $\{\mathbf{X}_{min}, \mathbf{X}_{max}\} \in P$, as in Eq. 2.3. Example bounds include maximum and minimum battery capacity and maximum data storage capacity.

Operational mission requirements are represented in Eq. 2.4 by enforcing a minimum change in system state over a specific time period. For example, there may be a mission requirement that a minimum amount of state (such as energy) must be acquired or consumed during a certain period of time. Each interval $i \in I_k$, where I_k is the set of intervals

spanning the full planning horizon for state x_k , has a start time, $0 \leq t_i \leq T_f$, where the end of interval i corresponds to the start of interval $i + 1$. Eq. 2.4 enforces a minimum change of state x_k during every interval $i \in I_k$, represented as $\Theta_{k,i}$. The change in state during interval i is its integrated time rate of change from t_i to t_{i+1} . For states without requirements, $\Theta_{k,i}$ will be zero $\forall i \in I_k$.

Another perspective for describing spacecraft operations is to consider subsystem functions individually. In particular, consider the analytical relationship between inputs and outputs specific to subsystem s and function j , $\mathbf{Z}_{s,j} = g_{s,j}(\mathbf{Y}_{s,j}, U, P, t)$, where the vector of inputs is $\mathbf{Y}_{s,j}$ and the vector of outputs is $\mathbf{Z}_{s,j}$, which are both comprised of components of \mathbf{X} . The function $g_{s,j}$ is the combination of $f_{s,j,k} \forall k \in K$, i.e. it models the impact of subsystem s and function j on all state inputs and outputs. A diagram representing these relationships for a single subsystem is given in Figure 2.1.

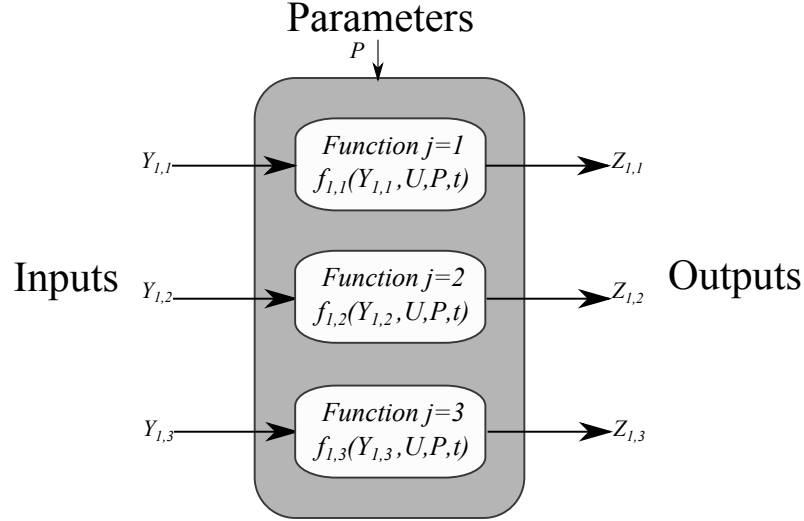


Figure 2.1: A generic representation of the subsystem function $\mathbf{Z}_{s,j} = g_{s,j}(\mathbf{Y}_{s,j}, U, P_{s,j}, t)$ for $s = 1$ and $j = 1, 2, 3$. All values are time dependent.

2.1.3 Block Diagram Representation

We represent the model framework using a conventional control system block diagram to demonstrate the interaction of the various model elements in Figure 2.2. The set of parameters, P , are provided to the input block, which identifies opportunities for subsystem functions, O , and interprets the mission requirements, \mathbf{R} , as control inputs. The error signal is expressed as $\mathbf{E} = \mathbf{R} - \mathbf{M}$, where \mathbf{M} is estimated state values, which are measured by on-board or ground sensors. \mathbf{E} , P and \mathbf{R} are provided to the scheduler, which

generates the operational schedule, U . Note that U is an output of the controller and an input to the dynamic system. The states evolve according to both the nominal dynamics and subsystem functions as prescribed by the U , where updated states (after time Δt) are denoted $\mathbf{X}(t + \Delta t)$. Unmodeled realistic disturbances, \mathbf{D} , may be injected into the system and modify the state. Mission performance is evaluated by measuring the states and verifying if the mission requirements are satisfied and comparing realized objectives to their expected values. Feedback control exists when the scheduler updates U according to mission performance, i.e. uses \mathbf{E} in for future scheduling decisions.

2.2 Framework Application to Communication-Focused Model

Motivated by the communication-centric research goals stated earlier, the modeling framework is applied to compose a *communication-focused* model. The model consists of an operational ground node and space node and enables assessment and optimization of the communication potential of a spacecraft mission.

The elements and parameters of the ground station network and spacecraft models are summarized in Tables 2.5 and 2.2, respectively. Position and attitude of the ground network and spacecraft do not appear in Table 2.2 as they are modeled in our simulation environment and not explicitly captured in the analytic formulations. The ground network and spacecraft models interact in three ways: 1) the download opportunities are a function of the combined orbital dynamics of the spacecraft and location of the ground nodes; 2) the communication link budget is a function of parameters from both systems (which determines feasible data rates and power for download); and 3) the schedule prescribes how the ground network and satellite interact operationally.

Several important definitions necessary for this *communication-focused* model are introduced in Section 2.2.1. Next, the modeling framework is applied to develop a ground station network model and spacecraft model in Sections 2.2.2 and 2.2.3, respectively.

2.2.1 Definitions

Capacity is the total amount of data exchanged across a network over a given time span and a *network* is a collection of nodes that exchange data over links. A *node* is any device entirely contained within a spatially local volume with the ability to transmit, receive, store, or catalogue data at the cost of energy. The nodes can be considered transmitters and

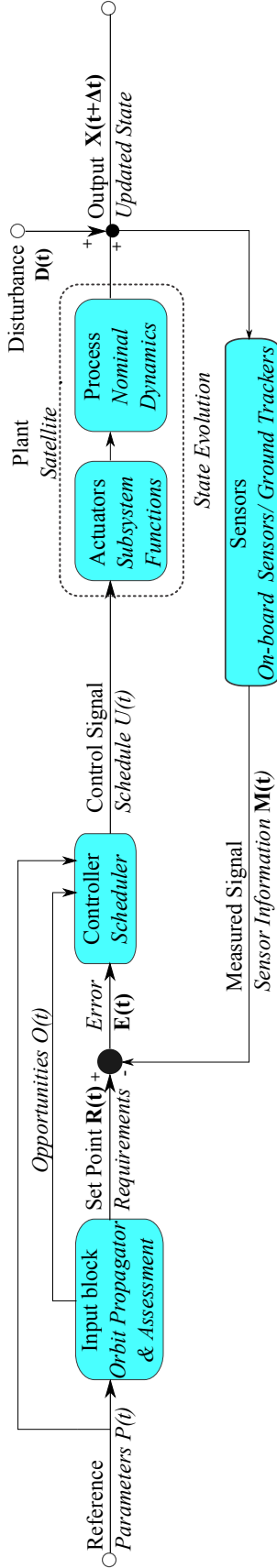


Figure 2.2: Elements and dynamics of the system model represented with a conventional feedback control loop diagram. The non-italicized labels are the conventional elements of a control feedback loop. The italicized labels are the elements of the modeling framework.

Table 2.1: Application of modeling framework to ground station and spacecraft models. All elements are time-dependent.

Elements	Ground Station Model	Spacecraft Model
Parameters	See Table 2.2	
States	position (l), attitude (q), downloaded data (d_{dl})	position (l), attitude (q), on-board energy (e), on-board data (d), downloaded data (d_{dl})
Opportunities	downloads to ground (o_{dl})	solar illuminations (o_{sol}), payload operations (o_{pl}) downloads to grounds (o_{dl})
Subsystems	communication	payload, communication, power collection, power management, data management, bus
Schedule	Assigns when and how to communicate (U)	

Table 2.2: Parameters that are explicitly used in the formulation of the *communication-focused* model.

Category	Parameter
Mission	$T_f, SNR_{min}, e_{min}, \Theta_i, I,$
Spacecraft Vehicle	$r_{op}, r_{pl}, r_{dl}, p_{op}, p_{pl}, p_{pr}, p_{dl}, A, \eta_p, d_{max}, e_{max}, G_t, L_l, \eta_r, e_{start}, d_{start}$
Ground Network	T_s, G_r, η_{dl}
Environment	E_{solar}

receivers while the links are communication channels, in parallel with information theory [148, 149].

Ground nodes are by definition located on the surface of a body with its own gravity field (such as the Earth, Mars, or the Moon), such as ground station antennas or ground-based science instruments. The location of the node is known with respect to a reference frame fixed and centered at the body’s center, such as the Earth Centered Earth Fixed (ECEF) frame, and designated by coordinates such as latitude, longitude and altitude. In our capacity assessment effort, we consider ground nodes as infinite sources or sinks of information where received messages can always be stored for later access and messages to transmit are always available. While we assume ground nodes are generally not energy limited, there may be power constraints on the communication system to consider in the link equation evaluation. The models presented can be augmented with constraints to satisfy realistic data or energy limitations.

A *space node* is located in orbit above a body’s surface and is usually mobile such that its position is not necessarily stationary relative to the frame fixed to the central body it

orbits, such as the ECEF frame. In the examples used in this paper, we employ deterministic orbital dynamics models to describe the motion of space nodes; however the exact position and orientation of the spacecraft body may not always be known, detectable, or controllable due to unmodeled disturbance forces, and the limits/absence of a determination and control system. For our ground-centric capacity assessment, we assume space nodes are capable of sourcing and sinking infinite amounts of data without energy limitations. This enables us to isolate ground station and spacecraft characteristics that influence network capacity.

A *link* is a means of connecting one node to another for the purpose of exchanging data. This work focuses on bidirectional links between ground and space nodes, which are generally wireless. For communication, a line-of-sight must exist between the two communication nodes. This requires an unobstructed straight line segment connecting the two communication nodes, where an obscuring feature (*e.g.* geography, obstruction) depends on the frequency of the desired communication method (*e.g.* weather may interfere with communication at certain frequencies).¹

We refer to *communication* as the exchange of information. *Information* is comprised of mission-specific data (telemetry, operational, science, etc.) along with the communication protocol overhead. We are interested in the overall ability to move data (bits) across the network, and are not concerned with the type of data (telemetry or payload). Each message can be decoded to obtain meaning, and perfect communication is when the intended message is identical to the decoded message. Communication may be partially successful if a fraction of the desired message is received and successfully decoded, and performance metrics may be implemented to measure the success of communication.

Communication *rate* is the speed at which information is transferred over links. This is also referred to as the net bit rate, or maximum throughput of a communication path. Data rate is often expressed in bits per second (bps), and in the sense of Shannon’s channel capacity, this is the maximum bandwidth of a communication channel at which information can be sent to meet minimum Signal-to-Noise Ratio (SNR) requirements [153].

2.2.2 Ground Station Network Model

This ground station network model consists of a ground network, consisting of at least one station, with deterministic elements and dynamics.

¹Communication may be feasible without a direct line of sight, for example with the spacecraft-based IRIDIUM wireless communications network enabling telephone service [150], moon-reflected signals between Earth and mobile space nodes [151], and extremely low SNR meteor communication [152]. These types of communication are not directly considered in our current work.

Elements

The elements of the model are summarized in Table 2.5. We assume a deterministic operational spacecraft exists and interacts with the ground network and downloads data when in view of the stations according to the operational schedule.

This model captures a single subsystem, the Communication subsystem, with a single function, to receive downloaded data. Each ground station is an immobile *ground node*, thus its position is fixed in ECEF. The ground station antenna attitude is generally controllable such that it can slew and track an orbiting spacecraft. There are constraints on the feasible attitude configurations and feasible rate of change in this configuration (i.e. slew rates). Antenna gain varies as a function of attitude. Availability of power is generally not a concern at the ground station, thus it is not modeled. We assume that infinite information is available for exchange. Spacecraft generally have the ability to collect great amounts of data (including telemetry, on-board sensor data, images and video, and status and performance updates), and similarly large amounts of data may be available at ground nodes for uplink (including code configuration updates, parameters for on-board processing, and propagated state variables). Ground nodes (excluding mobile or remote stations) have very high bandwidth to other ground nodes due to their connectivity to other stations. Data exchange between ground nodes via the globally-distributed Internet does not constrain this system because average Internet data rates exceed the space to ground communication data rates by several orders of magnitude. The existence and bandwidth of the links between the ground and space nodes are time variant, related to the dynamics and constraints of the spacecraft orbit and communication systems.

Formulation

The objective is to maximize the cumulative amount of downloaded data, which is the total capacity of a given network N ,

$$\max C_N, \tag{2.7}$$

$$C_N = \sum_{j \in J^m} C_j, \tag{2.8}$$

where J^m is the set of all ground stations, where there are m stations. C_j is the capacity of ground station j , defined as,

$$C_j = \sum_{i \in I^n} \int_0^T a_{ij}(t) l_{ij}(t) r_{ij}(t) \eta_j(t) dt, \tag{2.9}$$

and I^n is the set of all spacecraft of dimension n . In Eq. 2.9, $a_{ij}(t)$ represents the availability of a link (the existence of a line-of-sight) between ground station j and spacecraft i at time t . The establishment of a communication link, $l_{ij}(t)$, is driven by the ground station schedule and constraints. The dynamic data transfer rate is represented $r_{ij}(t)$ and characteristic of the space and ground communication systems. The efficiency, $\eta_j(t)$, is a function of the ground station characteristics, where we assume in this assessment that the spacecraft maintains an ideal link and is 100% efficient. The total capacity of a single ground station in a network is computed by summing the integrated data transfer rates to each spacecraft over the full time period of interest, $t \in [0, T]$. Note that the total data transfer time between a spacecraft and ground station is comprised of multiple passes, which may have different data transfer rates and time intervals. The four components of the station capacity model may be used to populate the following matrices to aid in implementation: $a_{ij}(t) \in A(t)^{n \times m}$, $r_{ij}(t) \in R(t)^{n \times m}$, $l_{ij}(t) \in L(t)^{n \times m}$, and $\eta_j(t) \in E(t)^m$.

1. *Availability* The first component of the network capacity model is based exclusively on the existence of a communication link between a single ground station and single spacecraft. This is dependent on a line-of-sight between the two nodes as a function of time, the orbital dynamics of the spacecraft, and the minimum elevation visibility constraint of the ground station. The availability matrix is $A(t)^{n \times m}$, consisting of elements $a_{ij}(t) \in \{0, 1\}$, $\forall i \in I, j \in J, t \in [0, T]$, where an available link between a spacecraft i and ground station j at time t is expressed $a_{ij}(t) = 1$, and when there is no visibility, $a_{ij}(t) = 0$.

2. *Data Transfer Link*

Governed by the scheduling constraints of the ground station, a link to a given spacecraft may or may not be desired even if one is available. $L_{ij}(t)^{n \times m} \in \{0, 1\} \forall i \in I, j \in J, t \in [0, T]$ is the link matrix, for which $l_{ij}(t) = 1$ for a desired link between spacecraft i and ground station j at time t and $l_{ij}(t) = 0$ if the schedule will not allow for communication. As an example schedule constraint, consider a ground station j , which can communicate only with one spacecraft i at a given time t . If $l_{ij}(t) = 1$, when $i = p$, it follows that $l_{ij}(t) = 0, \forall i \neq p$. In our ground station centric model, we assume that the link parameter is independent of the spacecraft constraints and focus on the ground station constraints.

3. *Data Transfer Rate*

The data transfer rate matrix is defined as $R_{ij}(t)^{n \times m} \forall i \in I, j \in J, t \in [0, T]$, where the data transfer rate between spacecraft i and ground station j at time t is

$r_{ij}(t)$. Typically, rates are selected at design time based on expected performance, and may have the ability to be updated during the operation of ground stations and spacecraft. The data rates are constrained by the minimum SNR requirement through the link equation, see Ref. [61]. Optimal communication rate distributions may be selected that maximize throughput, exploiting the increased SNR from decreased range distance as the elevation angle increases [154].

When assessing capacity, we suggest two approaches to populate $R_{ij}(t)$. In the first, the data rate matrix is ground-centric and represents the maximum communication rate of the station for some standard communication scenario. Alternatively, $R_{ij}(t)$ can be populated with rates that reflect operational constraints of missions and the matching of spacecraft and ground stations.

4. *Ground Station Efficiency*

Successful data transfer from spacecraft to ground station is influenced by the ground station efficiency, $\eta_j(t) \in [0, 1] \forall j \in J, t \in [0, T]$. This value reflects the estimated fraction of contact time when the communication link is not maintained due to antenna slewing and acquisition maneuvers, key holing, ground station failures, and local noise emissions that degrade the SNR. A ground station which always operates perfectly has an efficiency factor $\eta_j(t) = 1 \forall t \in [0, T]$, while a ground station which establishes a successful link on average for 90% of the available spacecraft time, $\eta_j(t) = 0.9 \forall t \in [0, T]$. Given our ground station centric capacity model, the spacecraft are modeled as point masses with perfect communication systems which can always close the link to a ground station in view.

2.2.2.1 **Model Representations**

We now introduce four example, layered network capacity models where with every additional model layer, classes of constraints are progressively considered to approach a more realistic representation. Each layer has successively higher model fidelity, and therefore generally results in a reduction in overall capacity. Figure 2.3 shows each successive model enclosed within a smaller ellipse, where the area represents the network capacity. Table 2.3 describes the models and summarizes each component of the ground station capacity from Eq. 2.9.

The *maximum model* assumes constant line-of-sight availability between communication nodes such that $a_{ij}(t) = 1 \forall t \in [0, T]$ and that data is transmitted constantly. The first model level is useful to characterize the ground station system and network at the overall maximum throughput rate, r_{max} . Next, we assess the communication link availability

between spacecraft and ground stations as a function of geographical constraints, specifically ground station locations and spacecraft orbits. This falls into the framework of the *topological model*, where we consider the availability of a communication link as a function of time, specifically the matrix $A(t)$. Station scheduling constraints are introduced in the higher fidelity *scheduled model*. Ground station operational constraints and conflicting spacecraft schedules are key parameters in the scheduled level, particularly in populating the link matrix $L(t)$. Ground station efficiency is considered in the final *actualized model*, and includes parameters such as ground station antenna pointing accuracy, hardware reliability, and the mean time to failure and recovery of station systems. We consider these factors collectively in the single ground station efficiency term, η_j . In the future, we will model this parameter dynamically and as a function of communication parameters, while currently we use estimates based on experimental data (see for example Ref. [155]). Real-time changes to the availability, data transfer rate, links, and efficiency matrices may also be considered, driven by variable data transfer rates, demands from the ground station and spacecraft users, and failures.

Each model is captured schematically for an example communication pass in Table 2.3, where the shaded area represents the communication capacity. For each model, capacity is integrated as a function of the data rate over the time interval of communication, where the total pass duration is t_A . In the *maximum model*, we communicate at the maximum feasible rate throughout the entire pass duration, $C_{max} = r_{max}t_A$ (represented by the entire shaded square). In the *topological model*, the feasible data rate required to close the communication link is the sinusoidal shape shown in the plots in Table 2.3. The feasible rate is dynamic as it is a function of the range between the ground and space nodes, which varies with the spacecraft's elevation. The elevation to begin communicating controls the capacity by governing both the optimal data transmission rate, r_{opt} , and the length of time communication is maintained, t . An optimal elevation can be selected based on the problem parameters [154]. In our example, we begin transmitting when $r_c = \frac{1}{2}r_{max}$, and the shaded capacity area is reduced significantly relative to the *maximum model*.

The *scheduled model* considers that multiple spacecraft may be competing for ground station resources, yet the ground station can communicate only with a single spacecraft at one instant in time. We introduce the average link parameter, l_{avg} , to represent the relative amount of total access time dedicated to a single ground station and spacecraft link. The shaded area representing network capacity shrinks by a factor equivalent to this average link parameter of $l_{av} = \frac{3}{5}$ in our example. Finally, the ground station efficiency ($\eta_{av} = \frac{2}{5}$ in our example) further shortens the total data transmission time in the *actualized model*, reducing total network capacity. Note the significant reduction in window of communication in

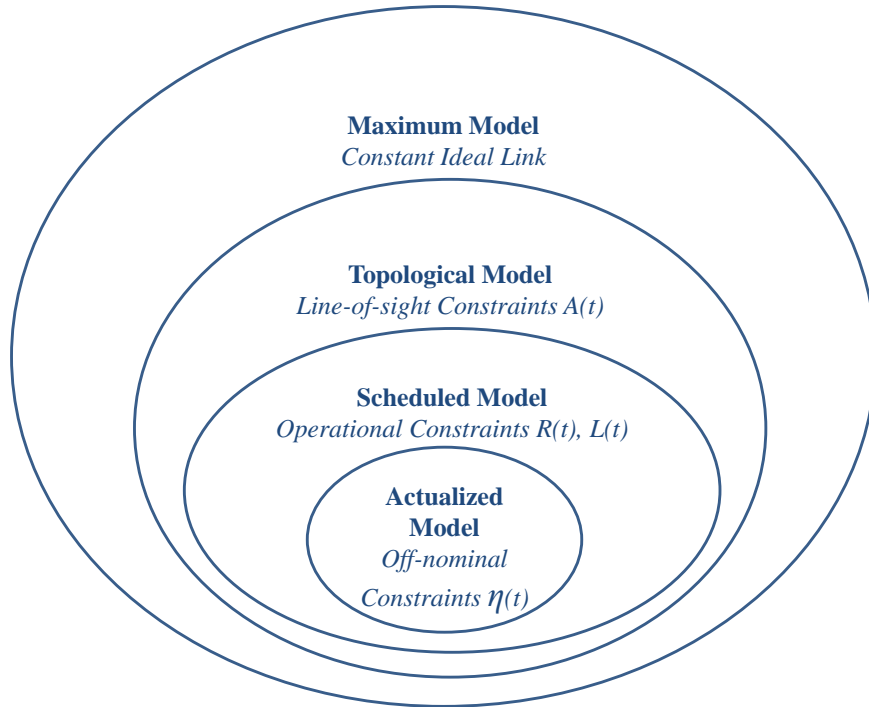


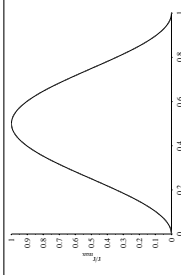
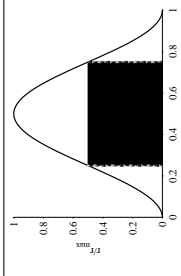
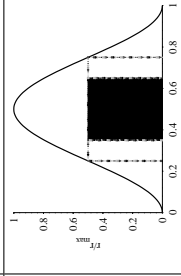
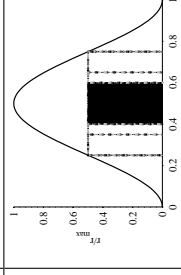
Figure 2.3: Schematic with increasingly higher fidelity ground station models within smaller ellipses, where the ellipse area represents network capacity. Note this diagram is not to scale.

the *actualized model* relative to the *maximum model* with the additional constraints. The example is simply for illustrative purposes; as a realistic scenario spans many download windows of opportunity.

2.2.3 Communication-Focused Spacecraft Model

This model consists of a single mobile space node, a spacecraft with deterministic elements and dynamics. We assume the inputs and dynamics are known a priori. In this section, we assume the schedule is an input to our simulator and developed with a simple heuristic or optimization algorithm. The model provides a detailed, analytical description of the energy, communication, and payload subsystems of a spacecraft and formulates an optimization problem to maximize data download. We assume a deterministic ground station network exists and interacts with the spacecraft by collecting downloaded data.

Table 2.3: Ground station (GS) models for spacecraft communication.

Model	Maximum	Topological	Scheduled	Actualized
Schematic				
Additive Constraints	GS Link Budget	Spacecraft Orbit, GS Locations	GS Operations	GS Downtime, Failures, Pointing, Keyholing
Parameters	$a = 1$ $r = r_{max}$ $l = 1$ $\eta = 1$	$a(t) = r_c = \frac{1}{2}r_{max}$ $l(t) = 1$ $\eta(t) = 1$	$a(t) = r_c = \frac{1}{2}r_{max}$ $l(t) = l_{avg} = \frac{3}{5}$ $\eta(t) = 1$	$a(t) = r_c = \frac{1}{2}r_{max}$ $l(t) = l_{avg} = \frac{3}{5}$ $\eta(t) = \eta_{avg} = \frac{2}{3}$
Capacity	$C = r_{max}tA$	$C = r_{max}t$	$C = r_{max}l_{avg}t$	$C = r_{max}l_{avg}\eta_{avg}t$

Elements

The elements of the spacecraft-specific model are summarized in Table 2.4. The *parameters* of the model are grouped into four categories: mission, vehicle, ground, and environment. Specific parameters are introduced in context of their usage later in this section and summarized Table 2.5. Several parameters, such as orbital properties of the spacecraft and atmospheric density, are not specifically mentioned in the formulation but are implicit in the simulation systems described in later sections. The *states* of the model are position, l , attitude, q , on-board energy, e , on-board data, d , downloaded data, d_{dl} , and their derivatives. There are opportunities for solar illuminations, o_{sol} , payload operations, o_{pl} , and downloads to grounds, o_{dl} . The model assumes e , d_{dl} , and d have no nominal dynamics. The *subsystems* are Communication, Energy Collection, Energy Management, Payload, Data Management, and the Spacecraft Bus. Subsystem functions are described in the next section.

Table 2.4: Elements in the formulation of the *communication-focused* model.

Type	Elements
Parameters	see Table 2.5
States	position, l , attitude, q , on-board energy, e , on-board data, d , downloaded data, d_{dl} , and derivatives
Subsystems	communication, energy collection/management, payload, data management, spacecraft bus
Schedule	defined a priori as an input

Table 2.5: Parameters that are explicitly used in the formulation of the *communication-focused* model.

Category	Parameter
Mission	$T_f, SNR_{min}, e_{min}, \Theta_i, I,$
Vehicle	$r_{op}, r_{pl}, r_{dl}, p_{op}, p_{pl}, p_{pr}, p_{dl}, A, \eta_p, d_{max}, e_{max}, G_t, L_l, \eta_r, e_{start}, d_{start}$
Ground	T_s, G_r, η_{dl}
Environment	E_{solar}

In this example model, the *schedule*, $U(t)$, is an input prescribed a priori by the user of the model. $U(t)$ defines when and how all events are executed. It is assumed to execute perfectly without feedback impacting scheduling decisions. The model simply captures deterministic dynamics and models the expected scenario. This is useful for determining feasibility of $U(t)$ to meet mission requirements, if and when constraints are active, and how sensitive the scenario is to input parameters.

Formulation

The objective of the mission, as represented in the optimization problem, is to maximize data downloaded, d_{dl} , which is expressed in Eq. 2.10. The model analytically captures state dynamics for energy, e , data downloaded, d_{dl} , and on-board data, d in Eqs. 2.12-2.14. Position, l , and attitude, q , are not explicitly modeled because there are no modeled subsystem functions that operate on these states. Simulators calculate l and q states as well as o_{sol} , o_{pl} , and o_{dl} , as will be shown in the following section. Upper and lower bounds on the on-board energy and data are enforced in Eqs. 2.15-2.16. A mission requirement on download latency in Eq. 2.17 defines the minimum data download amount, Θ_i , over specified time intervals, $i \in I$, where r_{dl} is the rate of data download. Subsystem functions and their impact on state evolution are described next.

$$\max_{U(t)} C_N \quad (2.10)$$

s.t.

$$C_N = d_{dl}(T_f) \quad (2.11)$$

$$e(t) = e_{start} + \int_0^t [p_{sol}(\tau) - p_{op}(\tau) - o_{pl}p_{pl}(\tau) - p_{pr}(\tau) - o_{dl}p_{dl}(\tau) - p_{sp}(\tau)]d\tau \quad \forall t \in [0, T_f] \quad (2.12)$$

$$d_{dl}(t) = \int_0^t \eta_{dl}o_{dl}(\tau)r_{dl}(\tau)d\tau \quad \forall t \in [0, T_f] \quad (2.13)$$

$$d(t) = d_{start} + \int_0^t [r_{op}(\tau) + o_{pl}r_{pl}(\tau) - r_{sp}(\tau)] d\tau - d_{dl}(t) \quad \forall t \in [0, T_f] \quad (2.14)$$

$$e_{min} \leq e(t) \leq e_{max} \quad \forall t \in [0, T_f] \quad (2.15)$$

$$0 \leq d(t) \leq d_{max} \quad \forall t \in [0, T_f] \quad (2.16)$$

$$\Theta_i \leq \int_{t_i^s}^{t_i^e} r_{dl}(t)dt \quad \forall i \in I \quad (2.17)$$

The function of the Communication subsystem is to download data from the spacecraft to a ground station network. Given download opportunities, i.e. when $o_{dl}=1$, the subsystem uses on-orbit energy at the rate p_{dl} , to downloaded data d_{dl} at the rate r_{dl} with transmit power, p_t . Consider the *link equation*, which defines the expected signal-to-noise ratio,

SNR , of the communication channel, as in Eq. 2.18 [61].

$$SNR = \frac{p_t G_t G_r L_l L_s L_a}{k_B T_s r_{dl}} \quad (2.18)$$

G_t and G_r are the gains of the transmit (spacecraft) and receive (ground station) antennas, respectively. Three sources of signal loss decrease SNR : L_l is the transmitter-to-antenna line loss, L_s is the free space loss, where L_s is inversely proportional to the transmission frequency and the square of the distance between the spacecraft and the ground station, and L_a is the transmission path loss resulting from the propagation medium and varies as a function of atmospheric weather. System noise is modeled as the product of the Boltzmann constant, k_B , and the noise temperature of the receiver, T_s . In Eq. 2.18, we assume perfect spectral efficiency, i.e. $\beta = r_{dl}/B = 1$, where B is the communication bandwidth, independent of modulation scheme. A minimum SNR , SNR_{min} , is required for maintaining an acceptable link quality such that $SNR \geq SNR_{min}$ [61].

Analysis is simplified by introducing a substitute variable, α , which is a measure of the energy per data (measured in Joules per bit) required to download data from the spacecraft to the ground at $SNR = SNR_{min}$, as in Eq. 2.19. Since L_s and L_a are time varying, α is dynamic. Antenna gain terms may be occasionally time varying as well, depending on relative orientation of the antennas.

$$\alpha = \frac{p_t}{r_{dl}} = \frac{k_B T_s (SNR_{min})}{G_t G_r L_l L_s L_a} \quad (2.19)$$

The transmitter efficiency, η_r , relates the power provided to the radio, p_{dl} , and the output power transmitted by the radio, p_t , such that $p_t = \eta_r p_{dl}$. Finally, the analytical relationship between the Communication subsystem parameters r_{dl} , p_{dl} , and p_t , is expressed as an inequality,

$$\frac{p_t}{\eta_r} = p_{dl} \geq \frac{\alpha r_{dl}}{\eta_r}, \quad (2.20)$$

since SNR_{min} is a minimum constraint. During operations, use of the minimum p_{dl} or maximum r_{dl} can be exploited to optimize communication links [154].

The function of the Energy Collection subsystem is to collect energy with solar panels when illuminated, i. e. when $o_{sol} = 1$, which is dependent on l and q . The illuminating solar energy density, E_{solar} , is converted to usable electrical energy at a rate p_{sol} and is a function of solar panel efficiency η_p , light incidence angle, $\theta(t)$, and the effective area of the panels, A . θ is a complex relationship between the spacecraft configuration and attitude,

and is determined by simulators in this model. Analytically, the function is described Eq. 2.21, where K is the set of spacecraft body faces.

$$p_{sol} = \sum_{k \in K} o_{sol} E_{solar} A_k \eta_p \cos(\theta_k) \quad (2.21)$$

The Energy Management subsystem stores and regulates on-board energy, e , according to the expression in Eq. 2.12. Its input is p_{sol} from the energy collection subsystem. It outputs power to the other subsystems and manages energy stored in the battery, which is e . Stored energy is added or consumed depending on whether p_{sol} exceeds the instantaneous power needs of the subsystems. The Spacecraft Bus requires continuous power, p_{op} . As prescribed by $U(t)$, the Payload requires p_{pl} during experiments, Data Management requires p_{pr} during processing, and Communication requires p_{dl} during downloading. If the battery reaches full capacity, i.e. $e = e_{max}$, the Energy Management subsystem diverts or spills surplus available solar energy at a rate of p_{sp} to prevent battery overcharging, as captured in Eq. 2.15. Depth of discharge management is enforced in Eq. 9 with a minimum battery storage capacity, $e_{min} = e_{max}(1 - \zeta)$, where ζ is the maximum allowable depth of discharge. At the start of the planning horizon, $t = 0$, the initial amount of stored energy is e_{start} .

The Payload subsystem collects or generates data to be processed and downloaded when prescribed by an input $U(t)$. The Payload consumes energy at a rate p_{pl} and outputs data at a rate r_{pl} .

The Data Management subsystem manages, stores, and regulates on-board data, d , according to Eq. 2.14. The inputs are data from the Bus and Payload subsystems at rates r_{op} and r_{pl} , respectively, and power at a rate p_{pr} to process the data (see Eq. 2.12). Processing includes compressing data (which may significantly reduce the amount of data such that it can be downloaded) at a rate r_l and outputting data at a rate r_{dl} . If data, d , reaches maximum level, d_{max} , as specified in Eq. 2.16, data is deleted at a rate r_{sp} to satisfy storage constraints. The initial amount of data on the spacecraft is d_{start} .

The cumulative amount of data downloaded from the spacecraft, d_{dl} , is analytically expressed in Eq. 2.13. d_{dl} is the integral of the product of the download rate, r_{dl} , and the download efficiency, η_{dl} , which is a function of the ground station communication subsystem [156].

As a model simplification technique, a Spacecraft Bus is included, which consists of the remaining spacecraft components not explicitly modeled by the previous subsystems. The input to the Bus subsystem is power, p_{op} , and it outputs telemetry data at a rate r_{op} .

2.3 Data Sets and Simulator

The model is executed in a simulation environment using mission data from the small satellite community, as described in the following subsections.

2.3.1 Satellite and Ground Station Data

To obtain realistic data sets, we have deployed two online surveys, one focusing on existing and upcoming small satellite missions and the other focusing on the ground stations supporting these satellites [1, 157]. These surveys were deployed to collect information on past, existing, and future small satellite missions and ground stations, which are critical to verify our models and tools for realistic applications. Furthermore, optimizing realistic data sets provides insights about the performance potential and constraints of existing and future missions and networks.

2.3.1.1 Small Satellite Survey

The Small Satellite Survey is a database of operational information about past and future small satellite missions [157]. The survey includes questions about the launch parameters, mission goals, download goals, as well as details on the spacecraft constraints such as on-board energy and data storage capacity.

Review of the survey indicated three representative satellite operational modes that capture how missions collect data: focused, opportunistic, and continuous. In the *focused* mode, all operational decisions, including data collection and download opportunities, are known a priori; the satellite has a deterministic schedule that does not vary. In the *opportunistic* mode, data collection is activated by stochastic events such as solar activity. In the *continuous* mode, data is collected or generated continually. Satellites may operate in a single or combination of these modes throughout its mission.

Three diverse missions from the satellite survey representing each of the operational modes are summarized in Table 2.6. The *focused* Radio Aurora eXplorer (RAX) measures ionospheric properties using bi-static radar measurements [158, 159]. RAX has the opportunity to collect radar data several times per day when it passes over the experimental zone located in Poker Flat, Alaska. After an experiment, RAX processes and downloads the data to eight globally distributed ground stations at a data rate of 9600 bps. The *opportunistic* myPocketQub 391 (myPQ) picosatellite mission will allow members of the public to upload software payloads and perform custom mission operations [160]. The myPQ satellite is designed to continuously download image and sensor data to a very large glob-

Table 2.6: Example missions representing the three operational modes.

Mission Parameters	Operational Mode		
	Focused	Continuous	Opportunistic
Mission Name	RAX-2	DICE	myPocketQub 391
Orbit Altitude (a)	350- 820 km	350- 820 km	700 km
Orbit Inclination (i)	102°	102°	98.8°
Spacecraft size	10 x 10 x 30 cm ³	10 x 10 x 15 cm ³	46 x 46 x 69 mm ³
Payload Data Collection Frequency	several times/day	Continuously	Random (50% of time)
Payload Data Collection Duration	≈5 minutes	One orbit	10 minutes
Payload Data Collected Amount (raw)	1.2 GB/day	10.5 MB/orbit	≤ 512 MB/orbit
Download Requirement	1 MB/day	0.29 MB/day	1 MB/day
Desired Download	5 MB/day	1 MB/day	1000 MB/day
Average Power (p_{sol})	8 / 5.5 / 3 W	6 / 4 / 3 W	3 / 2 / 1 W
Best/ Expected/ Worst (P1/ P2/ P3)			

ally distributed network (with over 100 stations) at a maximum data rate of 1200 bps. The *continuous* Dynamic Ionosphere CubeSat Experiment (DICE) mission consists of a pair of satellites that study the interactions of the Earth’s upper atmosphere and the Sun using a suite of on-board payload instruments including electric field probes, Langmuir Probes, and magnetometers. The two DICE satellites were launched with RAX in 2011 and download to two high-gain UHF antennas at a data rate greater than or equal to 1.5 Mbps. Additional survey results are given in Section 5.3 for a larger group of representative small satellites.

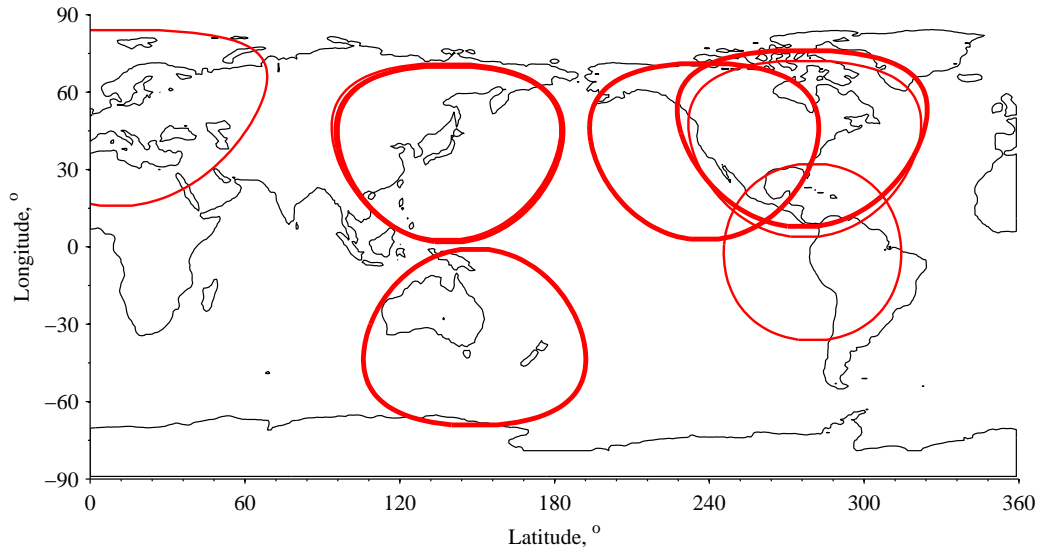
2.3.1.2 Ground Satellite Survey

The Ground Station Survey provides an database of information on existing ground stations from the CubeSat and amateur radio communities [1, 161]. The survey provides us with a online database containing necessary communication information for modeling these station, including their locations and capabilities. The ground station networks used to support small satellites are independently owned and operated and not centrally controlled.

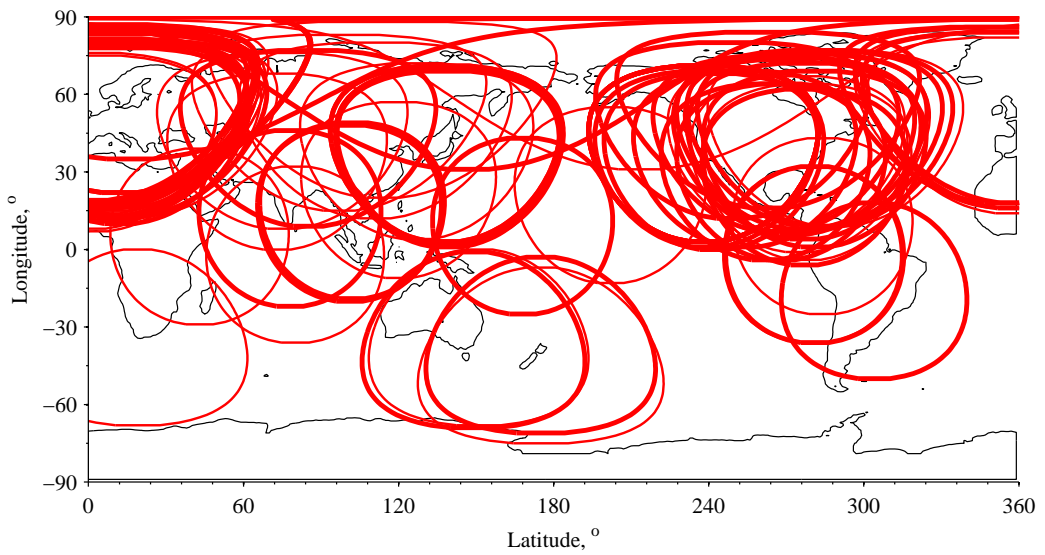
Three representative networks from the survey with variable global distribution and capabilities are summarized in Table 2.7 for representative radios and operational settings with a constant data rate and power level, where the link budget is satisfied for all elevation angles. Energy utilization for download is computed using the RAX parameters (antenna gains, losses, and efficiencies). For L_s calculations, the S-band and UHF stations are assumed to operate at frequencies of 2450 MHz and 437.505 MHz, respectively. The geographic distribution of the N2 and N3 stations are shown in Figure 2.4. The red perimeters are the ground station cone of visibility of a satellite with an altitude of 500 km projected onto the cartesian map assuming an elevation mask of 0° for the ground station.

Table 2.7: Representative ground station networks from the Ground Station Survey

Network	N1	N2	N3
Number of Stations	2	8	100
Locations	Wallops and SRI	RAX network	Globally distributed
Data Rate (r)	115.2 kbps	9600 bps	1200 bps
Frequency Type	S-band	UHF	UHF
Antenna Gain (G_r)	46 dBi	19 dBi	19 dBi
Energy Utilization (α)	.0441 mJ/bits	.4665 mJ/bits	.4665 mJ/bits



(a) RAX Network (N2)



(b) Global Network (N3)

Figure 2.4: Global locations and projected visibility cones of stations in N2 and N3 assuming a satellite altitude of 500 km and an elevation mask of 0° .

2.3.2 Simulator Description

We developed a suite of tools that implements and brings to life the modeling framework described in Section 2.1 and is used throughout the remainder of the thesis. This environment enables us to simulate, assess, and optimize a variety of diverse ground station networks and satellite populations. The simulation environment inherits much of the modularity from the modeling framework such that additional elements or interactions can be added and/or higher fidelity models can be implemented. The simulator consists of custom MATLAB[®] scripts integrated with the high-fidelity STK and other databases [162]. Most of the simulation environment is automated such that scenarios can be re-run with minimal operator involvement. For example, we have automated scripts that build STK scenarios and extract information relevant into Matlab for assessing or optimizing missions.

Our simulation tools extract data sets from a variety of sources to reflect realistic network examples. Information on ground station networks is drawn from the Ground Station Survey as well as default data sets in STK (e.g. AFSCN, USN, and NASA’s NEN). Satellite orbit information is drawn from the Small Satellite Survey and/or historical Two-Line Element (TLE) sets for launched satellites, which can be obtained from the Space-Track website [163]. For some applications, MATLAB scripts automatically extract historical TLE sets and load them into STK along with ground station locations from our database. TLEs or orbital elements of spacecraft, ground station locations, and experimental zones can also be loaded manually into STK. STK then propagates the orbit using an analytic propagator such as J4 Perturbation or Simplified General Perturbations Satellite Orbit Model 4 (SGP4) and computes contact times between the satellite and ground stations and targets of interest, as well as times the satellite is in the sun or eclipse. Additional access details such as the range, elevation, and azimuth of the satellite relative to ground targets can also be extracted. STK combines satellite orbit and attitude dynamics with information with locations of ground stations and targets of interest to determine opportunities for energy and data collection and data download, i.e. O_{sol} , O_{pl} , O_{dl} . STK-derived opportunities are exported to MATLAB and algorithms perform further processing and analysis, such as capacity, utilization, and constraint-based assessments.

In addition, a simulator was developed to execute the *communication-focused* model described in Eqs. 2.10-2.17. This simulator augments the analysis capabilities described in the previous paragraph by using the STK-derived opportunities to propagate the satellite states according to a prescribed schedule using input parameters specific to the mission. This enables verification of the schedule and allows us to perform sensitivity analysis relative to deterministic or stochastic problem parameters.

2.4 Application of Model and Simulator

In this section we demonstrate applicability of the model and simulator to a realistic mission scenario. In particular, we model the RAX-2 mission with power scenario P2 and in coordination with its dedicated ground network, N2. The mission is simulated according to a simple greedy heuristic schedule, $U(t)$, which is an input to the simulator. The satellite performs payload and download operations whenever there is an opportunity to do so. The simulation environment executes this schedule and propagates the satellite states as in Eq. 2.12-2.14 with input parameters from Table 2.6. The simulation parameters are: $p_{op} = 2$ W, $p_{pl} = 4$ W, $p_{dl} = 5$ W, $p_{pr} = 0$ W, $r_{dl} = 9.6$ kbps, $\eta_{dl} = 0.7$ and those from Table 2.6. In the simulations, we assume the nominal, payload, and download data and energy rates are constant during each interval. Furthermore, we assume energy collection occurs at a constant rate when in the sun and communication occurs whenever the satellite is above the horizon relative to a ground station in the network. These are reasonable approximations for most small satellites based on our operational experience.

The time histories of the on-board satellite energy and total downloaded data are shown in Figures 2.5 and 2.6. ² Figure 2.5 shows the scenario with an eclipse fraction of 0.35, which is the maximum annual eclipse time experienced by this orbit, and the nominal power collection, P2 ($p_{sol} = 3.3$ W). Figure 2.6 shows the scenario with the minimum annual eclipse time experienced by this orbit (zero eclipse) and the worst-case power collection, P1 ($p_{sol} = 3$ W). ³

The stored energy in Figures 2.5(a) and 2.6(a) evolves according to Eq. 2.12, where the time-dependent slope is a function of the combination of solar illumination (see sun indicator), nominal operations, and experiments and downloads (see shaded patches). The slope is generally positive when in the sun and negative when in eclipse, with lower slopes corresponding to times where experiments or downloads occur. In both cases, the energy level never reaches the lower energy bound as there is generally more energy available than required for operations in this scenario. The energy level exceeds the upper capacity and energy is spilled to satisfy the energy capacity constraint in Eq. 2.15 several times during the scenario (e.g. at 1.1, 1.6, 2.6, 3.2, 4.9, 6.5, and 8.1 hours in Figure 2.5(a) and nearly constantly in Figure 2.6(a)). The downloaded data in Figures 2.5(b) and 2.6(b) is the product of the rate of transmitted data (slope during downloads), duration of the downloads, and the download efficiency, as in Eq. 2.13. The no eclipse scenario with the worst-case power value (P1) simulation predicts that the RAX-2 mission should satisfy download

²Total on-board data is not shown because it is not informative. The total on-board data simply increases nearly monotonically because much more data is collected than could ever be downloaded.

³These dynamic eclipse trends are described in detail in Figure 3.12 in Section 2.7.

requirements (Eq. 2.17) since it nearly downloads the daily requirement (1 MegaByte) in the first 8 hours, see Figure 2.6(a). However, it is not clear if the maximum eclipse scenario with the expected power value (P2) will complete the download requirement, as it has only downloaded 0.3 MBytes in the first 8 hours, see Figure 2.5(a).

Next we compare the download results for scenarios with maximum and zero eclipse conditions with variable power collection values in Figure 2.7. The results are also summarized in Table 2.8 for the three power collection values (P1, P2, P3). When there is zero eclipse and $p_{sol} \geq 1$ W, all experiment and download opportunities are fully utilized and all constraints are satisfied, as shown in Figure 2.6. However, for scenarios that experience the maximum possible eclipse (eclipse fraction of 0.35), the RAX-2 mission is not feasible for $p_{sol} < 4$ W because there is insufficient energy to perform payload and download operations at every opportunity while satisfying Constraints 2.12-2.17. For scenarios with $p_{sol} \geq 4$ W, RAX-2 utilizes only a fraction of the download opportunities (due to limited available energy), for example only two download opportunities are used with the expected power value (P2), see Figure 2.5(b). These simulations have demonstrated that download performance is influenced significantly by the interaction of eclipse duration (which is dynamic throughout the year) and power collection value. Furthermore, these results motivate the development of optimal schedules, $U(t)$, to allocate resources to ensure constraints are satisfied and download is maximized for all possible scenarios.

Table 2.8: Total data downloaded for different eclipse and power scenarios.

Scenario Characteristics		Data Downloaded for each Power Setting		
Type	Eclipse Ratio	P1 ($p_{sol} = 3$ W)	P2 ($p_{sol} = 5.5$ W)	P3 ($p_{sol} = 8$ W)
Zero Eclipse	0	0.91 MBytes	0.91 MBytes	0.91 MBytes
Maximum Eclipse	0.35	Infeasible	0.30 MBytes	0.45 MBytes

2.5 Summary

In this chapter we have developed a fundamental analytical modeling framework, consisting of templates to model satellite states, subsystem functions, mission constraints, mission requirements, and the interactions of these elements. Different perspectives were used to express the model, including as a closed-loop control system and a multi-stage ground station capacity representation. We have implemented this model in a simulation environment that enables high-fidelity state propagation and evaluation of opportunities for collecting energy and data and downloading to ground station networks. We applied the model and

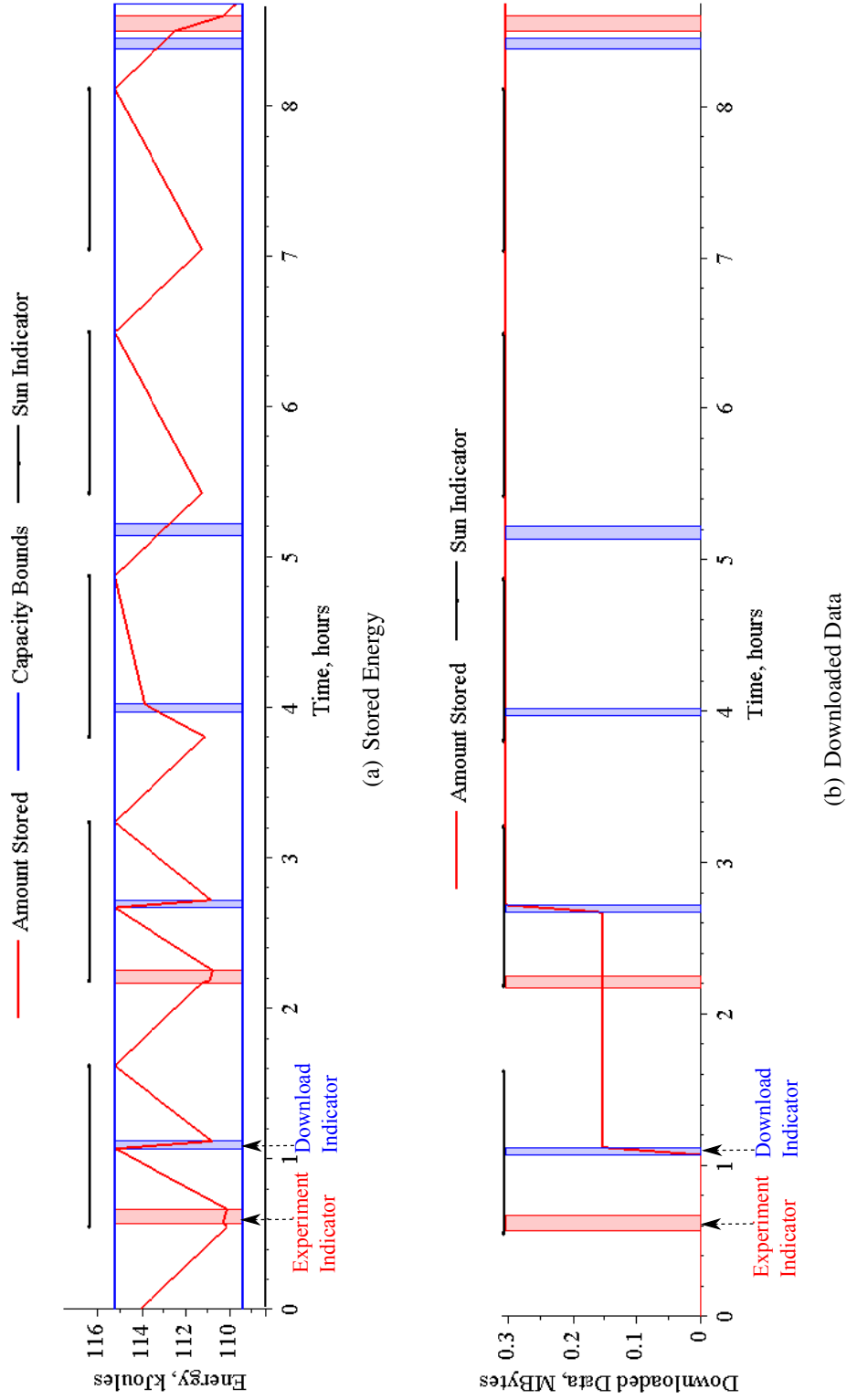


Figure 2.5: Time histories of on-board stored energy and total data downloaded for an instance of the RAX-2 mission with $p_{sol} = 5.5$ W and maximum eclipse duration of 35 % of orbit.

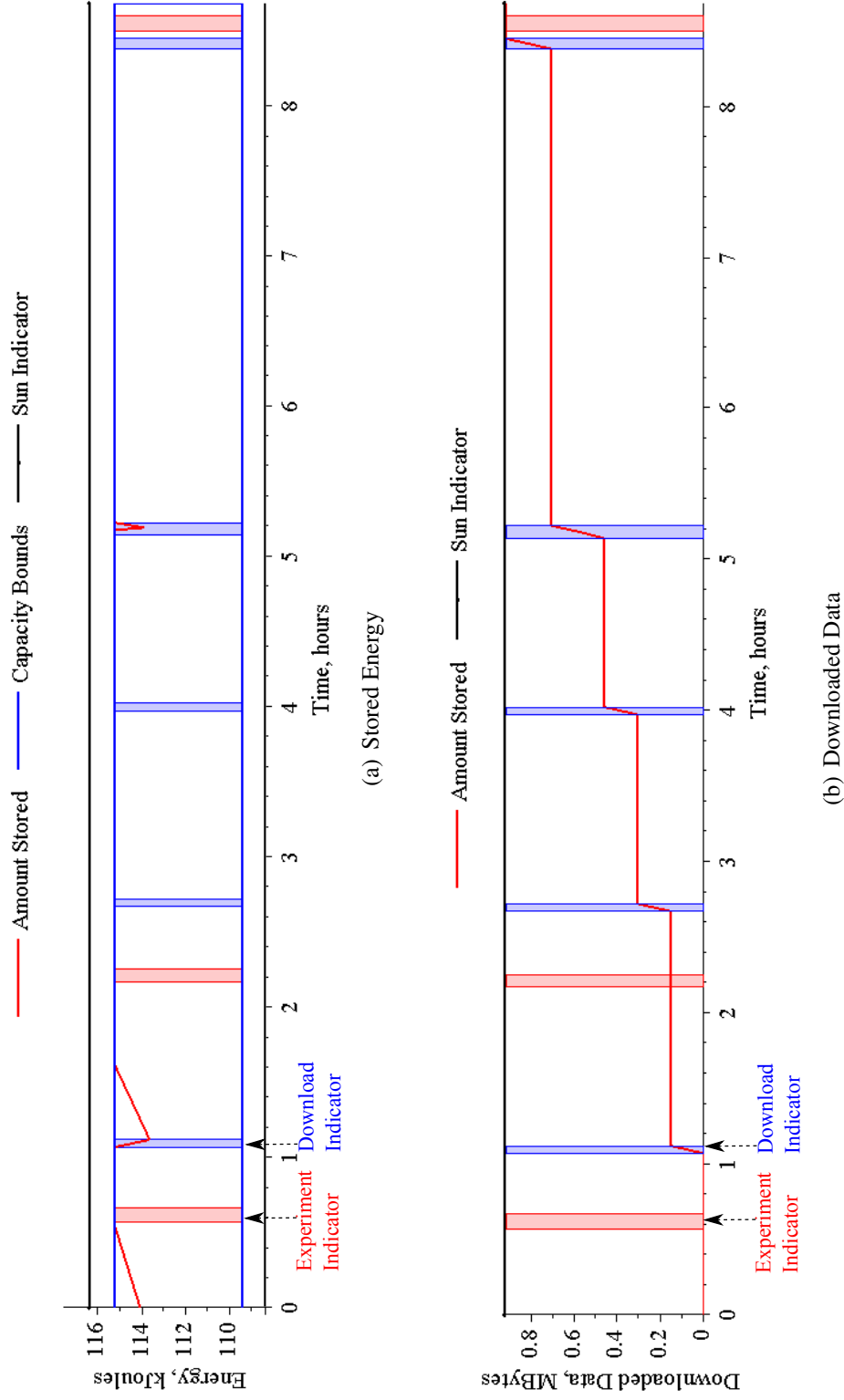


Figure 2.6: Time histories of on-board stored energy and total data downloaded for an instance of the RAX-2 mission with $p_{sol} = 3\text{ W}$ and zero eclipse duration (all 97 minutes of orbital period are in sunlight).

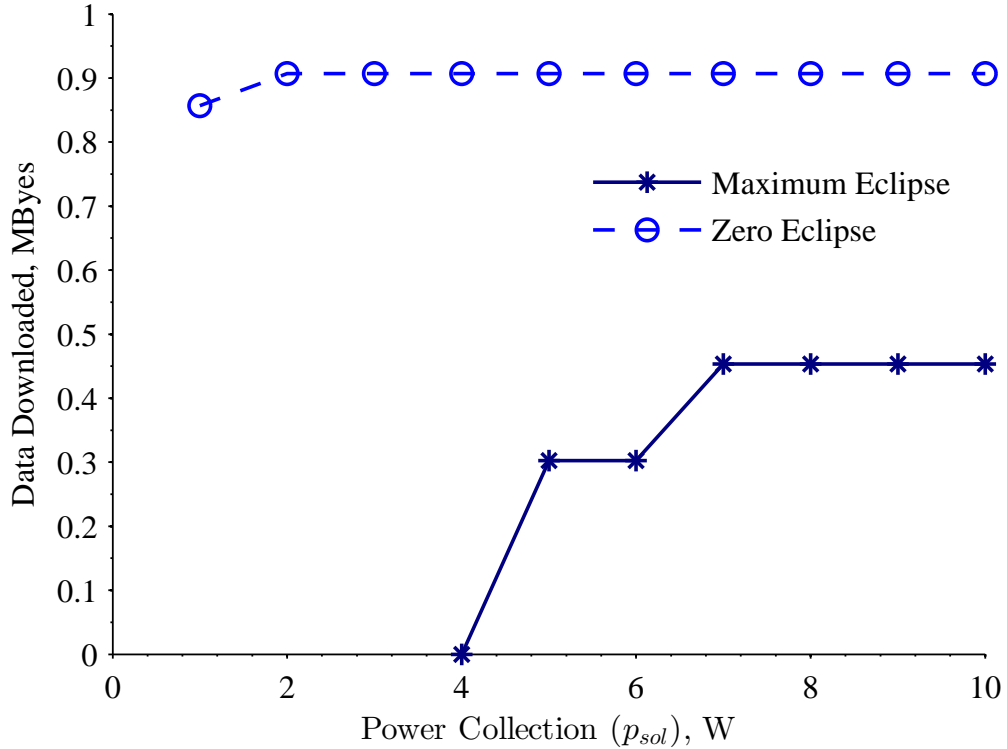


Figure 2.7: Data downloaded for variable solar power collection values (p_{sol}) for an instance of the RAX-2 mission. Results are compared when there is maximum eclipse, with an eclipse fraction of 0.35 (34 minutes of a 97 minute orbital period), and zero eclipse (always in sunlight). Note data is not plotted for infeasible mission scenarios.

simulator to develop a *communication-focused* model of a satellite that downloads data to a globally distributed ground station network and demonstrated how it can be used to evaluate system performance and constraints.

Furthermore, we have applied the framework to develop a model-based systems engineering (MBSE) representation of the RAX mission in Systems Modeling Language (SysML) [57]. This foundational model was used to develop a simulation environment that integrates STK and Matlab using Phoenix Integration Model Center to execute a specific mission scenario and perform trade studies [164]. Applying the framework to a specific mission instance requires minimal effort (several hours for an experienced systems engineer) once the problem is established.

Next we discuss the contributions of this chapter in the context of the future thesis chapters. The resource-constrained scenario discussed in Section 2.4 motivates the investigation of active constraints that limit overall performance for general classes of mission scenarios and networks. Therefore, in Chapter 3, we investigate the relationship between commu-

nication capacity and model parameters, such as orbital parameters and ground network characteristics. The modeling framework presented in this section lays the foundation for the development of future optimization formulations that are developed in Chapters 4 and 5. In particular, the foundational approach for capturing system dynamics and requirements as constraints in the context of an optimization formulation that maximizes the mission objective provides a useful template for specific optimization problems. The development of an analytic link budget is also be useful in future modeling approximations and applications to diverse LEO and interplanetary communication systems. The simulation environment and survey data enables future capacity and utilization studies in Chapter 3, as well as providing necessary inputs for optimizing schedules and testing their effectiveness in Chapters 4-6.

CHAPTER 3

Constraint-Based Capacity Assessment

Motivated by the research goals stated in Section 1.3, this chapter investigates trends in communication capacity as a function of constraints for representative satellite and ground station networks. The simulations performed in this chapter use the *communication-focused* model implemented in our simulation environment from Chapter 2. This enables the assessment of download potential relative to individual and combined parameters representative of existing and future spacecraft and ground networks. In this chapter we consider both generic orbits and ground stations and realistic example satellite missions and networks, with data sets from Section 2.3. Each of the simulations considers a single satellite communicating to a ground station network.

Constraint-based capacity is the maximum download capacity that can be achieved when a specific set of constraints is considered and all others are relaxed. Next we describe possible constraints and discuss their relative importance in the context of the missions we consider. The three primary resources necessary to support data download from the satellite perspective are energy (e), data (d), and downlink time (t). The resources are constrained by the limited opportunities to collect energy (o_{sol}), acquire data (o_{pl}), and download data to ground stations (o_{dl}). On-board data is usually abundant for space missions since payloads often generate more data than can be downloaded. For example, the RAX-2 CubeSat generates 1.2 gigabytes (GB) per day, while the satellite is capable of downloading only several megabytes (MB) per day.

The simulations in this chapter focus on constraints related to opportunities for downloading data to networks and collecting energy. We define *network constraints* as those that are exclusively a function of the download time between satellites and ground networks and *energy constraints* as those that are exclusively representing the total available energy for downloads, which are a function of the power collection (when in the sun) and eclipse time. Section 3.1 focuses on assessing communication capacity as a function of network constraints. Section 3.2 focuses on energy constraints for generic spacecraft missions. Section 3.3 compares the download capacity as constrained by available time and

energy for realistic mission examples. This section also compares the communication capacity relative to specific-mission download requirements, motivating the optimization of space networks introduced in Chapter 4.

3.1 Network Constraints

In this section, we are motivated to understand the capacity of a ground station networks, and therefore focus on a ground station centric capacity model, where capacity is modeled as a function of ground station constraints. We therefore assume reliable communication of the space node, such that the satellite can close the link to a visible ground station. We recognize that this is a simplified representation of the communication operation, and extend this model to include both the spacecraft and link constraints in Sections 3.2 and 3.3.

Coverage of orbits and ground station networks are studied in Section 3.1.1. This leads to the investigation of download time in Section 3.1.2. Download time is investigated in the context of the four model representations described in Section 2.2.2.

3.1.1 Orbit and Ground Station Coverage

This section studies the impact of dynamic node positions on availability in the context of the *topological model*. We first focus on the interaction of ground station location and satellite inclination, the major contributors to communication availability. We study capacity uniquely as a function of availability, assuming the other parameters of the link equation, $r(t)$, $l(t)$, and $\eta(t)$, are constant $\forall t \in [0, T]$. In the *topological model*, capacity is directly proportional to access time and the capacity can be expressed $C_j \propto \sum_{i \in I} \int_0^T a_{ij}(t) dt$ (from Eq. 2.9). We are interested in the dynamically varying duration of each space node pass over visible ground nodes to assess the total and average access times of the network. First we must establish if there is visibility between the ground and space node on each orbit, that is if the ground node location (latitude and longitude) is in the field of view of the space node inclination at some point during the orbit. When there is visibility on a given pass, we calculate the duration of the satellite pass, the length of time it is in view of the ground station.

Ref. [165] introduces an approximate method to determine the proportion of passes which will result in visibility between a space node and ground node. The percentage of passes which have ground station coverage is evaluated using simple geometric relationships as a function of space node inclination and altitude and ground node latitude. The

results using the algorithm for ground station latitudes and satellite inclinations in the upper hemisphere ($0^\circ \leq i \leq 90^\circ$) are shown in Figures 3.1 and 3.2. We study the variation in the results for two different minimum elevations at which the ground node establishes communication to an orbiting space node, $e_{min} = 0^\circ$ and $e_{min} = 10^\circ$ in Figures 3.1 and 3.2, respectively. The general trend is that coverage increases as the satellite inclination approaches the ground station latitude. Satellite orbits with low inclinations are visible only from equatorial or near-equatorial orbits, while polar orbits cover all of the ground station latitudes, although the percent coverage of low latitude stations is reduced as compared to near-polar latitudes. These plots can be used to extract useful information about the access times as a function of communication node parameters. For example, with the added topological constraint the satellite with $i = 60^\circ$ is no longer visible from ground stations with latitudes $> 85^\circ$ for a 650 km orbit and latitudes $> 80^\circ$ for a 300 km orbit (with $e_{min} = 0^\circ$ in Figure 3.1).

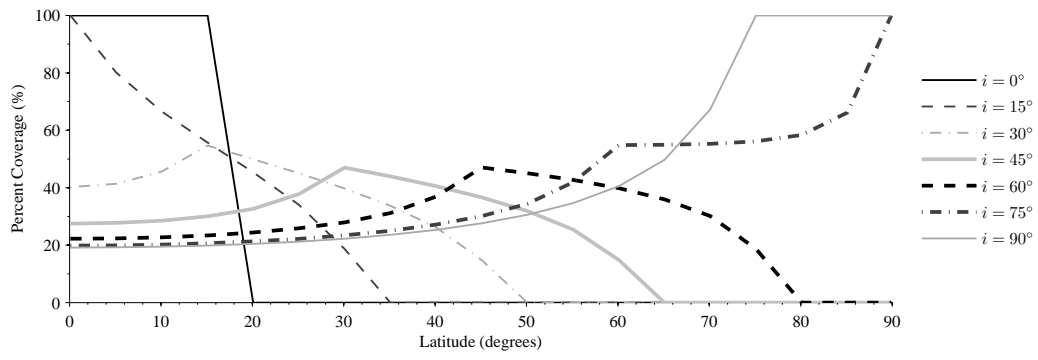
For low satellite inclinations, the percent coverage curves in Figures 3.1 and 3.2 generally shift towards greater latitudes (towards the right) with increasing altitude. The range of latitudes with zero coverage decreases by approximately 10° with each successive increment in inclination. With a higher minimum elevation constraint, shown in Figure 3.2, we note the distinct and expected reduction in available coverage times in general.

To compute total access time, we need to consider the length of each communication pass, which may be approximated for circular orbits by [166],

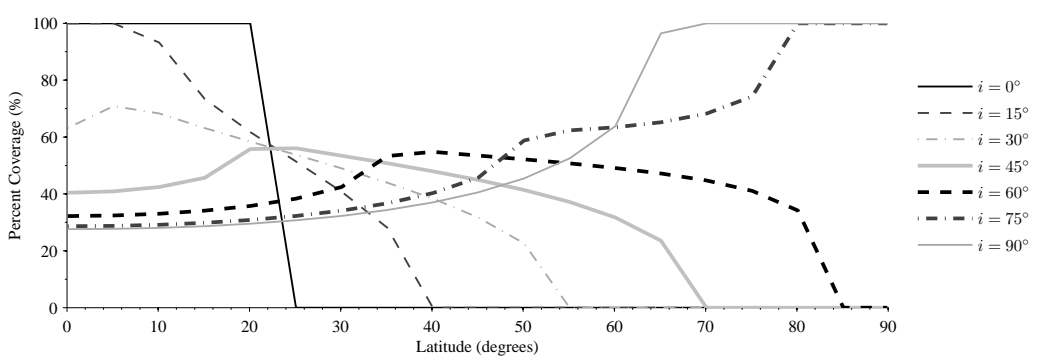
$$\tau(\theta_{max}) \approx \frac{2}{(\omega_s - \omega_E \cos i)} \cos^{-1} \left[\frac{\cos(\cos^{-1}(\frac{r_E}{r} \cos \theta_{min} - \theta_{min}))}{\cos(\cos^{-1}(\frac{r_E}{r} \cos \theta_{max} - \theta_{max}))} \right], \quad (3.1)$$

where ω_s and ω_E are the satellite and Earth rotation rates relative to an inertial frame, r and r_E are the radius of the satellite orbit and the Earth, i is satellite inclination, and θ_{max} is the maximum elevation of the space node relative to the ground node throughout the pass. Circular orbits are representative of most of the small satellite missions we consider, and these tools can be extended to the case of elliptical orbits.

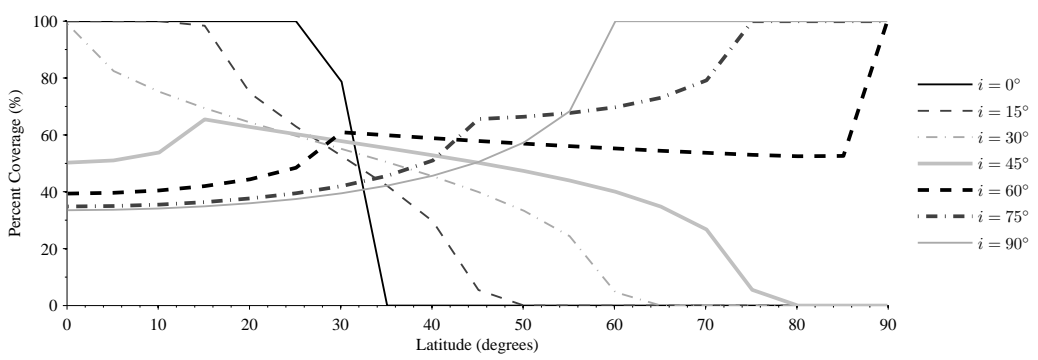
The key parameter from Eq. 3.2 governing access time is θ_{max} , which varies dynamically as a function of the orbital parameters of the space node and ground node location. The maximum elevation is related to the perpendicular distance between the orbital plane and the ground station location, and found analytically by exploiting the geometric relationships between the instantaneous positions of the nodes. Using the algorithm provided in Ref. [166], the length of each pass can be computed as a function of satellite inclination, ground station latitude, and minimum ground elevation constraint. For example, a ground station at a 20° latitude will achieve coverage to satellites with inclinations $\geq 20^\circ$ for a



(a) Altitude: 300 km

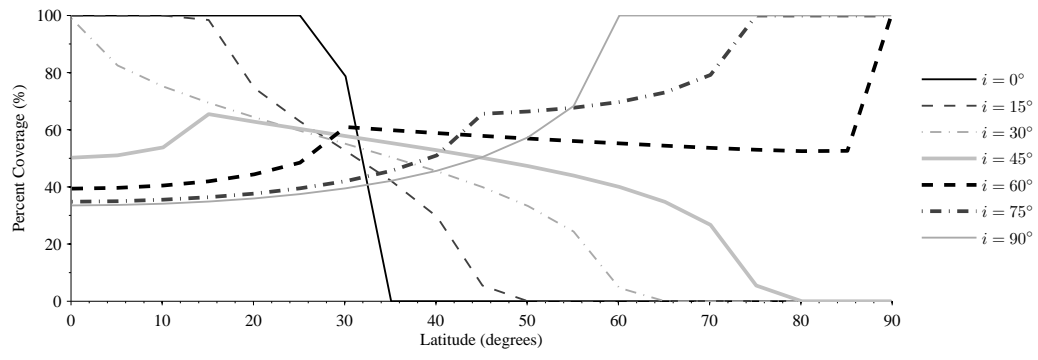


(b) Altitude: 650 km

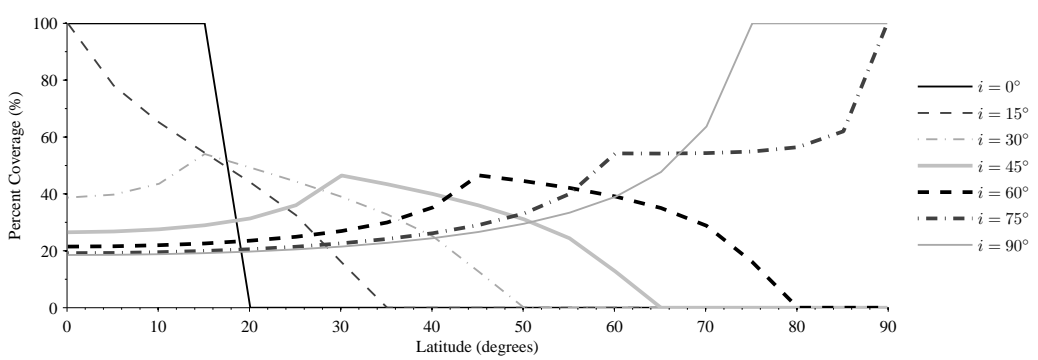


(c) Altitude: 1000 km

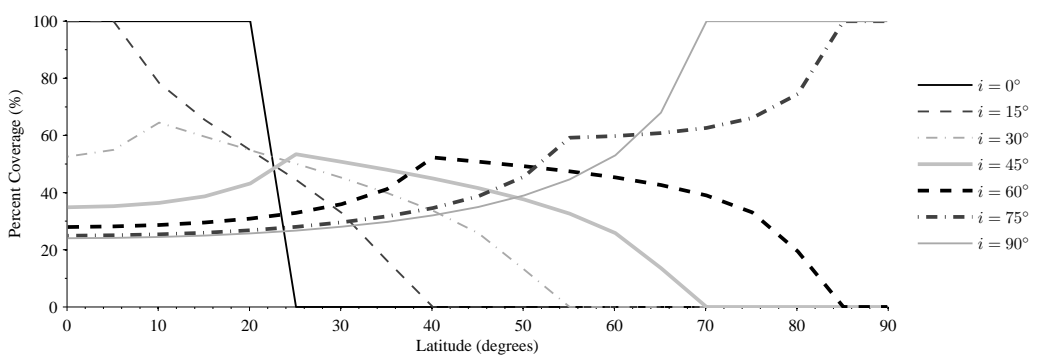
Figure 3.1: Percentage of satellite passes which have ground station coverage for different space node inclinations and ground node latitudes with a ground node minimum communication elevation of 0° .



(a) Altitude: 300 km



(b) Altitude: 650 km



(c) Altitude: 1000 km

Figure 3.2: Percentage of satellite passes which have ground station coverage for different space node inclinations and ground node latitudes with a ground node minimum communication elevation of 10° .

minimum of 60% of the orbits (for a 650 km altitude and minimum elevation of 0°).

Next we extend the investigation of coverage trends to include representative global ground networks using our simulation toolkit. Figures 3.3(a) and 3.3(b) show the global coverage of networks (i.e. defined as the percentage of Earth visible from a ground network) and the global coverage of orbits (i.e. defined as the percentage of Earth covered by satellite ground tracks) for 10° latitude ranges. For the orbits we study, the longitudinal affects average over long-duration scenarios. Note the high coverage of northern latitudes for the N3 network, as seen in Figure 2.4. The coverage for a satellite orbit and network combination is the product of the individual satellite and ground network coverage for a given latitude range. For both equatorial and polar orbits, the satellite spends approximately equal time in the latitude range it covers, which is $[-10, 10]^\circ$ ¹ for equatorial orbits and $[-90, 90]^\circ$ for polar orbits.² Thus, for these cases, coverage is simply the average ground network coverage in Figure 3.3(a) for the appropriate latitude ranges. Coverage results are summarized in Table 3.1 for equatorial and polar orbits and different ground network sizes.

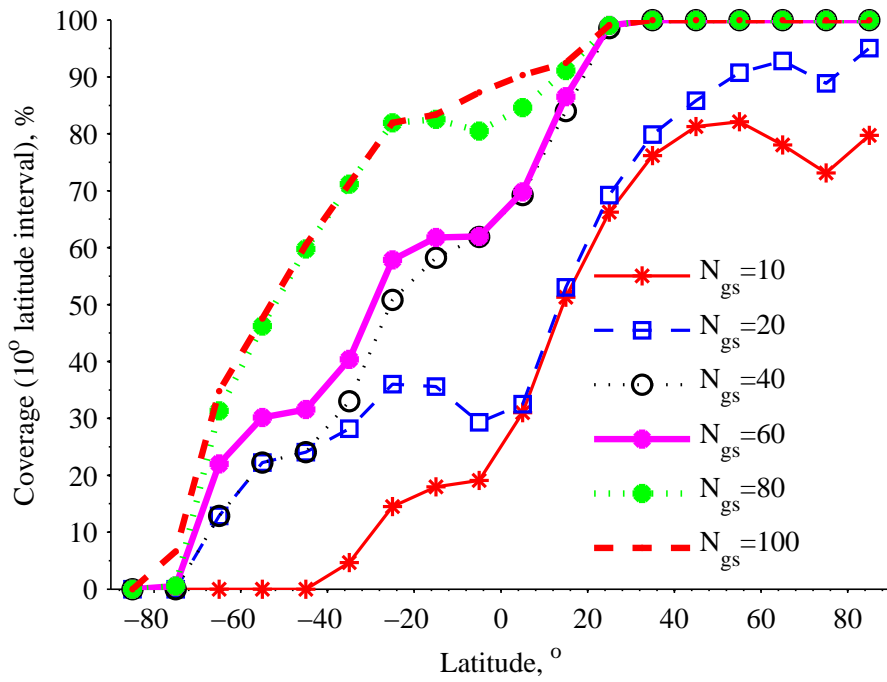
3.1.2 Availability of Download Time

Next we extend the coverage trends from the previous section to assess communication capacity for different combinations of orbits and ground networks. Communication capacity is directly proportional to download time for constant-rate communication. Thus in this section we show results as download time such that they are applicable to scenarios with diverse data rates. Capacity and download time are a direct function of coverage, for example if the ground network covers 50% of the area where the satellite orbits, there will be 12 hours of download time per day, and communication capacity is the product of download time and data rate.

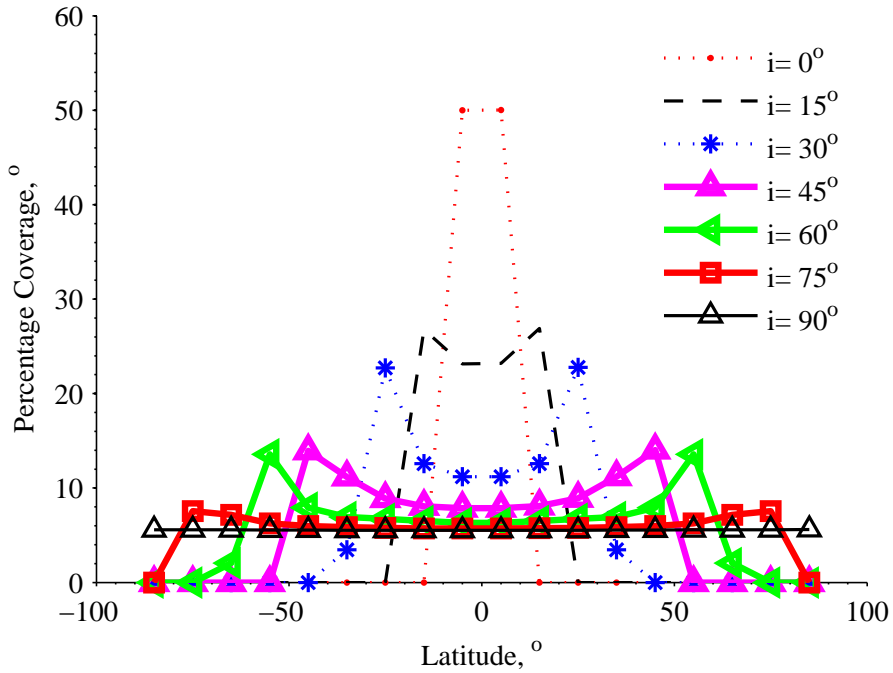
We address communication capacity trends in the context of the model representations in Section 2.2.2.1. In Section 3.1.2.1 we assess factors that impact how geographical constraints impact capacity and differentiate *maximum* and *topographical models*. Section 3.1.2.2 considers how and seasonal variations influence capacity in the context of the *topographical model*. Sections 3.1.2.3 and 3.1.2.4 discuss how multiple-satellite deployments and network utilization enforce constraints on possible download time and impact potential capacity in the context of the *scheduled model*. Finally, Section 3.1.2.5 discusses relevant factors in the *actualized model*.

¹The equatorial orbit has a latitude range slightly higher and lower than 0° due to solar perturbation effects.

²Note this is a function of the size of the latitude ranges selected.



(a) Ground Station Coverage



(b) Satellite Coverage

Figure 3.3: Earth coverage of latitude ranges for different numbers of ground stations (N_{gs}) and satellite inclinations assuming a circular orbit at an altitude of 500 km. See Figure 2.4 for the locations and footprints of the stations.

3.1.2.1 Geographical Constraints

The *maximum model* is used to characterize the ground communication system, quantifying the capabilities of each individual ground node and the summed capacity of all the nodes within the network. This model is used in quantifying the potential capabilities of ground station networks which operate under ideal conditions to aid in future network utilization analysis.

Figure 3.6 shows the geographical locations of the surveyed CubeSat global amateur radio ground stations. Typical communication systems in this network are capable of operating at 9600 bps using frequency shift keying (FSK) modulation. Thus, the network of 98 ground stations has the potential to exchange over 80 Gigabits of data on a daily basis, assuming satellites are constantly available for communication. As another example, the maximum capacity of the full AFSCN with 15 large dishes operating at 100 kbps can transfer 129.6 Gigabits of data on a daily basis.

Having established the basic approximations for access times using the principles of the Two-Body orbit problem, we now consider the higher fidelity SGP4 orbital propagator using STK simulation tools. We study a single communication link between a ground station and satellite and evaluate available access times as it is directly proportional to the capacity expression (Eq. 2.9) within the framework of the *topological model*. The interaction of ground station location and satellite inclination on total access time is shown in Figure 3.4. This plot considers 13 ground stations with longitude of 45° (chosen arbitrarily as the longitude effects average over several orbits) and latitudes distributed at even 15° intervals and 18 satellites with orbital inclinations distributed at even 10° intervals with altitudes of 650 km.

As expected, the distribution in Figure 3.4 is symmetric for ground stations located equidistance from the equator and for symmetric inclinations (i and $90^\circ + i$). This plot can be used as a simple look-up chart to aid in quantifying the average access time for any ground and space node combination. Different orbit altitudes can also be analyzed using these tools.

Satellite communication capacity is influenced by the size and distribution of the ground networks supporting the mission. To explore this effect, the model and simulator are employed to model the N3 network and operation of a DICE satellite with power level P2 (see Tables 2.6 and 2.7). To isolate the impact of ground network on capacity, it is assumed the satellite has sufficient energy to download whenever there is an opportunity. Furthermore, we assume the satellite can communicate with the ground station whenever it is above the horizon, which is representative of many small satellite missions. While we realize this is a simplifying assumption, it provides an estimate of the total communication potential using

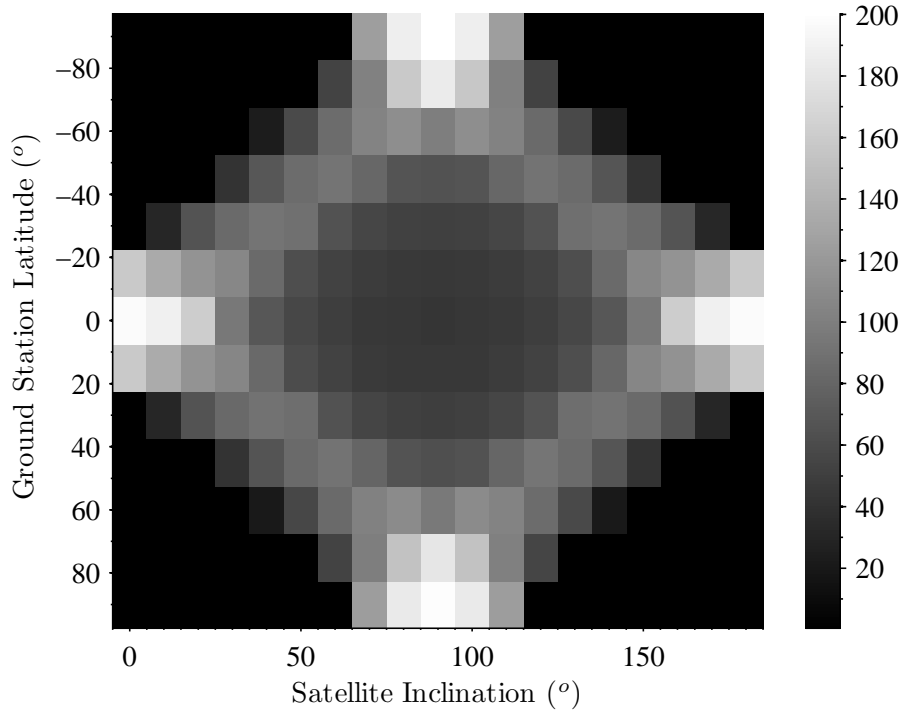
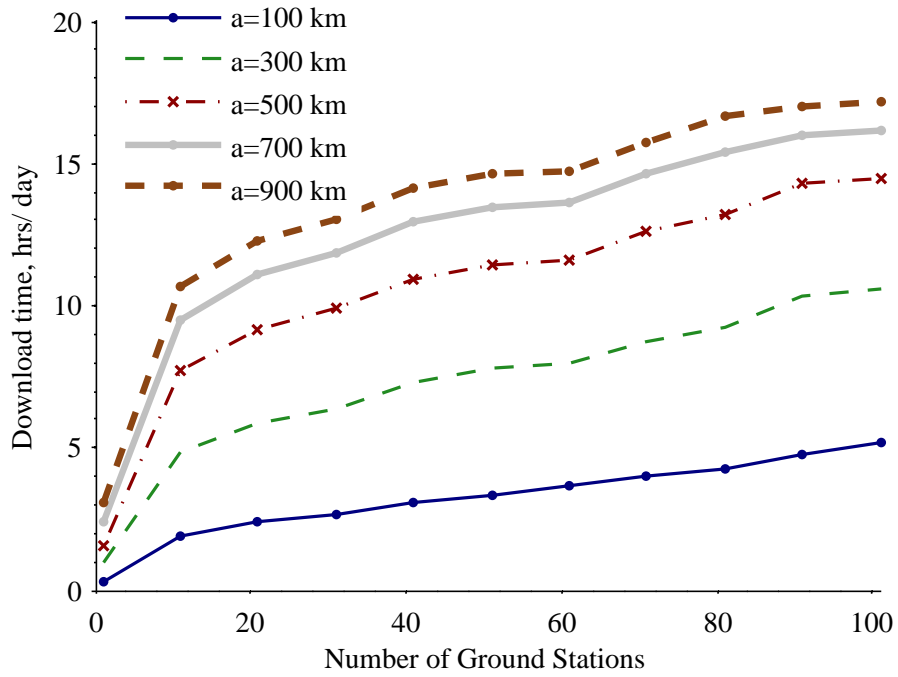


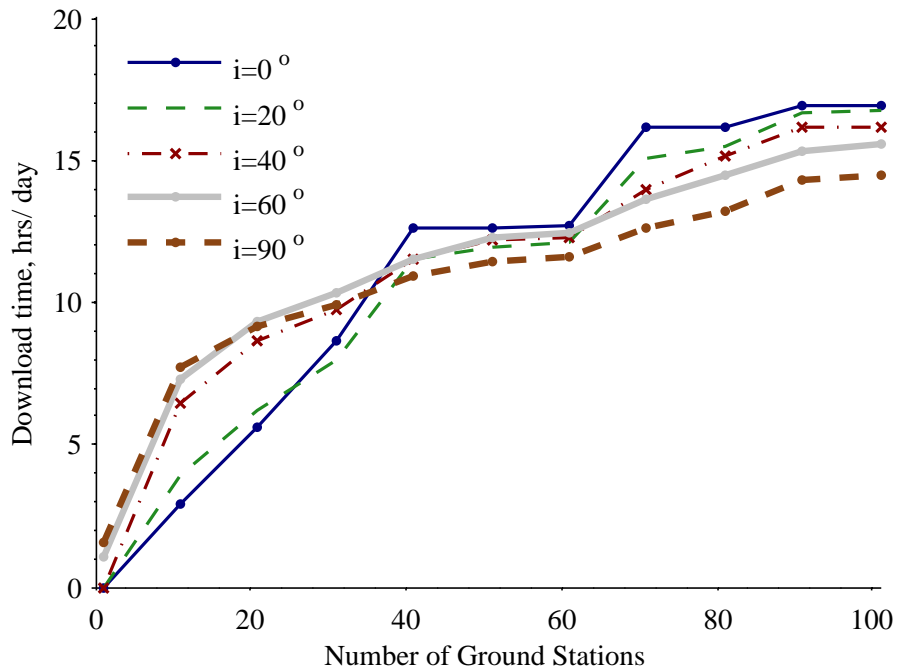
Figure 3.4: Average daily access time as a function of satellite inclination and ground station latitude for 650 km altitude circular orbits using SGP4 propagation method in STK.

a topological model (i.e. considering line-of-sight constraints) [156]. To obtain a more accurate estimate of capacity, the total download can also be multiplied by the fraction of time communication is feasible or the expected efficiency, if available. The simulator can also accommodate a minimum download requirement, as demonstrated in Ref. [144]. Results for different orbital parameters and network sizes are shown in Figure 3.5. Download time is plotted with respect to an increasing number of randomly selected stations from N3.

The combined effect of orbital altitude and network size on download time is demonstrated in Figure 3.5(a) for a polar orbit (i.e. $i = 90^\circ$). Higher altitude orbits have a larger ground footprint resulting in longer pass times and more pass opportunities, and thus greater download time. The addition of ground stations provides a nearly linear growth in communication capacity for low-altitude orbits (when the number of ground stations exceeds ten), while the growth rate does not increase consistently for higher altitude orbits. This is because low-altitude orbits have a smaller footprint on the Earth and therefore distributed ground stations are less likely to have overlapping footprints and be in view of an orbiting satellite simultaneously. Thus, the addition of stations provides improved coverage almost independent of network size for small and mid-sized networks; however this is not true for larger networks.



(a) Altitude Effects ($i = 90^\circ, e = 0$)



(b) Inclination Effects ($\alpha = 500 \text{ km}, e = 0$)

Figure 3.5: Communication capacity as a function of diverse orbital properties and a range of ground network sizes for a one year simulation.

Table 3.1: Satellite coverage for different satellite inclinations and number of ground stations (N_{gs}). These results assume the satellite spends equal time in the latitude range. The simulation is for two weeks using a J_4 orbital propagator.

Orbit Type	Equatorial	Polar
Inclination	0°	90°
Latitude Range	-10 to 10°	-90 to 90°
Coverage for $N_{gs}=10$	25.4%	37.9%
Coverage for $N_{gs}=40$	65.8%	62.1%
Coverage for $N_{gs}=100$	88.9%	75.4%

The combined effect of orbital inclination and network size on communication capacity is demonstrated in Figure 3.5(b). There is a monotonic, yet non-linear, growth in download time with an increase in the number of stations. This non-linearity is due to the complex relationship between satellite ground tracks and ground station coverage of randomly growing networks. For example, a new station near an existing station may not significantly increase capacity due to footprint overlap or if it is out of the latitude range of the satellite orbit.

The growth trend in Figure 3.5(b) can be explained more rigorously by connecting the coverage trends and observations from Section 3.1.1 with the download time results in Figure 3.5(b). For small ground networks ($N_{gs} < 40$ stations), higher inclination orbits have greater coverage, and thus download time, relative to lower inclination orbits. This is because the ground tracks of higher inclination orbits cover greater latitude range and thus have greater download time to globally distributed stations, as in Figure 3.3(b). Lower inclination orbits have a restricted latitude range, and thus are less likely to cover footprints of distributed stations, particularly those at high latitudes. There is an interesting inflection point at $N_{gs} \approx 40$ stations in Figure 3.5(b) and Table 3.1, where the download time of lower inclination orbits begins to exceed higher inclination orbits. This trend occurs because with $N_{gs} \geq 40$ stations, lower inclination orbits are able to exploit the nearly global coverage at low latitudes for larger networks, as shown by the footprints in Figure 2.4(b) and the coverage for low latitudes in Figure 3.3(a). Even with large number of stations, N3 does not cover the full globe, particularly southern latitudes because there are fewer ground stations there. Thus, high inclination orbits that cover a great latitude range achieve lower coverage relative to lower inclination orbits. The simulation environment has enabled an investigation into how global coverage impacts communication capacity trends for diverse networks and orbits.

We now apply our tools to a simple representative example ground station network and satellite mission and investigate the capacity properties. Consider a network consisting



Figure 3.6: CubeSat survey of existing ground stations [1].

Table 3.2: Geographical locations of sample AFSCN ground stations

Ground Station	Location	Latitude	Longitude	Latitude Category
Guam Tracking Station (GTS)	Anderson AFB, Guam	13.6°N	144.8°E	Low
New Hampshire Station (NHS)	New Boston AFS, NH	42.9°N	71.6°W	Mid
Thule Tracking Station (TTS)	Thule AB, Greenland	76.5°N	68.6°W	High

of three ground stations at different geographical locations in the AFSCN and a single satellite. Station locations are found in Table 3.2 and represent high, mid, and low latitude locations. We select a single representative satellite from the April 2007 Dnepr launch vehicle, AeroCube 3, deployed into an orbit with an inclination of 99° and an altitude of approximately 715 km.

Figure 3.7 shows the effect of ground station location, particularly latitude, on network capacity relative to the highly inclined satellite orbit. Average daily access time for the CubeSat is only 50 min/day relative to the low 13.6°N latitude ground station (GTS), and over three times larger (180 min/day) to the 76.5°N latitude station (TTS). The average access times are similar to the average values in Figure 3.4 for a 650 km orbit for the various ground and space node combinations.

The relationships presented in this section (Figures 3.1-3.4) can be used to determine average access times between unique space and ground node combinations and are useful in extracting trends with changes in inclination and latitude. In addition, the optimal location

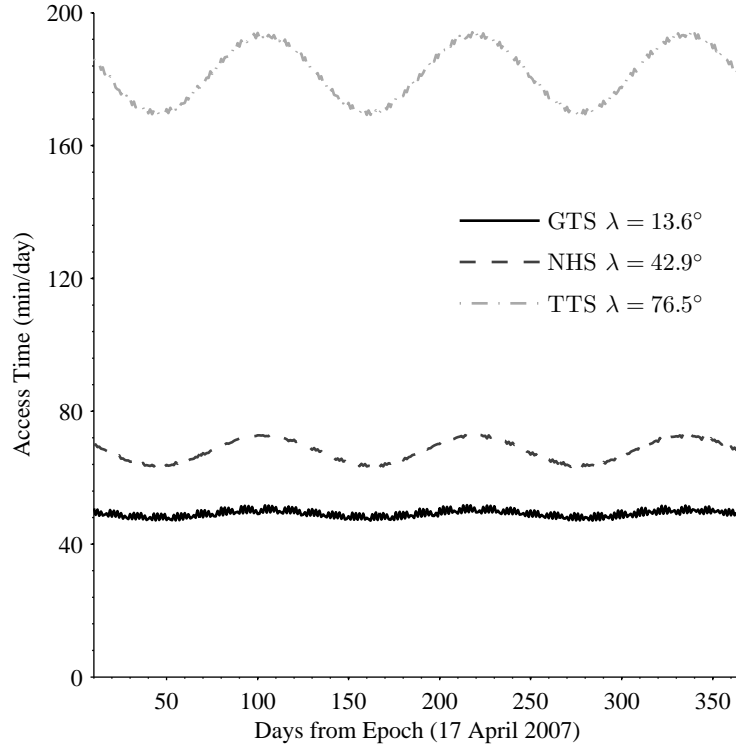


Figure 3.7: Effects of variation in AFSCN ground station latitudes from Table 3.2 on network capacity for the AeroCube 3 satellite with 99° inclination and 715 km altitude orbit.

of ground nodes for maximizing communication with a satellite constellation with orbits at unknown or distributed inclinations and/or altitudes may be selected using the plots and equations presented.

3.1.2.2 Seasonal Variations

Variations in satellite pass occurrence and duration due to orbital dynamics can have dramatic effects on the availability of communication links in the context of the *topological model*. The capacity distribution in Figure 3.7 shows oscillations with access time variations of approximately 15%, a very significant trend for constrained satellite communication applications. Missions are often designed for the worst-case scenario to guarantee that mission objectives will be satisfied, thereby sacrificing these potential communication advantages. From our examples, it is obvious that considerable improvements in network capacity may be realized with accommodation for these naturally occurring seasonal variations.

Oscillations in access times are largely determined by the interaction of the rotational rates of the satellite and Earth. As seen in Figure 3.7, distinct periods in the access time

variations emerge, which are explained by the Earth’s nominal rotation, perturbations due to Earth’s oblateness, characterized by the J_2 gravity coefficient [167], and the satellite inclination. The length of a communication pass is directly related to the maximum elevation angle from Eq. 3.1 [166],

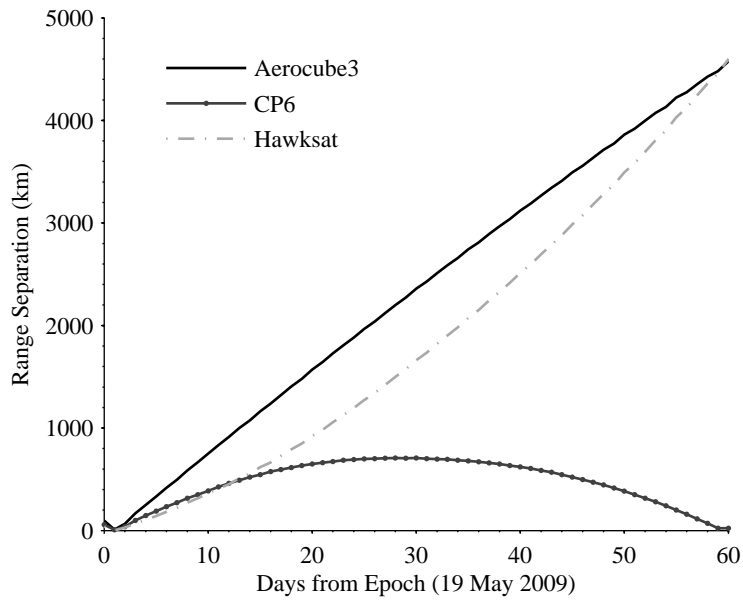
$$\theta_{max} = \cos^{-1} \left(\frac{r_E \sin \gamma}{\sqrt{(r_E^2 + r^2 - 2r_E r \cos \gamma)}} \right). \quad (3.2)$$

The Earth central angle, γ , is the angle measured at the center of the Earth from the subsatellite point to the ground station [61]. The *subsatellite point* is the intersection point of a straight line drawn from the satellite to the centre of the Earth and the Earth’s surface. These analytic relationships are essential in understanding the variation in access time. The maximum elevation angle, θ_{max} , controls the duration of each pass and varies with successive passes over a given ground station as a function of the subsatellite position. It is easily seen from Eq. 3.2 that a ground station which always lies in the plane of a satellite’s orbit has very little or no variation in access times between orbits (since γ will be zero at the instant of maximum elevation, and θ_{max} will be constant for all time). Consider two scenarios: an equatorial ground station and satellite with zero inclination or a polar ground station and satellite pair. Both of these cases will have no variation in maximum elevation angle and thus the pass duration remains constant for each successive pass.

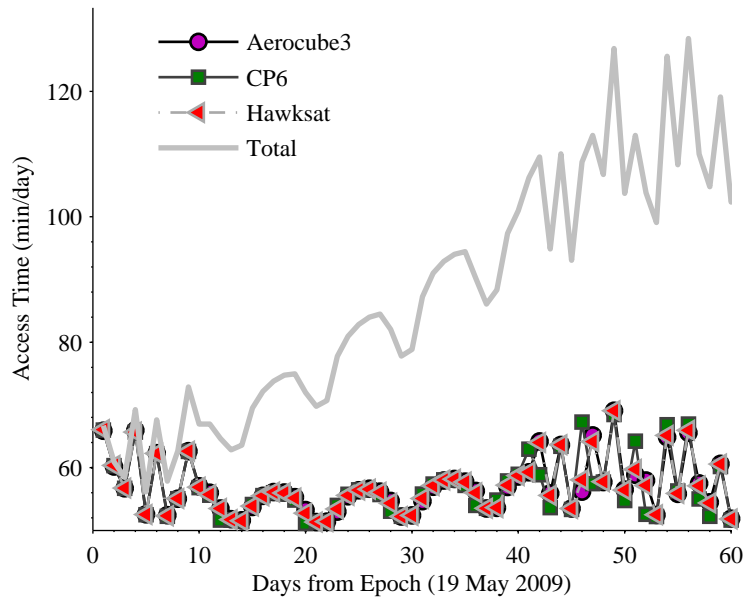
For those ground stations which do not lie directly in the orbital plane of the satellite of interest, we are interested in the frequency and amplitude of access times variability over successive passes. Those passes with the greatest variation in length between extrema are of most interest as they offer the potential for the largest advantage or disadvantage in capacity relative to the averages. Due to the problem complexity, the most direct way to evaluate access times and capture these seasonal variations is by using high fidelity numeric propagators such as those introduced earlier. As many of the trends are periodic in nature, with repeated periods due to the interaction of the rotational rates, we are developing approximate analytic and probabilistic models to characterize the variations [27].

3.1.2.3 Deployment Effects

Next we introduce a small-scale network with multiple satellites communicating to a single ground station, and realistic ground station scheduling constraints are now considered, thus it falls within the framework of the *scheduled model*. The main scheduling limitation



(a) Separation distance between pairs of spacecraft.



(b) Individual satellite and total network access time.

Figure 3.8: 2009 Minotaur-1 launched CubeSat group relative to Ann Arbor ground station ($42.27^{\circ}N$, $83.76^{\circ}W$) following epoch, the time the satellite emerges from the launch vehicle.

considered in our examples is the link constraint, where each ground station can communicate only with a single satellite at a given time instant. The actual network capacity will be less than or equal to the summed total available times a line-of-sight exists between all of the nodes within the network. This reduction in total network capacity is due to line-of-sight overlaps, when two satellites are in view of the same ground station, but only one communication link is feasible.

As an example, consider the representative small satellite family of three CubeSats deployed from the May 19, 2009 Minotaur-I launch at an altitude of approximately 460 km and inclination of 40.5° communicating to a single mid-latitude ground station located in Ann Arbor, MI (42.28°N , -83.74°W). The separation distances of the spacecraft and total network capacity of the deployment are shown in Figures 3.8(a) and 3.8(b), respectively. Satellite position information was obtained from online Keplerian element sets [163]. A particularly interesting feature of these satellites is their clustered deployment as they were launched simultaneously from a single launch vehicle interface [168] and separated following epoch (*i.e.* the time the satellite emerges from the launch vehicle). Note the dynamic distances between satellite pairs that results from orbital perturbations, initial separation velocity, and different drag coefficients. Within a month, the distances between the satellites have grown significantly, with a separation of nearly 5000 km between two of the satellite pairs. Note the Aerocube3 and Hawksat satellites initially separate to a maximum distance after 1 month and then re-approach one another, a function of the characteristics discussed above.

The effects of separation distance shown in Figure 3.8(a) largely influence the capacity shown in Figure 3.8(b) for the satellites deployed from the launch vehicle. The dotted lines are the total contact times between each unique satellite and ground station pair. The solid line represents the total daily capacity from the three satellites launched from the same Poly Picosatellite Orbital Deployer (P-POD). The total capacity is initially equivalent to the capacity of a single satellite (approximately 60 min/day), a function of the satellite's orbit and Ann Arbor's geographic location. The network capacity grows to an average of 115 min/day after two months and the satellites separate on-orbit, double the initial capacity for the clustered launch. The major influence of the additional scheduling constraints within the *scheduled model* is a reduction in feasible downlinking opportunities due to contact overlaps. With reduced windows of communication, it may be optimal to transmit at higher data rate and the resultant increased power (to meet the minimum SNRs) for a shorter duration. This trade-off may be considered both at design time and for tactical mission planning, enabling the mission to exploit the available access times.

3.1.2.4 Network Utilization

It is clear that larger networks provide greater opportunities for communication; however capacity does not scale linearly with the size of the network. For example, consider the global AFSCN, consisting of eight remote tracking stations with 15 antennas [169]. We calculate the access time of the three CubeSats launched on May 19, 2009 on a Minotaur-1 relative to the full AFSCN network. The plot in Figure 3.9 shows the increase in capacity for the clustered satellite deployment following epoch. After two months, the average daily capacity increases from approximately 830 min/day to a peak of 1130 min/day. The total network capacity decreases as the Aerocube3 and CP-6 re-approach one another, as seen in Figure 3.8(a) and discussed in the previous section. The obvious advantages of a combined network of satellites and the distributed ground station network is exemplified here, where a nearly tenfold improvement in total capacity occurs relative to the scenario with a single ground station. The improvement in capacity does not scale directly with the size of the ground station network due to the overlap effects, where multiple spacecraft are in view of a ground station only capable of single link communication.

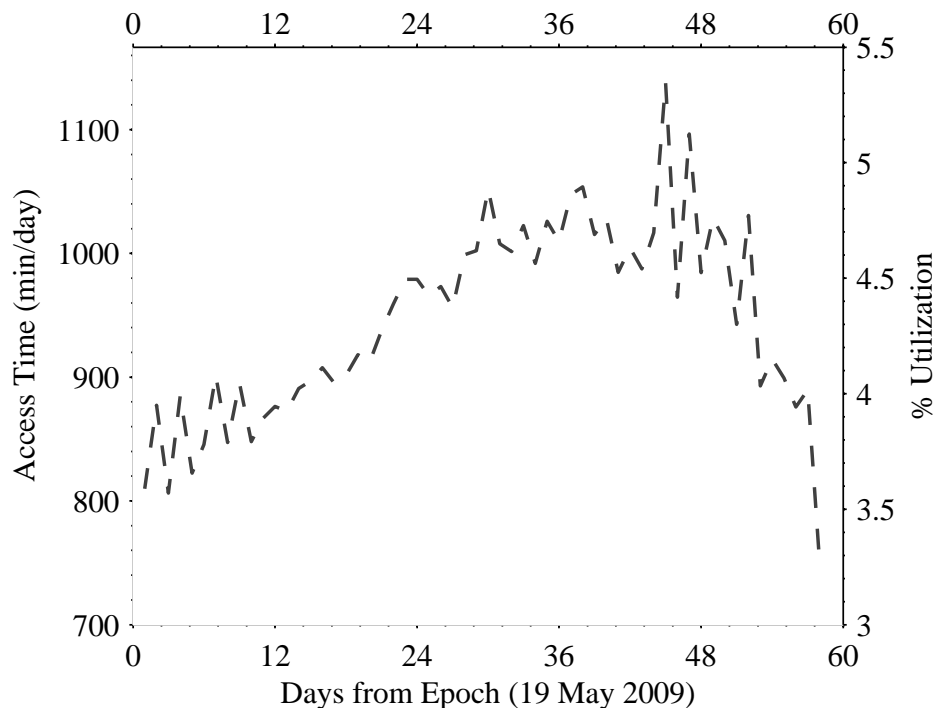


Figure 3.9: Total network capacity for 2009 Minotaur-1 launched CubeSat group relative to entire AFSCN with 15 antennas.

Demand for ground station capacity is increasing. Scientists are deploying satellite

constellations to collect scientific data over larger distances and longer time scales to avoid space and time aliasing. Single spacecraft provide only glimpses of space physics, and science is often patchy in space and bursty in time [170, 171]. Distributed constellations of sensors facilitate simultaneous multi-point remote sensing, which can revolutionize how space science missions are performed. Constellations of small satellites can be used to provide long-lived, systematic, multi-probe space weather data to study plasma or neutral density, temperature, composition, velocity, electric field, other fundamental geophysical parameters [172]. Small satellites can be used to explore and the data collected from their missions used to explain new phenomenon, and then initiate, constrain, and validate models and take predictive scientific algorithms to the next level. For example, the Naval Postgraduate School (NPS) has proposed a clustered launch of fifty science nanosatellites and the QB50 Project plans to set up and coordinate an international on-orbit CubeSat science network [173].

With growing satellite networks, there is increasing occurrence of overlap times, where multiple satellites are overhead a single ground station. Since each ground station can communicate only to a single satellite at a time, overlaps result in a significant reduction in communication utilization of total available line-of-sight opportunities. To simulate the growing small satellite population resulting from upcoming launches, consider a family of distributed satellites launched into LEO (distributed at altitudes from 400 to 800 km), relative to the full 15 antennas in the AFSCN. Figure 3.10 shows that as increasing numbers of CubeSats are injected into the network the ground station utilization increases until a saturation point is reached, where additional satellites can no longer be supported. Figure 3.11 shows excess access time, defined as the time where available links are not utilized due to the capacity constraints of the ground station network. The growth of excess time is exponential since the addition of satellites into a ground station network which is nearly or already at capacity rapidly reduces overall network utilization. It is clear that intelligent deployment methods and scheduling between ground stations and satellites is critical to maximize the network capacity of these growing networks.

3.1.2.5 Actualized Networks

The final model of highest fidelity, the *actualized model*, considers all the inefficiencies and constraints of the network, both from the ground station and satellite perspectives. In particular, the inefficiency discussion from Section 4 and the satellite energy and data constraints are considered as important elements in the problem. We are deploying monitoring systems in a growing number of ground stations to collect this data in real time for our ground station operations through deployment of the Mercury and GENSO networks [174] [175].

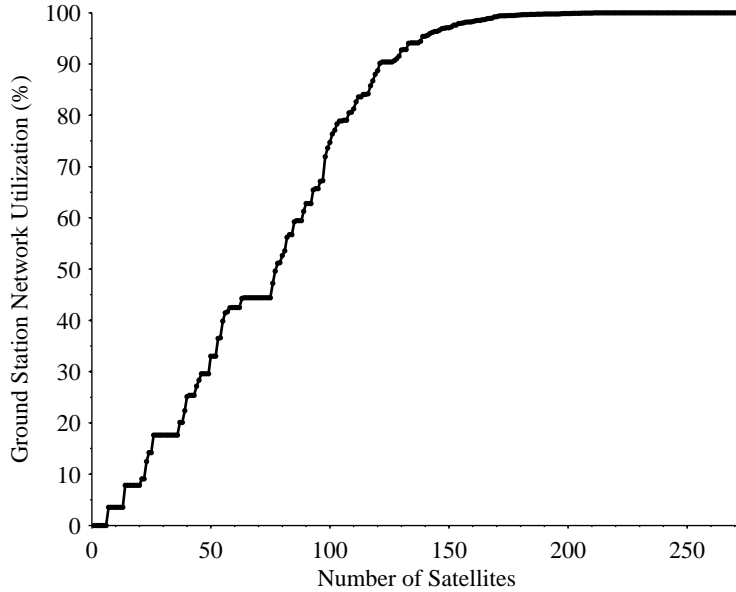


Figure 3.10: Effects of growing family of satellites on network utilization for a fixed ground station network. The CubeSats are launched into 400 to 800 km random orbits, and access time is computed relative to the 15 ground station antennas in the AFSCN. Excess access time is the time where available links are not utilized due to the capacity constraints of the ground station network.

Ref. [155] introduces a communication efficiency term (T_{eff}) representing the actual time a communication link is maintained relative to the total access time a satellite is in view of a ground station. Experimental data was collected for different maximum satellite elevation passes for low Earth orbits (LEO) satellites communicating to a Vienna ground station in January 2004. The reported efficiency term was just below 80% for a max elevation of $\theta_{max} = 10^\circ$, and increased nearly linearly until $T_{eff} = 97\%$ for $\theta_{max} = 90^\circ$. Based on minimum SNR requirements, conventionally small spacecraft communication links are established at a minimum elevation of $\theta_{min} \geq 10^\circ$, therefore we can model efficiencies using the values reported in this work. We are currently focusing on the ground station perspective, therefore we assume the satellite is an ideal node and always available for communication. Higher fidelity satellite models we are developing consider satellite operational constraints due to power and communication system inefficiencies, which will further decrease our network capacity and limit the solution space [176]. Download efficiencies for representative CubeSat missions are discussed in more detail in the context of stochastic optimization methods in Chapter 6.

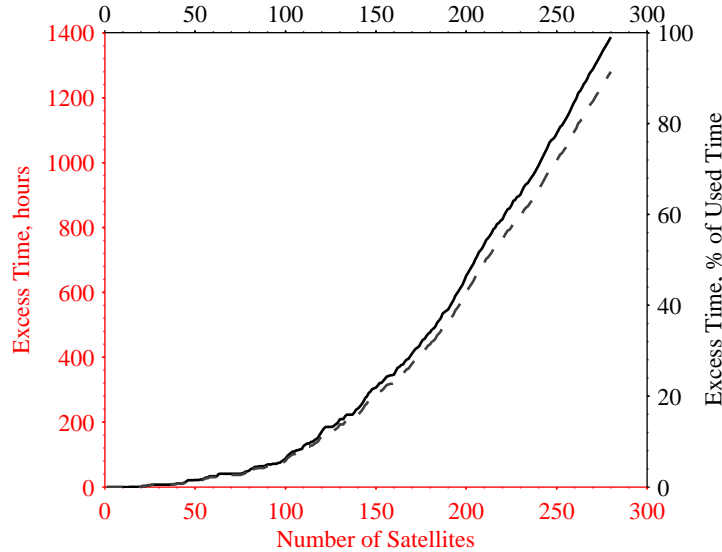


Figure 3.11: Effects of growing family of satellites on excess access time for a fixed ground station network. The scenario is identical to the one shown in Figure 3.10. Excess access time is the time where available links are not utilized due to the capacity constraints of the ground station network. The solid line represents the excess time in hours and the dashed line represents the excess time in % of used time.

3.2 Energy Constraints

This section investigates the set of constraints related to availability of energy. We focus on the energy collection potential, which is a function of eclipse time relative to orbit period.

Orbit parameters affect opportunities for power collection, which directly impacts the ability of the spacecraft to collect and download data. Since many small satellites are launched as secondary payloads, they have little control over orbit selection, and thus the orbital parameters of past and upcoming launches can vary greatly [157]. Motivated by this variability, the availability of energy for download from diverse low Earth orbits (LEO) is assessed.

The annual variation in eclipse for a satellite is highly dependent on orbit inclination. Figure 3.12(b) plots the eclipse fraction, which is the fraction of time the satellite is in shadow per orbit to the orbital period, for varying inclinations. The data for the plot was generated with a right of ascension, $\Omega_0 = 0^\circ$, an eccentricity, $e = 0$, and an altitude of 500 km. The eclipse fraction varies from nearly constant for equatorial orbits ($i = 0^\circ$) to a distinct bi-annual variation for polar orbits ($i = 90^\circ$). This variation is due to orbit precession; inclined orbits precess due to Earth’s non-uniform gravity field [61]. The sharp spikes at days 141 and 317 are due to the Moon’s penumbra.

There is also an annual variation in eclipse fraction as a function of altitude as plotted in Figure 3.12(a) where altitude ranges from 100 to 1000 km with a fixed inclination of 90° . As altitude increases, eclipse time decreases from 38 to 35 minutes, while orbital period increases from 86 to 105 minutes. Due to these combined effects, higher orbits have shorter relative eclipse fractions and greater seasonal variation.

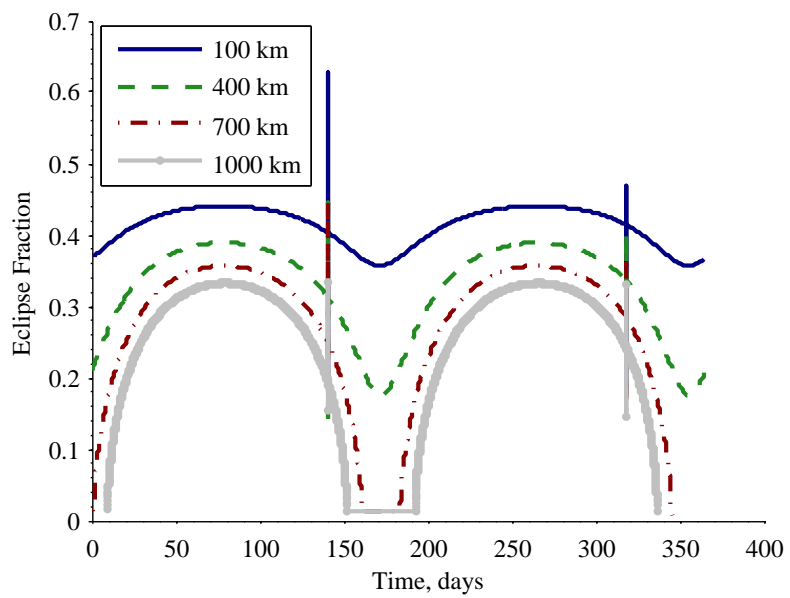
Additional trends are highlighted in Figure 3.13 where eclipse trends are plotted as contour lines for varying inclination, altitude, and right ascension of the ascending node of a circular orbit with simulation results run for one year. Eclipse trends are similar for $\Omega_0 = 0^\circ$ and $\Omega_0 = 180^\circ$ and for $\Omega_0 = 90^\circ$ and $\Omega_0 = 270^\circ$, so only two are plotted. The solid line at $i = 90^\circ$ marks the boundary between prograde and retrograde orbits. Variation occurs because Ω_0 describes the orbit plane orientation with respect to the vernal equinox, and as the inclination increases towards sun-synchronous, it varies the orbit between terminator (full sun) or noon-midnight (maximum eclipse) [61]. For near equatorial to mid-inclination orbits, corresponding prograde and retrograde orbits (i.e. orbits that share the same orbital plane such as $i = 30^\circ$ and $i = 150^\circ$) have similar eclipse trends. Near-polar retrograde orbits have lower average eclipse durations relative to corresponding prograde orbits that share the same orbital plane.

These simulations demonstrate model and toolkit utility in quantifying the sensitivity of eclipse fraction, and thus power collection, relative to orbital parameters. For operational planning of an active mission, these simulations highlight times during the year when excess energy is available for additional payload operations or higher-power downlink, or when the satellite must conserve energy to meet minimum requirements. These simulations are also useful during vehicle design since eclipse trends are important for the sizing and optimization of energy storage, energy collection, and thermal systems.

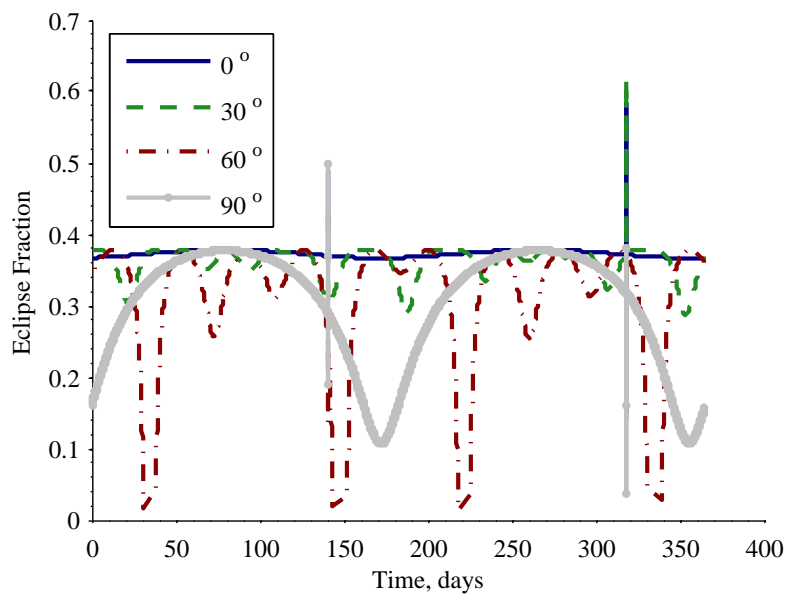
3.3 Comparison to Download Requirements

This section compares network and energy constraints to mission-specific download requirements, such as those in Table 2.6. Recall that network constraints are those that are exclusively a function of the download time between satellites and ground networks and energy constraints are those that are exclusively representing the total available energy for downloads.

The required and desired download capacities are compared to the energy and network constraint-based capacity in Figure 3.14 for the three missions from Table 2.6. The constraint-based capacity is plotted along the y-axis for the three missions that are shown along the x-axis. Horizontal lines represent the required and desired download and ver-

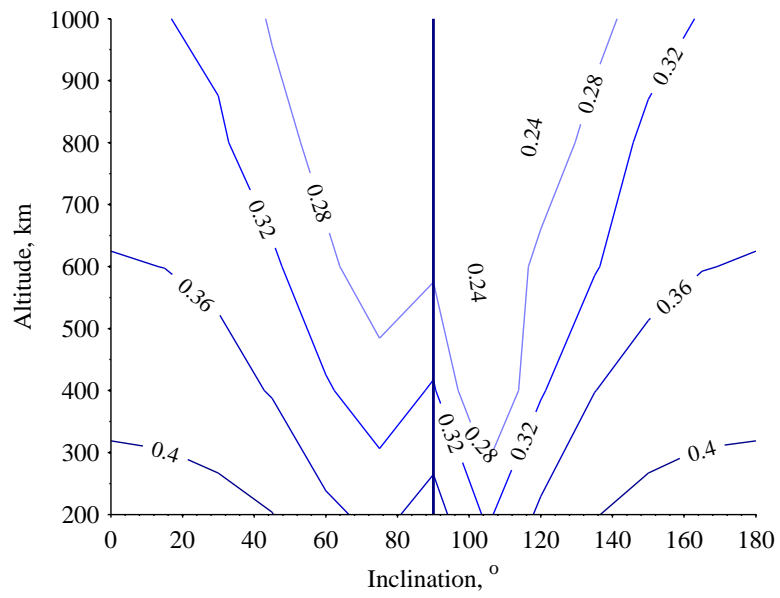


(a) Variable Altitude ($i = 90^\circ, e = 0$)

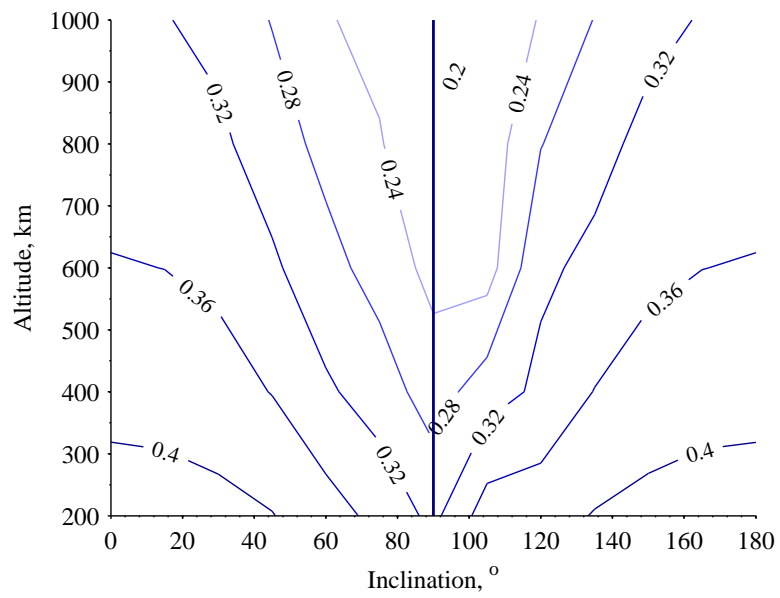


(b) Variable Inclination ($a = 500 \text{ km}, e = 0$)

Figure 3.12: Eclipse fractions for circular orbits with variable altitudes and inclinations for a one year scenario.



(a) $\Omega_0 = 0^\circ$



(b) $\Omega_0 = 90^\circ$

Figure 3.13: Eclipse fraction for circular orbits with variable orbital inclinations, altitudes, and right ascension of the ascending node.

tical lines represent the energy and network constraint-based capacity. In the constraint evaluation we assume that data is downloaded at a constant rate.

Considering energy constraints, each mission in Figure 3.14 achieves its download requirement with any of the three power scenarios (P1, P2, P3), and the RAX-1 and DICE scenarios also achieve the desired download using any of the power scenarios. These results provide snapshots of capacity, as they capture only a short planning horizon (two weeks) and eclipse trends vary throughout a year. Specifically, the satellites are all in near-polar orbits with mid LEO altitudes; thus, as shown earlier in Figure 3.12, the eclipse trends for these orbits vary significantly (from no eclipse time to $> 30\%$ of the orbit) throughout a year. Energy constraint results are informative for identifying constraint-based download capacity as a function of eclipse trends. They can also be useful for sizing solar panels to satisfy mission requirements during early design stages.

Considering network constraints, all three missions in Figure 3.14 achieve their respective minimum download requirements by communicating to either of the two larger networks (N2 or N3). Download potential increases by up to two orders of magnitude when the network grows from two ground stations in N1 to eight ground stations in N2, consistent with capacity growth trends for near-polar orbits and small networks shown earlier (see Figure 3.5(b)). The download capacity of RAX-1 and DICE communicating to N3 increases by over two orders of magnitude relative to communicating to N2, despite transmitting at a lower data rate (see Figures 3.14(a) and 3.14(b)). This is because the relative increase in download time exceeds the relative decrease in data rate. The myPQ mission achieves its desired download capacity only when communicating to the largest network (N3), see Figure 3.14(c). These simulation results demonstrate how the constraint-based analysis can quantify the ability of different networks to achieve different goals.

The results also highlight that network constraints are the limiting factor when a small network is used, and energy constraints are the limiting factor when larger networks are used for the scenarios studied in this section. The relative advantages of larger networks and/or higher power levels on the capacity have been quantified in this analysis and can enable the optimization of vehicle and network designs. This type of analysis enables the identification of mission scenarios that are infeasible and those with the potential to exploit surplus resources to reach and exceed desired download requirements.

3.4 Summary

This chapter has demonstrated the utility of the model and simulation framework to identify network and energy constraints, and how these constraints limit communication capacity.

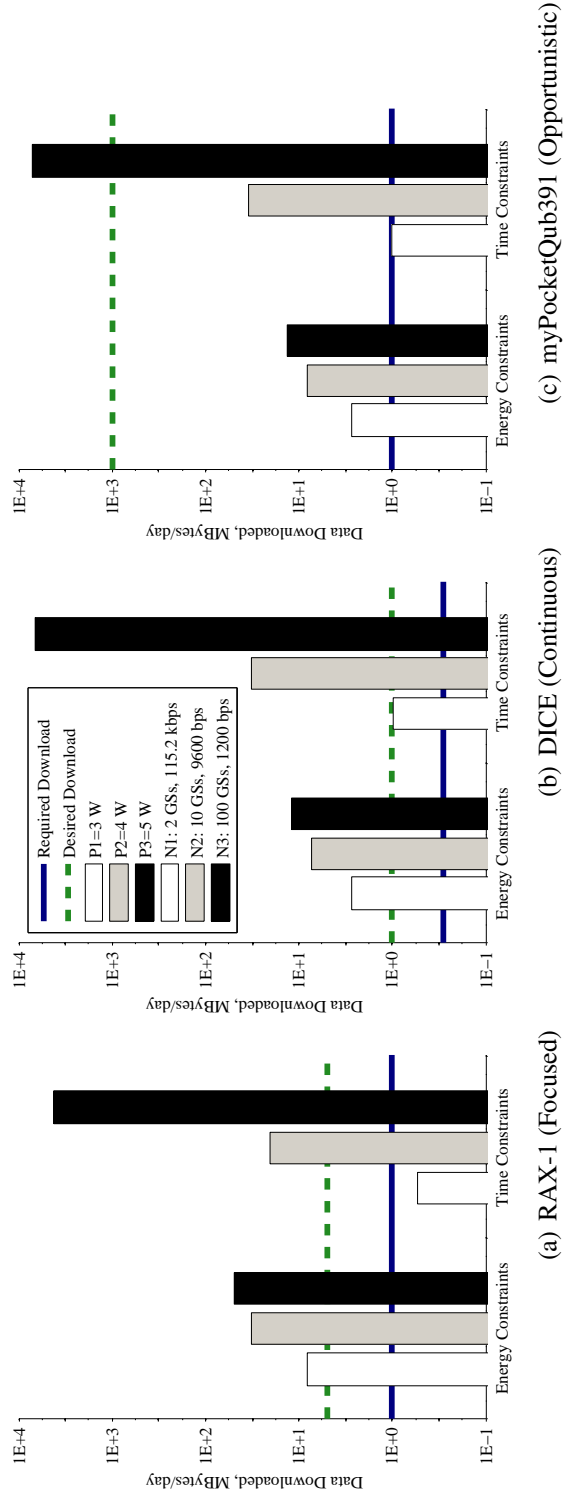


Figure 3.14: Communication capacity constraints compared to mission requirements and desired download for realistic satellites with diverse operational modes from Table 2.6. The simulation results are averaged for a two week period beginning.

This constraint information is useful for verifying the feasibility of satellite vehicle and network parameters relative to mission requirements. In addition, understanding these active constraints is useful for operational scheduling (i.e. making scheduling decisions for the operation of a constrained system) and designing satellite subsystems and ground station networks. For example, these constraints provide information on how to exploit excess energetic or ground resources in spacecraft scheduling to optimize communication capacity.

The examples in this chapter have shown the limitations due to ground network visibility constraints and energy availability for different spacecraft and networks when assessing communication capacity. In addition, the results have demonstrated the dynamic nature of these constraints. For example, download times vary as a function of the orbital track, and eclipse trends vary throughout the year. Consideration of the dynamic nature of these constraints is important for both short and long-term tactical planning.

This chapter has studied the characteristics of the constraints individually; however we require formulations and algorithms that consider all constraints simultaneously to develop realistic and implementable schedules. Operational scheduling is necessary to ensure constrained resources are optimally allocated throughout the planning horizon to ensure payload and download requirements are satisfied and mission goals are maximized, which is the topic of Chapters 4-5.

CHAPTER 4

Deterministic Optimization: Formulation and Results

In this chapter, we develop models and algorithms for solving the *Single-Satellite Multiple-Ground Station Scheduling Problem (SMSP)*, with the objective of maximizing the total amount of data downloaded from a single spacecraft. This work is motivated by the identification of active constraints in Chapters 2 and 3 and the need to optimally allocate constrained resources towards our goal of maximizing communication capacity.

Conventional approaches to satellite operations often use greedy scheduling, where the satellite performs mission operations and downloads data at every feasible opportunity. This approach may be sub-optimal since in some cases it may be preferable to store resources for a future time period with more beneficial download options. Thus, in this chapter, we aim to develop operational schedules that guarantee optimality (and thus feasibility) and overcome the drawbacks associated with greedy approaches. In particular, our goal is to develop algorithms that yield high-quality schedules in a timely fashion while accurately modeling on-board satellite energy and data dynamics as well as realistic constraints of the space environment and ground network. The optimization formulation in this chapter is derived from the *communication-focused* model in Section 2.2 based on the modeling framework from Section 2.1. The data sets and simulation environment described in Section 2.3 are used to generate and execute example problem instances. This chapter focuses on optimizing the schedule, which was described by $U(t)$ in Chapter 2, given opportunities on when we can collect solar energy, data, and download data to ground stations, as discussed in Chapter 3.

The contributions of this chapter are three-fold. First, we develop models and algorithms that enable us to solve real-world instances of *SMSP* in acceptable run times. Second, our research lays the foundation for solving more complex satellite scheduling problems, such as those in which there is stochasticity in the system (e.g. uncertainty as to the availability of the ground stations) as well as problems in which there are multiple satellites

competing for the same ground station resources. Third, the theoretical insights gained in this research have relevance for many other applications in which tasks must be scheduled subject to resource acquisition, storage, and utilization constraints.

The remainder of the chapter is outlined as follows. In Section 4.1, we formally state the problem and present a continuous-time model to demonstrate the physical dynamics of the system. In general, the associated non-linear optimization problem is not tractable except for limited special cases. We therefore develop a discretized MIP formulation in Section 4.2, which under-constrains the problem. A special case in which this model is guaranteed to yield optimal solutions to the original problem is discussed in Section 4.3, with corresponding computational results provided. For the more general case, we present an iterative algorithm in Section 4.4 that progressively tightens the constraints to converge to a feasible and thus optimal solution. Computational experiments are presented in this section as well, to demonstrate tractability and investigate problem structure. We summarize this chapter in Section 4.5.

4.1 Problem Description and System Dynamics

The goal of *SMSP* is to maximize the amount of data downloaded to a network of ground stations from a single spacecraft orbiting the Earth. In this problem, we assume the following:

- A single satellite is orbiting Earth, collecting both data (via on-board instruments) and energy (via solar panels) along its orbit.
- The satellite's data and energy acquisition rates may vary over time. For example, the collection of data depends on whether the spacecraft is in view of a target of interest (e.g. a science or surveillance target) and the collection of energy depends on the line of sight of the solar panels relative to the sun.
- Energy is required to conduct basic operational functions of the spacecraft and download data.
- The satellite has finite limits on the amount of data and energy that can be stored at any given time.
- We assume the spacecraft orbit is deterministic and known such that the access times to the globally distributed ground stations are known a priori.

- There are multiple ground stations which the satellite can download to, each of which periodically comes into view of the satellite, see Figure 4.2.
- More than one ground station may be in view of the satellite simultaneously, but the satellite can download only to one ground station and one data rate at a time.
- Ground stations may vary in their characteristics, both with respect to the rate with which they may receive data (bits-per-second) and the energy utilization from the satellite required to do so (Joules-per-bit). Ground stations also vary in the *efficiency* of the data download (i.e. the fraction of transmitted data that is actually captured by the ground station, where the losses are due to communication system and transmission inefficiencies [156]). Furthermore, each ground station may have multiple download *options* from which to choose, where an option is defined by a combination of values for data rate, energy utilization, and efficiency.

A *schedule* defines the time periods during which downloads are to take place and, for each of these time periods, which option is to be used. The objective is to maximize the total amount of data collected, subject to the constraints of the system. We assume the spacecraft orbit is deterministic and known such that the access times to the globally distributed ground stations are known a priori.

The mission scheduling horizon is the time for which we optimize the download schedule, $[0, T]$. Within this horizon, we define a set of *intervals*, I . Whenever a ground station comes in or out of view of the spacecraft, a new interval starts. The intervals may be of variable duration, however we define that within an interval the set of ground stations in view remains constant. Intervals in which there are one or more ground stations in view of the spacecraft are download *opportunities*. See Figure 4.2 for an example.

$O(t)$ denotes the set of download options at time $t \in [0, T]$. Each download option, $o(t)$, is defined by the data download rate, $\phi_{o(t)}$ (bits per second), the energy utilization, $\alpha_{o(t)}$ (Joules per bit), and the data download efficiency, $\eta_{o(t)}$ (fraction of transmitted data that is successfully received by the ground station). The no-download option, with $\{\phi_{o(t)}, \alpha_{o(t)}, \eta_{o(t)}\} = \{0, 0, 0\}$, is also an option at every instance. When there is not a download opportunity, the no-download option is the only available one. Since the spacecraft can download only to one ground station and single rate at a time, it can download only using a single option during every interval.

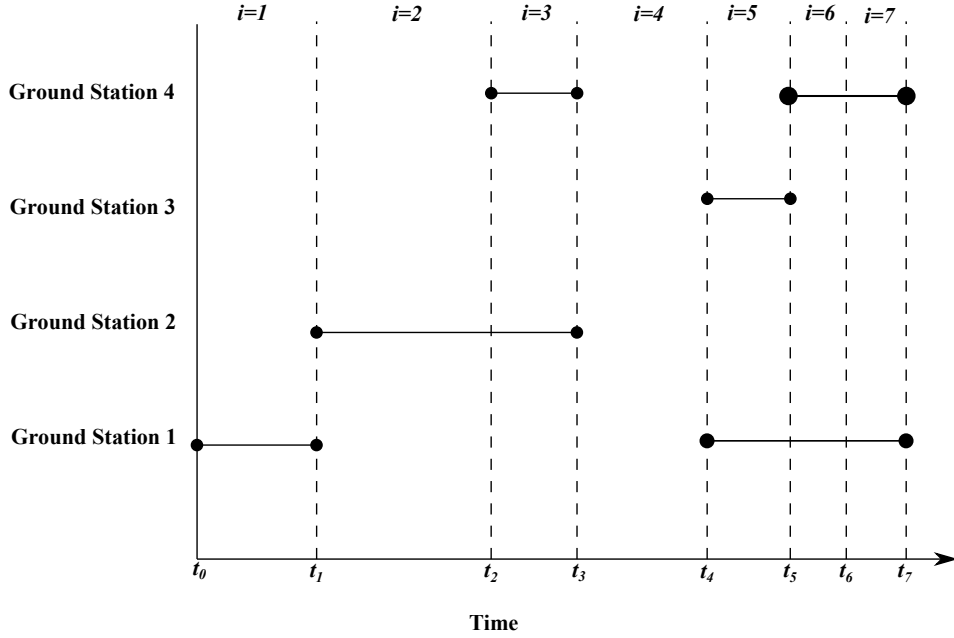


Figure 4.1: Horizontal lines indicate when a ground station is in view of the spacecraft as a function of time. Vertical lines indicate the boundaries between intervals.

4.1.1 Energy Dynamics

Spacecraft collect solar energy by solar panels and store energy in a battery as they orbit the Earth. Energy is consumed to support the basic functionality of the spacecraft and to download data. The upper and lower bounds on the amount of energy that can be stored in the spacecraft's battery are denoted e_{\min} and e_{\max} , respectively.

Let $e(0)$ be the initial energy level at time $t = 0$, and $o(t)$ be the option at time t associated with a given feasible data download schedule. We can then compute $e(t)$, the energy stored in the battery at time t , recursively in the following way:

$$e(t) = \min \left[\begin{array}{c} e(t - \tau) + [\pi^{e^+}(t - \tau) - \pi^{e^-}(t - \tau) - \alpha_{o(t-\tau)} \cdot \phi_{o(t-\tau)}] \tau, \\ e_{\max} \end{array} \right], \quad (4.1)$$

recognizing that if the energy levels exceeds the battery capacity, then the excess will be spilled.

In Eq. 4.34,

- τ is a unit of time approaching 0 in the limit, measured in seconds.

- $\pi^{e+}(t)$ is the rate of energy collection at time t , measured in Joules per second.
- $\pi^{e-}(t)$ is the rate of energy consumption for nominal (non-downloading) functional operations at time t , measured in Joules per second.
- $\alpha_{o(t)}$ is the energy utilization to download using option $o(t)$, measured in Joules per bit.
- $\phi_{o(t)}$ is the data download rate using option $o(t)$, measured in bits per second.
- e_{\max} is the upper limit on energy storage for the battery, measured in Joules.

Note that, by our assumption of the feasibility of the schedule, $e(t)$ will never drop below e_{\min} , which is defined to be the lower limit on energy storage for the battery, measured in Joules.

4.1.2 Data Dynamics

Data dynamics are analogous to energy dynamics, with the exception that we do not need the $\alpha_{o(t)}$ term. There are upper and lower bounds on the amount of data that can be stored in the spacecraft's data buffer, denoted d_{\min} and d_{\max} , respectively. Replacing π^{e+} and π^{e-} , with the analogous rates of data collection and consumption, π^{d+} and π^{d-} , we have,

$$d(t) = \min \left[\begin{array}{c} d(t - \tau) + [\pi^{d+}(t - \tau) - \pi^{d-}(t - \tau) - \phi_{o(t-\tau)}] \tau, \\ d_{\max} \end{array} \right]. \quad (4.2)$$

Note that, by our assumption of the feasibility of the schedule, $d(t)$ will never drop below d_{\min} , which is defined to be the lower limit on data storage, measured in bits.

4.1.3 System Optimization

Our goal is to find a feasible schedule that maximises the amount of data downloaded over the planning horizon. The discontinuities in the system dynamics caused by the upper and lower bounds on the allowable stored energy and data result in a difficult non-linear optimization problem. To overcome this challenge, we discretize the problem into a finite set of time periods to approximate the continuous-time dynamics. We initially require the energy and data constraints to be satisfied only at the start and end of each time period and ignore the dynamics within these time periods. We demonstrate that this under-constrained formulation may yield an infeasible solution, but if it is feasible then it is optimal. In Section

4.2.2 we present the under-constrained formulation. In Section 4.3 we discuss special cases where the under-constrained formulation is guaranteed to yield a feasible and thus optimal solution. In Section 4.4 we present an iterative algorithm that progressively tightens the constraints of the under-constrained formulation until a feasible and thus optimal solution is achieved.

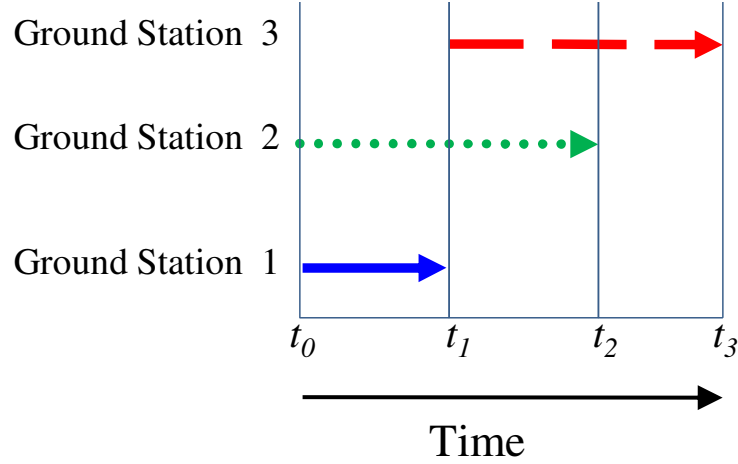


Figure 4.2: Ground stations in view of a single satellite for three intervals. The arrowed lines denotes the ground station is in view and available for communicating.

4.2 Problem Formulation

4.2.1 Notation

Sets and Subsets

- I is the set of intervals.
- O is the set of download options.
- $O_i \subseteq O$ is the subset of download options available during interval i , $\forall i \in I$.

Parameters

- η_{io} is the efficiency during interval i when downloading using option o , $\forall i \in I, o \in O_i$.
- t_i is the start time of interval i , measured in seconds, $\forall i \in I$.

- t_{i+1} is the end time of interval i , and coincides with the start of interval $i+1$, measured in seconds, $\forall i \in I$.
- Δt_i is the duration of time interval i ($\Delta t_i = t_{i+1} - t_i$), measured in seconds, $\forall i \in I$.
- ϕ_{io} is the data rate associated with downloading during interval i using option o , measured in bits/seconds, $\forall i \in I, o \in O_i$.
- α_{io} is the energy per data associated with downloading using option o during interval i , measured in Joules/bits, $\forall i \in I, o \in O_i$.
- e_{min} and e_{max} are the minimum and maximum allowable amount of energy to be stored in the battery, measured in Joules.
- e_{start} is the amount of energy stored in the battery at the beginning of the planning horizon, measured in Joules.
- d_{min} and d_{max} are the minimum and maximum allowable data stored in the data buffer, measured in bits.
- d_{start} is the amount of data stored in the data buffer at the beginning of the planning horizon, measured in bits.
- $\delta_i^{e+} \geq 0$ is the total amount of energy that can be acquired during interval i , measured in Joules, $\forall i \in I$.
- $\delta_i^{e-} \geq 0$ is the total amount of energy consumed during the interval i for non-download operations (e.g. data collection and nominal operations), measured in Joules, $\forall i \in I$.
- $\delta_i^{d+} \geq 0$ is the total amount of data that can be acquired during interval i , measured in bits, $\forall i \in I$.
- $\delta_i^{d-} \geq 0$ is the total amount of data lost during the interval i unrelated to data download (e.g. data degradation, expiration, etc.), measured in bits, $\forall i \in I$.

Variables

- $x_{io} \in \{0, 1\}$ is the binary value representing the decision of whether to download using option o during some portion of interval i , $\forall i \in I, o \in O_i$.
- $q_{io} \in \mathbb{R}^+$ is the amount of data downloaded during interval i using option o , measured in bits, $\forall i \in I, o \in O_i$.

- $e_i \in \mathbb{R}^+$ is the amount of energy available at the beginning of interval i , measured in Joules, $\forall i \in I$.
- $d_i \in \mathbb{R}^+$ is the amount of data available at the beginning of interval i , measured in bits, $\forall i \in I$.
- $h_i^e \in \mathbb{R}^+$ is the amount of excess energy spilled throughout interval i , measured in Joules, $\forall i \in I$.
- $h_i^d \in \mathbb{R}^+$ is the amount of excess data spilled throughout interval i , measured in bits, $\forall i \in I$.

4.2.2 Under-Constrained Formulation (*UCF*)

In the under-constrained formulation (*UCF*), we ignore the data and energy dynamics over the duration of the interval, checking that constraints are satisfied only at the start and end points. The formulation is as follows:

$$\max \sum_{i \in I} \sum_{o \in O_i} \eta_{io} q_{io} \quad (4.3)$$

s.t.

$$\sum_{o \in O_i} x_{io} \leq 1 \quad \forall i \in I \quad (4.4)$$

$$q_{io} \leq \Delta t_i \phi_{io} x_{io} \quad \forall i \in I, o \in O_i \quad (4.5)$$

$$e_0 = e_{start} \quad (4.6)$$

$$e_{min} \leq e_i \leq e_{max} \quad \forall i \in I \quad (4.7)$$

$$e_{i+1} = e_i + \delta_i^{e+} - \delta_i^{e-} - \sum_{o \in O_i} \alpha_{io} q_{io} - h_i^e \quad \forall i \in I \quad (4.8)$$

$$d_0 = d_{start} \quad (4.9)$$

$$d_{min} \leq d_i \leq d_{max} \quad \forall i \in I \quad (4.10)$$

$$d_{i+1} = d_i + \delta_i^{d+} - \delta_i^{d-} - \sum_{o \in O_i} q_{io} - h_i^d \quad \forall i \in I \quad (4.11)$$

$$x_{io} \in \{0, 1\} \quad \forall i \in I, o \in O_i \quad (4.12)$$

$$q_{io} \in \mathbb{R}^+ \quad \forall i \in I, o \in O_i \quad (4.13)$$

The objective, Eq. 4.3, maximizes the total amount of data received from a single satellite over the planning horizon. Constraint 4.4 enforces that the satellite may download using a single option o only during each time interval i . Constraint 4.5 enforces that data can only be downloaded using the chosen option for any given interval, and that the amount of data downloaded is limited by the time and chosen data rate of the interval. Constraint 4.6 initializes the amount of energy stored at the start of the planning horizon. Constraint 4.7 ensures that the amount of energy stored at the beginning of each interval is within the battery limits. Constraint 4.8 defines the amount of energy stored at the beginning of an interval to be the amount stored at the start of the preceding interval plus the amount acquired minus the amount consumed for nominal operations minus the amount used to support download minus the amount spilled. Constraint 4.9 initializes the amount of stored data. Constraint 4.10 ensures that the amount of data stored at the beginning of each interval is within the data buffer limits. Constraint 4.11 defines the amount of data stored at the beginning of an interval to be the amount stored at the start of the preceding interval plus the amount acquired minus the amount lost to degradation or expiration minus the amount downloaded minus the amount spilled.

UCF may result in infeasible solutions because the energy and data buffer constraints are imposed only at the start and end of each intervals, and are neglected throughout each interval. Consider the case in Figure 4.3 where an interval begins with an empty battery and a full data buffer. Assume that energy is acquired at twice the rate required to support download, however the acquisition occurs only during the second half of the interval. Then in *UCF*, the model would allow data to be continuously downloaded for the entire interval, because over the duration of the interval enough energy is acquired to support this download. However, in reality, downloading could begin only halfway through the interval (when energy begins to be acquired) and thus only half of the total data collected could be downloaded.

Note that because all feasible solutions to the true problem (i.e. with continuous dynamics) are also feasible for this under-constrained formulation, then if *UCF* is feasible for the true problem, then it is also optimal for the true problem.

4.3 A Special Case: Linear Dynamics

The previous section introduced a formulation to optimize instances of *SMSP* with general forms of energy and data dynamics. This section focuses on the special case in which the energy and data dynamics are linear, which has important implications for the optimality of *UCF* solutions. We begin here by presenting a theorem and proof that, for problems

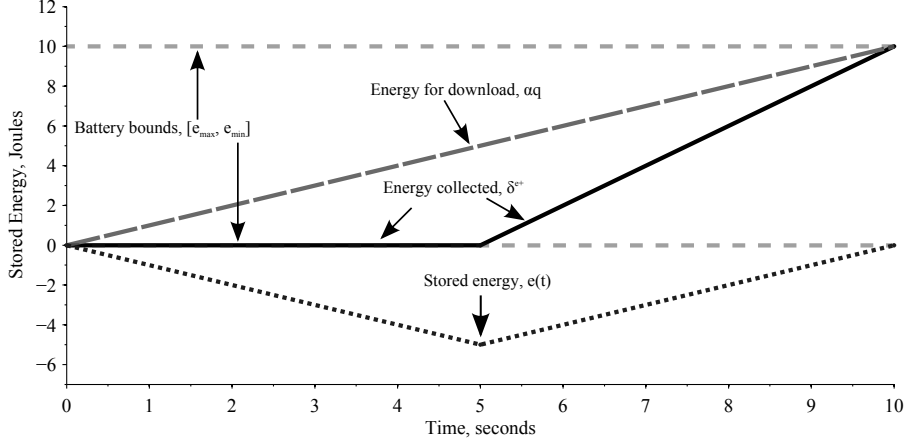


Figure 4.3: Single-interval instance of *SMSP* where a feasible solution to *UCF* results in infeasibilities when applied to the continuous-time dynamics.

with linear dynamics, *UCF* guarantees optimal solutions. In Section 4.3.1 we use this fact to study the tractability of several real-world instances of *SMSP* with a range of ground station networks and download options. We provide computational results in Section 4.3.2 for a wide range of instances of *SMSP* to show its general applicability beyond our specific application. We conclude in Section 4.3.3 with examples of special cases that may lead to computational intractability.

Theorem 1

Consider an instance of *SMSP* in which the rates of energy and data acquisition (π^{e+} and π^{d+}), nominal energy consumption (π^{e-}) (i.e. required for nominal operations but not to support downloading data), and data loss (π^{d-}) are each constant with respect to time over the duration of any given interval $i \in I$. For any such instance, an optimal solution to the corresponding instance of *UCF* will satisfy the continuous-time dynamics (see Sections 4.1.1 and 4.1.2) throughout the entire interval duration. This is a sufficient condition to yield a feasible, and thus optimal, solution to *SMSP*.

Proof 1

For simplicity of exposition, we focus here on the energy dynamics; the same logic applies to the data dynamics. Because energy acquisition and consumption are constant with respect to time over the duration of interval i , we replace $\pi^{e+}(t)$ and $\pi^{e-}(t)$ with π_i^{e+} and π_i^{e-} . By Lemma 1.1 (see Appendix C.1) we can convert any *UCF* solution in which data is downloaded for only a portion of a given interval i at a chosen constant rate to an equivalent and feasible solution in which the same amount of data is downloaded at a (lower) constant rate, which we denote by θ_i^e , for the entire duration of interval i . Similarly, for the case where energy is spilled for some fractional portion of interval i , we can assume

without loss of generality, by Lemma 1.2, that this spillage occurs at a constant rate, which we denote by s_i^e , for the entire duration of interval i .

Since the rates at which energy is acquired, nominally consumed, consumed to support download, and spilled are all constant over the duration of interval i , the net change in energy and data is linear with respect to time:

$$e(t) = e_i + [\pi_i^{e+} - \pi_i^{e-} - \theta_i^e - s_i^e](t - t_i) \quad \forall i \in I, t \in [t_i, t_{i+1}]. \quad (4.14)$$

Thus, the energy at any time $t \in (t_i, t_{i+1})$ will lie on the line segment connecting the two points (t_i, e_i) and (t_{i+1}, e_{i+1}) . Because the energy levels at times t_i and t_{i+1} (which are the end points of this line segment) are within the feasible energy range by Constraint 4.7, the energy level at any intermediate point along this line segment will be within this range and thus feasible as well. ■

4.3.1 Real-World Computational Experiments

Theorem 1 proves that using *UCF* for a problem with linear dynamics will yield an optimal solution to the underlying continuous-dynamics problem. In this section, we analyze the performance of this formulation, solved using standard branch-and-bound techniques, on real-world problem instances.

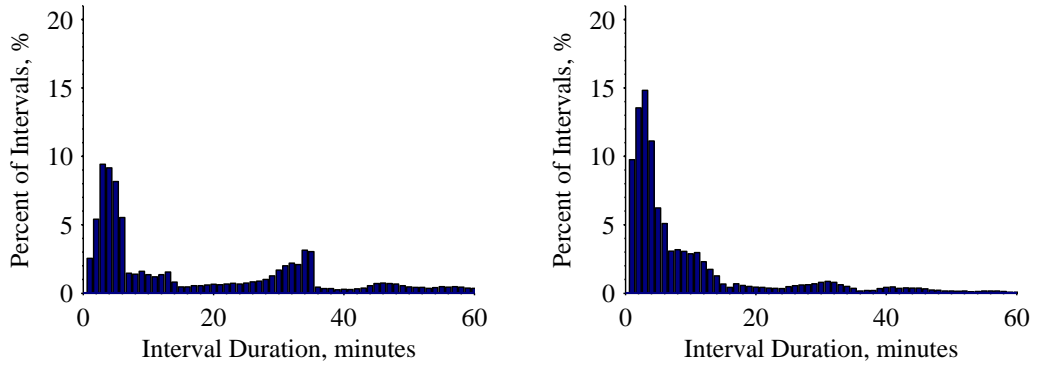
The computational experiments in this section are based on a real-world satellite mission studying space weather. Specifically, we model the Radio Aurora Explorer (RAX), the first NSF-funded CubeSat (a type of miniaturized satellite) [159]. RAX orbits the Earth every 97 minutes, with ground stations periodically coming in and out of view. RAX collects energy whenever it is in view of the sun. It collects scientific data whenever it is in view of its target of interest. This data collection occurs for short periods of time (≤ 5 minutes) at high data rates. These dynamics can be modeled by piece-wise linear functions. Thus, in accordance with *Theorem 1*, optimality is guaranteed when using *UCF*.

We analyze RAX in the context of three diverse and realistic networks, summarized in Table 4.1 [1]. N1 consists uniquely of RAX's primary ground station, located in Ann Arbor, MI. This ground station has a single download option. N2 is the ground station network primarily used by RAX. It consists of seven independently-owned and operated stations around the world. Each station has a single download option; downloading characteristics vary from station to station. N3 is comprised of 20 amateur radio ground stations with which RAX could potentially communicate. Each of these stations has three download options with unique characteristics.

Key properties of the three networks are shown in Figures 4.4 and 4.5 for a one year

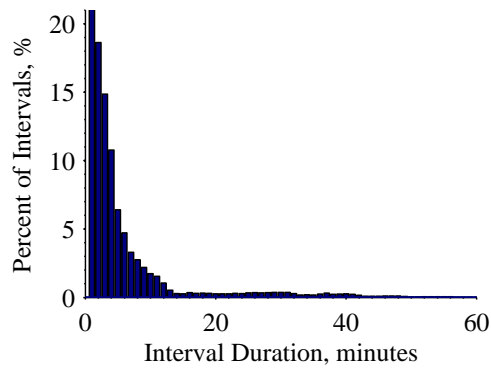
Table 4.1: Parameters for sample ground station (GS) networks

Network	Description	Number of GSs	Number of Options at each GS
N1	Single Station: Ann Arbor, MI	1	1
N2	RAX Network	7	1
N3	Global Amateur Radio Network	20	3



(a) N1

(b) N2



(c) N3

Figure 4.4: Distributions of interval duration for predicted RAX orbits and three diverse networks for a one year planning horizon.

planning horizon. The trends are similar for shorter planning horizons due to the periodicity of satellite orbits. The distribution of the duration of the intervals is shown in Figure 4.4. A new interval begins whenever a download opportunity starts (i.e. a ground station comes into view) or ends, or when there is a change in the rate at which energy or data is collected or nominally consumed. N1 has long duration intervals since it has a single station and

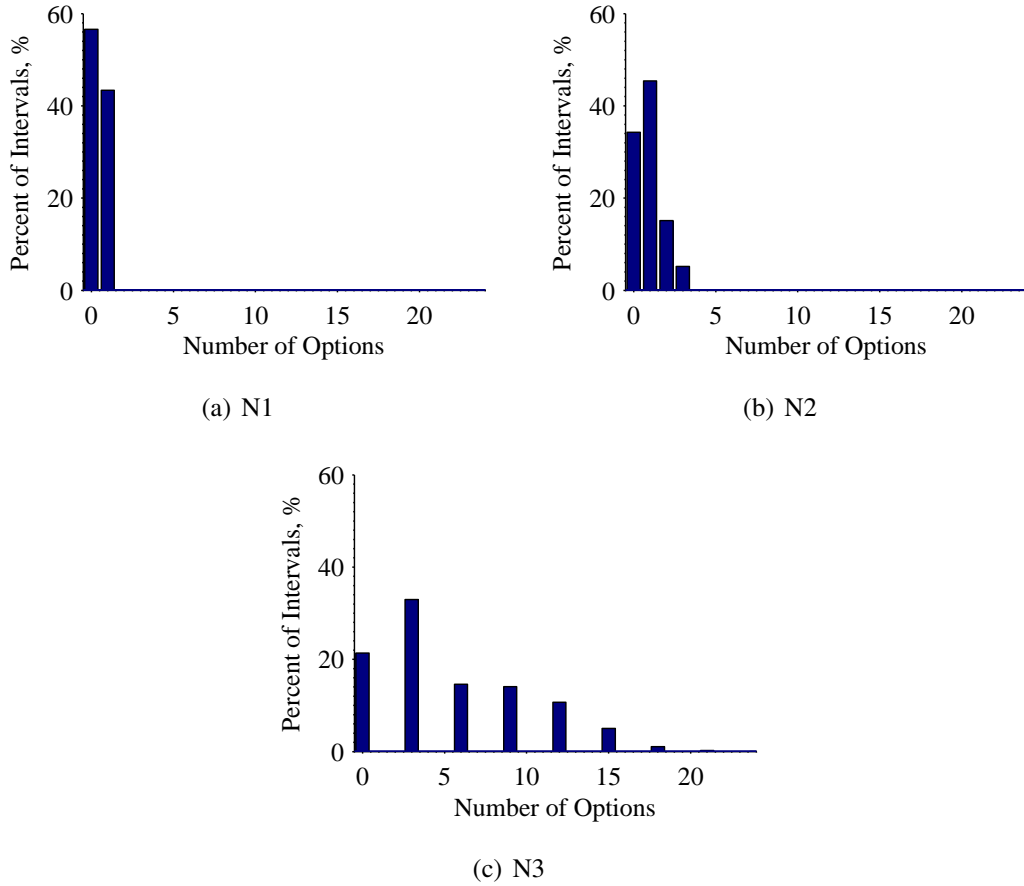


Figure 4.5: Distributions of number of download options for predicted RAX orbits and three diverse networks for a one year planning horizon.

eclipse durations can be up to 60 minutes. N2 and N3 have shorter intervals as they have greater number of ground stations, and thus more opportunities for these stations to come in and out of view. The distribution of the number of options per interval is shown in Figure 4.5. N1 and N2 each have a single option per network, where N2 has more options as it contains more ground stations relative to N1. N3 has three options per ground station, thus only multiples of three options occur.

For each of the three ground station networks, we evaluated several different planning horizons ranging from a few hours to two months. [Note that, in practice, the general planning horizon for RAX is on the order of days.] The computations were performed on an Intel Core i7 2.8 GHz processor with 8 GB of memory using the IBM ILOG Optimization Studio (CPLEX) 12.1 C++ API software package [177]. Instances of *SMSP* are MIPs because they have both integer and continuous variables. MIPs are NP-complete, which is a

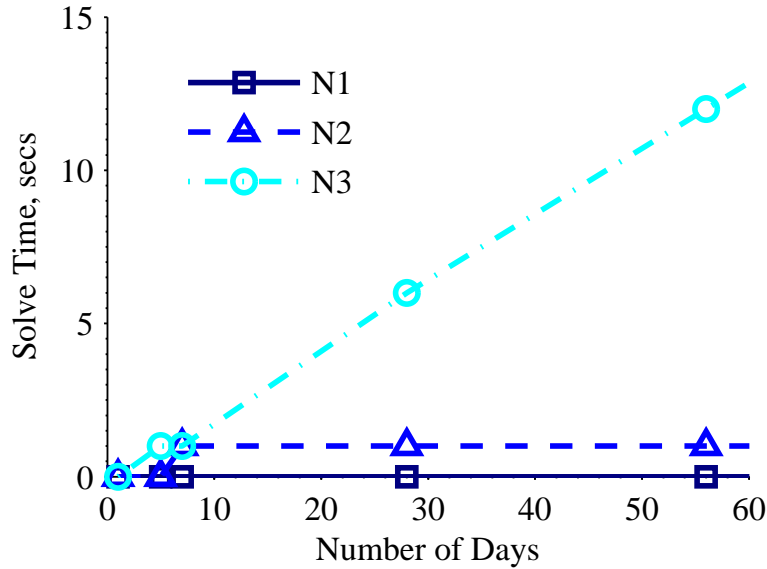


Figure 4.6: Solve time as a function of planning horizon

class of problems where the solutions can be verified quickly, however there is no known efficient way to find solutions. In the remainder of this section we investigate the computational tractability when solving instances of *SMSP* to verify how efficiently solutions can be found using CPLEX.

Results are shown in Figure 4.6. All problem instances solved quickly (i.e. in a matter of seconds), even those with planning horizons significantly longer than that required in practice. N3 did demonstrate significantly slower results relative to N1 and N2. This is not surprising, however, given that N3 not only has more ground stations, but each of these stations has multiple download options, thereby increasing both the size and complexity of the associated MIP. Even with the choice of multiple options per station, however, we observed no branching for any of the instances tested. This fact, in combination with the relatively small size of the MIPs, explains the fast run times.

To gain a more general understanding of our proposed formulation, we consider a collection of larger and more complex problem instances in Section 4.3.2, no longer limiting ourselves strictly to the real-world RAX applications. This enables us to test the boundaries of tractability for a more general class of problem instances. We then focus in Section 4.3.3 on exploring the integrality of the proposed formulation. In particular, we demonstrate problem instances for which significant branching (and thus slow run times) will occur.

4.3.2 General Case Computational Experiments

In the previous subsection, we observed that many real-world satellite communication scheduling problems could be solved via *Under-Constrained Formulation (UCF)*. Provably-optimal solutions were found in under one minute for all tested instances, including several instances with time horizons much longer than would be found in real-world practice. In this subsection, we consider whether these computational results are specific to the underlying RAX problem structure, or would be experienced in a broader class of problem instances as well.

For our broader computational experiments, we consider problem instances with parameters drawn from the distributions in Tables 4.2 and 4.3. Table 4.2 includes the distributions that are constant for all tests and Table 4.3 shows the distributions for 16 different test cases. Because all the parameters must be non-zero, if a sampled parameter drawn from a normal distribution is negative, it is rounded to zero. Similarly, if the efficiency parameter, η , exceeds its upper bound of one, it is rounded to one. We assume that no energy or data is nominally lost during each interval, i.e. $\delta_i^{e-} = 0$ and $\delta_i^{d-} = 0$. For each test case, we generated and solved 100 instances, sampling from the distributions in Tables 4.2 and 4.3.

Table 4.2: Parameters that are constant for all test cases 0-15

Parameter	Description	Distribution
I	Number of intervals	1000
t_i	Duration of interval i	$\approx N(10, 4)$
O_i	Number of options for interval i	$P(3)=P(5)=0.5$
e_{min}, e_{start}	Energy minimum and starting levels	0, 0
d_{min}, d_{start}	Data minimum and starting levels	0, 0
η_{io}	Efficiency of options	$\approx N(1, 0.5)$

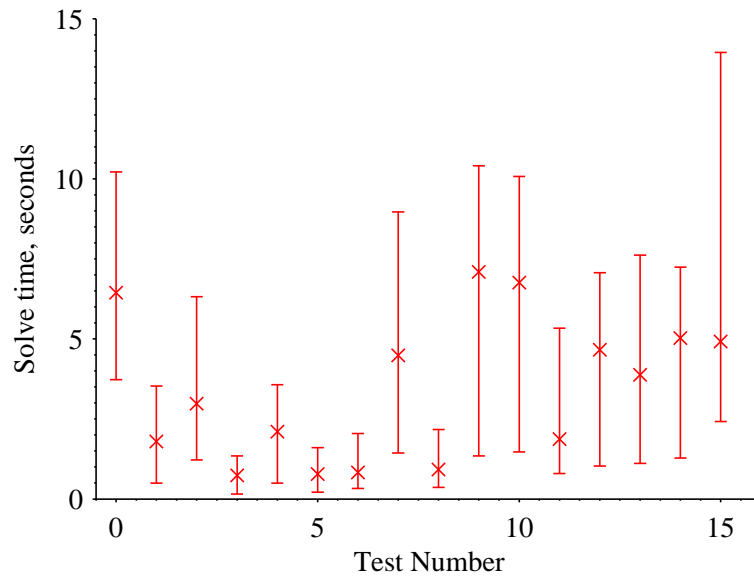
Computations were performed using an Intel Core i7 2.8 GHz processor with 8 GB of memory using CPLEX 12.2 C++ API with an optimality gap of 0.01%. The statistics of the computational results appear in Figure 4.7.

Figure 4.7 shows that, similar to the RAX application, all the instances solve very quickly – typically on the order of several seconds. Unlike the RAX application, where we observed no branching, there is branching observed in some of these tests. Most cases have a mean number of branch-and-bound nodes that is non-zero (often about 500 nodes). However, the minimum number of nodes is zero for all test cases, as shown in Figure 4.7, which indicates that at least one of the 100 instances for each test case had no branching.

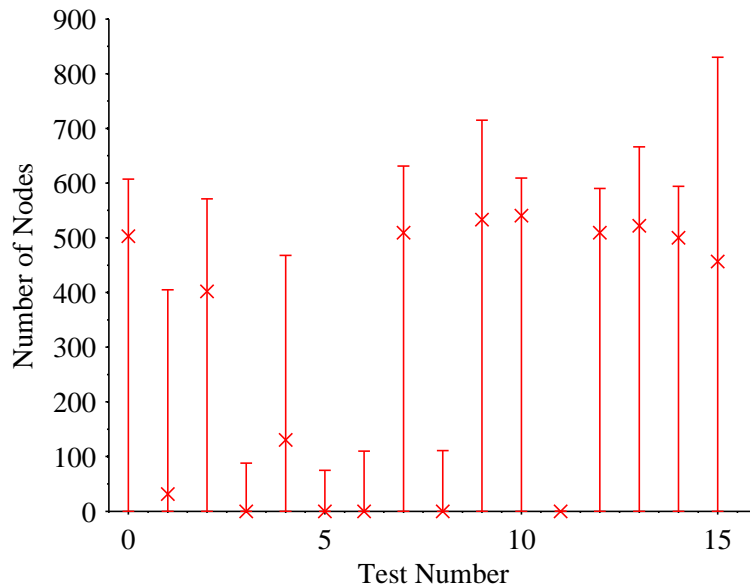
Table 4.3: Parameter distributions for test cases 0-15

Parameter description	Maximum energy buffer	Energy collection	Maximum data buffer	Data collection	Energy utilization	Data rate
Test	ϵ_{max}	δ^{e+}	d_{max}	δ^{d+}	α_{io}	ϕ_{io}
0	100	$\approx N(80,40)$	100	$\approx N(50,25)$	$\approx N(5,2.5)$	$\approx N(2,1)$
1	300	$\approx N(80,40)$	300	$\approx N(50,25)$	$\approx N(5,2.5)$	$\approx N(2,1)$
2	50	$\approx N(80,40)$	50	$\approx N(50,25)$	$\approx N(5,2.5)$	$\approx N(2,1)$
3	100	$\approx N(80,40)$	100	$\approx N(50,25)$	$\approx N(5,2.5)$	$\approx N(4,2)$
4	100	$\approx N(80,40)$	100	$\approx N(50,25)$	$\approx N(10,5)$	$\approx N(2,1)$
5	100	$\approx N(80,40)$	100	$\approx N(50,25)$	$\approx N(10,5)$	$\approx N(4,2)$
6	100	$\approx N(80,40)$	100	$\approx N(50,25)$	$\approx N(2.5,1.25)$	$\approx N(2,1)$
7	100	$\approx N(80,40)$	100	$\approx N(50,25)$	$\approx N(5,2.5)$	$\approx N(1,.5)$
8	100	$\approx N(80,40)$	100	$\approx N(50,25)$	$\approx N(2.5,1.25)$	$\approx N(1,.5)$
9	100	$\approx N(40,20)$	100	$\approx N(50,25)$	$\approx N(5,2.5)$	$\approx N(2,1)$
10	100	$\approx N(80,40)$	100	$\approx N(25,12.5)$	$\approx N(5,2.5)$	$\approx N(2,1)$
11	100	$\approx N(40,20)$	100	$\approx N(25,12.5)$	$\approx N(5,2.5)$	$\approx N(2,1)$
12	100	$\approx N(160,80)$	100	$\approx N(50,25)$	$\approx N(5,2.5)$	$\approx N(2,1)$
13	100	$\approx N(80,40)$	100	$\approx N(100,50)$	$\approx N(5,2.5)$	$\approx N(2,1)$
14	100	$\approx N(160,80)$	100	$\approx N(100,50)$	$\approx N(5,2.5)$	$\approx N(2,1)$
15	100	$\approx N(40,20)$	100	$\approx N(25,12.5)$	$\approx N(10,5)$	$\approx N(2,1)$

Although there is not a perfect correlation between the number of nodes and the run time, there is, as expected, a fairly strong correlation between them.



(a) Solve Time

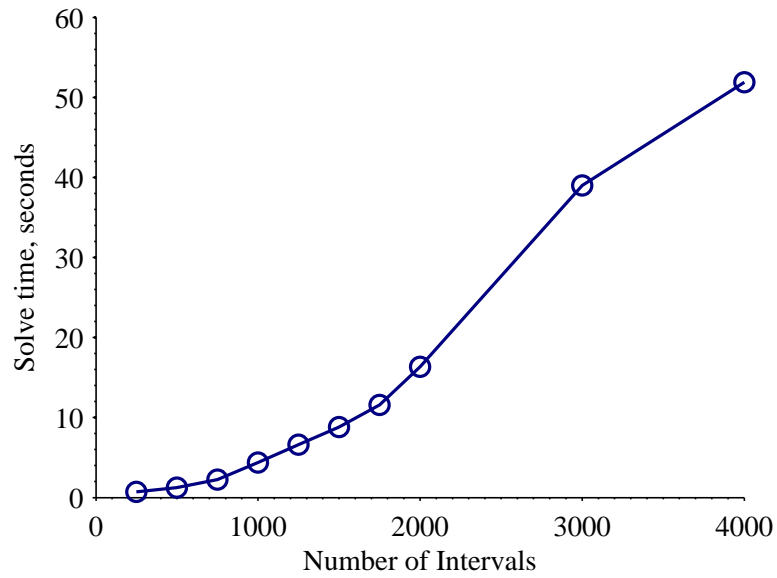


(b) Number of Branch-and-Bound Tree Nodes

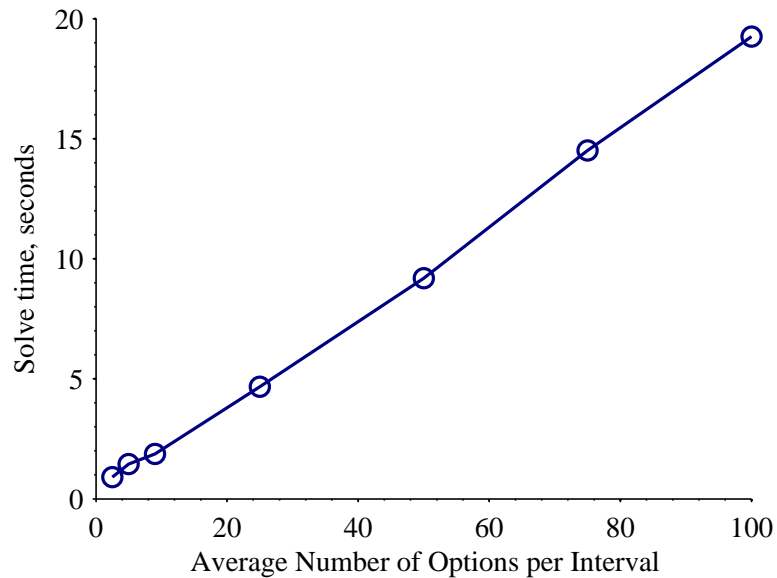
Figure 4.7: Computational statistics for test cases in Table 4.3. The x represents the mean, and the upper and lower bounds show the maximum and minimum values.

Next, we conducted tests specifically targeted at studying the impact of the number of problem intervals and number of download options, and therefore the size of the *MIP*,

on computational performance. These test are performed with the parameter distributions in Table 4.2 and parameter distributions for test case 0 in Table 4.3. The results for 100 randomly-generated instances with variations in the number of intervals and options are shown in Figs. 4.8(a) and 4.8(b), respectively.



(a) Effect of Intervals



(b) Effect of Options

Figure 4.8: The effects of the number of intervals and average number of options per interval on solve time

In Figure 4.8(a), we observe that the run time increases, as expected, with the number of intervals,. In our example, an instance with 1,000 intervals will have $\approx 4,000$ binary decision variables and $\approx 9,000$ constraints and an instance with 10,000 intervals will have $\approx 40,000$ binary decision variables and $\approx 90,000$ constraints. The lack of apparent exponential growth in solve time with significantly larger problems indicates there is a limited amount of branching. We hypothesize that this is because the impact of decisions in one time interval have limited impact on decisions several intervals into the future and thus the coupling between intervals is fairly loose except for this intervals very close together in time.

In Figure 4.8(b), we allow the number of options per interval to increase from one to 100 (recall that in the first set of experiments, each interval had either three or five download options from which to choose). In our example, an instance with 10 options per interval will yield $\approx 10,000$ binary decision variables and $\approx 15,000$ constraints and an instance with 100 options will yield $\approx 40,000$ binary decision variables and $\approx 105,000$ constraints. Despite a linear growth in the number of binary decision variables with increase in number of options, the problem may become more complex as, with more options, there are multiple competing options during every download opportunity. However, we continue to see run times remaining well within the range of tens of seconds and the solve time does not grow significantly as the number of options increases as there is limited branching. We suggest that although, in theory, there are many competing options at each interval that could lead to fractionality and associated branching, in practice as the number of options grows the likelihood of these new options being dominated by existing options grows as well. Thus, the incentive for fractional solutions does not increase as quickly as the number of options. In the next session, we focus on identifying cases where significant branching *does occur*.

The experiments in this section indicate that a broad class of problems that can be solved with *UCF* can be solved quickly. An important observation from these tests is that branching does occur in some instances. This confirms that integral optimal solutions, as observed in the RAX-specific examples, are not characteristic of the *UCF* problem structure, but are rather a consequence of the input data. All tests in this section have had fast run times, which is largely driven by a lack of significant branching, motivating the following section which addresses when branching occurs.

4.3.3 Non-Integral Solutions Resulting in Branching

In this section we investigate conditions under which branching occurs, and the implications of this for tractability.

4.3.3.1 Single Interval *UCF* Non-Integral Solution Example

For simplicity, we consider a single interval of *UCF*, so there is a single decision: what download option to use. If one option is more desirable in all aspects (i.e. most efficient, largest data-to-time ratio, and smallest energy-to-data ratio) compared to other available options, then the dominant option will be used exclusively, and the linear programming (*LP*) relaxation will yield an optimal integer solution. In this section, we investigate the case when there is no dominant solution and thus fractionality could potentially occur in an optimal solution to the *LP* relaxation.

Consider the two-option single-interval instance of *SMSP* in Table 4.4. The two feasible integer solutions and the optimal solution to the *LP* relaxation, which is non-integer, are given in Table 4.5.

Table 4.4: Two-Option example where *LP* relaxation yields a fractional optimal solution

Interval Parameters	Option 1	Option 2
$t=6$ secs	$\phi_1=2$ bits/sec	$\phi_2=3$ bits/sec
$e=36$ J	$\alpha_1=2$ J/bit	$\alpha_2=4$ J/bit
$d=14$ bits	$\eta_1=1$	$\eta_2=1$

Note that in this single-interval example, there is no advantage in reserving resources (energy, or data) for future intervals. Solution 1 exhausts the available energy by downloading at a high data rate, leaving behind unused time and data. Solution 2 exhausts the available time by downloading at a more energy-efficient rate, leaving behind unused energy and data. By taking a convex combination of these two options, Solution 3 is able to more effectively trade off time and energy to download a larger amount of data in a fractional solution.

Table 4.5: Feasible Solutions to the *MIP* and the optimal *LP* relaxation

Solution	x_1	x_2	q_1 (bits)	q_2 (bits)	Optimal Solution $q_1 + q_2$ (bits)	Excess Energy (J)	Excess Data (bits)	Excess Time (secs)
1	1	0	12	0	12 bits	0 J	5 bits	3 secs
2	0	1	0	9	9 bits	12 J	2 bits	0 secs
3	0.75	0.25	9.5	4	13.5 bits	0 J	0.5 bits	0 secs

4.3.3.2 Single Interval *UCF* Conditions for Non-Integral Solutions

Next we generalize this example to develop a set of sufficient conditions such that the optimal solution to the LP relaxation of *UCF* is non-integer. The constraints for the LP relaxation of *UCF* for a two-option single-interval scenario are explicitly enumerated in Eqs. 4.15-4.24, where Δe and Δd represent the total amount of available energy and data at the start of the interval, respectively.

$$\max\{\eta_1 q_1 + \eta_2 q_2\} \quad (4.15)$$

$$s.t. \quad q_1 \leq t\phi_1 x_1 \quad (4.16)$$

$$q_2 \leq t\phi_2 x_2 \quad (4.17)$$

$$\alpha_1 q_1 + \alpha_2 q_2 \leq \Delta e \quad (4.18)$$

$$q_1 + q_2 \leq \Delta d \quad (4.19)$$

$$x_1 + x_2 = 1 \quad (4.20)$$

$$q_1 \geq 0 \quad (4.21)$$

$$q_2 \geq 0 \quad (4.22)$$

$$x_1 \geq 0 \quad (4.23)$$

$$x_2 \geq 0 \quad (4.24)$$

To simplify the analysis, let the available data, Δd , be sufficiently large such that Constraint 4.19 is never binding. [Note that an analogous analysis can be conducted assuming that the energy level, Δe , is sufficiently large such that Constraint 4.18 is never binding.] In addition, assume that the efficiency of each option is equal to one. Finally, motivated by the observations of the relative option characteristics of our non-integer example, assume that Option 1 alone depletes time while leaving excess energy unused (i.e. if $x_1 = 1$, Constraint 4.16 will be binding and Constraint 4.18 will not be binding), and Option 2 alone depletes energy, leaving excess time (i.e. if $x_2 = 1$, Constraint 4.17 will not be binding and Constraint 4.18 will be binding). This set of conditions can be summarized as:

$$\alpha_1 \phi_1 < \frac{e}{t} < \alpha_2 \phi_2. \quad (4.25)$$

Eqs. 4.15-4.24 form a linear program with four decision variables, and thus a basic solution will have four binding constraints. Given the conditions in Eq. 4.25, the optimal solution to the problem may yield three possible outcomes, which are summarized in Table 4.6 and developed in more detail in Appendix B. Cases 1 and 2 each consider when a single option

is used, and their optimal solutions are computed based on Constraints 4.18 or 4.16 being active. In Case 3, both Options 1 and 2 are used, thus Constraints 4.19 and 4.21-4.24 are not active. As a result, the remaining four constraints 4.16, 4.17, 4.18, and 4.20 are active. The conditions for this case are $\phi_2 > \phi_1$, $\alpha_2 > \alpha_1$, see Appendix B for the procedure for computing these conditions.

The conditions for Case 3 are sufficient such that the optimal solution to the LP relaxation is non-integer. Note that branching may be caused by other conditions. For example, when energy is never a constraining resource, a similar argument can be made to show that the LP relaxation will yield fractional solution under the conditions $\phi_1 t < \Delta d < \phi_2 t$ and $\eta_2 < \eta_1$.

4.3.3.3 Fractionality with More than Two Options

We extend the logic for the single-interval two-option case to consider cases with multiple options. The LP relaxation of a single-interval instance of *UCF* with three options will have six variables, thus an optimal basic feasible solution will have six active constraints. A basic feasible solution that utilizes all three download options (and thus is fractional) will have non-binding non-negativity constraints on all variables, i.e. $x_1 > 0, x_2 > 0, x_3 > 0, q_1 > 0, q_2 > 0, q_3 > 0$. Thus, the six remaining equations must be binding, as in the system 4.26.

$$\begin{cases} q_1 = t\phi_1 x_1 \\ q_2 = t\phi_2 x_2 \\ q_3 = t\phi_3 x_3 \\ \alpha_1 q_1 + \alpha_2 q_2 + \alpha_3 q_3 = \Delta e \\ q_1 + q_2 + q_3 = \Delta d \\ x_1 + x_2 + x_3 = 1 \end{cases} \quad (4.26)$$

Since the system contains six variables and six equations, there exists a unique solution corresponding to this basis, $[q_1^*, q_2^*, q_3^*, x_1^*, x_2^*, x_3^*]$. Note that there may exist other basic feasible solutions with the same objective function value that use only one or two download options. We also note that as a basic feasible solution, there must be some cost vector $[q_1^*, q_2^*, q_3^*, x_1^*, x_2^*, x_3^*]$ for which this solution is uniquely optimal. Thus, so long as there are values of $[q_1^*, q_2^*, q_3^*, x_1^*, x_2^*, x_3^*]$ which are within the range $\{0, 1\}$ that make this basic feasible solution optional, then a fractional solution with three options used can be found.

We observe that it is *not* possible to have an optimal basic feasible solution for the

Table 4.6: Fractionality conditions for the single interval UCF with the conditions in Eq. 4.25.

Case	Conditions	x_1	x_2	q_1	q_2	Optimal Solution ($q_1 + q_2$)
1	$\alpha_2 \geq \alpha_1$ $\alpha_2 \phi_1 \geq \frac{e}{t}$	1	0	$t\phi_1$	0	$t\phi_1$
2	$\phi_2 \geq \phi_1$ $\alpha_2 \phi_1 \leq \frac{e}{t}$	0	$\frac{e}{\alpha_2 \phi_2 t}$	0	$\frac{e}{\alpha_2}$	$\frac{e}{\alpha_2}$
3	$\alpha_1 < \alpha_2$ $\phi_1 < \phi_2$	$\frac{\alpha_2 \phi_2 - \frac{\Delta e}{t}}{\alpha_2 \phi_2 - \alpha_1 \phi_1}$	$\frac{\frac{\Delta e}{t} - \alpha_1 \phi_1}{\alpha_2 \phi_2 - \alpha_1 \phi_1}$	$\frac{t\phi_1 (\alpha_2 \phi_2 - \frac{\Delta e}{t})}{\alpha_2 \phi_2 - \alpha_1 \phi_1}$	$\frac{t\phi_2 (\frac{\Delta e}{t} - \alpha_1 \phi_1)}{\alpha_2 \phi_2 - \alpha_1 \phi_1}$	$\frac{t\phi_1 \phi_2 (\alpha_2 - \alpha_1) + \Delta e (\phi_2 - \phi_1)}{\alpha_2 \phi_2 - \alpha_1 \phi_1}$

single interval case that uses more than three options, as stated and proven in the following theorem.

Theorem: For any single-interval instance of *UCF*, an optimal solution to the LP relaxation will at maximum have three download options.

Proof: For a single-interval instance of *UCF* with $|O|$ options, the number of variables is $n = 2|O|$ and the number of constraints is $m = 3 + 3|O|$. A solution that utilizes all $|O|$ options requires that all $2|O|$ non-negativity constraints are non-active. A basic solution requires that $2|O|$ of the remaining constraints (i.e. $3 + |O|$) must be active, therefore $(|O| + 3) \geq 2|O|$. The inequality holds only for $|O| = 0, 1, 2, 3$. If $|O| > 3$, then there are not enough remaining constraints to form a basis once the non-negativity constraints have been removed from consideration.

4.3.3.4 Computational Experiments

While all of the instances of *SMSP* that we have considered thus far solve very quickly, data instances *can* be contrived that branch extensively and thus solve slowly. Thus, we investigate the number of branch-and-bound nodes and solve times for different sizes of contrived cases with different optimality gaps, see the results in Figure 4.9 and Table 4.7. Results are shown in Figure 4.9 for up to 500 intervals to emphasize the trends for smaller numbers of intervals.

Large data sets, with up to 2,500 intervals can be solved in less than three minutes with an optimality gap of 1%, see Table 4.7. However, instances with more than 150 intervals with an optimality gap of 0.01% and with more than 300 intervals with an optimality gap of 0.1% are intractable, and thus do not appear in Figure 4.9 and Table 4.7. In these cases, the branch-and-bound tree becomes too large to store in allocated computer memory.

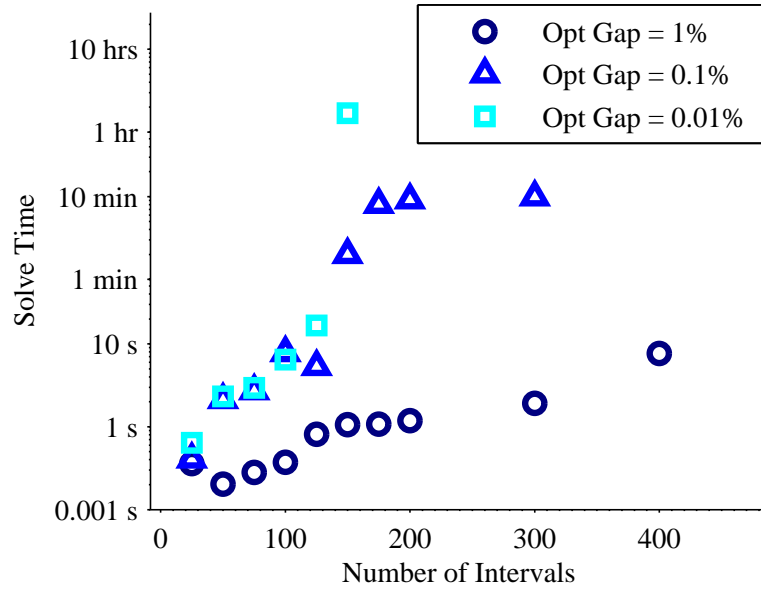
Figure 4.9 and Table 4.7 demonstrate that a tighter optimality gap increases the amount of branching and hence the run time. This is not surprising, as the problem instances were designed to have fractional solutions. On the other hand, the LP relaxation is quite weak in these problem instances, and the added run time serves only to tighten the upper bound, with little impact on the objective value.

4.4 Applications to Non-Linear Dynamics

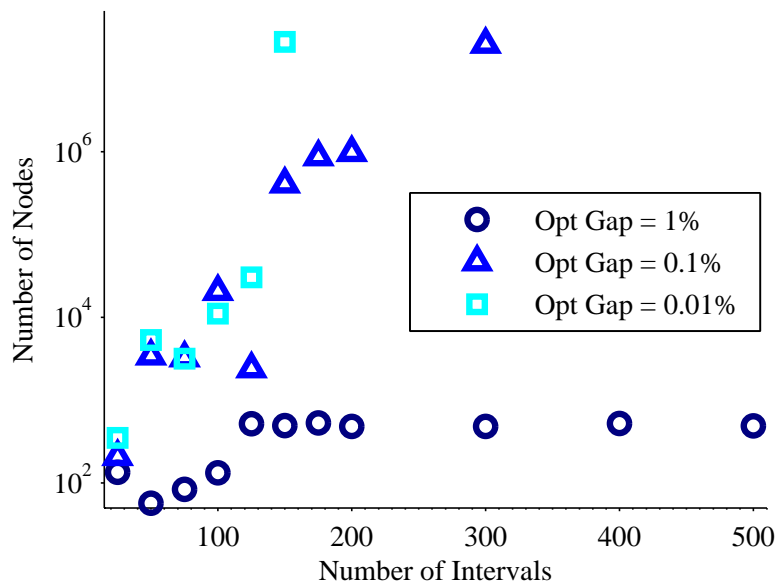
In *Theorem 1*, we proved that a solution to *UCF* for an instance of *SMSP* in which the data and energy dynamics are each constant over any given interval will always yield solutions

Table 4.7: Performance of worst-case instance of *UCF* under various optimality gaps. Numbers are rounded as relevant for our analysis.

Optimality Gaps	1.000%			0.100%			0.010%		
	Time	Number of Nodes	Obj. Val. (bits)	Time	Number of Nodes	Obj. Val. (bits)	Time	Number of Nodes	Obj. Val. (bits)
25.0	0.4 s	135	336	0.4 s	208	336	0.6 s	350	336
50.0	0.2 s	57	669	2.1 s	3406	672	2.3 s	5,321	672
75.0	0.3 s	84	1,005	2.7 s	3,244	1,011	2.9 s	3,166	1,011
100.0	0.4 s	133	1,338	7.8 s	20,471	1,350	6.4 s	11,115	1,350
125.0	0.8 s	520	1,671	5.3 s	2,393	1,686	16.8 s	30,341	1,686
150.0	1.1 s	492	2,010	2.0 min	407,241	2,022	1.7 hr	2,122,780	2,022
175.0	1.1 s	533	2,340	8.0 min	866,052	2,361	-	-	-
200.0	1.2 s	484	2,679	9.2 min	977,586	2700	-	-	-
300.0	1.9 s	483	4,011	9.9 min	1,973,530	4,047	-	-	-
400.0	7.7 s	523	5,352	-	-	-	-	-	-
500.0	8.0 s	490	6,684	-	-	-	-	-	-
2,500.0	3.0 min	29,682	33,441	-	-	-	-	-	-



(a) Solve Time



(b) Number of Nodes

Figure 4.9: Performance of worst-case instances of *UCF* under various optimality gaps.

that are feasible, and thus optimal, for the original problem. When the dynamics are non-linear, i.e. the rates of energy and data acquisition and consumption are not constant during an interval, a solution to *UCF* may be infeasible when applied to the true continuous-time problem. This occurs because *UCF* considers only the cumulative change in dynamics at the end of each interval and neglects the continuous-time dynamics that occur within the

interval.

For example, consider the single-interval case where the battery is empty at the start of the interval. Furthermore, suppose that for the first half of the interval no energy is acquired, and for the second half of the interval energy is acquired at the rate of 2 Joules/sec. In the *UCF* model, a feasible solution would be to use 1 Joule/sec throughout the entire interval, as the total amount of energy acquired is equal to the total amount consumed over the duration of the interval. In the true, continuous-time example, however, this would not be a feasible solution.

The infeasibilities that result from applying *UCF* solutions to instances with non-linear dynamics can be overcome by discretizing the continuous-time problem into sufficiently small intervals. As the interval sizes approach zero, *UCF* is guaranteed to yield feasible (and thus optimal) solutions. The resulting *MIP*, however, will be intractable.

In practice, much larger time intervals may be sufficient to ensure a valid solution. Specifically, so long as the solution defined by the boundary conditions of each interval also satisfies the dynamics within each interval, the solution will be valid. A generic approach for accomplishing this is described in Section 4.4.1. We present the approach specifically for problem instances with piece-wise linear dynamics (or that can be approximated with piece-wise linear dynamics) in Section 4.4.2.

4.4.1 Algorithm for Solving Non-Linear *SMSPs*

In this section we introduce Algorithm 1, *Non-Linear SMSP Algorithm (NLSA)*, an approach for solving non-linear instances of *SMSP*. This algorithm can theoretically be applied to solve any instance of *SMSP*. The tractability of the approach depends on the effort required to identify intervals with infeasibilities and the required degree of discretization required to achieve feasibility.

NLSA iteratively identifies intervals in a *UCF* solution that are infeasible and discretizes these intervals until feasibility (and thus optimality) is achieved. Let I^k be the set of intervals at the start of iteration k . At the start of *NLSA*, I^0 is the set of original intervals, where an *original interval* starts and ends when and only when there is a change in the set of download opportunities, i.e. a ground station comes in or out of view, see Figure 4.2. At every iteration k , *NLSA* solves an instance of *UCF* with I^k , and then checks the feasibility of each interval when the solution to the discretized problem is applied to the true, continuous-time dynamics. If the solution is feasible for all intervals, then it is optimal and the algorithm terminates. If any infeasibilities are found, then one or more infeasible intervals are split into sub-intervals. Constraints are then imposed to ensure that all sub-

intervals within an original interval use the same download option, i.e. x_i^k (the decision at iteration k for interval i) must be equivalent for all i within a common original interval. The process repeats until a valid solution is obtained.

Algorithm 1 *Non-Linear SMSP Algorithm (NLSA)*

1. Initialize $k = 0$ and I^0 to the set of original intervals
 2. Solve *UCF* with I^k
 3. Check for infeasibilities when applied to true dynamics
 - **if feasible** Terminate the algorithm because the solution is feasible (thus optimal)
 - **else**
 - Update $k = k + 1$
 - Split infeasible intervals, update $I^k \rightarrow I^{k+1}$
 - Impose constraints that all x_i^k must be equivalent for all i within a common original interval
 - Return to Step 2
-

The key design issues in the implementation of *NLSA* are the approaches for checking the feasibility of a given solution and splitting infeasible intervals.

4.4.2 Special Case: Piece-wise Linear Dynamics

In this section we demonstrate the applicability of **NSLA!** (**NSLA!**) to special cases of *SMSP* where the dynamics are piece-wise linear (PWL). These are instances where the time horizon can be broken into a discrete number of linear segments, which are time periods over which the energy and data rates are constant. See Figure 4.10 for an example of PWL energy and data dynamics.

4.4.2.1 Assessing the Feasibility of an Interval

A key component of *Non-Linear SMSP Algorithm (NLSA)* is the mechanism for testing the feasibility of an interval k for a given solution to *UCF*. That is, for a given iteration i , we have the optimal solution q_i^* from *UCF* which is guaranteed to satisfy system dynamics at the beginning and end of the interval, but not necessarily throughout the interval. In addition, in the case where the time required to download q_i^* is less than the full duration of the interval, *UCF* does not specify when within the interval the download should take place (note that a feasible download may include multiple starts and stops). In this section, we

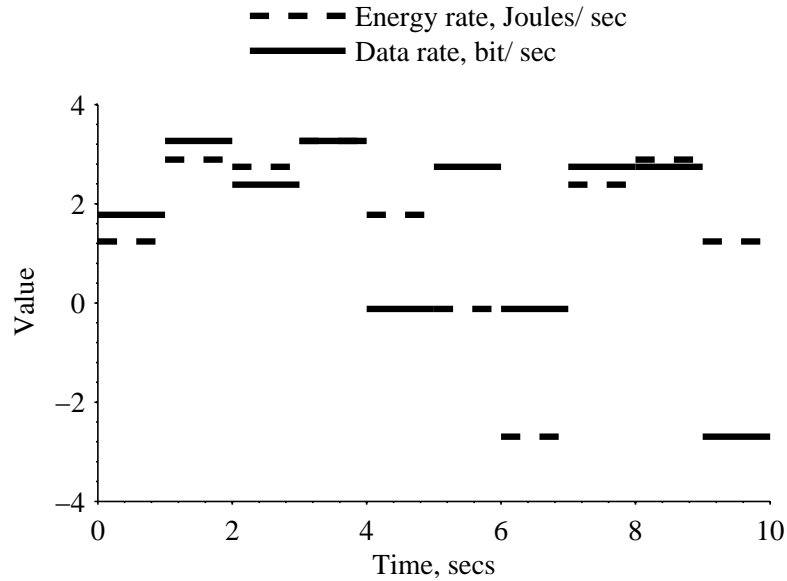
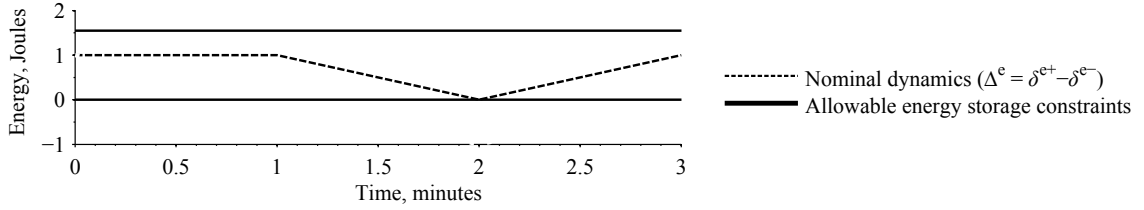


Figure 4.10: Example instance with PWL dynamics

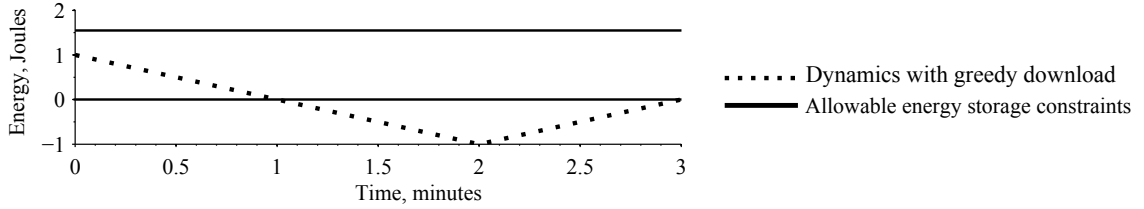
present a constructive algorithm to determine when to download within an interval and, in the process, assess the feasibility of a given interval with respect to a given *UCF* solution.

In general, a greedy approach to downloading seems advantageous: We should download whenever possible, starting from the beginning of the interval. Conversely, delaying download until later in the interval can have the negative impacts of: (a) energy and data buffer spillage (i.e. the need to spill energy and/or data that will be required for downloading or nominal operations at a future time); and (b) running out of time within the interval before completely downloading q_i^* . Such a purely greedy approach fails, however, when energy and/or data must be reserved to meet the nominal operating conditions of the satellite later during the interval. A simple example of this is given in Figure 4.11.

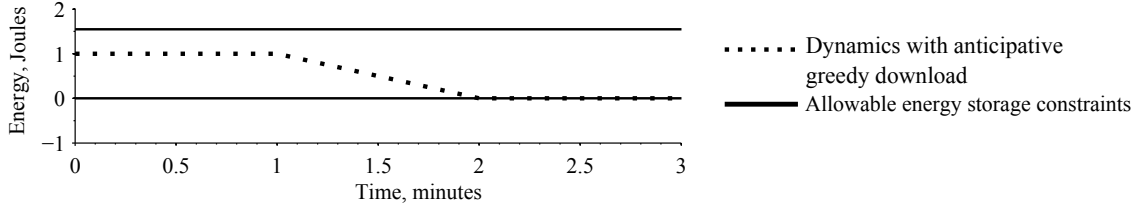
Figure 4.11(a) demonstrates a simple case where a greedy approach yields infeasibility, while other feasible download schedules exist. Figure 4.11(a) depicts the energy available for downloading and nominal operations. Specifically, the interval starts with one Joule of energy in storage and no net change in energy over the first linear segment. In the next linear segment, one Joule per minute is consumed for nominal operations. In the third linear segment, one Joule per minute is acquired by solar energy. Now suppose that the *UCF* solution is to download for one minute over this interval, at a corresponding energy utilization rate of 1 Joule per minute. In Figure 4.11(b), we see the energy level assuming that the download occurs during the first minute. Assume there is sufficient data available



(a) Nominal Energy Dynamics



(b) Dynamics with Greedy Download (Infeasible)



(c) Dynamics with Anticipative Greedy Download (Feasible)

Figure 4.11: Single interval instance of *SMSP* with piece-wise linear dynamics where a purely greedy download approach (downloading during the first segment) results in infeasibilities but an anticipative greedy approach (downloading during the third segment) is feasible.

at the start of the interval for the download and no nominal data consumption throughout the interval such that download can occur at any time during the interval. By using all of the available energy at the beginning of the interval (during the first linear segment), there is no energy available in the minute for nominal operations, leading to infeasibility. On the other hand, as seen in Figure 4.11(c), if we wait until the final third of the interval to download, then the initial energy can be used for the nominal operations, while the download uses the energy as it is acquired in the final minute.

For any given iteration of *NLSA*, we introduce *Anticipative Greedy Assign and Check Algorithm (AGACA)* as a way to check the feasibility throughout the duration of each interval i with respect to a solution $\{x_i^*, q_i^*, e_i^*, e_{i+1}^*, d_i^*, d_{i+1}^*\}$ from *UCF*. We begin by dividing the interval into its linear segments, starting a new segment whenever the energy or data rate changes. For example, Figure 4.11(a) depicts ten linear segments and Figure 4.11 de-

picts three linear segments. Let J_i denote the set of linear segments for interval I , and let the nominal energy and data acquisition and consumption (independent of download) during the interval i be denoted: $\Delta_j^e = \delta_j^{e+} - \delta_j^{e-}$ and $\Delta_j^d = \delta_j^{d+} - \delta_j^{d-}$, $\forall j \in J_i$.

By *Theorem 1*, we know that for an interval over which the data and energy rates have constant values, the system dynamics will be feasible throughout the duration of the interval so long as they are feasible at the beginning and end of the interval. Thus, our approach in *AGACA* is to move through each linear segment in sequence, determining how much to download throughout the segment while holding adequate data and energy in reserve for future linear segments within the current interval. We begin *AGACA* with e_i energy and d_i data, respectively. For each successive sub-interval $j = 1, 2, 3, \dots, |J_i|$, we first determine the maximum amount that can be downloaded during linear segment j . This amount is the minimum of four terms:

1.
$$\beta_{j,1} = q_i^* - \sum_{k=1}^{j-1} d_{dl,k}, \quad (4.27)$$

which is the remaining data to completely download q_i^* .

2.
$$\beta_{j,2}^* = \phi_i^*(t_{i,j+1} - t_{i,j}), \quad (4.28)$$

which is the maximum data that can feasibly be downloaded given the duration of the linear segment j .

3.
$$\beta_{j,3} = \frac{1}{\alpha_i^*} \left(e_j - e_{\min} + \min \left[0, \min_{p \in \{j, \dots, m-1\}} \left\{ \sum_{k=j}^p \Delta_k^e \right\} \right] \right), \quad (4.29)$$

which is the maximum data that can feasibly be downloaded to ensure all future energy constraints are satisfied. By *Theorem 1*, if the energy constraints are satisfied at the start and end of every linear segment, then feasibility is guaranteed throughout the complete segment. This expression ensures feasibility at the end of the linear segment with the lowest energy level, assuming we download $\beta_{j,3}$ and consider all future nominal dynamics.

4.
$$\beta_{j,4} = d_i - d_{\min} + \min \left[0, \min_{p \in \{j, \dots, m-1\}} \left\{ \sum_{k=j}^p \Delta_k^d \right\} \right], \quad (4.30)$$

which is the maximum data that can feasibly be downloaded to ensure all future energy constraints are satisfied, analogous to the energy constraint described above.

If the first term dominates, then we can fully download the proscribed amount q_i^* and we terminate *AGACA* with a certificate of feasibility for the interval and a corresponding feasible download schedule. Otherwise, it will be necessary to continue to linear segment $j + 1$. Before doing so, we update the amount of stored energy and data at the end of linear segment j ,

$$e_{j+1} = \min\{e_{max}, e_j + \Delta_j^e - \alpha_i^* d_{dl,j}\}, \quad (4.31)$$

$$d_{j+1} = \min\{e_{max}, e_j + \Delta_j^e - d_{dl,j}\}. \quad (4.32)$$

Note that this may include spilling if necessary.

At this point, we also check that we have enough remaining time, energy, and data to be potentially feasible over the remaining duration of the interval. In particular, Eqs. 4.33-4.35 must be satisfied, which enforce there must be sufficient time, energy, and data to completely download q_i^* during the remainder of the interval.

$$t_{i+1} - t_{j+1} \geq \frac{\beta_4}{\phi_{io}^*} \quad (4.33)$$

$$e_j - e_{i+1} + \sum_{k=j+1}^m \Delta_k^e \geq \frac{\alpha_i^*}{\beta_4} \quad (4.34)$$

$$d_j - d_{i+1} + \sum_{k=j+1}^m \Delta_k^d \geq \beta_4 \quad (4.35)$$

If these three conditions are satisfied, we repeat the above process for the next linear segment. Otherwise, we exit *AGACA* with a certificate of infeasibility.

4.4.2.2 *NLSA* Proof of Optimality

Theorem 2 *NLSA* yields a feasible (and thus optimal) solution in a finite number of iterations for instances of *SMSP* with PWL dynamics.

Proof 2 *NLSA* uses *UCF* to solve instances of *SMSP* with an updated set of current intervals, I^k , at each iteration. There are a finite number of linear segments in data over a finite planning horizon with PWL dynamics, thus there are a finite number of locations to be split. In the limit, when every linear segment is an interval, the *UCF* solution is feasible by *Theorem 1*. There are finite number of locations where the linear segments start or end, thus *NLSA* converges to a feasible (thus optimal) solution in a finite number of iterations.

4.4.2.3 Computational Results

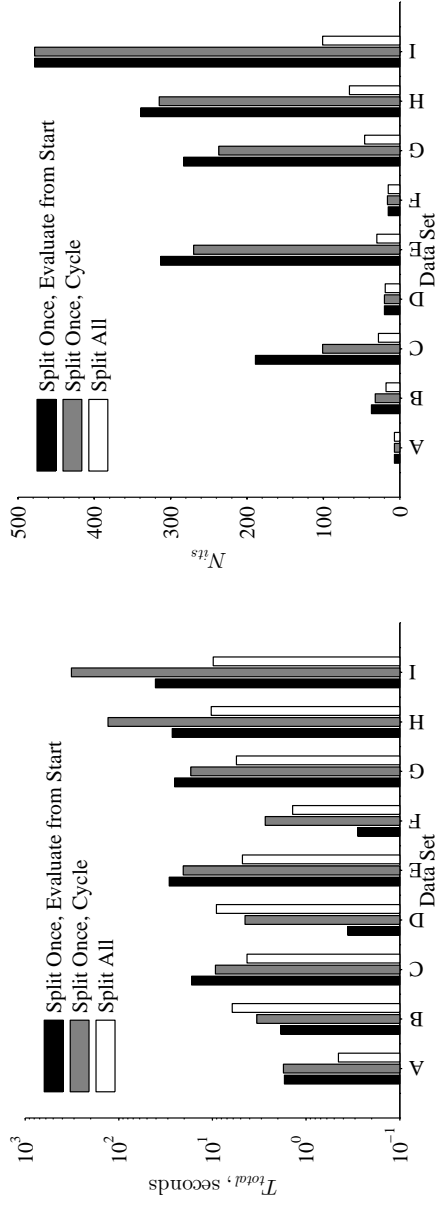
In this section we exploit *Theorem 2* to solve solving instances of *SMSP* with PWL dynamics. We provide computational results when *NLSA* is used to solve PWL instances of *SMSP* with diverse approaches for checking the feasibility of solutions.

To implement **NSLA!**, in addition to the strategy for checking the feasibility and splitting individual intervals, as described in Section 4.4.2.1, we require an approach for *which* intervals to check at every iteration, i.e. where to start checking and how many intervals to check. We have developed three approaches for determining which intervals to check:

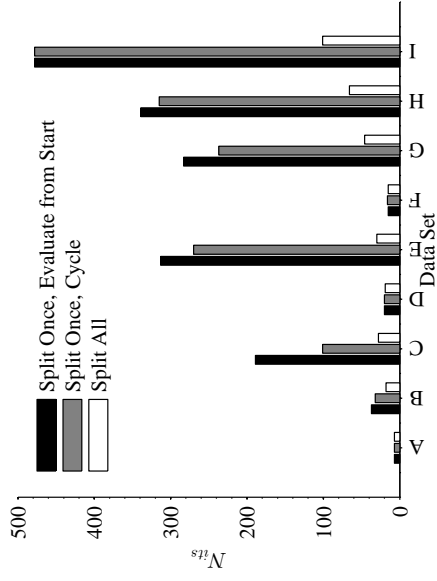
1. *Split Once, Evaluate from Start*: Starts at the first interval $i = 1$ and proceeds forward, checking each interval, until an infeasible interval is found and splits this and only this interval.
2. *Split Once, Cycle*: Starts at interval that was split in the previous algorithm iteration and proceeds forward, checking each interval, until an infeasible interval is found and splits this and only this interval.
3. *Split All*: Finds and splits all intervals with infeasibilities.

These three proposed approaches provide different potential advantages and disadvantages towards minimizing overall computational effort. To evaluate computational effort, we are mainly interested in the total solve time, $T_{total} = T_{ucf} + T_{eval}$, where T_{ucf} is the *UCF* solve time and T_{eval} is the evaluation time to find infeasible intervals. The *UCF*, evaluation, and total solve times are a function of the number of iterations, N_{its} . The *Split Once* approaches introduce only a single new interval per iteration and thus aim to minimize *UCF* problem size, and thus T_{ucf}/N_{its} . In the case that infeasibilities do not tend to appear earlier in the planning horizon than the location of the last split, the *Split Once, Cycle* strategy will be more efficient than the *Split Once, Evaluate from Start* strategy since it avoids redundant feasibility evaluations (and will yield smaller T_{eval}/N_{its} values). The *Cut All* strategy is designed to minimize N_{its} by splitting all infeasible intervals at each iteration, however this may yield larger T_{eval}/N_{its} and T_{ucf}/N_{its} values.

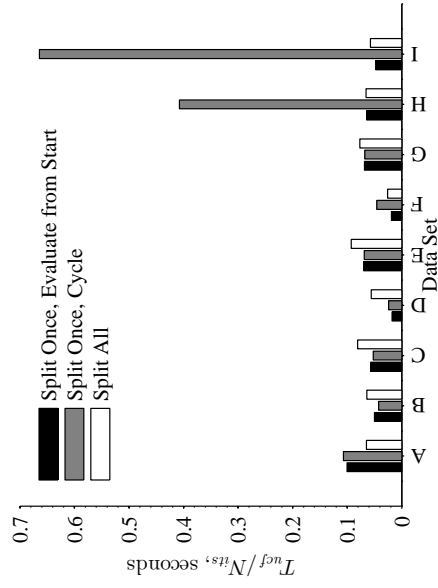
Results comparing the three approaches described above are summarized in Figure 4.12 for the data sets described in Table 4.8. There was no significant or consistent difference in T_{total} between the two *Split Once* approaches. As expected, the *Cut Once* approaches generally have higher T_{eval}/N_{its} values relative to the *Cut All* strategy, however this has a negligible contribution to T_{total} because it is up to two orders of magnitude smaller than T_{ucf}/N_{its} . The *Cut All* strategy requires up to five times fewer N_{its} relative to the *Cut Once* approaches, and thus has notably lower average T_{total} values for most data sets.



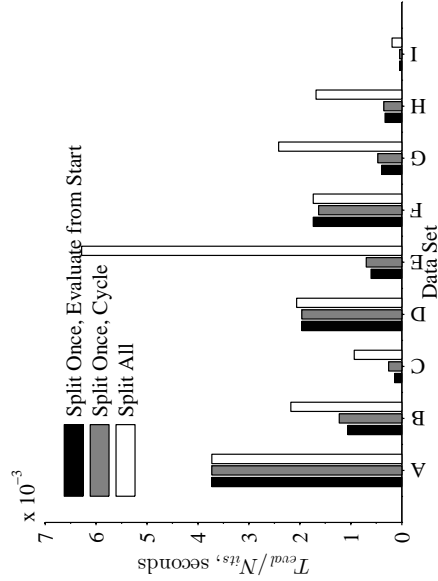
(a) Computational Solve Time (T_{total})



(b) Number of Algorithm Iterations (N_{its})

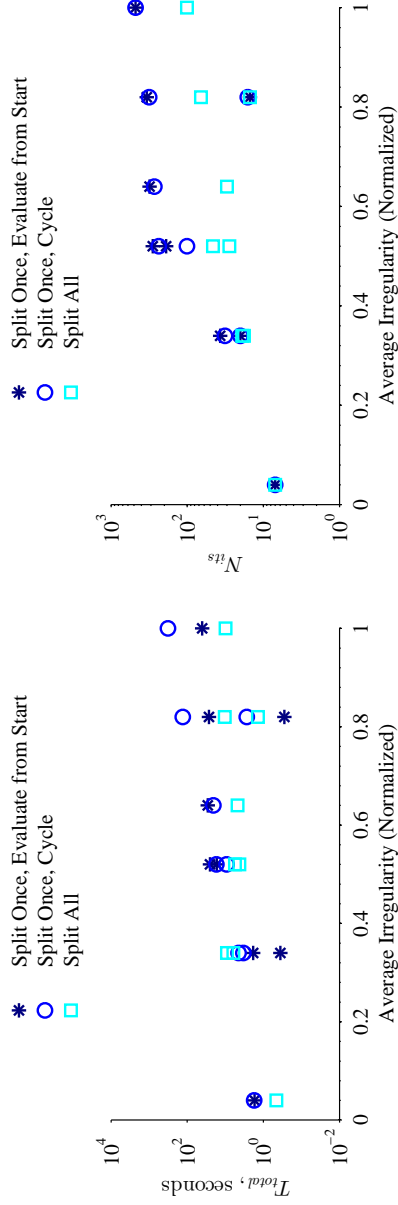


(c) Average UCF Solve Time per Iteration (T_{ucf}/N_{its})

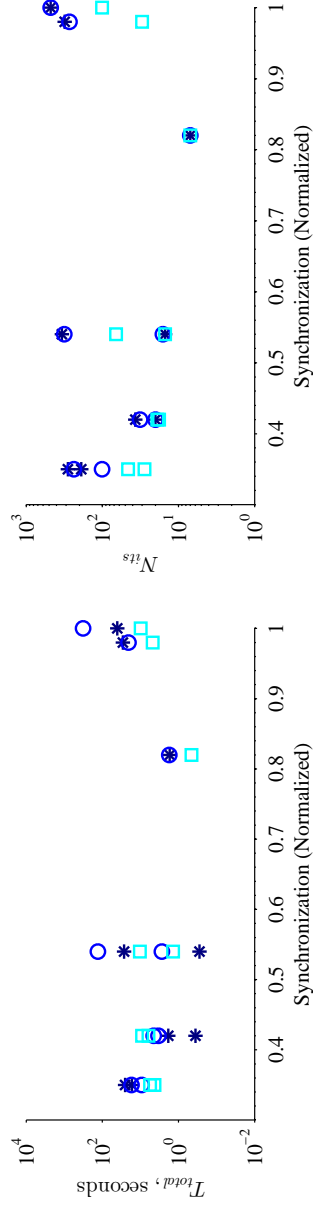


(d) Average Evaluation Time per Iteration (T_{eval}/N_{its})

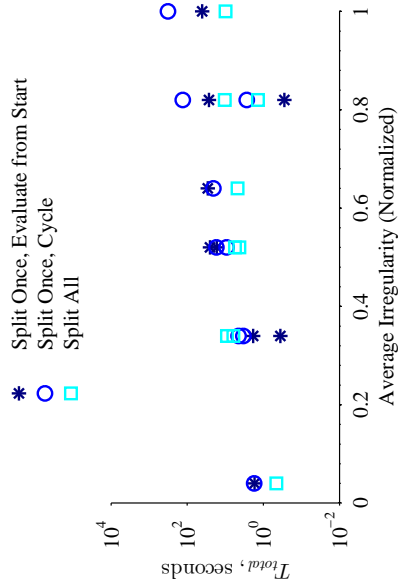
Figure 4.12: Comparison of *NLSA* approaches when applied to solve PWL instances of *SMSP* (data sets are summarized in Table 4.8).



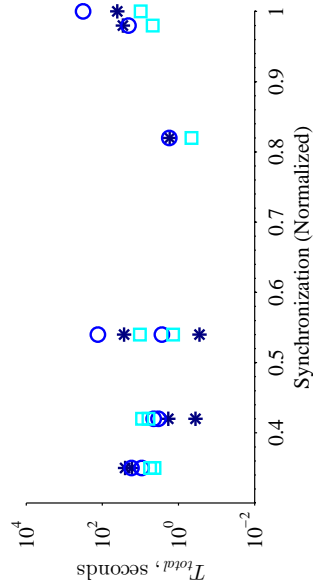
(a) Average Irregularity Effects on T_{total}



(b) Average Irregularity Effects on N_{itls}



(c) Synchronization Effects on T_{total}



(d) Synchronization Effects on N_{itls}

Figure 4.13: Effects of data characteristics on computational performance for *NLSA* with different check and split approaches. Results are shown for the data sets in Table 4.8.

Table 4.8: Data sets with diverse irregularity and synchronization characteristics. Normalized values are the raw values divided by the maximum raw value of the sets. Data sets have 100 linear segments per interval.

Data Set	Irregularity				Synchronization (ζ)	
	Energy (γ_e) (Raw)	Data (γ_d) (Raw)	Maximum (Normalized)	Average (Normalized)	(Raw)	(Normalized)
A	0.020	0.020	0.040	0.040	0.81	0.82
B	0.020	0.32	0.64	0.34	0.42	0.42
C	0.32	0.020	0.64	0.34	0.42	0.42
D	0.020	0.50	1.0	0.52	0.35	0.35
E	0.50	0.020	1.0	0.52	0.35	0.35
F	0.32	0.32	0.64	0.64	0.97	0.98
G	0.32	0.50	1.0	0.82	0.53	0.54
H	0.50	0.32	1.0	0.82	0.53	0.54
I	0.50	0.50	1.0	1.0	0.99	1.00

The data sets in Figure 4.12 have diverse data characteristics. In particular, they have diverse *irregularity*, a measure of the change in the dynamics during each interval relative to the buffer capacity (ζ), and *synchronization*, a measure of the similarity between energy and data dynamics (γ_e and γ_d), see analytic definitions in Appendix D. In Figure 4.13, the data sets are presented in order of increasing average irregularity from A to I (from left to right). As expected, T_{total} and N_{its} generally increase with irregularity. Data set F is an exception, with lower T_{total} and N_{its} relative to this trend, which we hypothesize is because its synchronization is relatively high, and thus it may be easier to solve. We expect data with higher synchronization to require lower T_{total} because the energy and data dynamics are similar.

Figure 4.13 shows the relationship between normalized data characteristics and T_{total} and N_{its} . There is a positive correlation between average irregularity and computational effort, see Figure 4.13(a). There is no observed correlation between maximum irregularity and computational effort (not shown). Furthermore, there was no observed correlation between synchronization and computational effort, see Figures 4.13(c) and 4.13(d).

4.5 Summary

Operating small spacecraft is challenging because they are highly constrained, with limited on-board power generation, control, storage, and computational resources. In Chapters 2

and 3 we observed how active constraints related to the spacecraft vehicle parameters, as well as combined orbit and network parameters, may limit mission potential. Thus, optimizing satellite schedules is become increasingly important to ensure resources, such as time, energy, and data, are efficiently allocated. In this chapter we presented an optimization formulation, algorithms, and theoretical and computational results that address the operational spacecraft scheduling problem.

We have defined the *Single-Satellite Scheduling Problem (SMSP)*, which maximizes the total data downloaded from a single satellite to a ground station network, subject to realistic constraints on available and stored energy and data. We presented the *Under-Constrained Formulation (UCF)* to solve instances of *SMSP*. We discussed theoretical and computational results when *UCF* is applied to solve linear real-world and generic problem instances. To address problem instances with non-linear dynamics, we developed a *Non-Linear SMSP Algorithm (NLSA)*. *NLSA* is guaranteed to yield feasible, and thus optimal, solutions for data instances with piece-wise linear dynamics and can prevent computational intractability when solving large problem instances. We discussed computational tractability of *NLSA* in the context of piece-wise linear instances with diverse data characteristics.

The optimization formulation presented in this chapter provides a fundamental approach for optimizing spacecraft missions that is used throughout the remainder of this thesis. Furthermore, the theoretical insights and computational results from this chapter are informative when solving problems with additional complexity. The formulation is extended in Chapter 5 to develop a formulation that can solve a general class of mission scenarios. Furthermore, we use the optimization formulation in this chapter to investigate how realistic sources of stochasticity impacts problem solutions in Chapter 6.

CHAPTER 5

Deterministic Optimization: Extensions and Applications

In the previous chapter we introduced an optimization formulation for modeling and solving instances of *SMSP*, where the decisions were when and how to download data to a global network of ground stations. In *SMSP*, we assumed the payload operations were determined a priori and the acquisition of data and consumption of energy related to these operations were incorporated in the nominal dynamics, δ^{d+} and δ^{e-} , respectively. However, for many highly-constrained missions, it is important to consider *both* download and payload decisions in the optimization problem as they are coupled through the constrained on-board energy and data. For example, consider the scenario where more payload data is collected throughout a planning horizon than can conceivably be downloaded due to on-board energy constraints. In this case, it may be advantageous to use less energy and time to collect payload data and more energy and time downloading to maximize the total amount of data download and the optimal allocation of energy resources.

In this chapter, we use the modeling framework from Chapter 2 to develop a generalized optimization formulation that can be applied to model a broad range of operational problems Section 5.1. Next, we describe a specific operational satellite problem that includes additional operational decisions (and the interactions of these decisions with the rest of the problem) in Section 5.2.1. The generalized formulation is applied to develop a specific formulation to solve this satellite operational problem in Section 5.2.2. We solve diverse mission scenarios and perform vehicle and network design trades in Sections 5.3 and 5.4, including real-world CubeSat missions and interplanetary mission architectures. The chapter is summarized and future extensions and applications are discussed in Section 5.5.

5.1 Generalized Under-Constrained Formulation (*GUCF*)

In this section we introduce the *Generalized Under-Constrained Formulation (GUCF)*, which is a generalized form of *UCF*. *GUCF* can be used to solve a broad class of operational satellite problems.

GUCF uses a generic approach for modeling decisions, variables, constraints, and the objective, such that it can be used as a template for specific optimization problems. Instead of directly modeling on-board energy and data as in *UCF*, *GUCF* models generic commodities, C . Satellite commodities, $c \in C$, are satellite states that are necessary to support the problem objectives and constraints. Commodities are collected from the environment, exchanged during functional operations by the subsystems, and may be stored on-board the spacecraft in finite buffers for future use. *GUCF* models commodity dynamics, constraints, and interactions with other commodities and the objective. *GUCF* also models generic functions, where F is the set of possible functions.

There are two challenging characteristics of the operational satellite scheduling problem. First, the realistic satellite operational problem is continuous in time because the orbit and attitude dynamics, and thus the commodity acquisition and consumption, are continuous. Second, there is a discontinuity in the problem dynamics caused by the upper bound on the allowable amount of stored commodities, which results in a difficult non-linear optimization problem. These challenges are managed by discretizing the problem into a finite set of intervals I and imposing a commodity storage constraint at the start and end of each interval, as done in the *UCF* in Section 4.2.2.

Note that F_i , the set of available functions during interval i , and O_{if} , the set of options for performing this function, are dependent on the opportunities during interval i , which depends on the geographical or temporal constraints of the problem (e.g. targets of interest or ground stations).

5.1.1 Notation

Sets and Subsets

- I is the set of intervals over the planning horizon
- F is the set of all satellite functions that have operational decisions
- $F_i \subseteq F$ is the subset of functions with opportunities to operate during interval i , $\forall i \in I$
- C is the set of all commodities

- $C_f \subseteq C$ is the subset of commodities operated on by function f , $\forall f \in F$
- O_f is the set of download options for function f , $\forall f \in F$
- $O_{if} \subseteq O$ is the subset of download options available during interval i using function f , $\forall i \in I, \forall f \in F$

Parameters

- t_i is the start time of interval i , measured in seconds, $\forall i \in I$
- t_{i+1} is the end time of interval i , which coincides with the start of interval $i + 1$, measured in seconds, $\forall i \in I$
- Δt_i is the duration of time interval i ($\Delta t_i = t_{i+1} - t_i$), measured in seconds, $\forall i \in I$
- $z_{c,min}$ and $z_{c,max}$ are the minimum and maximum storage capacity of commodity c , measured in units of commodity c , $\forall c \in C$
- $z_{c,start}$ is the amount of commodity c stored on-board the satellite at the beginning of the planning horizon, measured in units of that commodity, $\forall c \in C$
- $\delta_{ic}^+ \geq 0$ is the amount of commodity c acquired nominally (independent of operational decisions) during interval i , measured in units of that commodity, $\forall i \in I, c \in C$
- $\delta_{ic}^- \geq 0$ is the amount of commodity c consumed nominally (independent of operational decisions) during interval i , measured in units of that commodity, $\forall i \in I, c \in C$
- G_{ifco} is the maximum allowable transfer of commodity c due to function f using option o during interval i in units of that commodity, $\forall i \in I, f \in F_i, c \in C_f, o \in O_{if}$.
- ϕ_{ifco} is the linear rate of change of commodity c due to function f using option o during interval i in units of that commodity per second $\forall i \in I, f \in F_i, c \in C_f, o \in O_{if}$.¹

Variables

- $x_{ifco} \in \{0, 1\}$ is the binary variable representing the decision to perform function f using option o during interval i (1 represents the decision to perform the operation while 0 represents the decision not to), and it is unitless, $\forall i \in I, f \in F_i, o \in O_{if}$
- $q_{ifco} \in \mathbb{R}$ is the amount of commodity c transferred (acquired or consumed) by function f using option o during interval i , in units of that commodity, $\forall i \in I, f \in F_i, c \in C_f, o \in O_{if}$

¹This parameter exists only if the rate of change of the commodity c is constant.

- $z_{ic} \in \mathbb{R}^+$ is the amount of commodity c stored on-board the satellite at the beginning of interval i , measured in units of that commodity, $\forall i \in I, c \in C$
- $h_{ic} \in \mathbb{R}^+$ is the amount of excess commodity c spilled during the interval i , measured in units of that commodity, $\forall i \in I, c \in C$

5.1.2 Formulation

In *GUCF*, the constraints on commodities are enforced at the end of each interval, similar to the *UCF* from Section 4.2.2. The formulation is as follows:

$$\max z_{nc^*} \quad (5.1)$$

s.t.

$$\sum_{o \in O_i} x_{if_o} \leq 1 \quad \forall i \in I, f \in F_i, o \in O_{if} \quad (5.2)$$

$$z_{c,0} = z_{c,start} \quad \forall c \in C \quad (5.3)$$

$$z_{c,min} \leq z_{ic} \leq z_{c,max} \quad \forall i \in I, c \in C \quad (5.4)$$

$$z_{i+1,c} = z_{ic} + \delta_{ic}^+ - \delta_{ic}^- + \sum_{f \in F_i} \sum_{o \in O_{if}} q_{ifco} - h_{ic} \quad \forall i \in I, c \in C \quad (5.5)$$

$$q_{ifco} \leq G_{ifco}(x_{if_o}, \Delta t_i) \quad \forall i \in I, f \in F_i, o \in O_{if}, c \in C \quad (5.6)$$

$$q_{ifco} = f(q_{if\hat{c}o}) \quad \forall i \in I, f \in F_i, o \in O_{if}, \{c | c \neq \hat{c} \in C\} \quad (5.7)$$

$$x_{if_o} \in \{0, 1\} \quad \forall i \in I, f \in F_i, o \in O_{if} \quad (5.8)$$

$$q_{ifco} \in \mathbb{R}^+ \quad \forall i \in I, f \in F_i, o \in O_{if}, c \in C \quad (5.9)$$

The objective, 5.1, maximizes the total amount of a specific commodity, c^* , which is identified by *. Constraint 5.2 enforces that during every interval i , each function f can only operate using a single option o from the set of options available for that function during that

interval, O_{if} . Constraints 5.3-5.5 enforce the storage and dynamics for the set of commodities, C , for each interval $i \in I$. Constraint 5.3 initializes the quantity of stored commodity c stored on-board the satellite at the start of the planning horizon ($i = 0$) and Constraint 5.4 enforces that the amount of commodity c never violates the upper and lower bounds storage constraints. Constraint 5.5 updates the amount of stored commodity c on-board the satellite at every interval i based on nominal dynamics and the executed functions. The amount of commodity c stored on-board the satellite at interval $z_{i+1,c}$ is equal to the amount stored at the previous interval, z_{ic} , plus any commodity c acquired and consumed throughout the interval due to nominal dynamics (independent of subsystem functions), $\delta_{ic}^+ - \delta_{ic}^-$, plus any change in the commodity c due to subsystem functions (which may be positive or negative), minus any spillage of the commodity. Spillage of a commodity occurs to prevent overfilling a storage buffer, i.e. to prevent violating Constraint 5.4. Constraint 5.6 enforces that the amount of commodity c transferred during interval i by function f and option o is less than the maximum allowable transfer based on these conditions, $G_{ifco}(x_{ifo}, \Delta t_i)$. Constraint 5.7 enforces the relationship between the amount of commodity \hat{c} and all other commodities $c \in C$ for every interval i , function f , and option o .

In the case that commodity c is transferred at a constant rate, Constraint 5.6 is expressed:

$$G_{ifco}(x_{ifo}, \Delta t_i) = \phi_{ifco} \cdot x_{ifo} \cdot \Delta t_i \quad \forall i \in I, f \in F_i, c \in C_{if}, o \in O_{if}. \quad (5.10)$$

In the case where the relationships between the commodities transferred for a given interval, function, and option are linear, Constraint 5.7 is expressed:

$$q_{ifco} \propto q_{if\hat{c}o} \quad \forall i \in I, f \in F_i, o \in O_{if}, \{c | c \neq \hat{c} \in C\}. \quad (5.11)$$

When 5.10 and 5.11 are true, Constraints 5.6 and 5.7 are linear and the complete *GUCF* is linear.

Similar to *UCF*, *GUCF* has both integer variables, $x_{io} \forall i \in I, o \in O_i$, and continuous variables, $q_{io} \forall i \in I, o \in O_i$, thus it is a MIP. Due to its structural similarity with *UCF*, by applying the same logic as in Section 4.3, it can be shown that if *GUCF* is linear, it is guaranteed to yield a feasible, and optimal solution, if one exists for a given problem with linear dynamics. In addition, if the problem is linear, we can use linear programming techniques to solve the problem in a computationally tractable and theoretically satisfactory way. This can be accomplished using a commercial software package such as CPLEX [177].

To implement *GUCF*, the user must specify the problem they are trying to solve, including commodities, decisions, dynamics, constraints, and objectives. They must then

define all the appropriate notation, described above, and then apply this notation to the optimization formulation described above in 5.1-5.9.

5.2 Single Operational Satellite Problem (*SOSP*)

5.2.1 Problem Description

In this section we describe *Single-Satellite Operational Scheduling Problem (SOSP)*, which is an extension of *SMSP*. In addition to the objective, states, and constraints specified by the *SMSP*, the *SMSP* includes operational decisions related to the spacecraft payload. These operational decisions include when and how to collect payload data, for example by a science payload or surveillance imager. Despite the similarities between the two problems, in this section we provide a complete stand-alone description of *SOSP* for completeness.

The objective of *SOSP* is to maximize the amount of data collected from a single satellite and downloaded to a network of geographically and functionally diverse ground stations. The decisions include when and how to perform mission operations. Thus, there are two types of functional decisions in *SOSP*: payload functions, which consist of acquiring data, and download functions, which involve transmitting data to ground stations. Opportunities for payload and download functions occur according to spatial and/or temporal constraints. As discussed in Section 4.1, the satellite orbits the Earth, targets of interest and ground stations come in and out of view, providing these opportunities. Opportunities for solar power collection depend on when the satellite is in the sun or in eclipse. These opportunities can be estimated a priori since we have good knowledge of the satellites' orbital position.

Functionally, the satellite consists of the subsystems that keep it operating nominally and one or multiple payloads, which accomplish the scientific, observational, or engineering mission goals. Satellite states include position, attitude, on-board stored energy, data, temperatures, and their time-dependent derivatives. The subsystems have functions that operate on the satellite states. Sample functions include collecting energy from the sun, collecting data from targets of interest, and downloading on-board data to a ground stations. For example, the set of options for downloading include the available data rates for communication. The two commodities considered in *SOSP* are the amount of on-board stored energy and data.

Table 5.1: Comparison of *SMSP* and *SOSP* and corresponding formulations

Problem	Formulation	Decisions	Objective
<i>SMSP</i>	<i>UCF</i>	Download: when and option	Maximize download
<i>SOSP</i>	<i>OUCF</i>	Download: when and option Payload Operations: when and what option	Maximize download

5.2.2 Problem Formulation

In this section we utilize *GUCF* to develop an *Operational Under-Constrained Formulation (OUCF)*, to solve the *SOSP*. For clarity, the various problems and formulations introduced throughout this thesis are summarized in Table 5.1. Below we define the specific functions, commodities, and decisions for *OUCF*, which are summarized in Table 5.2:

- *Functions (F)*: collecting data from the payload subsystem(s) and downloading data to ground stations
- *Commodities (C)*: on-board energy, on-board data, and downloaded data
- *Commodities operated on by functions ($C_f \subseteq C$)*: defined in Table 5.2
- *Binary decisions (x_{if_o})*: decisions for which option to use when performing payload operations and downloading data (1 represents the decision to perform the operation while 0 represents the decision not to)
- *Continuous decisions ($q_{if_{co}}$)*: the amount of data acquired during payload operations and amount of data consumed during data download
- *Commodity that is maximized (c^*)*: downloaded data

Table 5.2: Relationship between commodities and functions in *OUCF*.

Function	Commodities (<i>C</i>)	
	Stored energy, $e(t)$	Stored data, $d(t)$
Collect Payload-Specific Data	consumes energy	acquires data
Download Data	consumes data	consumes data

These sets, subsets, parameters, and variables described above are then “plugged-in” to the objective and constraints described in 5.1 -5.9 to obtain *OUCF*. In *OUCF*, the amount of energy and data consumed or acquired are linearly related for both payload and download

operations (Eq. 5.7). For example, the amount of energy consumed for downloads is directly proportional to the amount of data downloaded, i.e. $\Delta d = \alpha_{ifco} q_{ifco} \forall i \in I, f \in F_i, o \in O_{if}, c \in C$, where α is measured in Joules-per-bit and was defined in Section 4.1.

For most satellite operational problems, it is reasonable to assume the dynamics are linear because the commodities we model (e.g. energy and data) are generally collected and consumed at constant rates throughout typical intervals. Thus, as discussed in Section 5.1, if the dynamics are linear, a solution is guaranteed to be feasible (and thus optimal). Furthermore, we can exploit the linear formulation and use commercial software to solve the problem. The computations in this section are performed on an Intel Core i7 2.8 GHz processor with 8 GB of memory using the IBM ILOG Optimization Studio (CPLEX) 12.1 Matlab API software package [177]. All test cases in this section solve in under ten minutes.

5.3 Diverse LEO CubeSat Missions Application

This section investigates solving instances of *SMSP* and *SOSP* with data representative of ten diverse small satellite mission instances. The representative missions are from the Small Satellite Survey, which are currently on-orbit or will be launched in the near future, and are summarized in Table 5.3. The Radio Aurora eXplorer (RAX) has Type 1 *targeted* data collection, collecting data when in view of ground-based radar stations, for example at Poker Flat, Alaska. The Naval Postgraduate School Solar Cell Array Tester (NPS-SCAT), Focused Investigations of Relativistic Electron Burst, Intensity, Range, and Dynamics (FIRE-BIRD), Michigan Multipurpose Minisatellite (MCubed), Kentucky Space’s KySat, Rapid-prototyped Microelectromechanical system Propulsion And Radiation Test CUBEflow Satellite (RAMPART) are Type 2 *repeating* data collection missions. The CubeSat Investigating Atmospheric Density Response to Extreme Driving (CADRE),² Colorado Student Space Weather Experiment (CSSWE), Dynamic Ionosphere Cubesat Experiment (DICE), Explorer - 1 [Prime] (E1P) are Type 3 *continuous* data collection missions. The satellite missions summarized in Table 5.3 are used in our optimization results in the coming sections.³

In the upcoming examples, we consider the satellite in Table 5.3 communicating to three types of networks:

²For the CADRE mission, we assume the science experiment is part of the nominal operations, and the operational decisions are related to the operation of the GPS subsystem.

³Note that these types differ from the Section 2.3.

Table 5.3: Satellite mission parameters from Small Satellite Survey. The data collection strategies are: Type 1 is *targeted*, Type 2 is *repeating*, and Type 3 is *continuous*.

Mission Parameter	Units	RAX-2	CADRE	CSSWE	NPS-SCAT	DICE	FIRE-BIRD	MCubed	KySat	RAMP-ART	EIP
Data Collection Type	-	1	3	3	2	3	2	2	2	2	3
Perigee Altitude	km	330	500	450	700	350	550	330	630	350	350
Apogee Altitude	km	810	500	800	500	820	550	810	640	1200	800
Orbital Inclination	degrees	101.8	81	55	75	102	60	102	98	72	101
Maximum Energy Capacity	kJoules	115	80.4	66.2	36.9	72.0	58.3	58.6	57.6	$1.35 \cdot 10^5$	$1.04 \cdot 10^5$
Maximum Data Capacity	Gbits	16	16	16	2	64	16	17	16	0.032	0.016
Maximum Depth of Discharge	%	20	30	85	10	29	6	20	3	10	4.5
Nominal Power Collection (sun)	Watts	7	15	7	1.5	3	3	1.5	1.5	3	3
Nominal Power Consumption	Watts	0.75	8.5	0.15	0.75	1.5	1.5	0.25	0.75	1.5	0.75
Experiment Power Consumption	Watts	3.6	1.3	0.25	0.25	1.5	1.5	5	0.25	1.5	0.25
Nominal Data Collection	Mbits/sec	$2 \cdot 10^{-4}$	0.2	$7 \cdot 10^{-6}$	$3 \cdot 10^{-6}$	0.01	$2 \cdot 10^{-4}$	$3 \cdot 10^{-6}$	$3 \cdot 10^{-5}$	$8 \cdot 10^{-6}$	10^{-6}
Payload Data Collection Rate	Mbits/sec	4	$2 \cdot 10^{-4}$	$5 \cdot 10^{-5}$	$3 \cdot 10^{-6}$	0.01	0.005	50	$3 \cdot 10^4$	$3 \cdot 10^{-5}$	$6 \cdot 10^{-6}$
Download Power Consumption	Watts	3	10	5	3	1.5	3	0.25	1.5	5	0.25
Download Data Rate	kbits/sec	9.6	250	9.6	9.6	1500	0.22	9.6	9.6	9.6	1.2
Minimum Required Download	MBytes/day	1.0	30	0.27	0.036	0.29	0.039	0.84	0.14	0.0021	0.016
Number of Ground Stations	-	7	1	1	1	2	1	1	1	3	1

- True Ground Network (TGN) ground stations that are intended to support each mission, see Table 5.3.
- Large Ground Network (LGN) unique collection of 14 ground stations that are intended to support all the missions in Table 5.3.⁴
- Very Large Ground Network (VLGN): collection of 30 globally distributed ground stations from the Ground Station Survey [1].

To determine the opportunities for communication, we assume the satellites must have an elevation angle (relative to the ground station horizon) that exceeds 30° , which is reasonable from our experience with the RAX-2 mission.

5.3.1 Optimal Results for Realistic CubeSat Missions

This section compares the optimal solutions of instances of *SMSP* and *SOSP* with data from the satellite missions in Table 5.3 communicating to the TGN, LGN, and VLGN. Recall that *SMSP* includes decisions on download operations while *SOSP* includes decisions on both payload and download operations (summarized in Table 5.1).

Figure 5.1 compares the minimum download requirements to the results for the optimized *SMSP* and *SOSP* schedules for representative satellites missions communicating to the TGN and VLGN. The ability of the optimized schedules to satisfy minimum download requirements can be assessed by comparison to the first bar for each mission in Figure 5.1. The RAX-2, CADRE, CSSWE, KySat, RAMPART, and E1P missions exceed the minimum download requirement using both the *SMSP* and *SOSP* formulations while the NPS-SCAT, FIREBIRD, and MCubed missions are not able to achieve the minimum download requirement using either formulation.

The CADRE, DICE, and FIREBIRD missions are infeasible when posed as an instance of *SMSP*, which is why they do not have the second bar representing the *SMSP* solution. This infeasibility indicates that these missions cannot support performing payload mission operations at every opportunity and must make payload operational decisions considering the full mission and upcoming events. For example, by performing energy-consuming operations earlier in the mission, the satellite may not be able to survive eclipse, when there is no opportunity for solar power collection. These infeasible scenarios are motivation for posing the problem as a *SOSP*, where operational decisions are considered.

Many of the missions show a significant increase in download potential when communicating to the Very Large Ground Network (VLGN), relative to when communicating to

⁴This network consists of the unique collection of the ground stations used for the missions in Table 5.3.

their TGNs, see Figures 5.1 to compare the networks. The NPS-SCAT and MCubed missions do not satisfy minimum download requirements when communicating to their TGNs; however exceed their requirements when communicating to the VLGN.

This type of analysis allows satellite designers and operators to realize the download potential when communicating to diverse networks and may motivate the need to modify the vehicle, ground network, or operational parameters. Next, we investigate how information on the feasibility and download potential based on optimized schedules varies with problem parameters. Throughout the remainder of this section we focus solely on *SOSP* which yielded feasible solutions for all the example mission scenarios in Figure 5.1

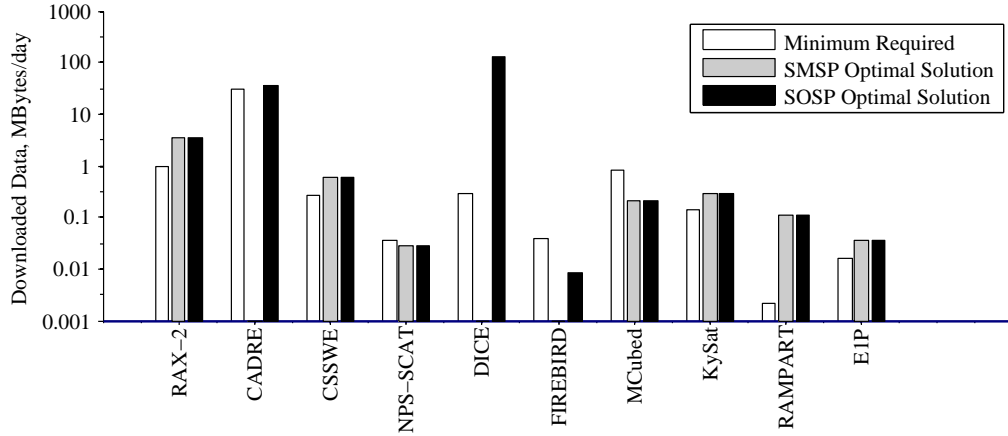
5.3.2 Sensitivity to Deterministic Problem Parameters

Beyond enabling the optimization of spacecraft mission schedules, as seen in the previous section, *SOSP* also enables exploration of the spacecraft vehicle and operational design space. For small representative examples, *SOSP* solves very quickly using linear programming techniques (i.e. most of the problems in the previous section solve in less than one minute). This enables the rapid optimization and assessment of spacecraft scheduling problems with variable input parameters such that spacecraft designers and operators can gain insight into the sensitivity of the solutions to these inputs. Using optimized schedules to perform design space trades in early design stages enables making informed and accurate spacecraft component sizing and logistical decisions. This is advantageous since design decisions can have great impact on later design and operational constraints and thus mission performance. In this section, we solve instances of *SOSP* due to its success in solving all missions in the previous subsection.

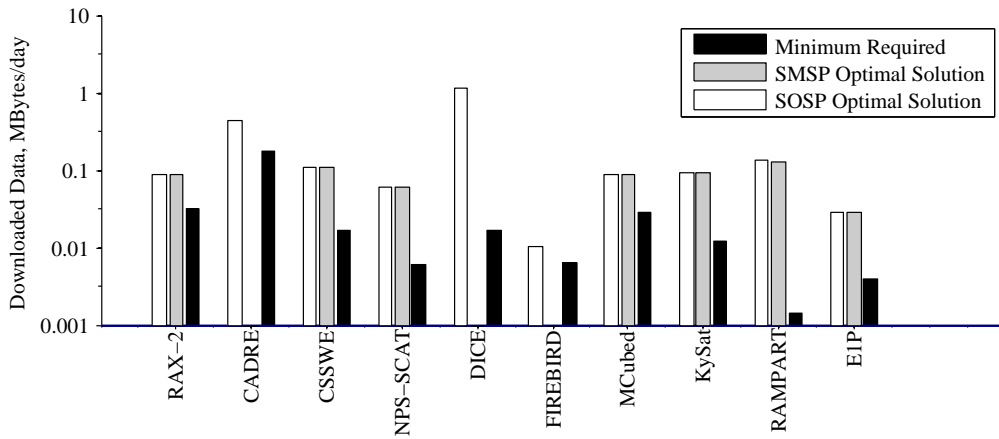
Next we investigate the sensitivity of optimized download solutions relative to deterministic vehicle and network parameters. The parameters are deterministic because they have a known, constant, and quantifiable impact on the solution objective, which is the quantity of downloaded data. This is useful for identifying the relationship between mission parameters and the active constraints limiting the feasibility and download potential of the missions. Throughout this section we assume the download efficiency is perfect ($\eta = 1$).

5.3.2.1 Power Generation and Battery Storage

During spacecraft design, sizing the solar cell array and battery is challenging. This is because the energy state is dynamic and constrained by the coupling of the solar cell array and battery decisions, and difficult to model prior to making operational decisions (i.e.



(a) Results when satellites are communicating to their TGNs with the number of ground stations from Table 5.3.



(b) Results when satellites are communicating to the VLGN consisting of 30 ground stations.

Figure 5.1: Comparison of mission download requirements to data downloaded with optimized schedules for the satellite missions from Table 5.3 for a one day mission scenario assuming perfect download efficiencies ($\eta = 1$). Results are shown for instances of *SMSP*, which considers only the download decisions, and *SOSP*, which considers both payload and download decisions. Optimal data downloads are shown on a log scale.

when the spacecraft will perform payload and download functions). Thus, we optimize the operational schedule for point designs, with a given battery and solar cell array, to investigate how these two factors impact download performance. This can help make informed design decisions in the presence of minimum download requirements and constraints on spacecraft volume, mass, or cost.

Figure 5.2 shows the optimal download solutions for two representative missions from Table 5.3 with variable power generation and battery storage values. The combined effect of variation in battery depth of discharge (along the x-axis) and collected power from the solar cells (along the y-axis) demonstrates the coupling between these two parameters in constraining the energy available to support spacecraft operations. For example, higher power collection results in more available energy, but battery storage capacity may limit the amount of energy that can be stored for future use.

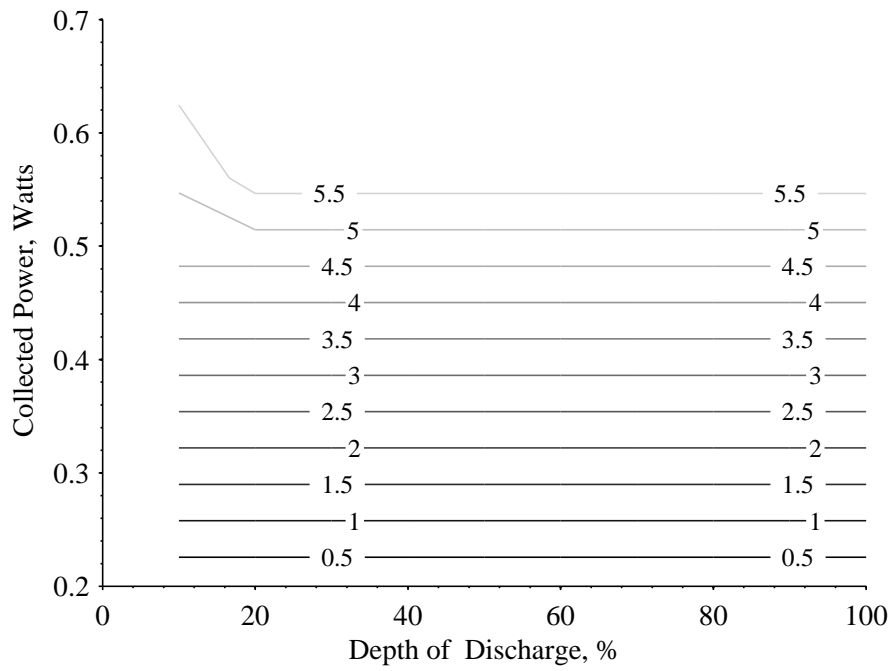
For both the CSSWE and DICE missions, the data downloaded is nearly independent of battery depth of discharge when the depth of discharge is sufficiently high. For high values of depth of discharge, the data downloaded scales linearly with collected power. For lower depths of discharge values, there's a non-linear trend in the download potential related to the trade-off between energy storage capacity and power collection. The same amount of data can be downloaded with less power and a greater depth of discharge as with higher power and a lower depth of discharge. This trade-off is represented by the line labeled 5.5 (representing data downloaded in MBytes/ day) for the CSSWE mission in Figure 5.2(a) which curves dramatically for low battery depths of discharge.

The other spacecraft missions from Table 5.3 were not as sensitive to combined variability in power generation and battery storage. In most cases, once sufficient power is collected and there is a sufficient battery capacity, the solution was feasible, the same quantity of downloaded data was achieved, independent of energy-related parameters.

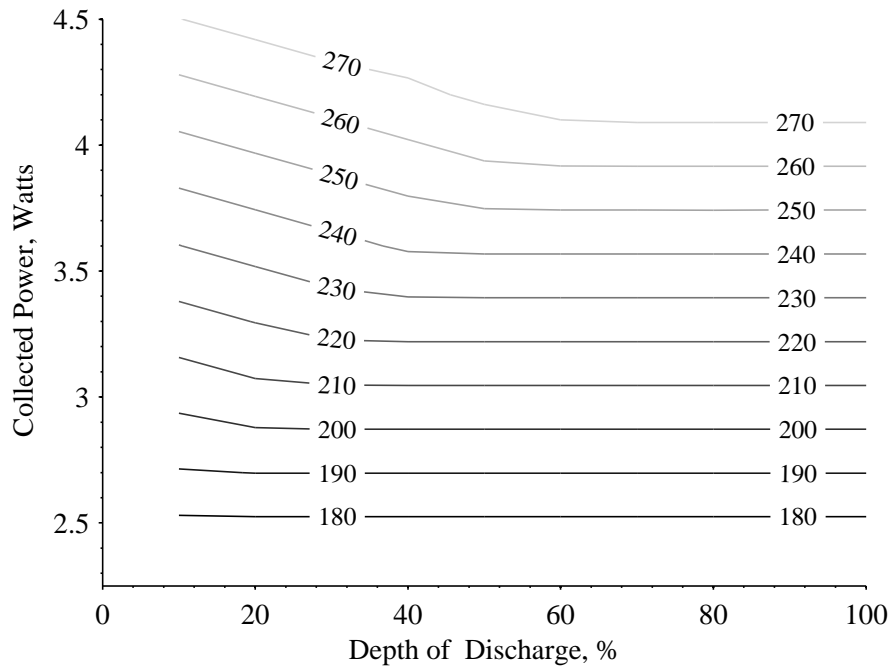
5.3.2.2 Download Data Rate

Most of the surveyed small spacecraft missions use relatively low- and constant-rate communication. This is due to the lack of existing higher-rate and variable-rate radios that are flight-ready or have flight heritage, and the lack of ground station infrastructure to support this type of communication. However, emerging radio technology and ground station capabilities are enabling the future generation of spacecraft to use higher and variable-rate radios, which may be particularly important for interplanetary missions (because the range distances, which impact the feasible data rates, can vary significantly). This section compares different communication data rates.

Figures 5.3 and 5.4 shows how the download potential scales with data rate for rep-



(a) CSSWE



(b) DICE

Figure 5.2: The sensitivity of optimized schedules relative to variations in power generation and battery depth of discharge. The lines on the contour plot represent the amount of data downloaded per day in MBytes/ day.

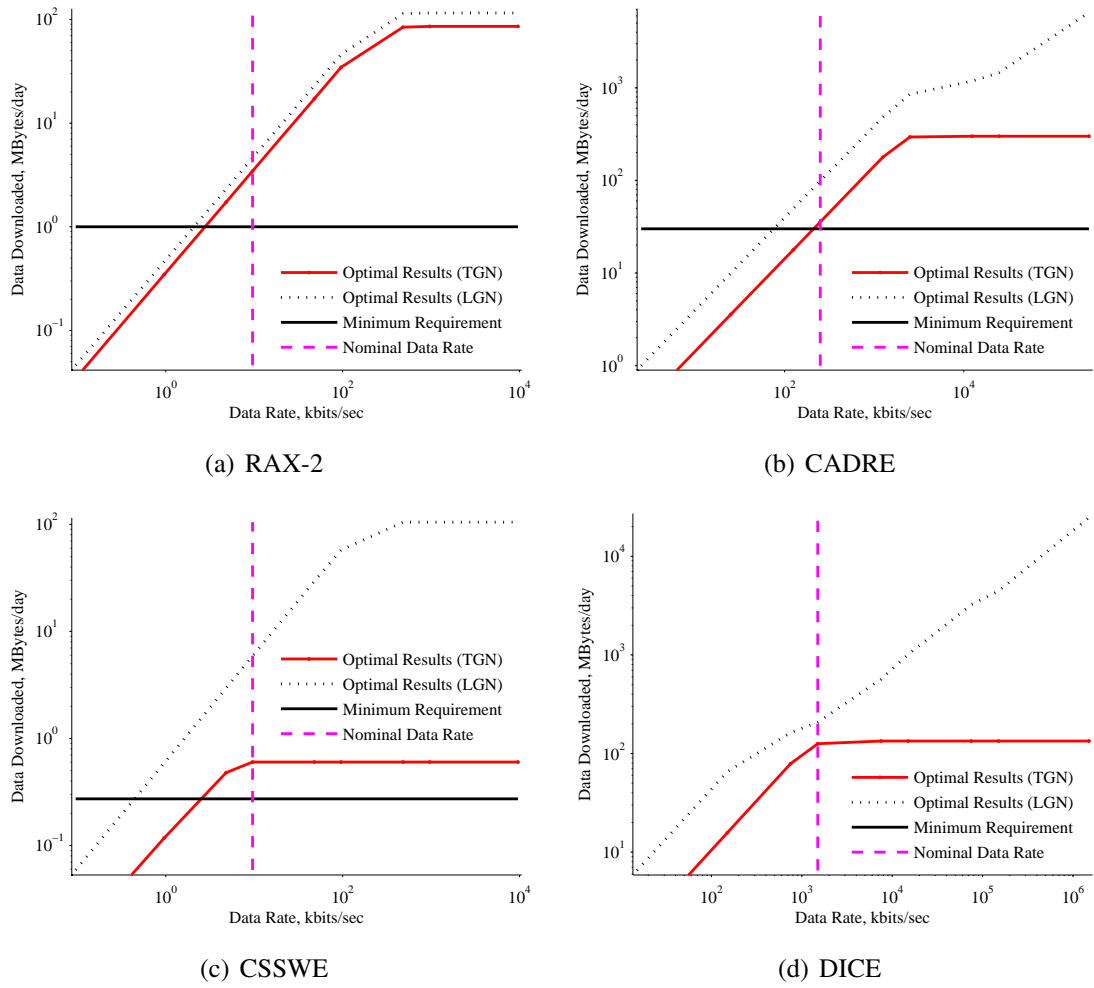
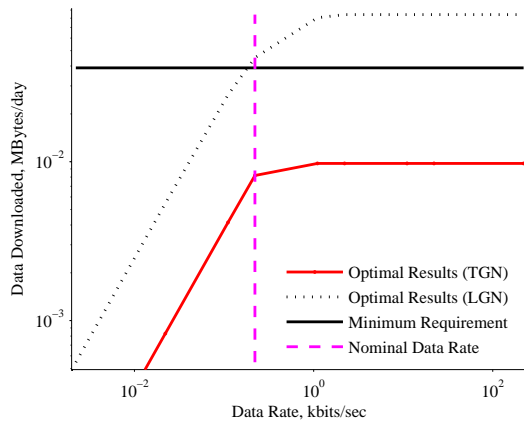
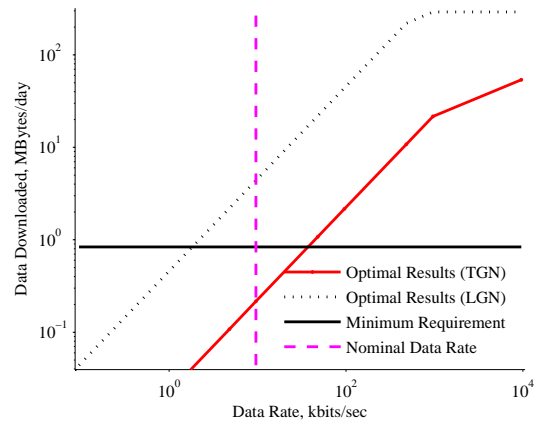


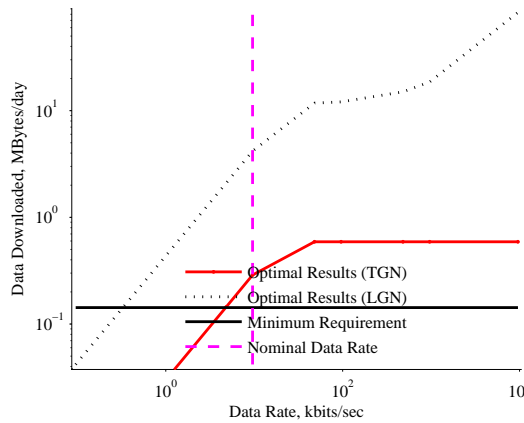
Figure 5.3: Sensitivity of data downloaded from optimized schedules to variations in download data rates. Optimal results are shown for each spacecraft communicating to its TGN and the LGN for a one day scenario. Results are shown on a log-log scale.



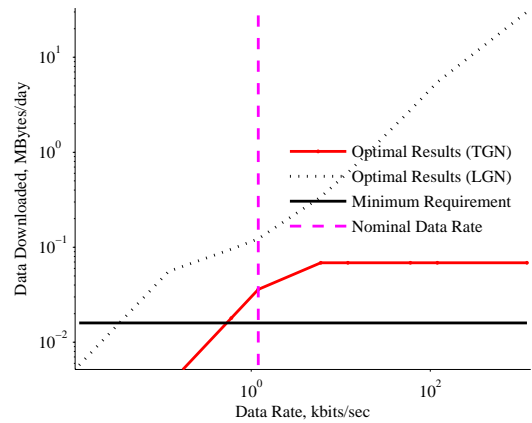
(a) FIREBIRD



(b) MCubed



(c) KySat



(d) E1P

Figure 5.4: Sensitivity of data downloaded from optimized schedules to variations in download data rates. Optimal results are shown for each spacecraft communicating to its TGN and the VLGN for a one day scenario. Results are shown on a log-log scale.

representative missions from the small spacecraft community from Table 5.3. We assume that the power required to support download scales proportionally to the download rate, which is a reasonable approximation for small spacecraft radios. Results are shown for each spacecraft mission communicating to its TGN and LGN. These results help to identify if the nominal data rate (dotted vertical line) achieves the required download (solid horizontal line), and how variations in data rate impact download potential. The advantages of increasing the data rate can be extracted from these figures. For example, the data downloaded for the RAX-2 mission scales linearly with data rate for data rates less than 1 Mbit/sec, see Figures 5.3. When the data downloaded versus data rate curve plateaus, there are no longer any gains with increasing the data rate, as seen for the RAX-2, CSSWE, and FIREBIRD missions. This occurs since these missions encounter an active constraint, such as limited availability to ground stations or energy limitations, which prevent an increase in download with data rate.

5.3.2.3 Number of Ground Stations

Most of the recently launched and upcoming small spacecraft missions rely on communicating to only a single or handful of ground stations, often at the university or institution where the spacecraft was built. However, there are existing and emerging ground station networks that have the potential to support multiple spacecraft missions, such as the loosely connected network of amateur ground stations [9], and more organized networks such as GENSO [36]. Thus we investigated the impact of growing ground station networks on download potential for representative missions. In this analysis, the 14 ground station network (LGN) is used. The network grows by adding stations in a random order to represent realistic networks which grow by the random addition of independently owned and operated ground stations. The download potential is computed for each network size.

The relationship between number of ground stations and quantity of downloaded data for representative missions is shown for both low data rate (< 10 kbits/sec) and high data rate (> 100 kbits/sec) missions in Figures 5.5(a) and 5.5(b), respectively. Although the download potential grows monotonically with additional stations, the trend is not continuous due to the random location of additional stations and availability of spacecraft commodities (such as energy and data) to support download. Ground station location and availability of energy impact these results, as investigated in Chapter 3. The impact of larger networks on download potential is significant for spacecraft that are not energy-constrained, such as RAX-2, CSSWE, KySat, CADRE, and DICE. However, the impact of larger networks is not as significant for spacecraft that are constrained by available energy or data, which have similar download potential with a single stations as with a larger network, such

as NPS-SCAT, FIREBIRD, and RAMPART in Figure 5.5(a).

5.4 Interplanetary Mission Application

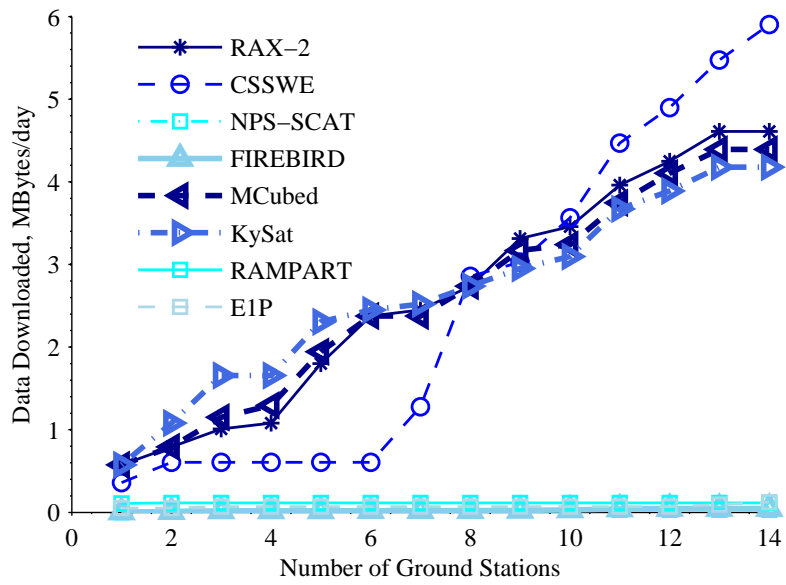
Next we apply our simulation environment and formulations to assess and optimize an interplanetary mission. This demonstrates and validates the applicability of our modeling approach to mission architectures beyond those in LEO. Interplanetary missions have different characteristics in the opportunities for collecting energy and communicate relative to LEO missions because of the difference in interplanetary orbits coupled with additional planetary bodies and moons, that may obstruct the line-of-sight to the sun, Earth, or communication relay. For example, opportunities may occur with different frequency and duration, which may cause scheduling challenges, particularly when opportunities are irregular or infrequent. In this section we also consider variable-rate communication, which is important for interplanetary missions due to the potential for large differences in range distances and the impact on data rates and communication efficiency.

The interplanetary mission is described and two potential communication architectures for this mission are introduced in Section 5.4.1. Using the simulation environment, we perform a high-level mission analysis in Section 5.4.2 by investigating both availability of download time and energy over long and short time scales, similar to the constraint-based approach for diverse LEO missions in Chapter 2. Next, we solve instances of *SOSP* with constant and dynamic communication data rates in Section 5.4.3. Throughout this section we use the optimization formulation and simulation environment to compare the two proposed communication architectures. We also perform a simple trade to consider the feasibility of this mission on a variety of planets within our solar system.

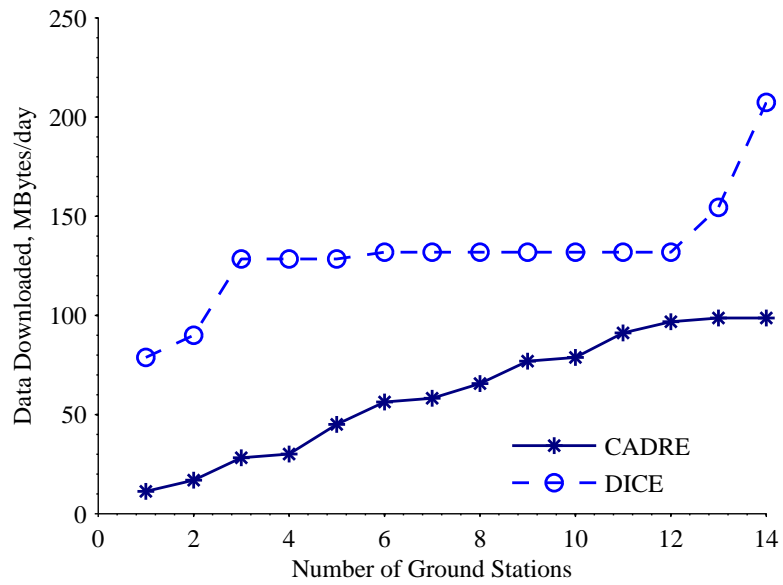
5.4.1 Mission Description and Proposed Communication Architectures

We consider a CubeSat mission that lands on the Martian moon, Phobos. This mission seeks to address the primary themes from the *Planetary Science Decadal Survey* in the context of small body exploration [178]. In particular, the goal is to investigate the chemical and physical properties and processes of Phobos towards an improved understanding of the origins and evolution of the solar system. The ultimate goal is to better understand Phobos and the possibility of human exploration on this moon.

The Phobos lander will be the first satellite to investigate the environment and surface material of Phobos using an in-situ approach. The primary mission objective is to collect samples of the Phobian soil and perform a composition analysis to improve our understand-



(a) Low Data Rate (< 10 kbits/sec) Missions



(b) High Data Rate (> 100 kbits/sec) Missions

Figure 5.5: Sensitivity of data downloaded from optimized schedules to the number of ground stations in the LGN from Table 5.3.

ing of how Phobos originated and evolved. Secondary mission objectives include radiation, thermal, and seismic characterization [179]. The payload includes a penetrator designed to retrieve samples, a mass spectrometer, a seismometer, a heat probe, and a docimeter. The seismometer will measure seismic reflections, monitor background noise, and detect seismic events and the mass spectrometer will measure elemental isotope ratios. The thermal conductivity probe will measure heat flow over the surface of Phobos for information on how the chemical and physical processes interact and have evolved. The docimeter will monitor the radiation dosage on Phobos to understand the potential radiation risk to humans if they were to visit this moon.

The Phobos lander consists of the conventional power, propulsion, attitude determination and control, flight computer, and thermal control subsystems. Although details of the mission design are beyond the scope of this thesis, several of the components have been sized to support the payload in the Phobian environment. For example, the solar panels occupy an area of 0.06 cm², the surface of a single 2U CubeSat or 1 U CubeSat with deployed solar panels and the battery is sized to sustain nominal operations for at least as long as the average eclipse duration (200 minutes) with a factor of safety of two. Mission specifications are provided in Table 5.4. We assume that the transmit radios are 30% power efficient, such that the transmit power is 30% of the input DC power. We also assume perfect communication efficiency for our analysis and optimization examples.

The CubeSat will land on Phobos at a latitude = 0°, longitude = 0°, such that it is always facing Mars (Phobos is tidally locked with Mars, thus the same surface always faces Mars). This location was selected to maximize access time to the communication orbiter relay. Details of the trajectory to get the lander to Phobos, including hitching a ride on the ExoMars mission as a secondary payload, self-jettison, and rendezvous, are beyond the scope of this thesis; however are provided in Ref. [179].

Table 5.4: Phobos lander mission parameters.

Parameter	Value	Units
Maximum Battery Capacity	336	kJ
Maximum Depth of Discharge	20	%
Maximum Data Capacity	25	MBytes
Power Collected in Sun on Phobos	9.4	W
Power Required for Nominal Operations	2.8	W
Power Required During Experiments	7.5	W
Data Collection Rate for Nominal Operations	0.5	kbps
Data Collection Rate During Experiments	400	kbps

Throughout this section we compare two communication architectures, where their pa-

rameters and representative link budget are summarized in Table 5.6.

The first architecture consists of the Phobos lander communicating to the ExoMars Trace Gas Orbiter (TGO), a Martian orbiter that is proposed to become operational in 2016. The science objective of ExoMars TGO mission is to detect, map, and characterize the atmospheric composition and state of the atmosphere, in particular investigating methane in the Martian atmosphere. The ExoMars TGO will be in a near-circular low altitude orbit around Mars to satisfy the coverage and inclination requirements of the mission [145]. Based on the orbital period, minimum altitude, inclination and argument of perigee of the proposed orbital approach, and the mean equatorial radius of Mars from the available literature, we deduced that the orbiter has the following orbital properties: perigee of 255 km, apogee of 320 km, and inclination of 92.7° [180, 181]. The ExoMars TGO will perform both proximity Ultra-High Frequency (UHF) and deep space X-band communication with the DSN and is expected to communicate 250 Mbits/sol in proximity operations and return over 2 Gbit/day to the DSN. Proximity communication between the Phobos lander and the ExoMars TGO will be using the UHF Electra Lite transceivers on both the lander and orbiter and then the lander data will be relayed via “piggyback” to Earth via the X-band communication link from the ExoMars TGO to the DSN. Electra Lite was designed for Mars relay telecommunications and navigation [2]. Electra Lite is proposed to weigh less than 2.1 kg with a volume of less than 2200 cm³ and can achieve data rates ranging from 1 to 2048 kbps with a transmit of approximately 10 W. While this transceiver is likely too large for a conventional 3U CubeSat, it may be feasible for larger 2-6 U CubeSats, or future versions of this technology may be sufficiently small to fit in a CubeSat formfactor. A CubeSat is capable of supporting a 10 W transmission power for short periods of time, depending on the available energy, which is considered in the upcoming optimization problem. We assume a low-gain (0 dBi) UHF antenna is used, such as the deployable dipole antennas with considerable flight heritage on the CubeSat form factor [182]. The receive antenna is a 12 dBi gain quadrafilar helix antenna with flight heritage on the Mars Global Surveyor, Mars Odyssey, and the International Space Station (ISS) [2].

The second architecture consists of the Phobos lander communicating via X-Band frequencies directly to the DSN [4]. X-Band is the preferred method of deep space communication since it is less susceptible to ranging distortions from interplanetary media. Although X-Band communication technology is less developed for interplanetary small satellite applications, there are emerging solutions that have been proposed and are being developed, such as the transceiver proposed in Ref. [183]. We assume an X-Band transmitter with 12 W transmit power [5] and a low-gain patch antenna are used [6] (in the upcoming examples we examine antenna patterns ranging from 0-7 dBi to represent possible antenna

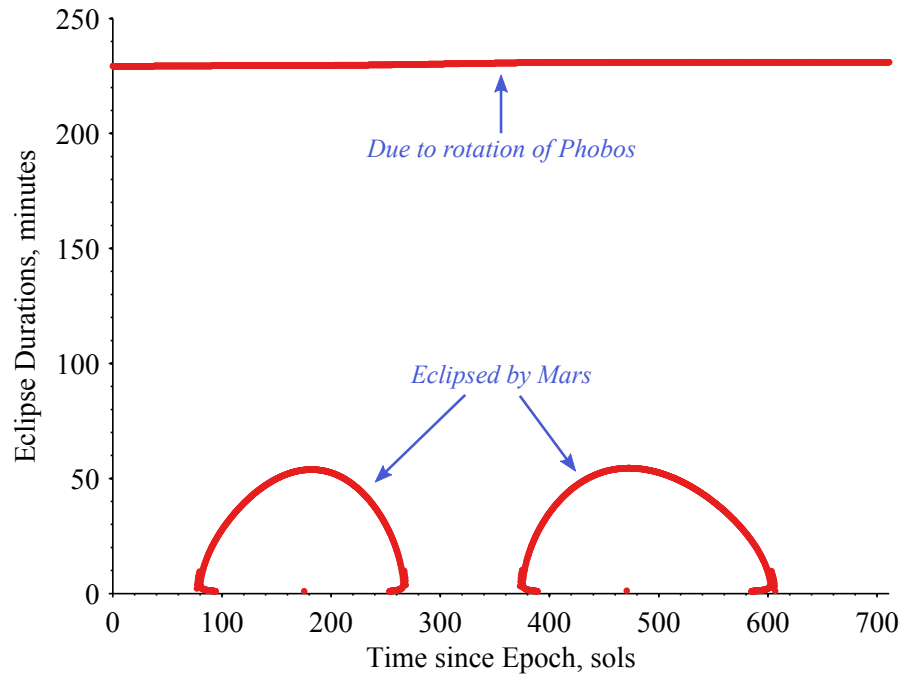
designs). The DSN consists of three 66.8 dBi gain 34 meter DSN ground stations located approximately 120° apart around the world at Goldstone, in California's Mojave Desert; near Madrid, Spain; and near Canberra, Australia.

5.4.2 Mission Assessment

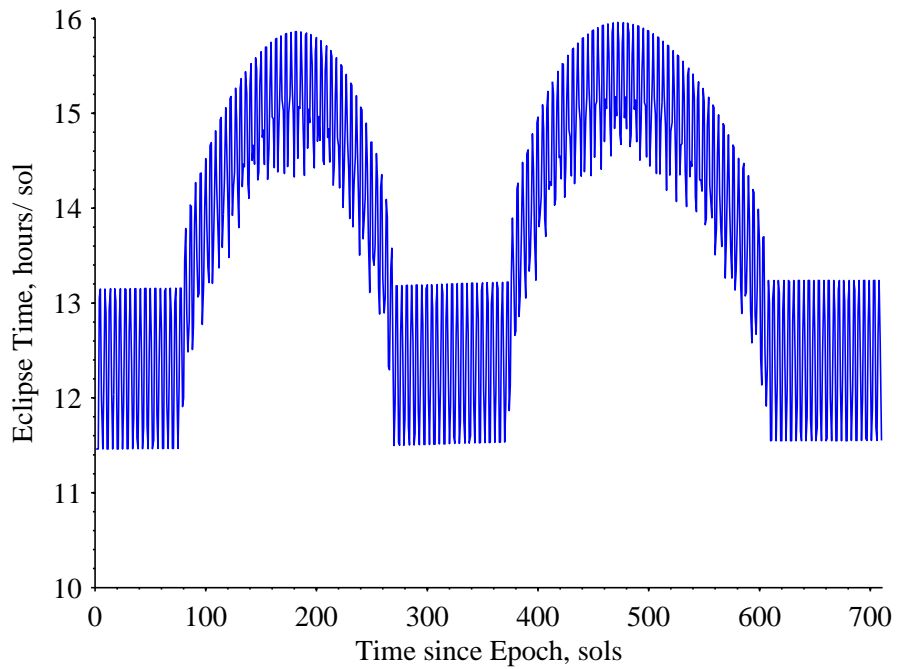
Next we assess the potential of the Phobos-lander mission to acquire energy and download data to identify average and extreme conditions which provide insight for operational planning. This analysis is similar to the constraint-based assessments for LEO spacecraft in Chapter 3, and is performed using our integrated simulation environment described in Chapter 2. In particular, STK is used to model realistic orbital motion of Phobos, Mars, and the ExoMars TGO relative to the Earth and Sun. STK also identifies opportunities to collect energy from the sun and communicate and relay information back to the ExoMars TGO and the DSN. Our custom Matlab scripts then process and use this information for operational planning.

The Phobos-lander acquires solar energy by body-fixed solar panels. This energy is necessary to support nominal, payload, and download operations. Energy collection is a function of eclipse time relative to the orbital period, and we assume energy collection is constant when the lander is in the sunlight (i.e. we neglect solar incidence angles). Figure 5.6 shows the eclipse times for the Phobos landers for a long-duration two year simulation. The constant longer periods of eclipse in Figure 5.6(a) are due to the Phobian orbital period, which lasts approximately half of Phobos' 7 hour 39 minute orbit. The shorter eclipse periods are due to Mars obstructing the line-of-sight to the sun at certain times during the year. The combined effect of these eclipse times on average daily eclipse time is shown in Figure 5.6(b). There are times during the year with considerably longer eclipse time per Martian sol (up to 16 hours of a 24.66 hour sol). Long eclipse times present operational challenges because the spacecraft has limited time to collect energy and long durations with no opportunity to replenish energy reserves.

We show the communication access times for the two communication architectures in Figure 5.7 for a two week scenario. Short-duration trends are shown as the access times are approximately consistent throughout the year. The average daily access times per Martian sol (24.66 hours) for the two architectures are summarized in Table 5.5. Access times to the ExoMars TGO are shorter and more frequent (about 7 passes/ sol all lasting less than 2 hours) relative to access times to the DSN, which are consistently approximately three hours (with access to all three DSN ground stations). Overall there is greater daily access time to the ExoMars TGO relative to the DSN.

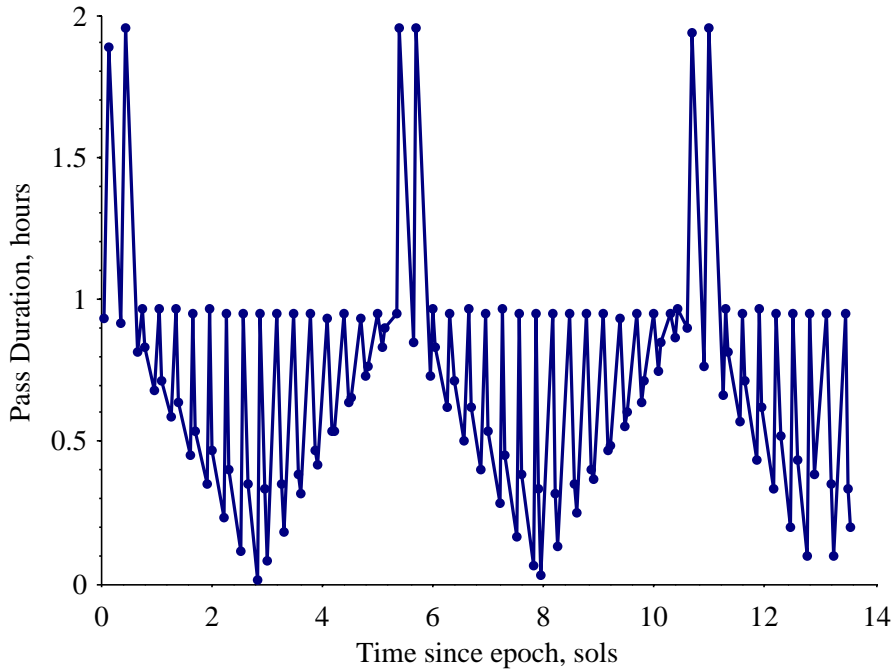


(a) Eclipse Durations

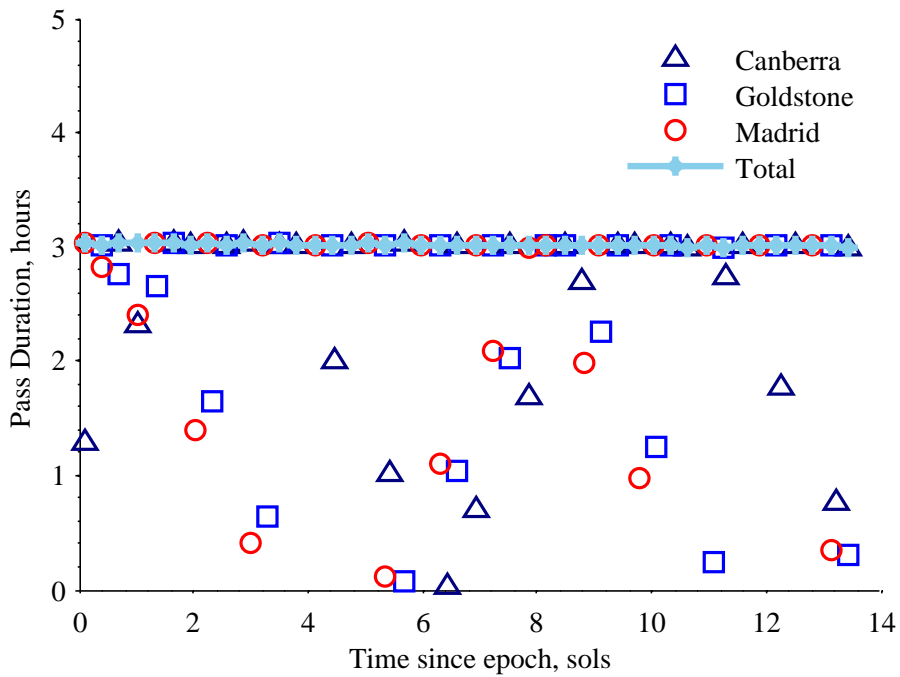


(b) Daily Eclipse Times

Figure 5.6: Long-duration eclipse characteristics for Phobos lander. A Martian sol is 24.66 hours.



(a) ExoMars TGO Pass Durations



(b) DSN Pass Durations

Figure 5.7: Short-duration access time characteristics for Phobos lander communicating to ExoMars TGO and DSN (consisting of three 34 m dishes). A Martian sol is 24.66 hours.

Table 5.5: Short-duration access time statistics for Phobos lander access time for 14 days (as shown in Figure 5.7).

Receiving Nodes	Pass Time/ sol (hours)		
	Minimum	Average	Maximum
ExoMars TG! (TG!)	4.67	6.55	9.12
DSN	9.05	9.74	10.76

Table 5.6 summarizes the link budget for communication between the Phobos lander and the two proposed communication architectures. To perform these analysis, we use the analytic link budget developed in Appendix A, which is similar to the link budget derived in Section 2.2.3. Figures 5.8 and 5.9 shows the predicted path distances and the feasible data rates for communication between the Phobos lander and the two communication architectures. The path distance from the Phobos lander to the ExoMars TGO varies on a daily basis while the path distance between the Phobos lander the DSN varies over longer time scales as a function of the position of the Earth relative to Mars (Mars has an orbital period of 667 sols or 687 days) . Data rates achievable for communication to the DSN are extremely low (<30 bps), while data rates achieved for communication to the ExoMars TGO are considerably higher (>100 kbps) and are comparable to data rates achieved by small satellites in LEO communicating to Earth stations.

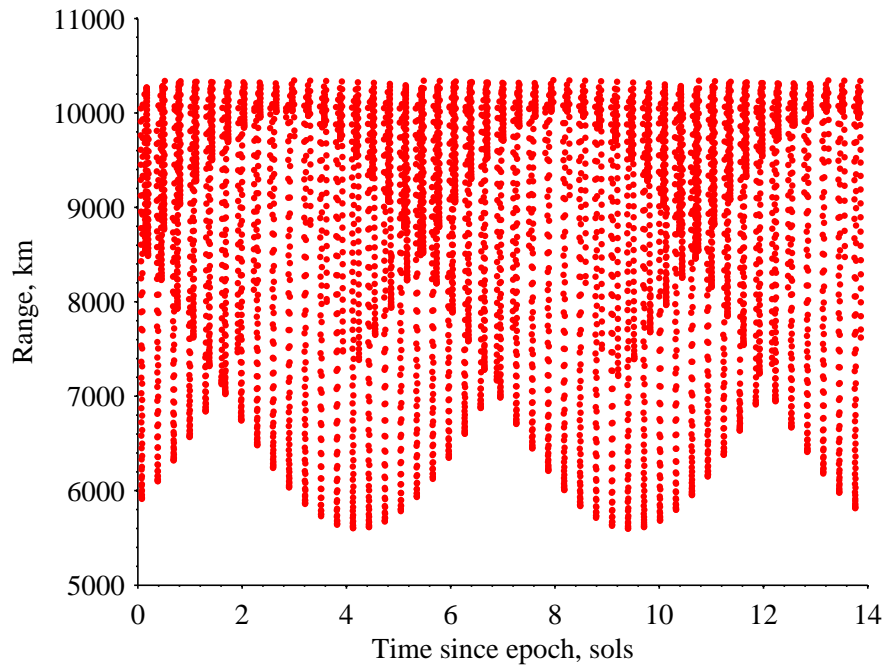
5.4.3 Optimization Results

We've assessed mission constraints at a high level, in particular the availability of energy and download time, as well as the feasible data rates for communication for the two proposed communication architectures. We can use this information to extract average, best, and worst-case scenarios, to roughly approximate energy and data budgets and performance metrics. However, analytic scheduling optimization is necessary to make intelligent operational decisions to ensure resources are allocated to maximise mission performance given the constraints of the problem. In this section we optimize operational schedules for the Phobos mission.

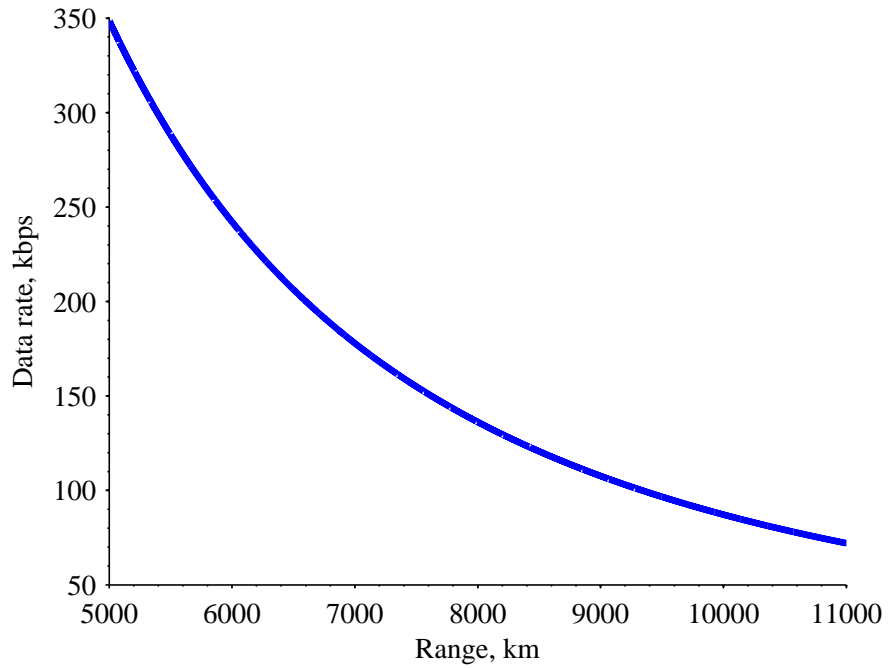
5.4.3.1 Comparison of Constant and Dynamic Data Rates

The problem we address is an instance *SOSP*, where the decisions are when and how much data to download. Thus, the problem is solved with *OUCF*.

We considered two variations of the problem. In the first, the communication data rate is dynamic, and varies as a function of distance between the transmitter and receiver, i.e.

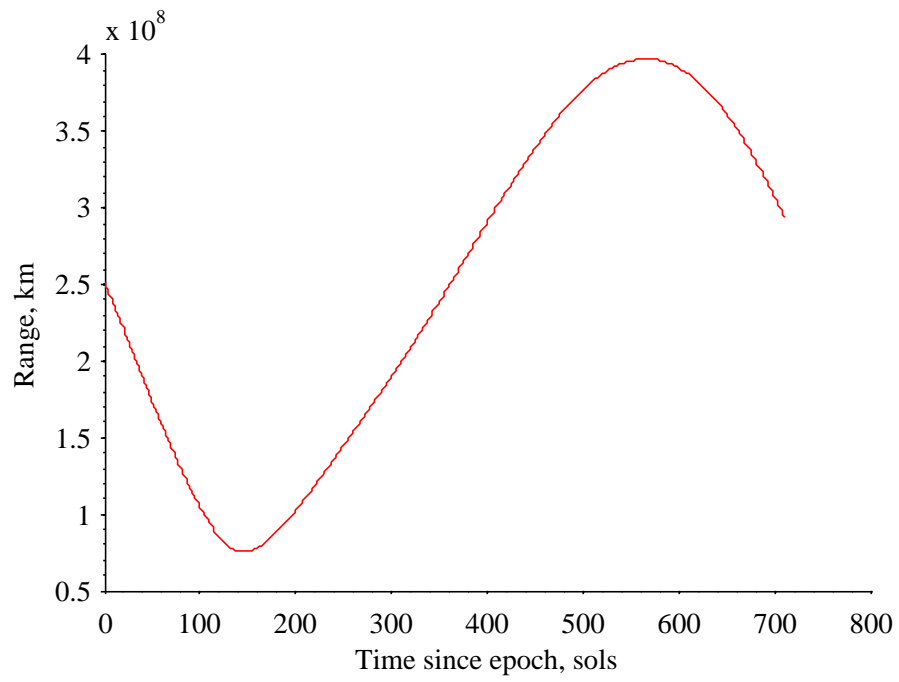


(a) ExoMars TGO Path Distances

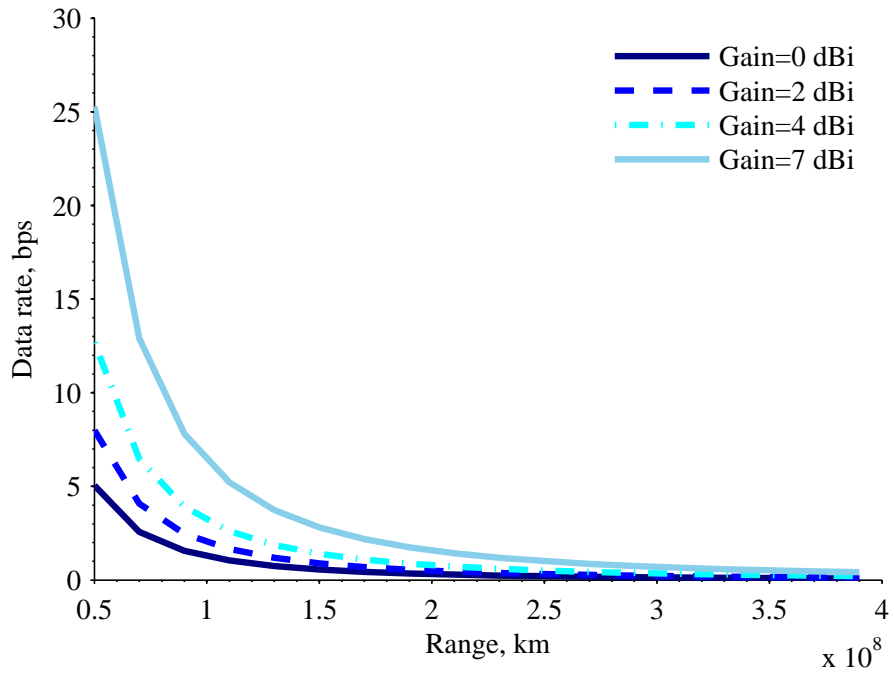


(b) UHF Data Rates

Figure 5.8: Path distances and feasible data rates for Phobos lander communicating to ExoMars TGO. Data rates are computed using the link budget in Table 5.6.



(a) DSN Path Distances



(b) X-Band Data Rates

Figure 5.9: Path distances and feasible data rates for Phobos lander communicating to DSN ground stations. Data rates are computed using the link budget in Table 5.6.

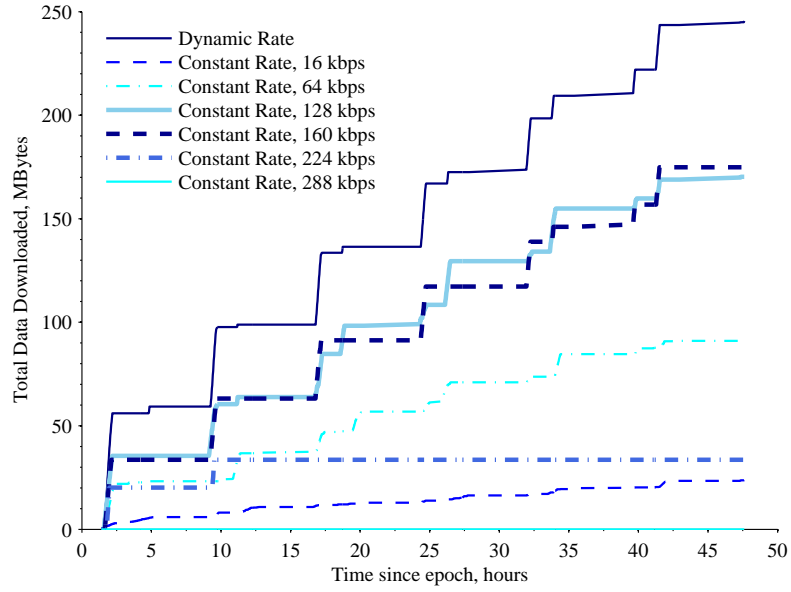
for any download interval the only opportunity is to download at the maximum feasible rate that satisfies the link budget. In the second variation, we assume only a constant rate is feasible, and there is only a communication opportunity if the link budget is satisfied. The link budget being satisfied is a function of the range distance between the transmitter and receiver.

Properties of the optimal solutions for UHF proximity communication between the Phobos lander and the ExoMars TGO are shown in Figure 5.10. The optimal amount of data downloaded for communication between the Phobos lander and the ExoMars TGO architecture is shown in Figure 5.10(a). It is not surprising that the dynamic rate solutions outperform the constant rate solutions since they are able to exploit the change in range distance and download at the maximum feasible rate throughout the mission. There is an interesting relationship between the constant rates and the total data downloaded. As the rate increases from 16-160 kbps, the data downloaded increases. However, for greater constant data rates the data downloaded decreases significantly (see the drop to less than 50 MBytes for the 224 kbps case and zero downloaded data for higher rates). This trend is due to the fact that there are fewer opportunities to download at higher rates (i.e. less time as closer ranges), as observed in Figure 5.8. In Figure 5.10(b), higher constant data rates have less feasible download times since they are limited to times where the range distance is sufficiently small.

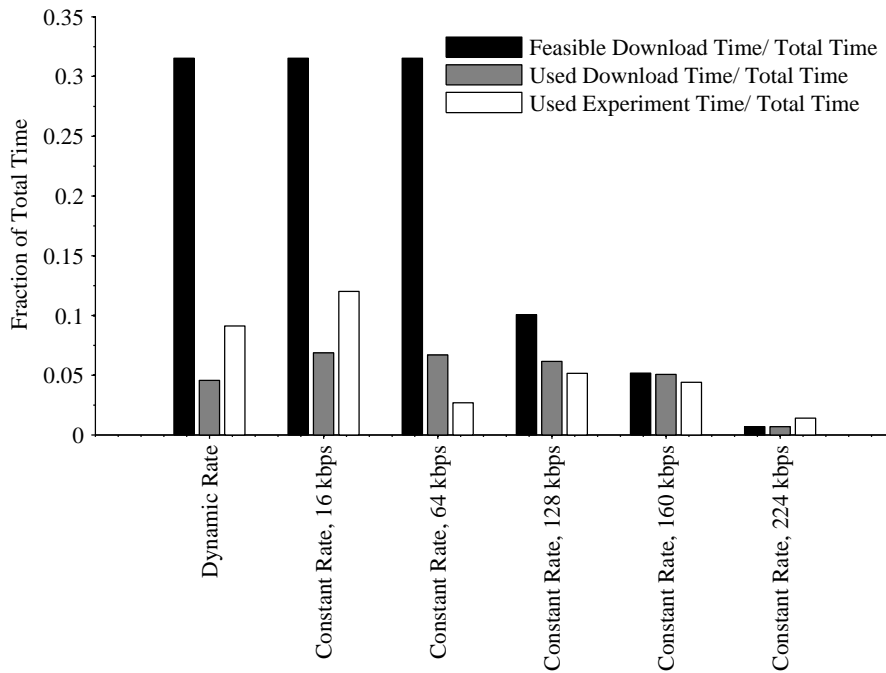
Optimal results for X-Band communication to the DSN are shown in Figure 5.11 for variable transmit antenna gains, where the achievable data rates are shown in Figure 5.9(b). Dynamic data rates are used; however the actual rates are relatively constant since the range distance does not change significantly over the two day planning horizon. Communication through the ExoMars TGO is the preferred communication architecture because data downloaded using the DSN is significantly reduced. Note that there will be additional routing and scheduling challenges in transmitting Phobos lander data via ExoMars TGO to the DSN, as discussed in Section 7.2.

5.4.3.2 Sensitivity to Power Collection

Next we investigate the sensitivity of the amount of data downloaded to energy collection for optimized operational schedules. The goal is to assess if the Phobos-lander mission would be feasible on other planets or with different solar panel configurations or efficiencies. Figure 5.12 shows the optimal results when the communication optimization problem is solved with diverse solar collection values, summarized in Table 5.7. We scale the energy collected when in the sun, assuming constant eclipse durations. Since this simulation is performed early in the mission, the short-term Martian effects on eclipse (see Figure



(a) Data Downloaded



(b) Fraction of Opportunities

Figure 5.10: Properties of optimal solutions for dynamic and constant rate communication for Phobos lander to ExoMars TGO for a 48 hour planning horizon.

5.6(a)) are not present. This analysis demonstrates how the data downloaded scales with collected power. The amount of data that can be downloaded when the collected power is

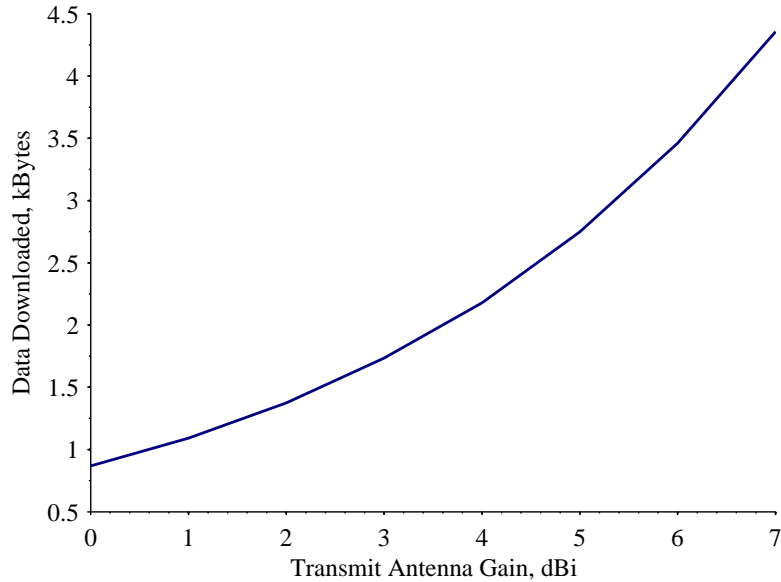


Figure 5.11: Data downloaded with different receive antenna gains for Phobos lander communicating to the DSN for a 48 hour planning horizon with optimized operational schedules.

100 Watts is over four times what is achievable when the collected power is only 10 Watts. For planets in the solar system that are further away than Mars, this mission is not feasible.

5.5 Summary

In this chapter, we have extended and generalized the scheduling formulation from Chapter 4 using the general modeling framework from Chapter 2. The new optimization formulation couples key operational decisions, such as how and when to perform payload operations and download to available communication nodes. The formulation may be applied to problems with diverse satellite commodities (such as on-board energy, data, and thermal states), subsystem functions (such as data collection by the payload and data downlink by the communication system), and mission constraints (such as buffer limitations and minimum data collection requirements).

We’ve demonstrated the applicability of this new formulation to solve a broad range of challenging spacecraft mission scheduling problems with diverse types of payload operations with both near-Earth and interplanetary orbits. The importance and advantages of considering the coupling between payload and download operations in satellite scheduling relative to considering download decisions only has been demonstrated. In particular,

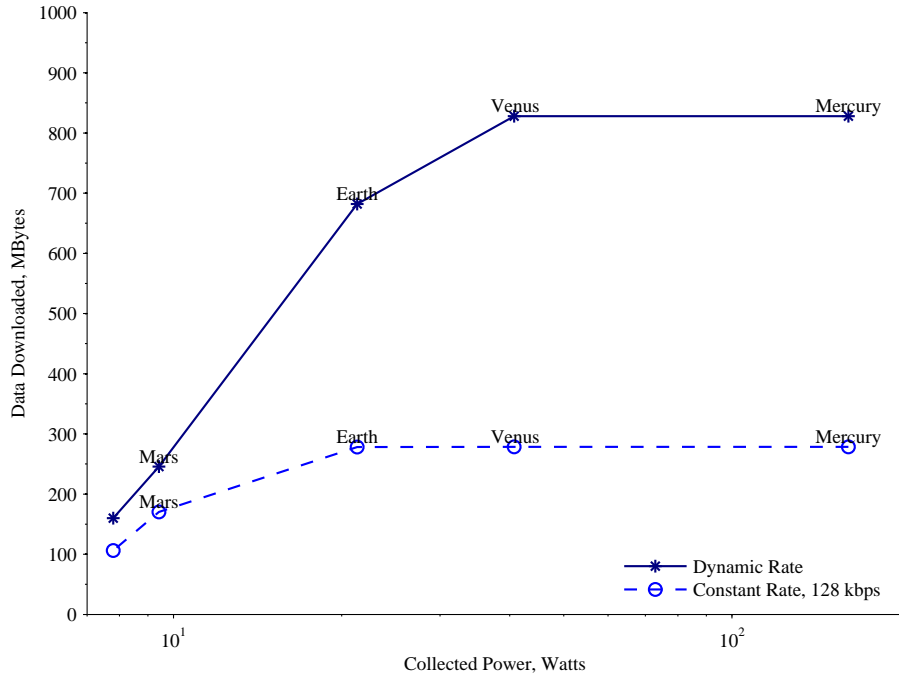


Figure 5.12: Optimal downloaded data with variable collected power representative of different planets for communication between Phobos lander and the ExoMars TGO for a 48 hour planning horizon.

some LEO CubeSat missions were found to be infeasible when we ignore payload decisions and assumed these operations occurred at every opportunity. By comparing the optimized schedule downloads to the minimum download requirements for each mission, we’ve demonstrated the advantages of optimizing operational schedules relative to using conventional requirements-driven approaches for operational planning, where designers typically design only to meet requirements and neglect the potential to exceed them.

Beyond optimizing existing satellite schedules, the optimization formulation presented in this chapter enables satellite designers and operators to answer key operational and satellite design questions. Conventional trade-space approaches in vehicle and network design typically use back-of-the-envelope approximations to model operational constraints, and tend to be highly inaccurate. The design space exploration technique demonstrated in this chapter has distinct advantages over these conventional approaches because it optimizes the schedule for each design option to enable accurate comparisons (and ensure each option is feasible). For example, we’ve demonstrated how rapid computation of optimal solutions enables design space exploration to size solar panels and batteries, and determine the impact of varying the size of the ground station network and data rates on the potential data

throughput. The optimization formulation has been used for a simple early-design trade for an interplanetary mission. We compared two potential communication architectures using both dynamic and constant data rates using optimized schedules. The advantages of dynamic link budgets has been demonstrated, where nearly double the data returns can be achieved relative to constant rate communication links in our examples.

In this chapter we have investigated how deterministic parameters impact solutions. However, beyond these parameters, there are multiple sources of uncertainty in operating spacecraft for both in near-Earth and interplanetary applications. For example, factors that impact the download efficiency include imperfect attitude pointing that modifies the expected gain, additional noise sources, and inaccurate ground tracking, due to their impact on signal strength in the link budget and thus probability of successful data transmission. The sensitivity of solutions to these stochastic inputs is investigated in Chapter 6.

Table 5.6: Link budget for Phobos lander communicating to the ExoMars TGO with the Electra Lite UHF transceivers [2] and to three 34 meter DSN ground stations [3, 4]. with an X-Band transponder [5] and patch antenna [6].

Link Budget Parameter	Symbol	UHF to ExoMars TGO	X-Band to DSN	Units
Frequency	f	401	8,425	MHz
Propagation Path Distance	S	$8 \cdot 10^3$	$1 \cdot 10^8$	km
Transmitter Power	P_t	10	12	dBW
Transmit Antenna Gain	G_t	0	0	dB
Transmit Line Losses	L_t	2	1	dB
Equivalent Isotropic Radiated Power (Eq. A.1)	$EIRP$	8	9.8	dBW
Receive Antenna Gain	G_r	12	66.8	dB
Pointing Error Loss	L_e	1	1	dB
Polarization losses	L_p	1	1	dB
Receive Line Losses	L_r	2	0.4	dB
Space Loss	L_s	162.6	270.9	dB
Received Power	P_r	-144.6	-196.4	dBW
System Noise Temperature	T_s	500	500	K
Carrier-to-Noise Spectral Power Density Ratio (Eq. A.4)	C/N_0	57.0	5.2	dB
Minimum E_b/N_0	$E_b/N_{0,min}$	2.7	1.24	dB
Link Margin	M	3	3	dB
Feasible Data Rate	r	136,239	1.3	bits/sec

Table 5.7: Phobos lander solar radiation properties on different planets assuming surface area $A = 0.06m^2$.

Planet	Units	Mercury	Venus	Earth	Mars
Average Solar Radiation	W/m ²	10,359	2,611.5	1,367	603.5
Collected Power	W	161.6004	40.74	21.33	9.41

CHAPTER 6

Sensitivity to Stochasticity in Download Efficiency

In Chapter 4 we developed a formulation to optimize deterministic instances of *SMSP*, where the goal was to maximize the data downloaded from a single satellite to a ground station network. However, there are several sources of stochasticity in real-world instances of *SMSP*. Below we list three main types of stochasticity in *SMSP* parameters and discuss their potential impact on solutions. Note that these stochastic parameters may be caused by multiple sources, as discussed in more detail in Section 6.2 in the context of stochastic download efficiency.

- *Stochasticity in download efficiency* will impact the actualized download, the data that is successfully downloaded when the schedule is executed in a real-world scenario (see Section 2.2.2.1). The actualized download may vary from the expected value because it is a function of the individual efficiencies, which may be stochastic. However, the total data downloaded may exceed the expected download, which is advantageous as the goal is to maximize the data downloaded. The total download may fall below the expected download, which may result in the mission failing to satisfy its minimum download requirements.
- *Stochasticity in energy and data generation* may impact the satisfaction of *UCF* constraints and may yield infeasibilities for real-world instances of *SMSP*. If more energy or data is collected than expected, then excess energy or data may be spilled to satisfy the upper buffer constraints. This is generally not problematic except in the case where a large quantity of excess energy or data causes other problems. For example, excess energy can cause heating that can be difficult to manage. In this case where there is excess energy and/or data, it may be exploited to support additional downloading; however, this would require a heuristic or scheduler to manage excess resources and dynamically modify the schedule to plan more downloads. On

the other hand, if less energy is collected than expected, there may be insufficient energy to support on-board nominal, payload, or download operations. If less data is collected than expected, there may be insufficient data available for downloads; however this is unlikely to be problematic as there is generally an excess amount of payload or telemetry data to be downloaded. The nature of the stochasticity (i.e. frequency and magnitude of the variation relative to the expected value) is also important. Stochastic energy collection with a small standard deviation may not impact solutions significantly, while energy collection that varies greatly may cause significant problems. For example, if the satellite does not have a chance to recharge its battery prior to long durations of eclipse when it may need to support nominal or download operations, the satellite may run out of energy, which may cause serious operational issues (e.g. damaging the battery or causing a spacecraft reset).

- *Stochasticity in opportunities for data collection and data download* will impact the amount of data that is successfully collected and downloaded. In this case, the satellite is scheduled to collect or download data according to a timeline, but the target of interest or ground station is not available or active (due to scheduling conflicts, environmental factors, or failures). As a result, the spacecraft may not be able to satisfy its minimum data collection or download mission requirements, and it will also waste on-board resources such as energy and time that can not be recovered. Note that if the spacecraft or ground network is able to detect and re-plan dynamically, the impact of stochasticity in opportunities may be managed (e.g. aborting operations and reserving resources for future opportunities).

Because we have access to realistic download efficiency data and this information provides the potential to improve on-going RAX-2 operations, we focus on stochasticity in download efficiency throughout the remainder of this chapter. In particular, we are interested in the amount of data that is scheduled to be downloaded that is successfully received by the ground station network. Essentially, we are considering the *actualized model* from Section 2.2.2.1, where scheduled events and realistic models for download efficiency are considered. Stochastic factors that contribute to download efficiency include the antenna gain and orientation of the satellite relative to the ground station, the ability of the antenna to slew and track the satellite at high rates, weather effects, and other noise or temporary obstructions (discussed in more detail in Section 6.2). Towards understanding the impact of stochastic download efficiency on *SMSP* solutions, the goal of this chapter is to answer the following question: *What is an approach to model and quantify how stochastic download efficiencies impact SMSP solutions?*

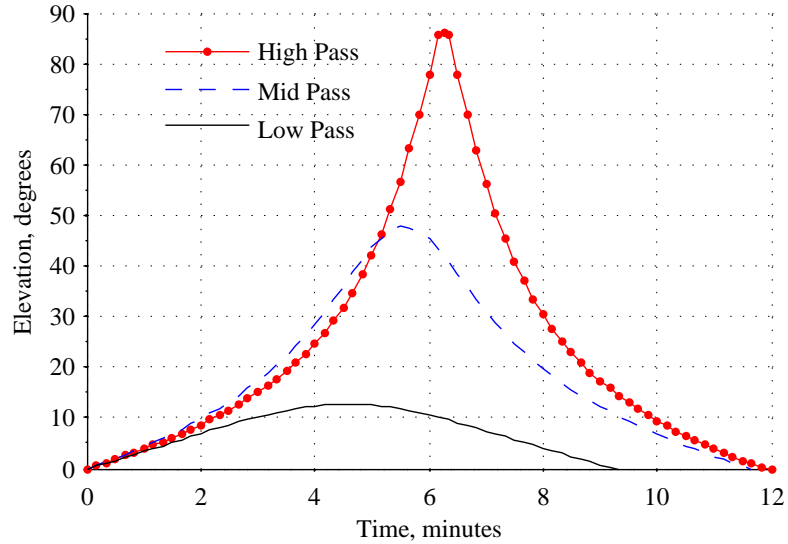


Figure 6.1: RAX-2 elevation profiles for representative high, mid, and low elevation passes over the primary RAX-2 ground station in Ann Arbor, MI.

This chapter is outlined as follows. In Section 6.1 we describe how historic efficiency data from the RAX-2 mission is collected, which provides a rich source of historic data to use in our analysis and modeling. In Section 6.2 we describe approaches to model RAX-2 realistic historic efficiency data and hypothesize explanations for some of the trends observed. In Section 6.3 we conclude and discuss future work.

6.1 Scheduling and Collecting Download Efficiency Data

In this section we describe the approach used to schedule downloads and extract download efficiency statistics for the RAX-2 mission (mission details are provided in Section 2.3). Note that this scheduling problem is similar to *SMSP*, yet it has some important properties and additional constraints. The goal of the RAX-2 scheduling problem is to maximize the amount of data downloaded to a global network of ground stations. We assume downloads are scheduled only during periods of time when there are no scheduled experiments (but after experiments have occurred). As a result, there is always sufficient energy and data to download and the optimal solution is to download whenever there is an opportunity, i.e. a ground station in view. We assume that ground station availability is known, because most of the ground stations operators that collect RAX-2 data inform the RAX-2 operations team about their availability.

The RAX-2 scheduling problem has two additional constraints relative to *SMSP*. First,

data is scheduled to be downloaded throughout a satellite pass, a period of time during which the satellite is continuously above the horizon relative to the ground station. Elevation angles, defined as the angle between the ground station horizon and the satellite in orbit, are dynamic throughout a satellite pass, for example see the representative RAX-2 passes in Figure 6.1. RAX-2 beacons telemetry at regular intervals (e.g. every 10 seconds), except when downloading. We want to collect telemetry both before and after scheduled downloads during every pass to check the health and status of the spacecraft. Thus, we impose a minimum elevation angle constraint for each ground stations, where downloads can be planned only at higher elevations. This constraint is imposed as we expect that at least one (and hopefully more) beacons will be successfully downloaded prior to this minimum elevation angle.

Second, the RAX-2 scheduling problem has an additional constraint on the number of total bytes in a given schedule due to memory constraints on-board the spacecraft. The available data to download is indexed by file parts, where there are often missing file parts from previous downloads that must be downloaded, which makes selecting the file parts to request (i.e. to be downloaded) challenging. File parts may be downloaded as lists (e.g. file parts 1, 2, 3, 4), or as ranges (e.g. file parts 1-4), where the number of bytes in a schedule is a function of the number of lists, the size of those lists, and the number of ranges requested. In some cases, it may be more efficient to request a range of file parts (e.g. 1-4 in the case that file part 1, 2, and 4 are missing), instead of individual file parts (e.g. only 1, 2, and 4) because it results in fewer total bytes in the schedule to be uploaded. Note, however, that requesting a range of file parts that includes parts that have already been successfully downloaded will waste possible download time (e.g. file part 3 in the previous example), resulting in an inefficient use of available download time. An intelligent schedule will determine which lists and ranges to request and when to request them to maximize the total data downloaded. Thus, we impose a constraint that only a fraction of file parts requested during each download opportunity may be redundant (i.e. file parts that have already been downloaded). This constraint ensures that download opportunities are utilized efficiently.

We generally develop download schedules for planning horizons of 12 or 24 hours, depending on other operational constraints (such as performing experiments) and upcoming possible passes where the satellite can be commanded (and a new schedule uploaded). The approach to schedule and extract download efficiency data is described below:

1. Select the file part lists and ranges to be downloaded that maximizes the total number of file parts subject to the byte constraint and a constraint on the fraction of redundant file parts requested during each opportunity.

2. Identify the download opportunities over the planning horizon, which are the times the satellite is at an elevation greater than a minimum elevation angle specific to each ground station (such that beacons are decoded before and after the download during each pass).
3. Rank the download opportunities according to their expected average efficiencies throughout the full pass.
4. Schedule list(s) at the highest priority station(s), then schedule ranges during the download opportunities, downloading the “oldest” data with highest priority over the stations with the highest performance (to download earlier data as soon as possible). If there are more download opportunities than the expected time needed to download the selected file part lists or ranges and the byte constraint is not active, re-schedule lists and ranges such that they are requested a second or third time (this technique is also important when completing the download of a file as it increases the chances of completely downloading the file and being able to move on to the next file on the next schedule).
5. Upload the schedule during a command pass (generally from the primary command station in Ann Arbor, MI). If the schedule is successfully uploaded, the schedule will execute. Note that we generally do not change the schedule after it has been uploaded (although it is possible to abort and re-upload the schedule).
6. Record all scheduled downloads and all received file parts (the received file parts are recorded locally by each ground station and the information is sent instantly to the RAX-2 operations team if there is an Internet connection). Based on the time the file parts are received, extract the position and elevation of the satellite relative to the ground station using propagators and known orbital elements (e.g. published TLE). Note that the times file parts are requested and actually received may not agree perfectly due to timing issues, processing problems, or execution delays issues on-board the spacecraft; however we assume the true time of the transmission is the time it was received (note the time for the data to travel from LEO to Earth is negligible for this analysis).
7. Compute efficiency statistics based on the requested and received file parts. These statistics can be computed as a function of several properties, including the ground station, elevation angle, local time of day, thermal state of the spacecraft, expected attitude of the spacecraft, etc.

This approach is used to schedule all the downloads and extract historic download efficiency data used throughout the rest of this chapter.

6.2 Modeling Download Efficiency Data

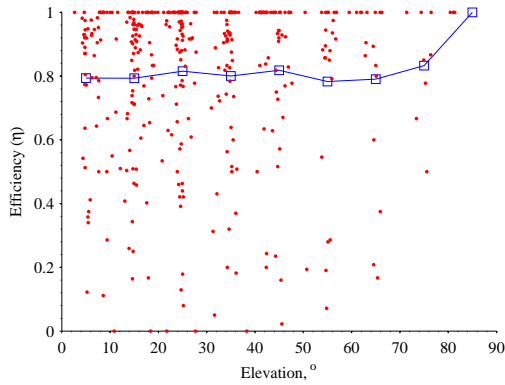
In this section, we examine real-world download efficiency data to extract trends and gain insight into how to model download efficiencies. Throughout the remainder of this chapter, we define a download *opportunity* as an interval with at least one download option. Each option corresponds to a specific data rate, ground station, and average elevation angle of the satellite relative to the ground station, which we assume are all known for every option and opportunity.

In this analysis, we examine efficiency as a function of ground station and elevation as we expect these to be the main factors that influence efficiency trends. There is a well understood relationship between efficiency and elevation angle (related to the link budget described in Section 2.2), which has also been observed in RAX-2 operations. Thus, we bin the data into 10° elevation ranges for convenience of analysis and for planning purposes. Each 10° elevation range corresponds to a time period that is less than or equal to approximately 10 seconds in duration, which corresponds to the shortest time period that is practical for operational decisions. In Figure 6.2 we show realistic efficiency data as a function of elevation angle for RAX-2 communicating to six representative ground stations from its network, which consists of low-cost ground stations at universities and amateur radio operators around the world [184]. Each point in Figure 6.2 represents a portion of a single pass when the satellite has an elevation angle within a 10° range (e.g. $e \in (0, 10]^\circ$). The average efficiency is plotted relative to the average elevation angle for that portion of a pass.¹

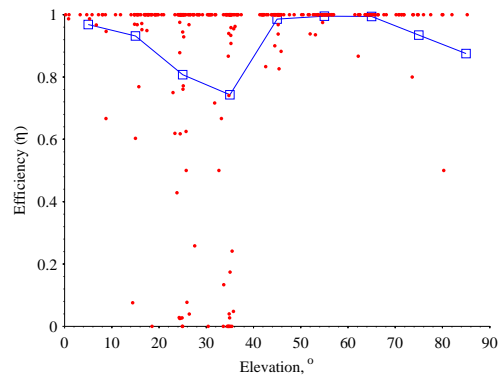
While the distributions in Figure 6.2 are interesting, they are not particularly informative since the range of efficiency values is large, often spanning the complete range of possible values. In addition, the standard deviation is often large and the statistics for a given elevation range is not available. Overall, mean and standard deviation statistics are insufficient to characterize the data, motivating an approach that uses more comprehensive statistical properties of the data.

For improved understanding of the efficiency data, we compute the probability distribution function (PDF), which provides statistics about the likelihood that the efficiency (a random variable) will have a certain efficiency. Figure 6.3 shows the PDFs for all ground

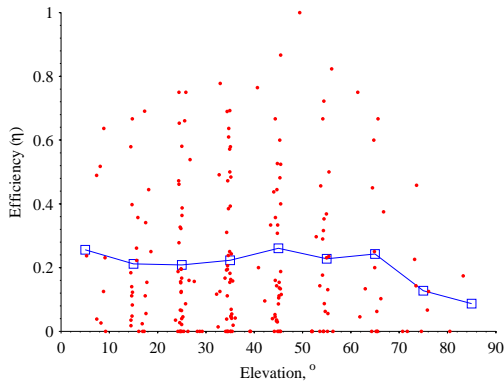
¹For ground station and elevation combinations where we do not have and historic data, we extrapolate the mean and standard deviation statistics using the closest elevations.



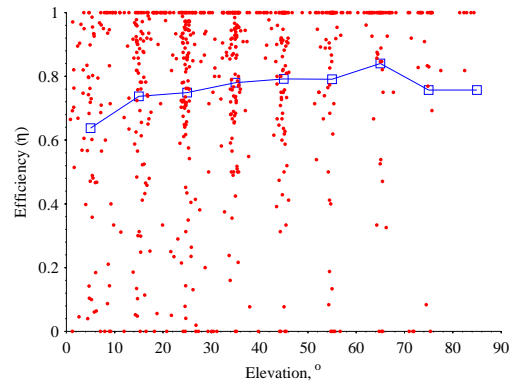
(a) Station ID 29



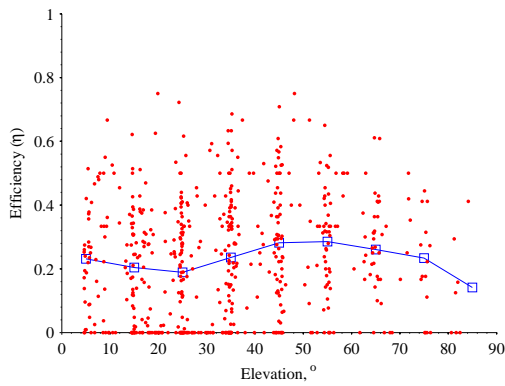
(b) Station ID 40



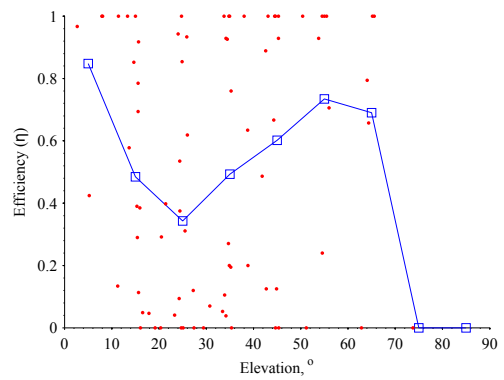
(c) Station ID 46



(d) Station ID 55

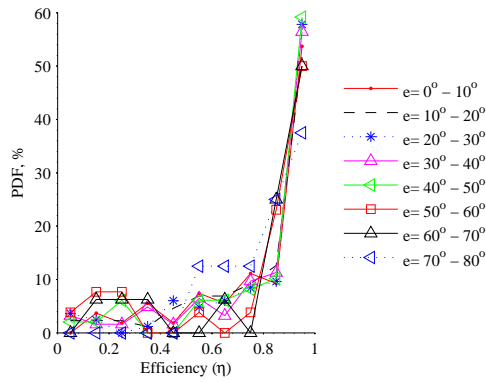


(e) Station ID 100

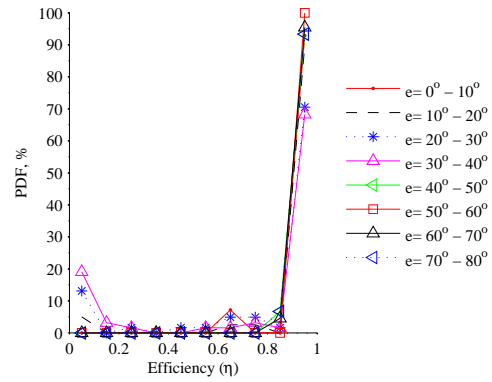


(f) Station ID 163

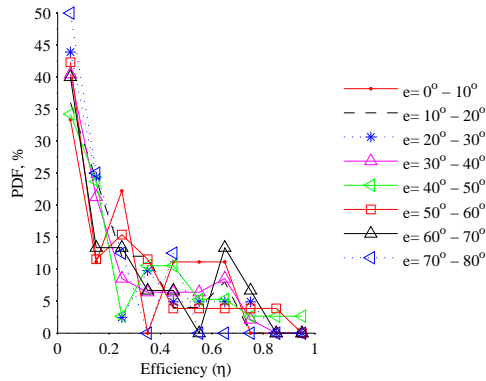
Figure 6.2: RAX-2 efficiency as a function of elevation for download to representative ground stations over about a three month period. Each point represents averaged data over a single time interval that the elevation is within a 10° range. The horizontal lines show the mean for each elevation range.



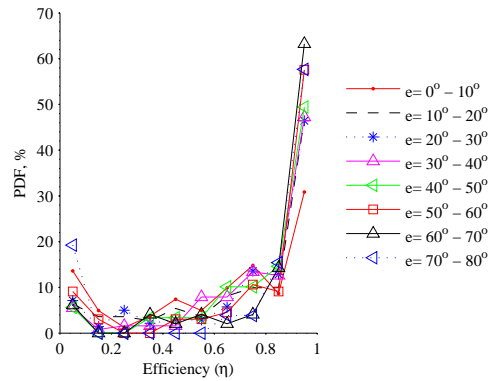
(a) Station ID 29



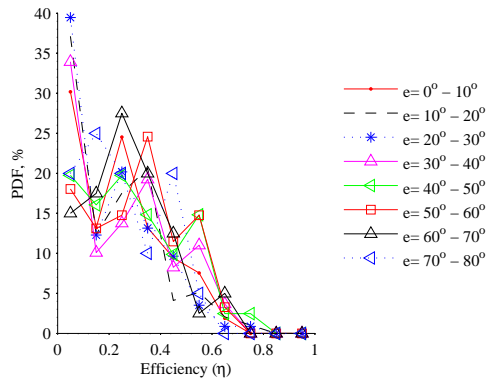
(b) Station ID 40



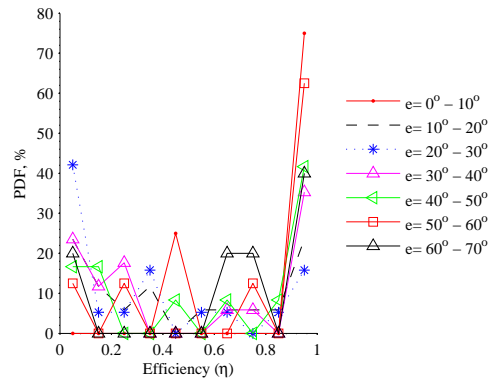
(c) Station ID 46



(d) Station ID 55



(e) Station ID 100



(f) Station ID 163

Figure 6.3: RAX-2 efficiency data probability distribution functions (PDFs) as a function of elevation (e) for download to representative ground stations.

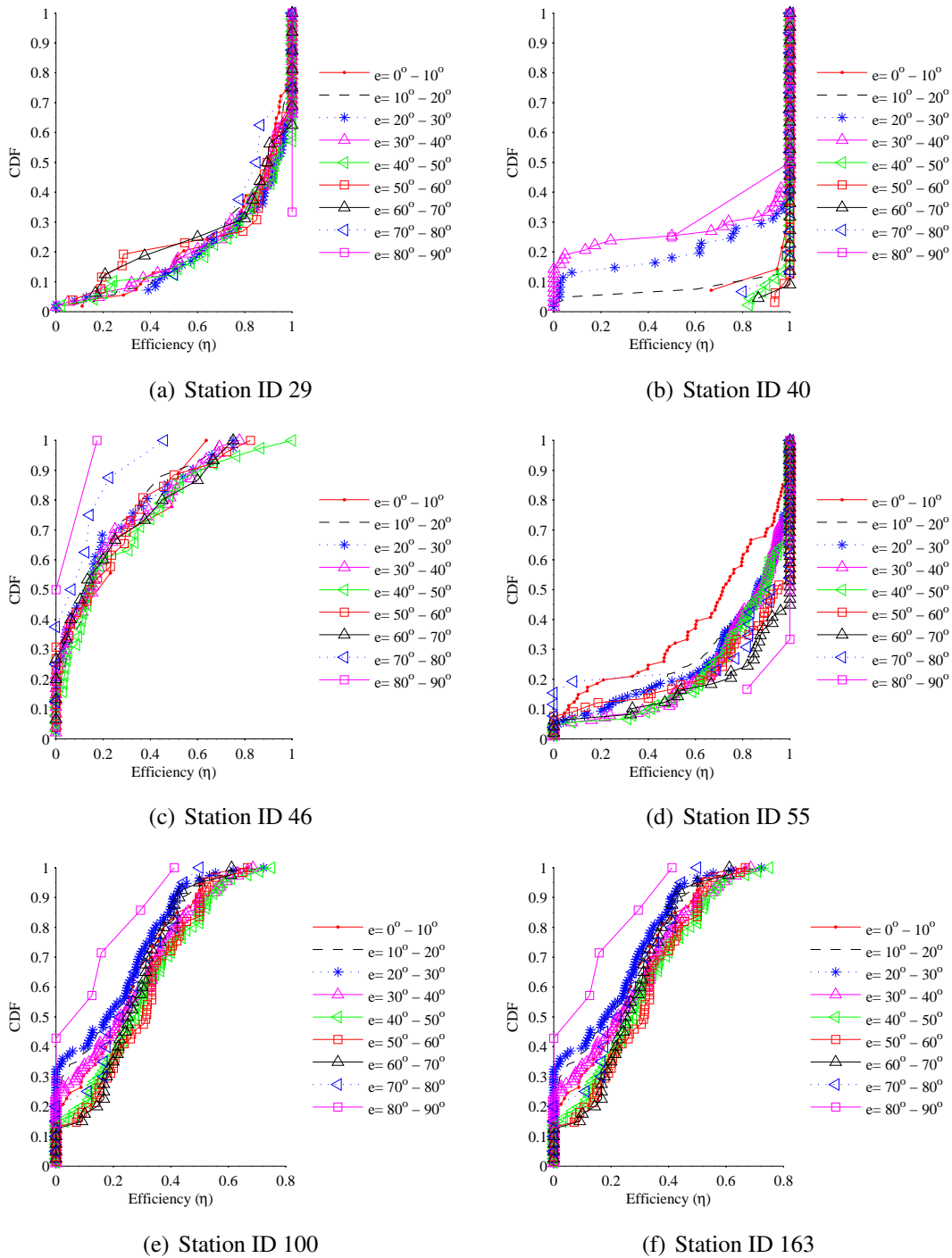


Figure 6.4: Cumulative distribution functions (CDFs) for RAX-2 download efficiency as a function of elevation (e) for download to representative ground stations over about a three month period.

stations and elevation angle ranges in Figure 6.2. The cumulative distribution function (CDF), which is the cumulative sum of the PDF, provides information about the probability that a random variable X will have a value less than or equal to x , i.e. $P(x \leq X)$. This provides information about the fraction of time (or the fraction of possible outcomes) that a given efficiency will be achieved or exceeded. Figure 6.4 shows the CDFs for the data in Figures 6.2 and 6.3. A CDF with a concave up shape is representative of a station and elevation with high performance most of the time, as in Figures 6.4(a) and 6.4(d). A CDF with a concave down shape indicates that the station and elevation performs poorly, as in Figure 6.4(e). Finally, a CDF with an inflection (i.e. where the curvature changes from concave down to concave up), as in Figure 6.4(b), indicates that there is a significant fraction of the solutions have low efficiencies and a significant fraction have high efficiencies. Note that the CDF characteristics are relatively consistent for the various elevation angles for a given station, where higher elevations tend to have slightly higher efficiencies relative to lower elevations.

There are several sources that may contribute to the probability distribution trends noted in the previous figures. A detailed investigation into the causes of the efficiency trends in Figures 6.2- 6.3 is outside the scope of this chapter. However, below we list several of these sources, how they impact download efficiency, and provide examples from the distributions in Figures 6.2- 6.3.

- Ground station efficiency may be proportional to elevation angle if the link budget margin is satisfied for high elevations but not for low elevations due to the physics of the problem (and this is the dominating factor influencing efficiency). In particular, at higher elevations there is a smaller range distance between the ground station and satellite, and thus the signal strength is higher and more likely to have higher efficiency, as modeled in the link budget in Chapter 2 [61]. This trend is observed in the mean efficiency values in 6.2 and has been observed anecdotally by RAX-2 spacecraft operators. For example, ground station ID 29 in Figure 6.3(a) has higher efficiencies for high elevation angles ($80 - 90^\circ$) and ground station ID 55 in Figure 6.3(d) also has higher efficiencies for high elevation angles.
- Local noise lowers signal strength and thus efficiency, which may impact efficiencies at lower elevations due to the reduced signal strength. This will yield the same effects as the previous source. We have observed anecdotally that local noise may be day or time dependent (e.g. caused by radio activity in the area at a certain time of the day or week); however have not yet extracted these trends from the efficiency data.
- Ground stations with low antenna gains will have consistently low signal strengths

and thus low efficiencies. For example, ground station ID 46 in Figure 6.3(c) and ground station ID 100 in Figure 6.3(e) have consistently low efficiencies for all elevation angles.

- Ground obstructions may limit the field of view of some ground stations for some azimuth angles and will result in low efficiencies at low elevation angles for some passes (i.e. those passes with acquisition of sight or loss of sight within the range of azimuth angles where the obstruction exists). Note that ground obstructions are less likely to impact efficiencies at higher elevations, where they are no longer in view. For example, there is a tree that obstructs certain azimuthal angles for ground station ID 163 at elevations $\leq 30^\circ$. Although we have generally avoided scheduling downloads where the tree would obstruct the download (which would minimize the effect in the data), this may account for the 20-40% of solutions at low elevation angles with low efficiencies in Figure 6.3(f). This may also explain the zero efficiencies observed for a small percentage ($< 20\%$) of passes for ground station ID 40 in Figure 6.3(b), which are inconsistent with the trends at higher elevations.
- The spacecraft travels at a higher angular rate relative to the ground station at high elevations when compared to low elevations. Thus, if the antenna is unable to slew at sufficiently high rates, this can result in inefficiencies at high elevation angles. For example, ground station ID 163 has relatively high efficiency for all elevation angles except high elevations (60-70°), which may be explained by this phenomenon. Ground station IDs 46 and 100 also perform worse at higher elevations, see Figures 6.3(c) and 6.3(e). Note this is the exact opposite effect that we may expect from the first two effects, thus may be difficult to extract.

These results indicate there is a clear relationship between download efficiency and the elevation angle. There is also a relationship between download efficiency data and azimuth angle (caused by reasons discussed above, such as obstructions). However, in a preliminary investigation, the efficiency data was found to be relatively constant for most azimuth angles, thus in this analysis we focus on the elevation data.

6.3 Summary and Future Directions

In this chapter, we described the approach used for scheduling the RAX-2 mission and to extract realistic download efficiency statistical data. We used PDFs and CDFs to model the statistical data in an analytically useful way. This enabled us to identify trends related to possible sources of stochasticity, as well as providing a useful tool for sensitivity analyses.

Future work should first aim to quantify the impact of stochastic download efficiency on the total download for *SMSP* solutions, i.e. mapping download efficiency stochasticity to solution performance. Methods to evaluate performance of solutions should be developed and applied to evaluate instances of *SMSP* with diverse types of download efficiency probability distributions. Example approaches to consider include sampling-based techniques or those that analytically combine probability distribution functions (e.g. convolution techniques in the case where the relationship between the sources of stochasticity and objective are linear). Finally, future work should investigate how probabilistic knowledge of download efficiency can be exploited to improve formulating and solving stochastic optimization problems.

CHAPTER 7

Conclusions and Future Work

7.1 Conclusions

We conclude by describing how the three main thesis contributions have been accomplished relative to the thesis goals. We also emphasize our work in the context of the existing literature and the larger implications of our work in the small space community.

A novel analytical modeling framework for generic space systems is the first contribution of this thesis. This addressed the need for a modular, extensible, and flexible modeling approach, which we have not found in the literature. This framework is novel because it enables both analytical and numerical assessment and optimization of spacecraft missions, and thus has great potential to improve the operations and design of existing and future spacecraft missions. We applied the framework to develop a *communication-focused* model and simulation environment of a satellite that downloads data to a globally distributed ground station network. This model enables the assess and optimize communication capacity, which is particularly challenging for highly-constrained small spacecraft missions architectures. An integrated simulation environment was developed that executes the model, which inherits modularity and flexibility from the modeling framework. The analysis and optimization formulations throughout the thesis use this foundational modeling and simulation framework.

The second contribution of this thesis was the constraint-based communication capacity assessment of spacecraft and ground networks. Constraint-based capacity is the maximum download capacity that can be achieved when a specific set of constraints is considered and all others are relaxed. Constraint-based analysis was addressed independently of schedule optimization in Chapter 3, and within the context of optimized schedules in Chapter 5. This contribution addressed the need for an approach to verify the feasibility of missions (by comparing to mission requirements), and to identify deficient and excess resources (such as on-board energy, data, and download time). In particular, we identified, quantified,

and compared the active constraints that limit communication capacity relative to mission requirements. This has implications for operational scheduling as well as the future design of spacecraft subsystems and ground station networks. For example, the active constraints for a given mission provide insight into the parameters (e.g. battery size, ground network size, data rate) that are limiting the mission from accomplishing its requirements and/or objectives, as well as how to exploit excess resources to optimize mission performance. Constraint-based assessments motivate the need for scheduling algorithms to ensure that constrained resources, such as time, energy, data, and ground resources, are efficiently allocated.

The final contribution of this thesis was an approach to formulate and solve operational scheduling problems for spacecraft missions and ground networks, which was addressed in Chapters 4-6. We presented optimization formulations that capture relevant decisions, key constraints, and the dynamic nature of spacecraft operations in Chapters 4-5 using the modeling framework from Chapter 2. Our formulations include realistic temporal and geographical constraints, on-board state constraints, multiple options for downloading (e.g. to select between available ground stations, data rates, etc.), and realistic efficiencies, which are elements often neglected in the satellite modeling, simulating, and scheduling literature. We developed algorithms to solve deterministic linear and non-linear instances of this scheduling problem and provided theoretical and computational results for generic and real-world mission problem instances in Chapter 4. The advantages of optimizing operational schedules relative to using conventional requirement-driven approaches (which are often sub-optimal) for operational planning have been demonstrated. This contribution has significant impact for small spacecraft operational scheduling as it provides an approach for optimal scheduling of spacecraft. For example, the RAX mission is able to exceed download mission requirements by a factor of five with an optimized schedule. We've extended the modeling framework and optimization formulation to solve a more complex problem that includes payload decisions and solved a variety of real-world LEO and interplanetary mission scenarios in Chapter 5. The results have highlighted the importance of considering the coupling of payload (e.g. related to on-board instruments) and download (e.g. related to transmitting data to ground stations) decisions for certain highly-constrained mission scenarios, particularly because some missions are infeasible or sub-optimal when important payload decisions are not considered. We've applied a variation of our scheduling algorithm to schedule the operational RAX-2 CubeSat. Furthermore, we've collected real-world efficiency data from the execution of the schedule, which provides information to improve our modeling and scheduling algorithms, see Chapter 6. Beyond optimizing schedules for specific mission scenarios, the model, simulation environment, and optimiza-

tion techniques in this thesis also enable exploration of the spacecraft and ground network design space. For example, we quantified how decisions and parameters related to satellite orbits, ground networks, vehicle design, and operations constrain communication capacity in the context of optimized schedules in Chapter 5. Finally, we discussed how stochasticity in problem parameters may impact total download and discussed approaches for modeling stochasticity in download efficiency in Chapter 6.

The work in this thesis provides fundamental approaches for modeling, assessing, and optimizing spacecraft and network problems towards enhanced communication capacity. This work captures key elements that are necessary (and often neglected) in the literature for modeling and simulating small spacecraft problem instances; including storage constraints, limited opportunities, and specific goals. We have motivated the need for optimal scheduling by exploring the design space and realizing the impact of realistic constraints on download potential. We have developed formulations and algorithms to optimally allocate these constrained resources to improve communication capacity and demonstrated the advantages of optimal scheduling for realistic and operational small satellite missions.

7.2 Future Work

The foundational modeling and simulation framework developed in this thesis lays the groundwork for several possible extensions.

7.2.1 Verification and Validation (V&V)

An important area of future work is to verify and validate (V&V) the models, simulators, and algorithms developed in this thesis. These are independent procedures that are used to check that a system satisfies requirements and objectives.

For small spacecraft systems (with limited resources), we suggest that V&V should first occur by comparing the outputs from the models/simulators in this thesis to those found by employing different simulation environments, such as those with a higher level of fidelity. The sensitivity to uncertain problem parameters should also be investigated to test the robustness of expected performance for a given solution. Beyond this, testing the generated schedules on realistic mission scenarios will provide critical information on its accuracy and performance. For example, it may be possible to have the spacecraft execute the schedule and then measure actual performance with ground-based or spacecraft sensors. This information, when compared to the expected performance, will provide necessary feedback to improve the scheduling algorithms, as in Figure 2.2 in Section 2.1.3. We have

collected some RAX-specific download efficiency data, as shown in Chapter 6, which has provided good information for future modeling and scheduling decisions. In the future, additional data should be collected and analyzed to perform a system-level V&V of the models, simulators, and schedulers, particularly as they become more complex.

7.2.2 Operational Planning for Complex Spacecraft

Upcoming small spacecraft are increasingly complex, which results in new operational planning challenges relative to those addressed in this thesis. Many emerging and future small spacecraft use active orbit and attitude control, deployable and controllable solar panels and antennas, peak-power tracking energy collection systems, maneuverable science instruments and antennas, on-board processing capabilities, and variable-rate radios. By contrast, for many existing small spacecraft these subsystems are passive, and do not actively change their operating points (i.e. data rates, power levels, attitude). Subsystems that have the potential to dynamically change their operating points introduce new states, dynamics, constraints, decisions, and interactions between these elements relative to the operational problems addressed in this thesis. For example, slewing the spacecraft to improve the antenna gain in the direction of a transmitting or receiving communication node (e.g. another spacecraft or ground station) will improve signal strength, yielding increased data rate or communication transmission efficiency. However, to accomplish this maneuver, additional power is consumed by the attitude control system and slewing may result in a loss of time to collect science data if the instrument is not pointed in the optimal direction.

The framework presented in this thesis models states, decisions, dynamics, constraints, and objectives using a generic template, thus we believe it can be extended to consider more complex space systems. For example, we demonstrated how the payload subsystem, and its decisions, dynamics, constraints, and interactions with other subsystems, were added to create an extended formulation in Chapter 5 relative to the simpler communication-focused formulation presented in Chapter 4. This new model allowed us to make trade-offs between the energy and data acquired and consumed in both payload and communication operations towards the operational objective of the mission. Future models will integrate communication, payload, data processing, position determination and control, attitude determination and control, thermal determination and control, and power management subsystems (and possibly others) to capture the complexity of the future generation of spacecraft systems. Developing optimization algorithms to make intelligent decisions about when and how to perform spacecraft operational decisions that impact multiple subsystems and states should be investigated in future work. We expect new challenges in these more complex mission

scenarios; such as non-linear relationships between states and decisions as well as problems that scale significantly with problem size due to complex interactions between the decisions and states.

Future work should address the computational issues of solving the more complex problems proposed in this section. In particular, the efficiency of algorithms used to solve these problems should be investigated, both computationally and theoretically.

7.2.3 Applications to Multi-Satellite Missions

There are several space mission architectures being developed and proposed that consist of multiple spacecraft. Examples include formation flying small spacecraft [185, 186], spacecraft that have inter-satellite communication links [187], constellation missions [25, 188], and spacecraft that act as motherships that deploy and possibly act as a communication relay for multiple smaller spacecraft [189]. In addition, ground stations are being networked in new ways to create globally distributed, independently owned and operated networks [9]. Many of the models, simulations, and optimization work presented in this thesis should be extended to capture these multi-mission, multi-network scenarios.

The work in this thesis lays the ground work for optimization algorithms to distribute excess ground station capacity to satellite users through intelligent deployment coordination and flexible scheduling. In future work, a network-wide model and simulation environment will be developed by integrating the ground station modeling and capacity analysis tools, spacecraft models, and high-fidelity simulators that can manage multiple spacecraft.

First, future work should investigate communication capacity trends for these larger and more complex space mission problems. The impact of single spacecraft orbits and ground networks on communication capacity discussed in Section 3.1.2.1 (see Figure 3.5) should be extended to consider multiple individual, constellation, and networked spacecraft missions communicating to networked ground stations. Overall, we expect different capacity trends to emerge when multiple spacecraft are considered relative to the trends observed with a single spacecraft. For example, multiple spacecraft in equatorial orbits will quickly saturate globally distributed or equatorial ground networks. However, multiple spacecraft in polar orbits may be able to exploit distributed ground networks and yield higher communication capacity. Sensitivity to orbital altitude, inclination, and eccentricity, as well as ground station distribution, are also important to consider in these multi-mission studies.

Second, future research should investigate scheduling multiple independent or networked satellites communicating to one another or multiple global ground networks. The optimization formulations presented in this thesis may be extended to accommodate these

types of problems by adding more space nodes and additional constraints that enforce that communication nodes (e.g. spacecraft or ground station) communicate only to a single other communication node at a time (depending on the practical constraints of the system). Scheduling multiple satellites is challenging because individual spacecraft or network goals and constraints may conflict with the goals and constraints of other spacecraft, missions, or networks. This problem will need to consider additional elements such as network availability, conflict, priority, and financial cost constraints. The framework presented in this thesis may be applied to manage additional constraints, for example, financial cost could be an additional state that increases with every pass over a ground station used for download to represent the cost of reserving that ground station. This financial cost state could be subject to a constraint (e.g. upper bound on total cost), or alternatively could be optimized (e.g. minimize the total cost of the mission).

7.2.4 Stochasticity in Operational Scheduling Problem

Beyond the initial investigation into the effects of stochasticity on download performance in Chapter 6, there are several additional sources of stochasticity in the satellite operational problem that should be studied, such as those described at the beginning of Chapter 6. Realistic data representative of these sources should be collected and characterized for both LEO and interplanetary missions to enhance the modeling and optimization algorithms to solve these types of problems.

The impact of stochasticity in problem parameters on solution performance when there are diverse sources of stochasticity should be addressed. An initial investigation in the context of stochastic download efficiencies for LEO satellites was demonstrated in Chapter 6; however this should be extended to include a more in-depth analysis and other sources of stochasticity should be considered. Future work should bound the solution space to understand how stochastic effects improve or worsen solutions. In addition, future work should investigate the relationship between deterministic causes for inefficiencies (e.g. azimuth angles, obstructions, local time of day) and incorporate these causes into the scheduling models.

Future work should investigate approaches to measure the impact of multiple correlated sources of stochasticity on *SMSP* solutions. Future work should investigate theoretical and computational issues related to the use of convolutions to assess and optimize stochastic problem instances. Applicability to a variety of real-world and generic problem instances should also be studied.

Future research should investigate approaches to scheduling when stochastic problem

parameters may cause infeasibilities in the solutions. Feasibility of solutions is an issue when there is stochasticity in the constraint parameters such that decisions early in the planning horizon may impact the ability to satisfy future constraints (e.g. on-board energy and data capacity constraints). Dynamic approaches are required to accommodate this type of stochasticity. Many dynamic approaches that include re-planning, such as planning with recourse, have been discussed in the literature review, which may be applicable to this problem. Future work will extend the work in this area to address stochastic satellite operations problems.

7.2.5 Coupled Vehicle and Operations Optimization

Future work should extend the simple trade-space exploration concepts addressed in Chapter 5. In particular, future work should develop models and algorithms to simultaneously optimize vehicle, network, and operational decisions while considering realistic constraints, dynamics, and objectives. Conventional trade-study approaches for vehicle and network design typically use back-of-the-envelope approximations, neglect the dynamic nature of realistic constraints, and do not use scheduling techniques. This often yields operational vehicle, and network design solutions that are sub-optimal. An integrated approach that couples vehicle and operational decisions has distinct advantages over conventional approaches because it optimizes the schedule for each design option. This verifies the feasibility of possible solutions and enables accurate comparisons of design solutions. For example, consider the significant variation in eclipse duration throughout a year, as demonstrated in Figure 3.12. Designing solar panels, batteries, or operational schedules with either the best-case, average-case, or worst-case eclipse conditions would each yield infeasible or sub-optimal solutions at some time throughout the year. This example demonstrates the importance of considering the dynamic nature of eclipse durations and other environmental factors when designing vehicles and operational plans. This approach will also enable incorporating additional constraints into the model, such as mass, volume, or financial cost limitations, which become important when the complete mission is considered. Non-linear optimization techniques may be required to solve these types of problems due to the expected non-linearities in the constraints and/or objectives.

7.2.6 Applications to Interplanetary Missions

The models and algorithms presented in this thesis are applicable to space architectures beyond LEO missions, as demonstrated in Chapter 5. However, our models and algorithms

should be extended to address challenges and constraints specific to interplanetary communication.

One of the key challenges for current and upcoming interplanetary missions is managing the high demand and competition for access to deep space networks (e.g. DSN), which support interplanetary missions and orbiters (e.g. ExoMars at Mars), that are used as inter-satellite communication nodes to relay data back to Earth. There is significant financial cost and logistical limitations associated with using these types of communication architecture. For example, acquiring time on the DSN requires special authorization and the cost to use the DSN, the *Apeture Cost*, can be a significant fraction of mission costs [190]. Conflict resolution for these types of missions, including routing and prioritization [191], is an active area of research [116, 131, 132]. However, most of the work in this area focuses on scheduling spacecraft requests from the perspective of the DSN. Future work should investigate conflict resolution scheduling that considers realistic on-board constraints and objectives for individual spacecraft/ground stations, as well as financial, and temporal constraints (e.g. due to limited opportunities for communication links).

Other challenges in interplanetary communication are related to the great communication distances for these mission architectures. For example, the transmission time for communication between Mars and Earth ranges from approximately 3 to 20 minutes, compared to about 0.0022 seconds for LEO communication with a satellite at an altitude of 650 km. In addition, feasible data rates are low because the signal strength is proportional to the inverse of the square of the distance between the communication nodes [61]. This results in challenges in sending, acknowledging, re-sending, and changing commands on-the-fly. Delay Tolerant Networking (DTN), a standard, secure, store and forward mechanism for high latency and bandwidth delay product has been proposed to provide reliable deep space links [191]. In addition, due to the orbital constraints of interplanetary missions, there are often limited uplink opportunities, which must be considered when scheduling. As a result, generating schedules for long-term planning horizons may be necessary, which will yield larger scheduling problems that may require a significant increase in computational effort to solve. Furthermore, accommodation of non-deterministic events must be incorporated into long-term schedules due to the higher probability of diverging from a prescribed schedule with a longer planning horizon. Future work should incorporate the effects of long transmission times, low data rates, DTN concepts, and limited uplink opportunities into the scheduling problems addressed in this thesis.

7.2.7 Summary

The work in this thesis has applicability not only in the space communication domain, but in other domains as well. In particular, the modeling and simulation framework in this thesis captures key elements that are common to many real-world problem instances such as opportunities, states, constraints, objectives, and stochasticity for a system with multiple interacting nodes (e.g. satellites, ground stations). The models and algorithms can be used to represent a variety of problem instances, and the constraint-based analysis and optimization techniques presented in this thesis are applicable to these problems. For example, the work in this thesis can be applied to other complex vehicles with constrained resources and opportunities to perform mission-specific goals such as UAVs, robots, high-altitude balloons, and spacecraft constellations, as well as the multi-satellite and interplanetary mission architectures described above.

APPENDIX A

Analytic Link Budget

We derive an analytic link budget to assess the communication potential and for use in our optimization algorithms. We use the link budget to solve for the feasible data rate for communication between two communication nodes with known communication systems and a given path distance, S .

The equivalent isotropic radiated power, EIRP, is a function of the transmit power, P_t , transmit line losses, L_t , and gain of the transmitting antenna, G_t ,

$$\text{EIRP} = P_t + G_t - L_t \quad (\text{A.1})$$

The power at the receiver, P_r , is a function of EIRP, the receiver antenna gain, G_r , the pointing error losses, L_e , the polarization losses, L_p , the receive line losses, L_l , and the space loss, L_s ,

$$P_r = \text{EIRP} + G_r - L_e - L_p - L_l - L_s \quad (\text{A.2})$$

L_s is the space loss which varies dynamically throughout the orbit and is defined,

$$L_s = \left(\frac{c}{4f\pi S} \right)^2, \quad (\text{A.3})$$

where f is the frequency of the transmitted signal, c is the speed of light, and S is the path length. The received carrier-to-noise received power ratio, C/N_0 , is then computed as a function of P_r , the system noise temperature, T_s , and the Boltzmann constant, k ,

$$C/N_0 = P_r - 10 \log_{10}(T_s) - 10 \log_{10}(k) \quad (\text{A.4})$$

The signal-to-noise ratio is a function of C/N_0 and the data rate, r ,

$$E_b/N_0 = C/N_0 - 10 \log_{10}(r) \quad (\text{A.5})$$

The minimum required signal-to-noise ratio, $E_b/N_{0,min}$, is a function of the implementation and modulation scheme. In accordance with conservative design practices, the link margin, the difference between E_b/N_0 and the minimum required carrier-to-noise spectral power density ratio, $E_b/N_{0,min}$, must exceed the required link margin, M [61],

$$M \leq E_b/N_0 - E_b/N_{0,min}. \quad (\text{A.6})$$

There are two approaches for modeling data rate in our analytic optimization formulation, using a *constant* or *variable* data rate. When the data rate is constant, a single data rate is selected, and opportunities for communication exist only when the analytic link budget in Eqs. A.1-A.6 is satisfied. Alternatively, when the data rate is modeled as variable, the rate is computed based on the time-dependent distance between the communication links. In this case, we solve Eqs. A.5-A.6 to solve for the maximum feasible data rate as a function of path distance, S .

APPENDIX B

Derivation of Three Branching Cases

In this section we derive the three possible outcomes summarized in Table 4.6. A linear program with four decision variables will have four binding constraints in the optimal solution. This yields three possible cases:

1. **Option 2 is not used** When Option 2 is not used, $x_2 = 0$ and $q_2 = 0$. By 4.25, the solution is,

$$q_1 = \min(t\phi_1, \frac{e}{\alpha_1}) = t\phi_1 q_2 = 0 \quad (\text{B.1})$$

with objective value,

$$q_1 + q_2 = t\phi_1 \quad (\text{B.2})$$

and binding constraints 4.17, 4.18, 4.20, 4.22, and 4.24

2. **Option 1 is not used** When Option 1 is not used $x_1 = 0$ and $q_1 = 0$. By 4.25, the solution is,

$$q_1 = 0 \quad (\text{B.3})$$

$$q_2 = \min(t\phi_2, \frac{e}{\alpha_2}) = \frac{e}{\alpha_2} \quad (\text{B.4})$$

with the objective value,

$$q_1 + q_2 = \frac{e}{\alpha_2} \quad (\text{B.5})$$

and binding constraints 4.16, 4.18, 4.20, 4.21, and 4.23.

3. **Option 1 and Option 2 are used** When both options are used to download data, the variable constraints 4.21-4.24 are not active. Also, by the assumption that data

is unlimited, constraint 4.19 is not active. Therefore, the remaining four constraints 4.16, 4.17, 4.18, and 4.20 must be. Since x_1 and x_2 are a convex combination define,

$$\lambda = x_1 \quad (\text{B.6})$$

$$1 - \lambda = x_2 \quad (\text{B.7})$$

By solving the system of equations formed by the three binding constraints,

$$q_1 = t\phi_1\lambda \quad (\text{B.8})$$

$$q_2 = t\phi_2(1 - \lambda) \quad (\text{B.9})$$

$$\alpha_1q_1 + \alpha_2q_2 = e \quad (\text{B.10})$$

we get the solution,

$$\lambda = \frac{\alpha_2\phi_2 - \frac{e}{t}}{\alpha_2\phi_2 - \alpha_1\phi_1} \quad (\text{B.11})$$

and objective value,

$$q_1 + q_2 = \frac{t\phi_1\phi_2(\alpha_2 - \alpha_1) + e(\phi_2 - \phi_1)}{\alpha_2\phi_2 - \alpha_1\phi_1} \quad (\text{B.12})$$

The difference between the objective value for Case 3 and Case 1 is,

$$\Delta_1 = B.12 - B.2 = \frac{\frac{e}{t} - \alpha_1\phi_1}{\alpha_2\phi_2 - \alpha_1\phi_1} t(\phi_2 - \phi_1) \quad (\text{B.13})$$

If $\phi_2 > \phi_1$, then $\Delta_1 > 0$ and Case 3 provides a better solution than Case 1. Similarly, the difference between the objective value for Case 3 and Case 2 is,

$$\Delta_2 = B.12 - B.5 = \frac{t\phi_1}{\alpha_2} \frac{\alpha_2\phi_2 - \frac{e}{t}}{\alpha_2\phi_2 - \alpha_1\phi_1} (\alpha_2 - \alpha_1) \quad (\text{B.14})$$

$$(\text{B.15})$$

If $\alpha_2 > \alpha_1$, then $\Delta_2 > 0$ and Case 3 provides a better solution than Case 2. Therefore, if 4.25 holds in addition to $\phi_2 > \phi_1$ and $\alpha_2 > \alpha_1$, then a fractional solution is uniquely optimal.

APPENDIX C

Lemmas and Theorems

C.1 Proof to Lemma 1.1

Lemma 1.1 Assume we are given a solution to an instance of *UCF* in which rate ϕ_{io}^* is used to download a total of q_{io}^* data during an interval i (where $*$ denotes the solution). Then, download occurs for $p = \frac{q_{io}^*}{\phi_{io}^* \cdot (t_{i+1} - t_1)}$ fraction of interval i , and we can construct an equivalent solution in which data is downloaded at the constant rate $\phi_{io}^* \cdot p$ over the entire duration of the interval.

Proof 1.1 A solution to *UCF* specifies the amount of data to download during an interval (q_{io}^*) and the rate at which this download occurs (ϕ_{io}^*). For those cases where the specified amount of data and download rate will not fill the entire duration of the interval (i.e. $0 < p < 1$), the solution does not specify when in the interval the download must occur. In addition, the physics of the system do not require that the download occur over a single continuous interval. We may thus assume without loss of generality that the full interval is split into smaller sub-intervals of equal duration, and that during each interval we download for the first p fraction of time and then do not download for the remainder of this sub-interval. As the size of these sub-intervals approaches zero, the download rate approaches the constant download rate $\phi_{io}^* \cdot p$ over the entire duration of the interval i .

Lemma 1.2 Given a solution to an instance of *UCF* in which a total of s_i^{e*} energy is spilled during interval i (where $*$ denotes the solution), then spillage occurs for $f = \frac{s_i^{e*}}{\phi_{io}^* \cdot (t_{i+1} - t_1)}$ fraction of interval i . We can construct an equivalent solution in which energy is spilled at the constant rate $s_i^{e*} \cdot f$ over the entire interval duration.

Proof 1.2 The identical approach taken in *Lemma 1.1* can be used to prove *Lemma 1.2*.

C.2 Algorithm 2 Theorem

Theorem 4.4.1 An anticipative greedy approach is an optimal approach to assigning download schedules within an interval and verifying the feasibility of solutions to instances of *SMSP*.

Proof 4.4.1 An anticipative greedy approach ensures the assigned downloads are *feasible* by construction because it uses a constraint to guarantee feasibility for all future sub-intervals within the interval. Algorithm 2 is *optimal* since it uses a greedy approach and thus maximizes download and there is no advantage in delaying download.

Now that we have established the general approach for assigning downloads, we describe Algorithm 2 in more detail. Algorithm 2 starts at the first sub-interval and proceeds forward, performing two important functions for each sub-interval. First, it computes the maximum achievable download using an anticipative greedy approach, constrained by the interval-specific optimal download by the *UCF* solution. Second, assuming this amount of data is downloaded, it verifies that sufficient energy, data, and time remain during this interval to support the optimal download by *UCF*. Algorithm 2 checks every sub-interval and returns if the download is feasible or not, and in the latter case, returns the first infeasible sub-interval using the greedy anticipative approach. By simultaneously assigning downloads and checking feasibility during each sub-interval and stopping when any infeasibilities arise, Algorithm 2 minimizes the required effort to check the feasibility of each interval for a given solution.

APPENDIX D

Non-linear Data Descriptions

The data characterizations are defined below.

- *Irregularity* is a measure of the change in the dynamics during each sub-interval relative to the buffer capacity. This is measured independently for energy and data dynamics. Dynamics with a high level of irregularity will increase or decrease by as much as the buffer size over an interval, while data with low irregularity will be relatively constant (i.e. $\delta^{e+} + \delta^{e-} \approx 0$). Irregularity for energy and data dynamics, γ_e and γ_d , respectively, are defined analytically,

$$\gamma_e = \frac{1}{n} \sum_{i=1}^n \frac{(\delta_i^{e+} - \delta_i^{e-}) \Delta t_{av}}{(e_{max} - e_{min})(t_{i+1} - t_i)}, \quad (\text{D.1})$$

$$\gamma_d = \frac{1}{n} \sum_{i=1}^n \frac{(\delta_i^{d+} - \delta_i^{d-}) \Delta t_{av}}{(d_{max} - d_{min})(t_{i+1} - t_i)}, \quad (\text{D.2})$$

where changes in dynamics during interval i are normalized by the average interval duration of the planning horizon, Δt_{av} , divided by the duration of that interval, $t_{i+1} - t_i$. The number of intervals in the planning horizon is n .

- *Synchronization* is a measure of the similarity between energy and data dynamics. Note this only characterizes the nominal dynamics of a problem, and not the effects due to data download. Data with a high level of synchronization is characterized by energy and data dynamics that have slopes with the same sign, i.e. both increasing or decreasing, while for dynamics with low level of synchronization, the dynamics will have opposite signs, i.e. one resource will be increasing while the other is decreasing. Synchronization, ζ , is defined analytically as the sum of the intervals where the

dynamics are synchronized normalized by the number of intervals in the planning horizon, n . $\{S\}$ represents the sum of the set S , where $\delta_i^{e+} \cdot \delta_i^{e-}$ checks that the dynamics have the same sign.

$$\zeta = \frac{1}{n} |\{i \in I | (\delta_i^{e+} - \delta_i^{e-}) > 0, (\delta_i^{e+} \cdot \delta_i^{e-}) > 0\}| \quad (\text{D.3})$$

BIBLIOGRAPHY

- [1] Burlacu, M.-M. and Lorenz, P., “A survey of small satellites domain: challenges, applications and communications key issues,” *iCaST ICST’s Global Community Magazine*, Vol. 18, September 2010.
- [2] Helvajian, H. and Janson, S., *Small Satellites: Past, Present, and Future*, The Aerospace Press, August 2009.
- [3] Cutler, J., Linder, P., and Fox, A., “A Federated Ground Station Network,” *SpaceOps Conference*, AIAA, Houston, TX, October 2002.
- [4] de Milliano, M. and Verhoeven, C., “Towards the next generation of nanosatellite communication systems,” *Acta Astronautica*, Vol. 66, No. 9-10, 2010, pp. 1425 – 1433.
- [5] Woellert, K., Ehrenfreund, P., Ricco, A. J., and Hertzfeld, H., “Cubesats: Cost-effective science and technology platforms for emerging and developing nations,” *Advances in Space Research*, Vol. 47, No. 4, 2011, pp. 663 – 684.
- [6] Baker, D. N. and Worden, S. P., “The Large Benefits of Small-Satellite Missions,” *Transactions American Geophysical Union*, Vol. 89, No. 33, Aug. 2008, pp. 301.
- [7] Moretto, T. and Robinson, R. M., “Small Satellites for Space Weather Research,” *Space Weather Journal*, Vol. 6, No. 5, May 2008, pp. S05007.
- [8] Moretto, T., “CubeSat Mission to Investigate Ionospheric Irregularities,” *Space Weather*, Vol. 6, No. 11, 2008.
- [9] Crowley, G., Fish, C. S., Bust, G. S., Swenson, C., Barjatya, A., and Larsen, M. F., “Dynamic Ionosphere Cubesat Experiment (DICE),” *American Geophysical Union (AGU) Fall Meeting*, Dec. 2009, pp. A6.
- [10] Rowland, D., Weatherwax, A., Klenzing, J., and Hill, J., “The NSF Firefly Cubesat: Progress and Status,” *American Geophysical Union (AGU) Fall Meeting*, Dec. 2009, pp. A7.
- [11] Lin, R., Parks, G., Halekas, J., Larson, D., Eastwood, J., Wang, L., Sample, J., Horbury, T., Roelof, E., Lee, D., Seon, J., Hines, J., Vo, H., Tindall, C., Ho, J., Lee, J., and Kim, K., “CINEMA (Cubesat for Ion, Neutral, Electron, MAgnetic fields),” *American Geophysical Union (AGU) Fall Meeting*, Dec. 2009, pp. A9.

- [12] Li, X., Palo, S. E., Turner, D. L., Gerhardt, D., Redick, T., and Tao, J., “CubeSat: Colorado Student Space Weather Experiment,” *American Geophysical Union (AGU) Fall Meeting*, Dec. 2009, pp. C1585.
- [13] Klumpar, D. M., Spence, H. E., Larsen, B. A., Blake, J. B., Springer, L., Crew, A. B., Mosleh, E., and Mashburn, K. W., “FIREBIRD: A Dual Satellite Mission to Examine the Spatial and Energy Coherence Scales of Radiation Belt Electron Microbursts,” *American Geophysical Union (AGU) Fall Meeting*, Dec. 2009, pp. A8.
- [14] Ricco, A., Hines, J., Piccini, M., Parra, M., Timucin, L., Barker, V., Storment, C., Friedericks, C., Agasid, E., Beasley, C., Giovangrandi, L., Henschke, M., Kitts, C., Levine, L., Luzzi, E., Ly, D., Mas, I., McIntyre, M., Oswell, D., Rasay, R., Ricks, R., Ronzano, K., Squires, D., Swaiss, G., Tucker, J., and Yost, B., “Autonomous Genetic Analysis System to Study Space Effects on Microorganisms: Results from Orbit,” *Solid-State Sensors, Actuators and Microsystems Conference (Transducers)*, 2007, pp. 33–37.
- [15] Diaz-Aguado, M., Ghassemieh, S., Van Outryve, C., Beasley, C., and Schooley, A., “Small Class-D spacecraft thermal design, test and analysis - PharmaSat biological experiment,” *IEEE Aerospace Conference*, March 2009.
- [16] Minelli, G., Ricco, A., and Kitts, C., “O/OREOS Nanosatellite: A Multi-Payload Technology Demonstration,” *24th Small Satellites Conference*, Logan, Utah, August 2010.
- [17] Johnson, L., Whorton, M., Heaton, A., Pinson, R., Laue, G., and Adams, C., “NanoSail-D: A solar sail demonstration mission,” *Acta Astronautica*, Vol. 68, No. 5-6, 2011, pp. 571–575, Special Issue: Aosta 2009 Symposium.
- [18] Borowski, H., Reese, K., and Motola, M., “Responsive access to space: Space Test Program Mission S26,” *IEEE Aerospace Conference*, Big Sky, MT, March 2010.
- [19] Ridley, A., Forbes, J., Cutler, J., Nicholas, A., Thayer, J., Fuller-Rowell, T., Matsuo, T., Bristow, W., Conde, M., Drob, D., Paxton, L., Chappie, S., Osborn, M., Dobbs, M., Roth, J., and Armada Mission Team, “The Armada mission: Determining the dynamic and spatial response of the thermosphere/ionosphere system to energy inputs on global and regional scales,” *American Geophysical Union (AGU) Fall Meeting*, Dec. 2010, pp. A7.
- [20] Cutler, J. and Fox, A., “A Framework for Robust and Flexible Ground Station Networks,” *AIAA Journal of Aerospace Computing, Information, and Communication*, Vol. 3, March 2006, pp. 73–92.
- [21] Spangelo, S., Boone, D., and Cutler, J., “Assessing the Capacity of a Federated Ground Station Network,” *IEEE Aerospace Conference*, Big Sky, MT, March 2010.
- [22] Cutler, J., Fox, A., and Bhasin, K., “Applying the lessons of Internet Services to Space Systems,” *IEEE Aerospace Conference*, Big Sky, MT, March 2009.

- [23] Pietras, J., "Air Force Satellite Control Network Interoperability Progress Report," Presented at the Ground Systems Architecture Workshop (GSAW), March 2003.
- [24] Group, S., "Universal Space Network," 2012.
- [25] Cutler, J., Linder, P., and Fox, A., "A Federated Ground Station Network," *SpaceOps Conference*, October 2002.
- [26] Towns, C., "History of project OSCAR," *73 Amateur Radio*, Vol. 332, 1998, pp. 27–29.
- [27] Cutler, J. and Hutchins, G., "OPAL: Smaller, Simpler, and Just Plain Luckier," *Small Satellite Conference*, Logan, UT, August 2000.
- [28] Anderson, C. and Kitts, C., "A MATLAB expert system for ground-based satellite operations," *IEEE Aerospace Conference*, Big Sky, MT, March 2005.
- [29] Cutler, J., "Ground Station Virtualization," *The Fifth International Symposium on Reducing the Cost of Spacecraft Ground Systems and Operations (RCSGSO)*, Pasadena, CA, July 2003.
- [30] "Global Educational Network for Satellite Operations," <http://www.genso.org/>, July 2012.
- [31] Rash, J., Parise, R., Hogie, K., Criscuolo, E., Langston, J., Jackson, C., and Price, H., "Internet Access to Spacecraft," *Small Satellite Conference*, Logan, UT, August 2000.
- [32] Hogie, K., Rash, J., and Casasanta, R., "Internet Data Delivery for Future Space Systems," *Earth Science Technology Conference*, Pasadena, CA, June 2002.
- [33] Janicik, J. and Wolff, J., "The CHIPSat Spacecraft Design - Significant Science on a Low Budget," *International Society for Optical Engineering (SPIE)*, San Diego, CA, August 2004.
- [34] Jiong, L., Zhigang, C., and Junaid, K., "TP-Satellite: A New Transport Protocol for Satellite IP Networks," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 45, No. 2, 2009, pp. 502–515.
- [35] Wang, R., Gutha, B., and Rapet, P., "Window-based and rate-based transmission control mechanisms over space-Internet links," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 44, No. 1, 2008, pp. 157–170.
- [36] Hogie, K., "STS-107 Mission: End-to-End IP Space Communication Results," *Third Space Internet Workshop*, June 2004.
- [37] Hogie, K., Parise, R., and Criscuolo, E., "Link and Routing Issues for Internet Protocols in Space," *IEEE Aerospace Conference*, Big Sky, MT, March 2001.

- [38] Will Ivancic, W. M. E. and Stewart, D., “Experience with delay-tolerant networking from orbit,” *Advanced Satellite Mobile Systems Conference*, August 2008.
- [39] Wang, R., Shrestha, B., Wu, X., Wang, T., Ayyagari, A., Tade, E., Horan, S., and Hou, J., “Unreliable CCSDS File Delivery Protocol (CFDP) over Cislunar Communication Links,” *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 46, No. 1, 2010, pp. 147–169.
- [40] Zimmermann, H., “OSI Reference Model–The ISO Model of Architecture for Open Systems Interconnection,” *IEEE Transactions on Communication*, Vol. 28, No. 4, April 1980, pp. 425 – 432.
- [41] “NASA Systems Engineering Handbook,” National Aeronautics and Space Administration (NASA), SP-610S, 1995.
- [42] Wymore, A. W., *Model-Based Systems Engineering*, CRC Press, 1993.
- [43] Technical Operations, I. C. o. S. E. I., “Systems Engineering Vision 2020,” INCOSE-TP-2004-004-02, September 2007.
- [44] Dickerson, C. and Mavris, D. N., *Architecture and Principles of Systems Engineering*, CRC Press, 2009.
- [45] Estefan, J. A., “Developing a Strategy and Roadmap for Advancing the State-of-the-Practice of MBSE within Your Organization,” Plenary Briefing (IC-MBSE), 2010.
- [46] Hein, A. M., Lopez, R. P., Herzig, S., and Brandstatter, M., “Object-Oriented System Models in Spacecraft Design: First Steps towards an Application in Phase B,” *4th International Workshop on System & Concurrent Engineering for Space Applications (SECESA)*, 2011.
- [47] Ogren, I., “On principles for model-based systems engineering,” *Systems Engineering*, Vol. 3, No. 1, 2000, pp. 38–49.
- [48] Jansma, P. A. T. and Jones, R. M., “Advancing the Practice of Systems Engineering at JPL,” *IEEE Aerospace Conference*, Big Sky, MT, March 2006.
- [49] Delp, C. L., “INCOSE MBSE Grand Challenge,” Space Systems Working Group Entry and MBSE, 2008.
- [50] Delp, C., Bindschadler, D., Wollaeger, R., Carrion, C., McCullar, M., Jackson, M., Sarrel, M., Anderson, L., and Lam, D., “MOS 2.0 - modeling the next revolutionary Mission Operations System,” *IEEE Aerospace Conference*, March 2011.
- [51] Spangelo, S., Kaslow, D., Delp, C., Cole, B., Anderson, L., Fosse, E., Hartman, L., Gilbert, B., and Cutler, J., “Applying Model Based Systems Engineering (MBSE) to a Standard CubeSat,” *IEEE Aerospace Conference*, Big Sky, MT, March 2012.
- [52] Qamar, A., “An Integrated Approach towards Model-Based Mechatronic Design,” KTH Royal Institute of Technology, 2011.

- [53] Karban, R., Weilkens, T., Hauber, R., and Diekmann, R., “MBSE Practices in Telescope Modeling,” INCOSE MBSE Challenge Team, February 2010.
- [54] Brandsttter, M. and Eckl, C., “Multi-disciplinary System Engineering and the Compatibility Modeling Language (U)CML,” *Journal on Systemics, Cybernetics and Informatics*, Vol. 7, 11-16 2009.
- [55] Wertz, J. and Larson, W., *Space Mission Analysis and Design*, Microcosm Press, 3rd ed., 1999.
- [56] Salmasi, A. and Rahmat-Samii, Y., “Beam Area Determination for Multiple-Beam Satellite Communication Applications,” *IEEE Transactions on Aerospace and Electronic Systems*, Vol. AES-19, No. 3, May 1983, pp. 405 –412.
- [57] Beste, D., “Design of Satellite Constellations for Optimal Continuous Coverage,” *IEEE Transactions on Aerospace and Electronic Systems*, Vol. AES-14, No. 3, May 1978, pp. 466 –473.
- [58] Ballard, A., “Rosette Constellations of Earth Satellites,” *IEEE Transactions on Aerospace and Electronic Systems*, Vol. AES-16, No. 5, 1980, pp. 656 –673.
- [59] Ganz, A., Gong, Y., and Li, B., “Performance study of low Earth-orbit satellite systems,” *IEEE Transactions on Communication*, May 1994.
- [60] Zhu, K., Li, J., and Baoyin, H., “Satellite Scheduling Considering Maximum Observation Coverage Time and Minimum Orbital Transfer Fuel Cost,” *Acta Astronautica Sinica*, Vol. 66, 2010, pp. 220 – 229.
- [61] Horan, S., Minnix, T., and Vigil, J., “Small satellite access of the Space Network,” *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 35, No. 4, Oct. 1999, pp. 1173 –1182.
- [62] Beering, D., Tseng, S., Hayden, J., Corder, A., Ooi, T., Elwell, D., Grabowski, H., Frederic, R., Franks, J., Fish, R., Johnson, A., and Gavin, N., “RF Communication Data Model for Satellite Networks,” *IEEE Military Communications Conference*, Piscataway, NJ, USA, 2009.
- [63] Cutler, J. and Boone, D., “Assessing Global Ground Station Capacity,” CubeSat Developers’ Workshop, April 2009.
- [64] Boone, D., “A Study of CubeSat Orbital Separation and Ground Station Capacity using Satellite Tool Kit,” AERO/AOSS 590 Final Report, Space Systems Project, University of Michigan, Ann Arbor, April 2009.
- [65] Weck, O. D., Neufville, R. D., and Chaize, M., “Staged Deployment of Communications Satellite Constellations in Low Earth Orbit,” *Journal of Aerospace Computing, Information, and Communication*, 2004, pp. 119–136.

- [66] McFadden, J., Ergun, R., Carlson, C., Herrick, W., Loran, J., Verneti, J., Teitler, W., Bromund, K., and Quinn, T., "Science Operations and Data Handling for the FAST Satellite," *Space Science Reviews*, Vol. 98, No. 1-2, 2001, pp. 169 – 96.
- [67] Fu, A. C., Modiano, E., and Tsitsiklis, J. N., "Optimal energy allocation and admission control for communications satellites," *IEEE/ACM Transactions on Networks*, Vol. 11, No. 3, June 2003, pp. 488–500.
- [68] Vassaki, S., Panagopoulos, A., and Constantinou, P., "Effective Capacity and Optimal Power Allocation for Mobile Satellite Systems and Services," *IEEE Communications Letters*, Vol. 16, No. 1, January 2012, pp. 60–63.
- [69] Modiano, E., "Satellite Data Networks," *AIAA Journal of Aerospace Computing, Information, and Communication*, Vol. 1, October 2004, pp. 395–398.
- [70] Mosher, T., Barrera, M., Bearden, D., and Lao, N., "Integration of Small Satellite Cost and Design Models for Improved Conceptual Design-to-Cost," *IEEE Aerospace Conference*, Vol. 3, New York, NY, USA, 1998, pp. 97 – 103.
- [71] Lemaitre, M., Verfaillie, G., Jouhaud, F., Lachiver, J.-M., and Bataille, N., "Selecting and scheduling observations of agile satellites," *Aerospace Science and Technology*, Vol. 6, No. 5, Sept. 2002, pp. 367 – 81.
- [72] Martin, W., "Satellite image collection optimization," *Optical Engineering*, Vol. 41, No. 9, 2002, pp. 2083 – 2087.
- [73] Defence, H. P., Harrison, S. A., Price, M. E., and Philpott, M. S., "Task Scheduling for Satellite Based Imagery," *UK Planning and Scheduling Workshop, Special Interest Group, University of Salford, UK*, 1999, pp. 64–78.
- [74] Wolfe, W. and Sorensen, S., "Three Scheduling Algorithms Applied to the Earth Observing Systems Domain," *Management Science*, Vol. 46, No. 1, 2000, pp. 148–166.
- [75] Potter, W. and Gasch, J., "A Photo Album of Earth: Scheduling Landsat 7 Mission Daily Activities," *5th International Conference on Space Operations*, Tokyo, Japan, 1998.
- [76] Burrowbridge, S., "Optimal Allocation of Satellite Network Resources," *Master of Science In Mathematics, Virginia Polytechnic Institute and State University*, 1999.
- [77] Rao, J., Soma, P., and Padmashree, G.
- [78] Pemberton, J. and Galiber, F., "A Constraint-Based Approach to Satellite Scheduling," *DIMACS Workshop on Constraint Programming and Large Scale Discrete Optimization*, Providence, RI, 2001, pp. 101 – 14.
- [79] Barbulescu, L., Watson, J.-P., Whitley, L., and Howe, A., "Scheduling space-ground communications for the Air Force satellite control network," *Journal of Scheduling*, Vol. 7, 2004, pp. 7–34.

- [80] Sun, B., Mao, L., Wang, W., Xie, X., and Qin, Q., “Satellite mission scheduling algorithm based on Genetic Algorithm,” *Second International Conference on Space Information Technology (SICE)*, Vol. 6795, 2007, p. 67950U.
- [81] Frank, J., Johnsson, A., Morris, R., and Smith, D. E., “Planning and scheduling for fleets of earth observing satellites,” *Sixth International Symposium on Artificial Intelligence, Robotics, Automation and Space*, 2001.
- [82] Globus, A., Crawford, J., Lohn, J., and Pryor, A., “A comparison of techniques for scheduling earth observing satellites,” *Innovative Applications of Artificial Intelligence Conference, IAAI’04*, AAAI Press, 2004, pp. 836–843.
- [83] Baek, S.-W., Cho, K.-R., Lee, D.-W., Bainum, P. M., and Kim, H.-D., “Development of scheduling algorithm and GUI for the autonomous satellite mission operation,” *60th International Astronautical Congress (IAC)*, Vol. 7, Republic of Daejeon, Korea, 2009, pp. 5527–5534.
- [84] Vasquez, M. and Hao, J., “A Logic-Constrained Knapsack Formulation and a Tabu Algorithm for the Daily Photograph Scheduling of an Earth Observation Satellite,” *Computational Optimization and Applications*, Vol. 20, 2001, pp. 137–157.
- [85] Bensana, E., Lemaitre, M., and Verfaillie, G., “Earth Observation Satellite Management,” *Journal of Constraints*, Vol. 4, pp. 293–299.
- [86] Jian, L. and Cheng, W., “Resource planning and scheduling of payload for satellite with particle swarm optimization,” Vol. 6795, USA, 2007, pp. 67951 – 1.
- [87] Verfaillie, G., Pralet, C., and Lemaitre, M., “How to model planning and scheduling problems using constraint networks on timelines,” *Knowledge Engineering Review*, Vol. 25, No. 3, 2010, pp. 319 – 336.
- [88] Hiroyasu Muraoka, Ronald H. Cohen, T. O. and Doi, N., “Aster Observation Scheduling Algorithm,” *International Symposium Space Mission Operations and Ground Data Systems*, 1998.
- [89] mi Han, S., woo Beak, S., rae Cho, K., woo Lee, D., and dong Kim, H., “Satellite mission scheduling using genetic algorithm,” *SICE Annual Conference*, 2008, pp. 1226–1230.
- [90] Rabideau, G., Knight, R., Chien, S., Fukunaga, A., and Govindjee, A., “Iterative repair planning for spacecraft operations using the ASPEN system,” *Fifth International Symposium on Artificial Intelligence, Robotics and Automation in Space (ESA SP-440)*, Noordwijk, Netherlands, 1999, pp. 99 – 106.
- [91] Marinelli, F., Nocella, S., Rossi, F., and Smriglio, S., “A Lagrangian Heuristic for Satellite Range Scheduling with Resource Constraints,” *Dipartimento di Informatica, Universitita degli Studi di L’Aquila, Technical Report TRCS 004*, 2005.

- [92] Dutta, A. and Rama, D. V., “An optimization model of communications satellite planning,” *IEEE Transactions on Communications*, Vol. 40, No. 9, 1992, pp. 1463 – 1473.
- [93] Bianchessi, N., Cordeau, J.-F., Desrosiers, J., Laporte, G., and Raymond, V., “A heuristic for the multi-satellite, multi-orbit and multi-user management of Earth observation satellites,” *European Journal of Operational Research*, Vol. 177, No. 2, 2007, pp. 750 – 762.
- [94] Harrison, I. and Schwuttke, U., “Optimizing satellite, network, and ground station operations with next generation data visualization,” *IEEE Aerospace Conference*, March 2000.
- [95] Lin, W.-C., Liao, D.-Y., Liu, C.-Y., and Lee, Y.-Y., “Daily imaging scheduling of an Earth observation satellite,” *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, 2005, pp. 213–223.
- [96] A. Fukunaga, G. Radideau, S. C. and Yan, D., “ASPEN: A Framework for Automated Planning and Scheduling of Spacecraft Control and Operations,” *International Symposium on AI, Robotics and Automation in Space (i-SAIRAS)*, Tokyo, Japan, 1997.
- [97] Dungan, J., Frank, J., Jonsson, A., Morris, R., and Smith, D. E., “Advances in Planning and Scheduling of Remote Sensing Instruments for Fleets of Earth Orbiting Satellites,” *Earth Science Technology Conference*, Pasadena, CA, June 2002.
- [98] Arkali, G., Dawande, M., and Sriskandarajah, C., “Scheduling Support Times for Satellites with Overlapping Visibilities,” *Production and Operations Management*, Vol. 17, 2009, pp. 224–234.
- [99] Cheung, K.-M., Lee, C., Gearhart, W., Vo, T., and Sindi, S., “Link-capability driven network planning and operation,” *IEEE Aerospace Conference*, Vol. 7, 2002, pp. 7–3281 – 7–3285 vol.7.
- [100] Zheng, W., Meng, X., and Huan, H., “Genetic Algorithm for TDRS Communication Scheduling with Resource Constraints,” *International Conference on Computer Science and Software Engineering*, December 2008.
- [101] Gathmann, T. P. and Raslavicius, L., “Systems approach to the satellite operations problem,” *IEEE Aerospace and Electronic Systems Magazine*, Vol. 5, No. 12, 1990, pp. 20 – 24.
- [102] Dantzig, G. B., “Linear programming under uncertainty,” *Management Science*, Vol. 1, pp. pp. 197–206.
- [103] Beale, E. M. L., “On Minimizing A Convex Function Subject to Linear Inequalities,” *Journal of the Royal Statistical Society*, Vol. 17, No. 2, 1955, pp. pp. 173–184.

- [104] Kleywegt, A. and Shapiro, A., *Chapter 101: Stochastic Optimization*, School of Industrial and Systems Engineering, Georgia Institute of Technology, 2000.
- [105] Kutanoglu, E. and Wu, S. D., “Improving Schedule Robustness via Stochastic Analysis and Dynamic Adaptation,” Manufacturing Logistics Institute, Department of Industrial and Manufacturing Systems Engineering, Lehigh University.
- [106] Birge, J. R., “Decomposition and Partitioning Methods for Multistage Stochastic Linear Programs,” *Operations Research*, Vol. 33, No. 5, 1985, pp. 989–1007.
- [107] Birge, J. and Glazebrook, K., “Bounds on Optimal Values in Stochastic Scheduling,” 1996.
- [108] Drummond, M., Bresina, J., and Swanson, K., “Just-In-Case Scheduling,” *AAAI-94*, 1994.
- [109] Pinedo, M., “Stochastic Scheduling with Release Dates and Due Dates,” *Operations Research*, Vol. 31, No. 3, May/June 1983, pp. 559–572.
- [110] Johnston, M. D., “Deep Space Network Scheduling Using Multi-Objective Optimization With Uncertainty,” *SpaceOps Conference*, 2008.
- [111] Liao, D.-Y. and Yang, Y.-T., “Satellite imaging order scheduling with stochastic weather condition forecast,” *Systems, Man and Cybernetics, 2005 IEEE International Conference on*, Vol. 3, oct. 2005, pp. 2524 – 2529 Vol. 3.
- [112] Swamy, C. and Shmoys, D. B., “Approximation algorithms for 2-stage stochastic optimization problems,” *SIGACT News*, Vol. 37, 2006, pp. 33–46.
- [113] Bertsimas, D., Doan, X. V., Natarajan, K., and Teo, C.-P., “Models for Minimax Stochastic Linear Optimization Problems with Risk Aversion,” *MATHEMATICS OF OPERATIONS RESEARCH*, 2010, pp. 580–602.
- [114] Pemberton, J. C. and Greenwald, L., “On the Need for Dynamic Scheduling of Imaging Satellites,” *AAAI Conference*, 2004, pp. 723–728.
- [115] Lapp, M., “Methods for Improving Robustness and Recovery in Aviation Planning,” Thesis, Industrial and Operations Engineering, University of Michigan.
- [116] Bean, J. and Birge, J., “Match-Up Real-Time Scheduling,” 1996.
- [117] Dean, B. C., Goemans, M. X., and Vondrk, J., “Approximating the Stochastic Knapsack Problem: The Benefit of Adaptivity,” *Mathematics of Operations Research*, Vol. 33, No. 4, 2008, pp. 945–964.
- [118] Mora, J. N., “Stochastic Scheduling,” *Encyclopedia of Optimization*, Vol. V, 2005, pp. 367–372.
- [119] Birge, J. and Holmes, D., “Efficient Solution of Two Stage Stochastic Linear Programs using Interior Point Methods,” 1992.

- [120] Pemberton, J., “Toward Scheduling Over-Constrained Remote-Sensing Satellites,” *Second NASA International Workshop on Planning and Scheduling for Space*, San Francisco, CA, 2000.
- [121] A. Segun Adeyefa, M. K. L., “Multiobjective Stochastic Linear Programming: An Overview,” *American Journal of Operations Research*, Vol. 1, 2011, pp. 203–213.
- [122] Daintith, J. and Wright, E., *Oxford Dictionaries*, Oxford University Press, 2010.
- [123] Smith, B., Sherwood, R., Govindjee, A., Yan, D., Rabideau, G., Chien, S., and Fukunaga, A., “Representing Spacecraft Mission Planning Knowledge in ASPEN,” *Artificial Intelligence Planning Systems Workshop on Knowledge Acquisition*, 1998.
- [124] Chien, S., Knight, R., and Rabideau, G., “CASPER: using local search for planning for embedded systems,” *ESTEC Meeting on Onboard Autonomy*, Noordwijk, The Netherlands, October 2001.
- [125] Johnston, M. D. and Giuliano, M., “MUSE: Multi-User Scheduling Environment for Multi-Objective Scheduling of Space Science Missions,” *IJCAI Workshop on Space Applications of AI*, 2009.
- [126] Mark D. Johnston, Daniel Tran, B. A. and Page, C., “Request-Driven Scheduling for NASAs Deep Space Network,” *International Workshop on Planning and Scheduling for Space (IWSS)*, 2009.
- [127] Aghevli, A., Bachmann, A., Bresina, J., Greene, K., Kanefsky, B., Kurien, J., McCurdy, M., Morris, P., Pyrzak, G., Ratterman, C., Vera, A., and Wragg, S., “Planning Applications for Three Mars Missions with Ensemble,” *Proceedings of The IEEE (PIEEE)*, 2009.
- [128] Norris, J. S., Powell, M. W., Vona, M. A., Backes, P. G., and Wick, J. V., “Mars Exploration Rover Operations with the Science Activity Planner,” *IEEE International Conference on Robotics and Automation*, 2005, pp. 4629–4634.
- [129] Ai-Chang, M., Bresina, J. L., Charest, L., Chase, A., jung Hsu, J. C., Jónsson, A. K., Kanefsky, B., Morris, P. H., Rajan, K., Yglesias, J., Chafin, B. G., Dias, W. C., and Maldague, P. F., “MAPGEN: Mixed-Initiative Planning and Scheduling for the Mars Exploration Rover Mission,” *IEEE Intelligent Systems*, Vol. 19, No. 1, 2004, pp. 8–12.
- [130] Dvorak, D., Rasmussen, R., Reeves, G., and Sacks, A., “Software Architecture Themes in JPLs Mission Data System,” *IEEE Aerospace Conference*, Big Sky, MT, March 2000.
- [131] Mettala, E. and Graham, M. H., *The Domain Specific Software Architecture Program*, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, June 1992.

- [132] Tracz, W., Coglianese, L., and Young, P., “A Domain-Specific Software Architecture Engineering Process Outline,” *ACM SIGSOFT Software Engineering Notes*, Vol. 20, 1993, pp. 49–62.
- [133] Green, W. B., “Multimission ground data system support of NASA’S planetary program,” *Acta Astronautica*, Vol. 37, No. 0, 1995, pp. 407 – 415.
- [134] Duane L. Bindschadler, C. L. D. and McCullar, M., “Principles to Products: Toward Realizing MOS 2.0,” *Proceedings of Space Ops*, Stockholm, Sweden, June 2012.
- [135] Backes, P. G., Norris, J. S., Powell, M. W., Vona, M. A., Steinke, R., and Wick, J., “The Science Activity Planner for the WITS Database 2 Mars Exploration Rover Mission: FIDO Field Test Results,” *IEEE Aerospace Conference*, Big Sky, MT, March 2003.
- [136] Spangelo, S., Cutler, J., Gilson, K., and Cohn, A., “Optimization-Based Scheduling for the Single-Satellite, Multi-Ground Station Communication Problem,” *Operations Research (In Progress)*, 2012.
- [137] Spangelo, S. and Cutler, J., “Integrated Approach to Optimizing Small Spacecraft Vehicles and Operations,” *International Astronautical Congress*, Cape Town, South Africa, October 2011.
- [138] Spangelo, S. and Cutler, J., “Optimization of Single-Satellite Operational Schedules Towards Enhanced Communication Capacity,” *AIAA Guidance, Navigation, and Control Conference*, Minneapolis, MN, August 2012.
- [139] Smith, M., “Mars Trace Gas Mission Science Rationale & Concept,” Presentation to the NRC Decadal Survey Mars Panel, September 2009.
- [140] Reinholtz, W. K., “Timeline as Unifying Concept for Spacecraft Operations,” *Space Ops Conference*, Stockholm, Sweden, June 2012.
- [141] Chen, C.-T., *Linear System Theory and Design*, Oxford University Press, Inc., 3rd ed., 1999.
- [142] Jaynes, E. T., “Information Theory and Statistical Mechanics,” *Phys. Rev.*, Vol. 106, May 1957, pp. 620–630.
- [143] Shannon, C., “A Mathematical Theory of Communication,” *The Bell System Technical Journal*, Vol. 27, July 1948, pp. 379–423, 623–656.
- [144] Maine, K., Devieux, C., and Swan, P., “Overview of IRIDIUM satellite network,” *Wescon Conference Record Proceedings*, San Francisco, CA, USA, 1995, pp. 483 – 490.
- [145] Zwart, P., “‘Moon bounce’, another way of VHF communication,” *Philips Test and Measuring Notes*, , No. 2, 1972, pp. 13.

- [146] Zhi-yi, Z., Zan, L., Chang-xing, P., and Jue-ping, C., “Extremely low SNR meteor communication technology,” *Telecommunication Engineering*, Vol. 47, No. 2, 2007, pp. 44 – 47.
- [147] Cover, T. and Thomas, J., *Elements of information theory*, Sons, Inc., 2nd ed., 2006.
- [148] Klesh, A. and Cutler, J., “Exploiting the Link: Improving Satellite Communication through Higher Elevation Links,” *AIAA/AAS Astrodynamics Specialist Conference*, Toronto, Ontario, Canada, 2010.
- [149] Cakaj, S., Keim, W., and Malaric, K., “Communications Duration with Low Earth Orbiting Satellites,” *IASTED International Conference on Antennas, Radar, and Wave Propagation Proceedings*, Anaheim, CA, USA, 2007, pp. 85 – 88.
- [150] Spangelo, S., Cutler, J., Klesh, A., and Boone, D., “Models and Tools to Evaluate Space Communication Network Capacity,” *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 48, No. 3, July 2012, pp. 2387–2404.
- [151] Cutler, J. and Mann, J., “CubeSat Ground Station Survey,” http://gs.engin.umich.edu/g_s_survey, 2009-2011.
- [152] Spangelo, S. and Cutler, J., “Small Satellite Survey,” http://gs.engin.umich.edu/sat_survey, 2011-2012.
- [153] Cutler, J. W., Springmann, J. C., Spangelo, S., and Bahcivan, H., “Initial Flight Assessment of the Radio Aurora Explorer,” *25th Annual Small Satellite Conference*, Logan, UT, August 2011.
- [154] Bahcivan, H., Kelley, M., and Cutler, J., “Radar and rocket comparison of UHF radar scattering from auroral electrojet irregularities: implications for a nano-satellite radar,” *Journal of Geophysical Research*, Vol. 114, June 2009.
- [155] Johnson, M., “CubeSat-on-Demand,” CubeSat Developers Summer Workshop, April 2011.
- [156] Mann, J. and Cutler, J., “Global Ground Station Survey,” Summer Cubesat Workshop, August 2008.
- [157] Analytical Graphics, I. A., “Systems Tool Kit (STK),” <http://www.stk.com/>.
- [158] “Space Surveillance Data,” www.space-track.org, 2009.
- [159] Spangelo, S., Kaslow, D., Delp, C., Anderson, L., Cole, B., Foyse, E., Cheng, L., Yntema, R., Bajaj, M., Soremekum, G., and Cutler, J., “Model Based Systems Engineering (MBSE) Applied to Radio Aurora Explorer (RAX) CubeSat Mission Operational Scenarios,” *IEEE Aerospace Conference*, Big Sky, MT, March 2013.
- [160] Wertz, J., editor, *Orbit and Constellation Design and Management*, Microcosm Press, Hawthorne, CA, 3rd ed., 2001.

- [161] Ali, I., Al-Dhahir, N., and Hershey, J., “Predicting the visibility of LEO satellites,” *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 35, USA, 1999, pp. 1183 – 1190.
- [162] Scheeres, D., *AOSS 605: Astrodynamics Notes*, Department of Aerospace Engineering, University of Michigan, 2001.
- [163] Chin, A., Coelho, R., Nugent, R., Munakata, R., and Puig-Suari, J., “CubeSat: The Picosatellite Standard for Research and Education,” *AIAA SPACE Conference and Exposition*, September 2008.
- [164] Hodges, L. and Woll, R., “Air Force Satellite Control Network (AFSCN) Support for Operational Responsive Space (ORS),” *6th Responsive Space Conference*, April 2008.
- [165] Spence, H. and Moore, T., “A Retrospective Look Forward on Constellation-Class Geospace Missions,” Fall AGU Meeting, December 2009.
- [166] Jorgensen, T., “The NSF CubeSat Program: The Promise of Scientific Projects,” Fall AGU Meeting, December 2009.
- [167] Earle, G. and Davidson, R., “Challenges and Promise: CubeSat-Based Instrumentation for Thermal Plasma and Neutral Measurements,” Fall AGU Meeting, December 2009.
- [168] Bournes, P. and Williamson, D., “CubeSat Experiments (QBX),” CubeSat Workshop, April 2009.
- [169] Cutler, J. and Kitts, C., “Mercury: A Satellite Ground Station Control System,” *IEEE Aerospace Conference*, March 1999.
- [170] Biraud, M. G., Page, H., and Kurahara, N., “Global Educational Network for Satellite Operations (GENSO),” *60th International Astronautical Congress*, Vol. 5, Daejeon, Republic of Korea, 2009, pp. 3577 – 3582.
- [171] Spangelo, S. and Cutler, J., “Small Satellite Operations Model to Assess Data and Energy Flows,” *AIAA/AAS Astrodynamics Specialist Conference*, 2010.
- [172] IBM, “IBM ILOG CPLEX Optimization Studio,” <http://www-01.ibm.com/software/integration/optimization/cplex-optimization-studio/>, March 2010.
- [173] Solar System Exploration Survey, N. R. C., “Visions and Voyages for Planetary Science in the Decade 2013 - 2022,” National Academies Press, 2011.
- [174] “Habitation-Assessing Interplanetary Lander (HAIL),” Aerospace 483, University of Michigan, March 2012.
- [175] Agency, E. S., “ExoMars Orbiter: Experiment Proposal Information Package,” EXM-OM-IPA-ESA-00001, 2010.

- [176] Juan L. Cano, F. C., “Exomars Mission Analysis and Design - Launch, Cruise and Arrival Analyses,” Available Online, 2012, Deimos Space S.L.
- [177] Edwards, C. D., Jr., T. C. J., Schwartzbaum, E., Devereaux, A. S., DePaula, R., and Mark Dapore, T. W. F., “The Electra Proximity Link Payload for Mars Relay Telecommunications and Navigation,” *54th International Astronautical Congress*, Bremen, Germany, 2003.
- [178] Gao, S., Clark, K., Unwin, M., Zackrisson, J., Shiroma, W., Akagi, J., Maynard, K., Garner, P., Boccia, L., Amendola, G., Massa, G., Underwood, C., Brenchley, M., Pointer, M., and Sweeting, M., “Antennas for Modern Small Satellites,” *IEEE Antennas and Propagation Magazine*, Vol. 51, No. 4, 2009.
- [179] Imbriale, W. A., “Evolution of the Deep Space Network 34-M Diameter Antennas,” *IEEE Aerospace Conference*, Snowmass, Colorado, March 1998.
- [180] Duncan, C., “Low Mass Radio Science Transponder Navigation Anywhere,” *iCube-Sat Workshop*, Boston, MA, May 2012.
- [181] (SSTL), S. S. T. L., “XTx400 X-Band Transmitter,” Surrey Satellite Technology Limited, Specification Sheet, Retrieved 2012.
- [182] (SSTL), S. S. T. L., “S-Band Patch Antenna,” Surrey Satellite Technology Limited, Specification Sheet, Retrieved 2012.
- [183] Krikorian, Y. Y., Sue, M. K., Leon, G. V., Cooper, L., Do, S. K., Emmons, D. L., Dichmann, D. J., McVey, J. P., and Campbell, E. T., “A Dynamic Deep Space Communication Link Analysis Tool for the Deep Space Network (DSN),” *IEEE Aerospace Conference*, Big Sky, MT, March 2005.
- [184] Springmann, J., Kempke, B., Cutler, J., and Bahcivan, H., “Initial Flight Results of the RAX-2 Satellite,” *Proceedings of the 26th Annual Small Satellite Conference*, Logan, UT, August 2012.
- [185] Gill, E., Sundaramoorthy, P., Bouwmeester, J., Zandbergen, B., and Reinhard, R., “Formation flying within a constellation of nano-satellites: The QB50 mission,” *Acta Astronautica*, Vol. 82, No. 1, 2013, pp. 110 – 117.
- [186] Campbell, M. E. and Schetter, T., “Formation flying mission for the UW Dawgstar satellite,” *IEEE Aerospace Conference*, Vol. 7, 2000.
- [187] Nelson, J. M., “Persistent Military Satellite Communications Coverage Using a Cubesat Constellation in Low Earth Orbit,” Honors Thesis in the Major Program in Electrical Engineering, College of Engineering and Computer Science, Burnett Honors College, University of Central Florida.
- [188] “QB50 von Karman Institute for Fluid Dynamics,” www.vki.ac.be/QB50/project2.php, 2009.

- [189] Saks, N., Boonstra, A. J., Rajan, R. T., Bentum, M. J., Beliën, F., and van 't Klooster, K., “DARIS, a fleet of passive formation flying small satellites for low frequency radio astronomy,” *Small Satellite Systems and Services - The 4S Symposium 2010, Funchal, Madeira, Portugal*, ESA en CNES, Portugal, June 2010, pp. 1–15.
- [190] “Deep Space Network Services Catalog,” Jet Propulsion Laboratory, California Institute of Technology, DSN No. 820-100, Rev. D, JPL D-19002, May 2009.
- [191] Krupiarz, C. J., Jennings, E. H., Pang, J. N., Schoolcraft, J. B., Segu, J. S., and Torgerson, J. L., “Spacecraft Data and Relay Management Using Delay Tolerant Networking, AIAA 2006-5754,” SpaceOps 2006 Conference, 2006.