MISSION SCHEDULING AND OPTIMIZATION ALGORTIHM FOR SMALL SATELLITE CONSTELLATIONS

BREANNON LEWIS

A THESIS SUBMITTED TO THE FACULTY OF GRADUATE STUDIES IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE

GRADUATE PROGRAM IN EARTH AND SPACE SCIENCE
YORK UNIVERSITY
TORONTO, ONTARIO

January 2021

Abstract

CubeSats, a class of small satellite, offer a unique opportunity for training, technology demonstrations, Earth observation, and other space-based research. There has been a recent increase in their design and implementation in private industry. Private industries and agencies have begun to research and implement larger amounts of small satellites working together, referred to as a constellation. CubeSat Constellation missions use multiple satellites to complete complex and challenging tasks instead of one larger satellite. One of the keys to mission success for CubeSat constellation missions is mission scheduling. When implementing a large network of satellites in constellation operation, scheduling becomes a challenge due to the large amount of conflicts that need to be resolved. Conflicts occur when more than one satellite and/or ground resource can be used to complete a task. The following thesis describes and demonstrates an advanced mission scheduling algorithm to schedule Earth observation, data transfer, or relief aid missions. Each type of mission is given a test case and results show the algorithm's weighted-sum flexibility to solve multiple mission objectives. The weighted-sums optimization algorithm is used to test if the effectiveness of the chosen design variables (transfer time, age of data, spatial coverage, and temporal coverage) and to test if the blended objective functions are effective. This thesis presents the preliminary results of the mission scheduling and selection of ideal weighting of different objectives for CubeSat constellation missions.

Acknowledgements

I would like to express my deep gratitude to my supervisor Dr. Regina Lee for all her guidance, patience, advice, and encouragement throughout my masters. This thesis would not have been possible without her. She truly goes above and beyond to ensure the success of her student and I am eternally grateful.

I also want to thank Kepler Communications for working with me on this thesis and providing industry opinions and recommendations. Thank you for taking the time to provide guidance during such a busy time for the company.

I would also like to thank Dr. Franz Newland for all guidance provided throughout my masters. Thank you for staying late and always providing recommendations with such enthusiasm.

I want to thank my friends in the Nanosatellite Research Laboratory for their encouragement and aid. Whether it was letting me voice my thoughts out loud, giving advice or providing a cheerful distraction from my work, your role was instrumental to my time at York University.

I would also like to give a special thanks to Sarah Burns for the late-night trips to distract me and help me reset my mind to start the next day with a clear mind. I would not have been able to complete this thesis without her.

Finally, I want to express my gratitude to my family for their love and support during this process. I would not be where I am today without their encouragement and guidance.

Table of Contents

ABSTRACT .... **II**

ACKNOWLEDGEMENTS .... **III**

TABLE OF CONTENTS .... **IV**

LIST OF TABLES .... **VI**

LIST OF FIGURES .... **VII**

LIST OF ACRONYMS .... **IX**

CHAPTER 1 : INTRODUCTION **1**

1.1 TRENDS IN SMALLSAT TECHNOLOGY 1

1.2 MOTIVATION 7

1.3 RESEARCH OBJECTIVES 9

1.4 THESIS OUTLINE 9

CHAPTER 2 BACKGROUND **11**

2.1 EARLY MISSION SCHEDULING FOR AUTONOMY 11

2.2 EARTH OBSERVATION AND DISASTER AID MISSION PLANNING 13

2.3 COMMUNICATION SATELLITE MISSION PLANNING 15

2.4 CURRENT MISSION PLANNING TOOLS 16

2.5 MULTI-OBJECTIVE OPTIMIZATION FOR SATELLITE CONSTELLATIONS 17

CHAPTER 3 : MISSION SCHEDULING AND OPTIMIZATION ALGORITHM **23**

3.1 INPUT 24

3.2 ORBIT PROPAGATION 25

3.3 SCHEDULE MANAGER 25

3.4 OPTIMIZATION AND CONSTRAINTS 27

CHAPTER 4 ALGORITHM VALIDATION **40**

4.1 TEST CASE # 1 41

4.2 TEST CASE # 2 44

4.3 TEST CASE # 3 46

4.4 TEST CASE # 4 49

4.5 TEST CASE # 5 49

CHAPTER 5 RESULTS 50

5.1 TEST CASE 1 50

5.1.1 Transfer Time 52

5.1.2 Age of Data 55

5.1.3 Spatial Coverage 57

5.1.4 Temporal Coverage 58

5.2 TEST CASE 2 65

5.3 TEST CASE 3 67

5.4 TEST CASE 4 72

5.5 TEST CASE 5 72

CHAPTER 6 RESULTS SUMMARY 74

CHAPTER 7 CONCLUSION 77

CHAPTER 8 FUTURE WORK 80

CHAPTER 9 APPENDIX 82

9.1 A 82

9.2 B 82

CHAPTER 10 REFERENCES 85

List of Tables

Table 1 GA comparisons 19

Table 2 Input Parameters 24

Table 3 Test Case 1 Ground Resource Locations... 41

Table 4 Test Case 1 Input variables and constraints 43

Table 5 Test Case 1 Scenario Weighting 43

Table 6 Test Case 3 Ground resource locations 44

Table 7 Scenario Weights 51

Table 8 Additional Scenario weights... 51

Table 9 Scenario Results 52

Table 10 Test Case 2 Results... 66

Table 11 Test Case 4 Results... 67

Table 12 Test Case 4 Results... 72

Table 13 Test Case 5 Results... 73

List of Figures

Figure 1: Global Launches of Small Satellites by Application from NSR [32] 2

Figure 2 :Number of small satellites from the last 15 years. Reprinted from A Comprehensive Review of Small Satellites Communications and Networks. Wireless Communications and Mobile Computing. Burleigh et al. (2019). 4

Figure 3 Market forecast, 8th edition, approved for public release, SpaceWorks Enterprises, Inc. (SEI), Reprinted from A Comprehensive Review of Small Satellites Communications and Networks. Wireless Communications and Mobile Computing. Burleigh et al. (2019). 5

Figure 4 Decision Tree 21

Figure 5 Data Drain Rate vs Time for a test case. 29

Figure 6 Data Storage vs Time for a test case. 30

Figure 7 Power Consumption vs Time for a test case. 31

Figure 8 Battery Storage vs Time for a test case. 31

Figure 9 Flow chart for algorithm 38

Figure 10 Flow chart for cost function... 39

Figure 11 Test Case 1 Orbits 42

Figure 12 2D map of ground resources.. 42

Figure 13 3D representation of space and ground resources for Test Case 2 45

Figure 14 2D representation of ground resources for Test Case 2 45

Figure 15 3D representation of Test Case 3 and 4 satellite constellation and ground resources . 48

Figure 16 2D representation of ground resources for Test Case3 and 4 48

Figure 17 Transfer Time for all scenarios across all targets 53

Figure 18 Cost graph for Scenario 1. Cost 1 is the cost of the approved schedule and Cost 2 cost of the next considered schedule. It can be seen that if the Cost 2 value is lower than the current Cost 1 value, the approved schedule doesn't change. 54

Figure 19 Age of Data for all scenarios across all Targets 55

Figure 20 Cost graph for Scenario 2. Cost 1 is the value of the approved schedule and Cost 2 is the value for next considered... 56

Figure 21 Spatial Coverage for all scenarios across all Targets 57

Figure 22 Cost graph for Scenario 3. Cost 1 is the value for the approved schedule. Cost 2 is the value for the next schedule considered. 58

Figure 23 Temporal Coverage for all Scenarios.. 59

Figure 24 Cost Comparison for Scenario 4 Cost values.. 60

Figure 25 Cost Comparison for Scenario 5 61

Figure 26 Variation of design variables as a function of each other 63

Figure 27 Variation of design variables along the x-axis 64

Figure 28 Variation of design variables along the z-axis.. 65

Figure 29 Transfer Time for Test Case 2 66

Figure 30 Transfer Time for Test Case 3 68

Figure 31 Age of Data for Test Case 3 69

Figure 32 Spatial Coverage for Test Case 3 70

Figure 33 Temporal Coverage for Test Case 3 71

List of Acronyms

Acronym	Description
CNES	National Center of Space Studies
COTS	Commercial Off the Shelf
CPAW	Collection Planning and Analysis WorkStation
CSA	Canadian Space Agency
DMOEA	Dynamic Multi-objective Evolutionary Algorithm
DTN	Delay Tolerant Network
DS1	Deep Space 1
EO	Earth Observation
ESA	European Space Agency
GA	General Algorithm
GENSO	Global Educational Network for Satellite Operations
ISEB	International Education Board
JAXA	Japan Aerospace Exploration Agency
LEO	Low Earth Orbit
MIP	Mixed Integer Programming
MILP	Mixed Integer Programming
MOGA	Multi-objective Genetic Algorithms
MOO	Multi-Objective Optimization
M2M	Machine to Machine
NPGA	Niche Pareto Genetic Algorithm

NASA	National Aeronautics and Space Administration
NSGA	Nondominated sorting in Generic Algorithms
PAES	Pareto Archived Evolution Strategy
PESA	Pareto Envelope-based Selection Algorithm
POMDP	Partially Observable Markov Decision Process
RWGA	Random Weighted Genetic Algorithm
RDGA	Rank Density based Genetic Algorithms
SPEA	Strength Pareto Evolutionary Algorithms
SPEA-2	Strength Pareto Evolutionary Algorithms
SGP4	Standard General Perturbations Satellite Orbit model 4
STK	Systems Tool Kit
TLE	Two-Line Element Set
VEGA	Vector Evaluated Genetic Algorithms
WBGA	Weight Based Genetic Algorithms

Nomenclature

Age of Data - number of minutes between the end of the uplink and the beginning of the downlink.

Transfer Time - total transfer time in minutes across all satellites that data is uplinked from target to satellite and downlinked from satellite to ground.

Spatial Coverage - This is the variable that designs the schedule so that all targets are visited a uniform number of times.

Temporal Coverage - – this is the variable that designs the schedule so that a target is visited uniformly across the duration of the schedule.

Visits – When a satellite pass over the target and either transfers data or completes an EO task.

Uplink – Data transfer from target to satellite.

Downlink – Data transfer from satellite to ground.

Ground – Ground station that receives data from satellites.

Target – Location where data is uplinked from or location where EO task is completed.

Time slot – The 24hr schedule period is broken up into four 6-hour slots from 00:00 – 06:00, 06:00 – 12:00, 12:00-18:00, 18:00 – 23:59.

Request – Include the target ID, Satellite ID and start time. A request is a possible communication between target and satellite.

Chapter 1: Introduction

1.1 Trends in SmallSat Technology

In the last twenty years there have been vast improvements in microelectronics and microsystems technology. These improvements sparked a new interest in miniaturization of satellite technology while increasing the functionality of the spacecraft in general. This has then made commercial off the shelf (COTS) technology more abundant, useful, and affordable than ever before. These COTS solutions are the building blocks of small satellites (also often referred to as SmallSat). Therefore, the improvements in COTS have directly supported significant advancements in SmallSat Technology and mission objectives [4]. Gone are the days where government solely had management over satellites and the compatible ground resources, now the space industry is also driven by commercial enterprises and consumer demand [28]. SmallSats often follow a "fly-learn-re-fly" approach due to the shorter design cycles, lower costs, and smaller design teams necessary for SmallSat development and launch. The CubeSat (a class of SmallSat with mass in the range of 1 to 10 kilograms) standard was developed by California Polytechnic State University and Stanford University. Their goal was to simplify launch and deployment operations by creating a customizable satellite that had a standard shape and mass. A CubeSat consists of one or more units. Each unit is 10 cm X 10 cmX10 cm and has a mass of no larger than 1.33 kg per unit [4]

With the increasing popularity of CubeSat-class small satellites in recent years, the number of small satellites launched increases year-over-year as seen in the figure below (see Figure 1).

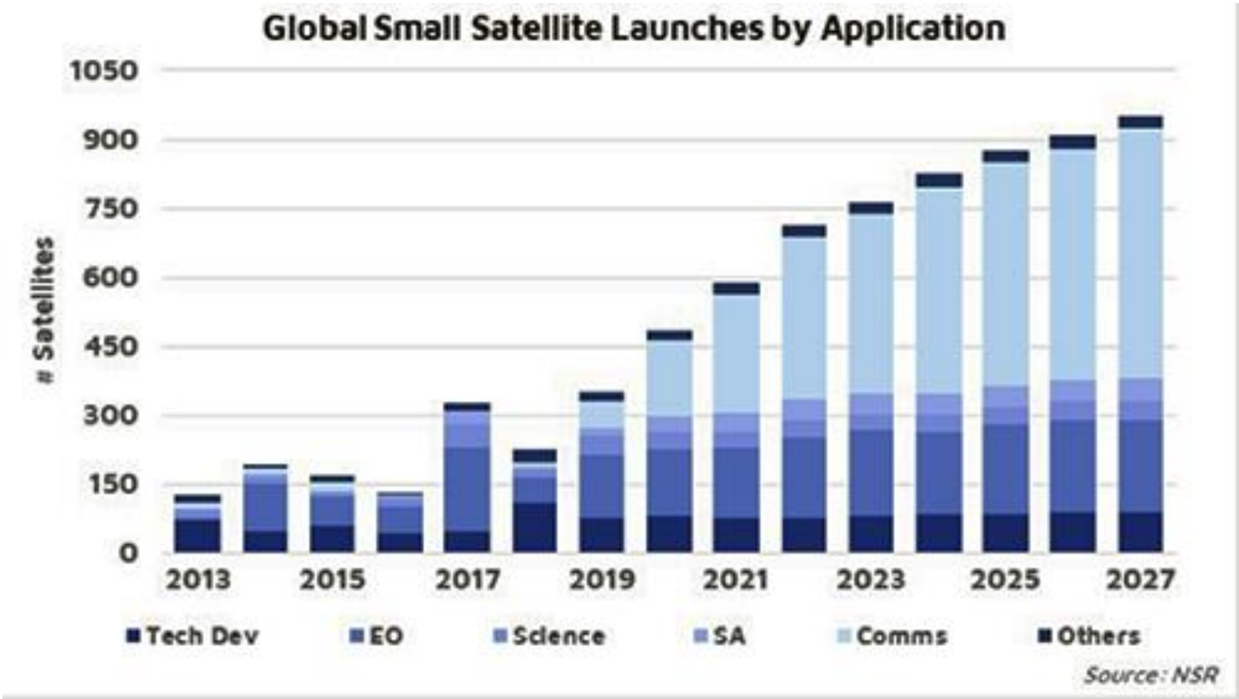

Figure 1: Global Launches of Small Satellites by Application from NSR [32]

Introduction of CubeSat standards has also substantially reduced costs of launch and operation so that it has gained popularity with universities, students, and commercial companies [28, 20]. These attributes, added with benefit of more flexible launch accommodations, have made CubeSats the most popular choice for the recently designed large constellation missions that have been proposed and developed.

The initial purpose of CubeSat missions was to test new technology or application demonstrations for validation purposes, since the satellite community prefers to use technology that has space heritage to lessen the probability of mission failure, either complete or partial [4]. The driving force behind the use and development of small satellites, after the success of these initial technology demonstrations missions, has been Earth observation and remote sensing. The data generated by the small satellite missions has been used in the agricultural sector, disaster management, forestry, and wildlife industries to name a few [4].

Today, CubeSats are becoming an increasingly popular scientific and technological endeavor for both academia and industry. Their light weight and low cost allow for technology and scientific principles to be demonstrated quickly and efficiently. It is estimated in [20] that the joint cost of production and launch of a CubeSat ranges from $100,000 to $200,000 USD. As the technologies aboard SmallSats become more complex and advanced, their scientific missions have become more involved. In a recent article in Nature [21] the author describes this phenomenon as "Mini satellites prove their scientific power – proliferation of 'CubeSats' offers fresh and fast ways to gather space data." The advancement of CubeSats requires the technological innovation of compatible payloads, as well as innovative and cost-effective ways to operate them so as to utilize them commercially. Figure 2 below shows the history of CubeSat launches and launch from 2010 to 2020.

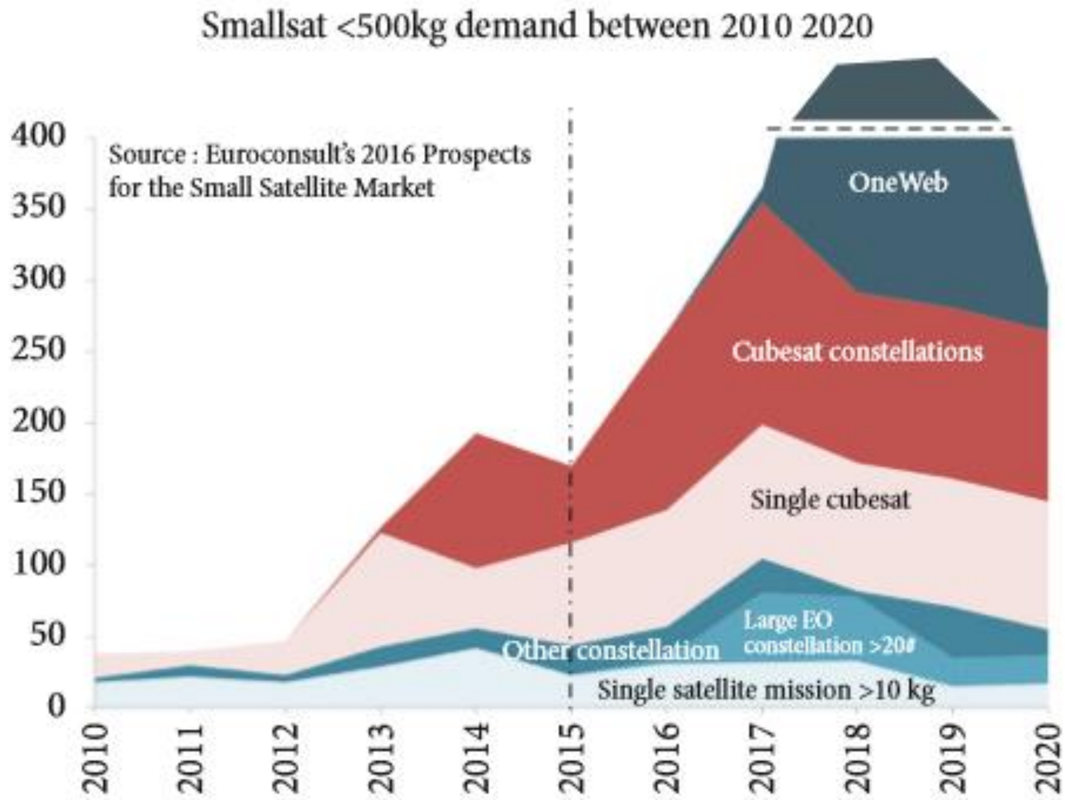

Figure 2 :Number of small satellites from the last 15 years. Reprinted from A Comprehensive Review of Small Satellites Communications and Networks. Wireless Communications and Mobile Computing. Burleigh et al. (2019).

Improvements in SmallSat technology have introduced new functionalities to the industry such as using CubeSat constellations missions where simultaneous multi-point mission objectives can be achieved [28]. Constellation missions are used to utilize multiple satellites in a coordinated operation [4]. Figure 3 summarizes the forecasted trends by application.

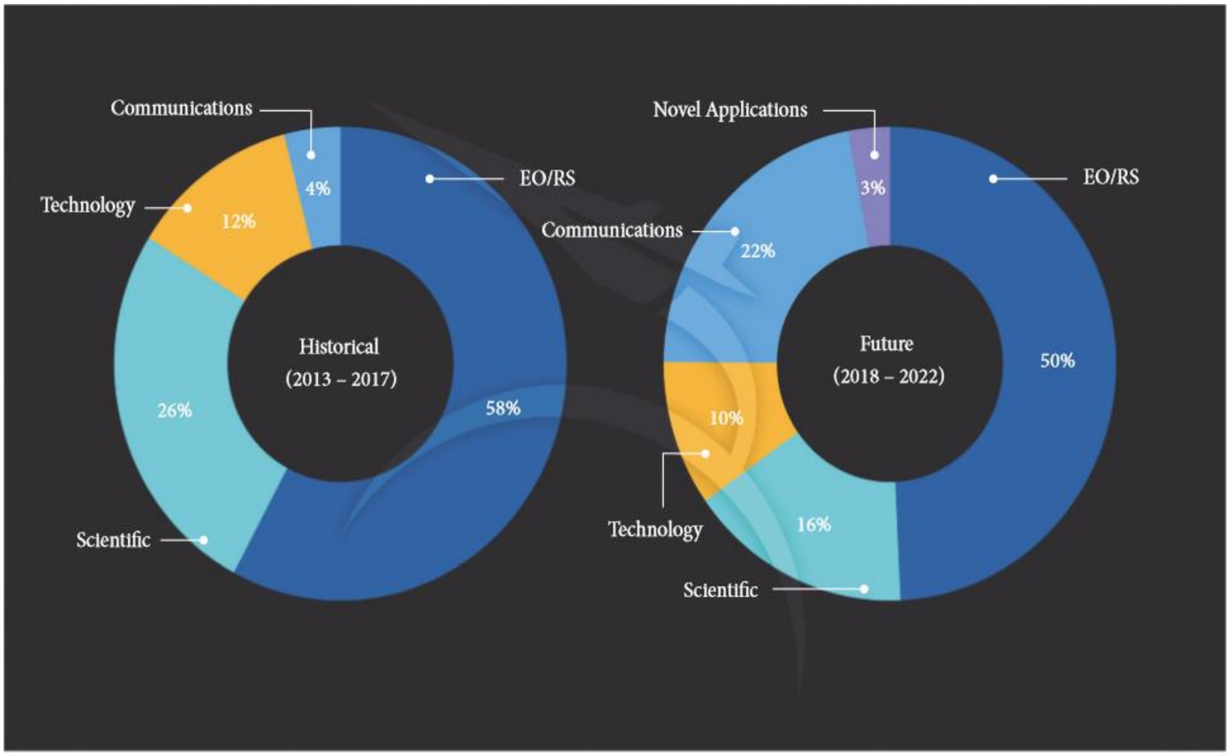

Figure 3 Market forecast, 8th edition, approved for public release, SpaceWorks Enterprises, Inc. (SEI), Reprinted from A Comprehensive Review of Small Satellites Communications and Networks. Wireless Communications and Mobile Computing. Burleigh et al. (2019).

Figure 3 shows that Earth observation and remote sensing will remain as the main objective for SmallSat missions, but communication missions will increase from 4% to 22% over 4 years.

There is a recent trend of increasing number of CubeSat/SmallSat missions with multiple satellites in constellation. In particular, there has been an increase in the number of CubeSat constellation missions with several commercial telecommunication missions such as broadcasting applications, internet of things and Machine-to-Machine paradigm, as well as more advanced EO (Earth Observation) such as Planet Labs' Dove satellite constellation [4][20]. The OneWeb constellation has a planned 720 satellites in its constellation; Boeing has 2,956, and Space X has 4,500 satellites and growing.

As expected, large satellite constellations have more complex and challenging constraints and complications compared to traditional single satellite missions. These constraints and complications are dependent on the mission's objectives, many of which revolve around the scheduling of payload operations and data transmission. In an EO satellite constellation, there can be constraints on when the image is taken, the coverage of the images taken, as well as the storage and battery constraints of the satellite. These constraints are decided by the operator and satellite design. When using such large numbers of satellites, complications arise in assigning ground communication times and resources. Often more than one satellite can communicate with the same ground station and often the ground stations can handle only one communication link at a time. The operator must decide which satellite is assigned the communication link. Deciding factors could be based on the satellite's storage capacity, other available ground communication options, cost of operation of both the ground station and the satellite and the communication time limit of each satellite's pass over the ground station.

In data transfer constellations, there are constraints on the battery capacity and storage capacity, as in the EO constellation, as well as the added constraints of scheduling data transfers within a time constraint to ensure that users receive data transfers within a specific duration and maximize the amount of data transferred. There are the same complications with ground communication times as well as the added complication associated with multiple users. Scheduling and planning for satellite constellation missions has been studied before; however, most if not all endeavor is to simply maximize communication times with ground stations via more satellites in the same orbits. There are very few studies that deal with the conflicts that arise when using large constellations[3][8]. Most of the literature deals with Earth Observation satellites only[8]. The objective is to schedule the mission's payload (such as cameras) while also satisfying the satellite's cost constraints. According to the literature review in [6], "*most studies neglect data and energy dynamics*" and while scheduling data transfers for larger constellations with multi-ground resources has been studied before the authors are unaware of any studies that add energy and data dynamics to the problem.

While the missions mentioned above assume the use of newly designed satellites that are launched in sequence to create the constellation matrix, there is also the possibility of repurposing and re-tasking resources already in orbit. The International Charter "Space and Major Disasters" is an international effort to use current space technology at the service of emergency and rescue responders during a major disaster. When the Charter becomes active, the member satellites are re-tasked to capture an image of the devasted regions. The data must be accurate and rapidly available. Canada's RADARSAT is part of the Charter [5] to support the global disaster monitoring initiative. The Global Educational Network for Satellite Operations (GENSO) is another example of re-tasking available space resource. It is a project promoted by the International Space Education Board (ISEB). The goal of GENSO is the cooperation on educational matters. Its members include ESA (European Space Agency), CSA (Canadian Space Agency), JAXA (Japan Aerospace Exploration Agency), NASA (National Aeronautics and Space Administration), and CNES (The National Center of Space Studies) [14].

1.2 Motivation

As noted earlier, CubeSats have multiple advantages over larger satellites. The smaller size of the CubeSat means less materials are used to construct the satellite and the development time is much shorter than traditional satellite programs. CubeSats are therefore more cost-effective than the traditional larger satellites. The disadvantages of CubeSats are that their smaller size often has the consequence of using smaller payload instruments, simpler missions due lack of available volume in the CubeSat for multiple payloads, and active guidance mechanism such as thrusters. To counteract these disadvantages, multiple satellites are used in formation to complete the tasks. These formations or constellations can range from 3 to over 1000s.

With larger constellations, scheduling becomes a challenging problem no matter the mission objective due to the increasing amount of data generated during the mission. Satellites in these large constellations are often in low earth orbits (LEO) to allow the satellites to share the same ground resources. Sharing ground resources has the consequence of scheduling conflicts where more than one satellite needs to communication with the same ground resource simultaneously. It then becomes an optimization problem to choose which satellite uses the conflicted communication time with the ground resource. In traditional scheduling methods, a human operator solves scheduling conflicts manually and relay the commands to the satellites. With such large (100s or 1000s of satellites) constellations the amount of conflicts quickly increases to numbers that would strain human operators. Therefore, an automated scheduling and optimization algorithm are required. The current tools available to large constellations are inadequate to be used effectively due to the limitations of scaling them to thousands of spacecraft, targets and ground stations.

Mission planning methods for satellite constellations involves multi-objective optimization method. In any optimization problem there are key design decisions and the formal constraints. For generic mission planning the decision variables and constraints are difficult to define quantitatively, as there is no physical link between them and the constraints that can be expressed mathematically. The problem is further complicated when the satellites in the constellation have different constraints and capabilities. Constraints for mission planning include data storage, attitude and orbit control and maneuvers, thermal, and power constraints. STK (Systems Tool Kit) is the traditional method used for mission planning; to use it for large constellations the operator would need to build a large constellations planning and management tool within STK.. There have also been several publications on optimization techniques, but the studies thus far all focus on a specific constellations mission objective and on a single satellite platform or a constellation under 10 satellites [8]. These publications treat the constellation as a small collection of satellites and therefore do not address the complications that are associated with many satellites working on the same parameters [20]. There are limited generic constellation planning tools that can be customized to fit a range of constellation structures and mission objectives [20].

1.3 Research Objectives

The primary research objective of this thesis is to design a generic constellation scheduling algorithm that produces an optimized schedule for a large constellation of CubeSats by testing the effectiveness of the chosen design variables and the blending of them. The algorithm must be sufficiently flexible for any number of satellites, ground locations and target locations, as well as individual satellite constraints and capabilities, such as battery power and data storage. The constraints considered are power and data storage. The decision variables (or the characteristics to be optimized) includes amount of data, age of data, spatial coverage of targets, and temporal coverage of targets.

1.4 Thesis outline

The following 6 chapters are structured as follows. In Chapter 2, a brief history of scheduling methods is described, as well as an overview of current scheduling tools available and a trade study on optimization techniques applicable for the CubeSat constellation scheduling.

In Chapter 3, the optimization algorithms are detailed, consisting of the logic flow including the input variables and structures, the orbit propagation, the scheduling controls, and the optimization calculation and constraints check. In Chapter 4, there is an explanation of the scenarios chosen to validate the success of the algorithm. Chapter 5 summarizes the results of the test cases from chapter 4. Conclusions and final remarks are provided in Chapter 6.

Chapter 2 Background

2.1 Early Mission Scheduling for Autonomy

Most of the satellites in orbit today do not possess much intelligence or decision-making capabilities. Instead, the current software for satellites measures sensor data, receives and completes ground commands usually constructed by human operators, and reboots rather than troubleshoot when an error occurs [36]. One of the first attempts at autonomous scheduling for satellites was made by NASA's Deep Space 1 (DS1) mission. It was launched in 1999 but due to technical difficulties the scheduling software was removed from the satellite. The patches to the software were later uplinked to the satellite and tested in monitored scheduled intervals. DS1 used an agent architecture that was based on a model-based programming method. DS1 had onboard deduction and search algorithms along with goal-directed closed-loop commanding. The agent architecture relied on tradition flight software instead of a hierarchy of intelligent agents [36].

ATLAS was another autonomous scheduling mission test launched by ESA. ATLAS uses agent-based software where the schedule of the constellation is calculated by the cooperation between different "agents" [3]. An agent is described in [42] as "…a computational entity that can be viewed as perceiving and acting upon its own environment and that is autonomous in that its behavior at least partially depends on its own experience". Each satellite in the constellation is an agent. The planning of the ATLAS constellation is computed by the cooperation between all agents. The agents exchange messages that are guided by criticality and cost. This cooperation between agents produces a mission plan that maximizes the number of scheduled requests completed while also balancing the load of requests within the constellation. ATLAS agents have three main steps; Perceive, decide, act. Perceive is the step where the satellite's current condition and environment is analyzed, decide is the step where ATLAS makes a plan on what actions to take in order to achieve the objective and act is the commands sent to the separate satellite systems to achieve the plan. [3].

DS1 is only one satellite and ATLAS staggered three satellite launches. The autonomy for larger satellite constellations is a much different problem. Some of the challenges for constellation autonomy are:

1) Instead of one satellite there will be multiple satellites that may be coordinating with each other to complete the same goal or working on different goals [36].
2) The proximity between multiple satellites creates more control challenges and resource allocation challenges [36].
3) If the algorithm is space-based, then all satellites must produce the same results [36].

Agent-based software differs in organization structure and functional distribution compared to traditional space software. Agent-based software can be used to test different software options as well as architectures where each agent is a virtual satellite. Part of the agent-based software is the decision-making process when there is a conflict between satellites or agents.

The missions mentioned above all have limited satellites in their constellations and therefore could have followed the traditional operation architecture for scheduling that requires significant operator involvement. This method of scheduling does not scale up well to larger constellation sizes [23]. The complexity of scheduling increases with increasing data amounts, increased space resources and limited ground resources. Adding in constraints such as energy and storage adds yet another layer of complexity.

2.2 Earth Observation and Disaster Aid Mission Planning

Data rates and customer demand are continuing to grow and consequently more importance is placed on the effectiveness of satellite management and scheduling methods. SmallSat constellations are a proposed method to meet the ever-increasing data demand. SmallSat constellations also have the measurement advantages of higher temporal resolution, multi-point instrument coordination, and low latency data availability [23]. Tropics is a six 3U CubeSat constellation that is managed by MIT Lincoln library. It is a low data mission that continually scans Earth's atmosphere to provide data to weather models that is rapidly updated. Each satellite transfers approximately 1.5 GB of data per day [23]. An example of a large data mission is Planet's Dove satellite constellation. This constellation transfers 6 terabytes of data a day [23]. Hyperspectral images require an even larger data transfer amount [23]. There are also large "instantaneous" data packets that need to be downlinked quickly for disaster relief aid [23]. Satellites from several nations can be repurposed for relief aid but this supplies a new set of complications to the scheduling problem since the satellites will have different capabilities and constraints.

To effectively schedule large data missions, custom objective functions have been proposed. Herald et al suggest a hierarchical system with a centralized coordination algorithm. The algorithm would interface with sub planners. The schedule is computed using a Mixed Integer Programming (MIP) model a solver. The algorithm addresses observation tasking but does not effectively consider the data route through the constellation space and ground resources [16]. Multi-agent cooperation has been studied and implement successfully outside of space applications. Now similar methods are being proposed for space applications. Agents are considered to be satellites, robots, or other decision-making entities. In Wang et al., the algorithm uses graph decomposition techniques to simplify the planning complexity and then uses a heuristic forward search to compute a schedule [40]. Gombolay et al. used a Mixed Integer Linear Programming (MILP) formulation for quick scheduling of tasks to multiple factory floor robots. For this situation, temporally evolving spatial constraints were critically important [14]. Both Wang et al and Gombolay lack the necessary considerations for delay tolerant satellite networks [23]. Choi et al use agents that come to a consensus on task allocation by passing messages between each agent. These messages contain information on each agent's decisions. The communication that is needed for this method is not found in delay tolerant networks [9]. Amato et al used a decentralized partially observable Markov Decision Process (POMDP) to find an optimal decentralized schedule across multiple agents using explicit reasoning about uncertainty in the agents' activities [1]. This method is very computationally heavy and complex and therefore does not scale up well for large satellite constellation missions. On board satellite task allocation discussed in Damiani et al has a system where the centralized ground system keeps track of the global tasks that are available and then distributes the tasks to the satellites during ground communication times. The satellites then use on board planning systems to fulfil task based on their status. This approach also takes into consideration energy, memory and, timing constraints [11]. Spangelo separates individual satellite responsibilities hierarchically from an analytical model of the constellation network and then constructs a simulation environment from the results [37]. This is again too computationally heavy for larger satellite constellations.

Most of the solutions, discussed in Chein et al., for scheduling algorithms maximize mission throughputs and resource utilization, and are developed for single or small number of platforms [8]. Constellations are treated as a small collection and the interface and operational issues associated with larger numbers of similar satellites in a constellation are not addressed [20].

2.3 Communication Satellite Mission Planning

Iridium is the canonical constellation example. Iridium has large scale inter-satellite networking and four direction communication links. Packet routing is scheduled in discrete time steps for the entire constellation network. The goal is for the packets to take the shortest "distance" i.e., minimize the number of transfers between satellites and ground stations, to the gateway ground station from the customer origins. This is a constellation that operates continuously instead of using a delay tolerant network [23]. More specific to small satellite constellations with a network of ground stations, Castaing proposes a MILP formation to address conflicts and schedule downlink transmission with resource constraints [6]. Parham et al propose a mesh network that forwards packets across a small cluster of sensor nodes aboard the satellites [33]. Wu et al applies a Tabu search algorithm with a GA (Genetic Algorithm) to minimize energy use and route latency [44]. Wang et al discusses a MILP formulation with results that guarantee ratios of observation performance and latency requirements for a 6-satellite constellation [42]. The algorithm delivers 60 % of the packets within a 120-minute delay requirement. Zhou et al proposes the use of two algorithms together, the first is the "Mission Aware Contact Plan" that uses MILP formulations to compute optimal throughput. The second is the "Algorithm based on conflict graft" which sequentially assigns data route decisions over a range of time slots. The results show 90 % of optimal data throughput across the constellation of six satellites within 2-hour planning windows [45].

Most of the current research into communication constellations does not account for the unique complications that arise when using large numbers of small satellites in LEO. There are several companies that are currently designing this type of communication constellation such as Kepler Communications, Boeing, and SpaceX.

2.4 Current Mission Planning Tools

Currently there are few generic constellation planning products that can be customized to fit a range of mission objectives. Planet Labs launched and manages the DOVE constellation and offers services to manage other entities' constellations [20]. Terra Bella is also managed by Planet Labs and allows for more human operator interaction and repeated scheduling. CPAW from orbit logic is the most advanced option available. It offers spacecraft simulations and scheduling algorithms with high fidelity image collection plans, a range of settings in satellite model definitions, ranging degrees of automation and graph search heuristic algorithms [20]. Orbit logic works as an extension of STK. STK is a graphical interface and high computationally heavy with high licensing fees. The satellite schedules are presented graphically and in text file format. Sa Voir Swath Planner was developed by Taitus Software. This planner analyzes the coverage of EO satellites. It uses a GRASP heuristic to compute coverage in near real time but cannot perform more advanced constraint modeling which makes it unsuitable for large constellation scheduling. Deimos' gs4EO software developed the Capacity Analysis and Mission Planning Tool. This tool is a greedy scheduler and GA optimization algorithm. It has low fidelity for the basic user and medium fidelity for the offline simulator for expert users. It is a collect of tools instead of an integrated path that is user friendly [20].

The four methods mentioned above are currently the most used tools available for constellation scheduling. These methods cannot be simply modified to be used for any constellation mission. There are similarly many publications on scheduling algorithms, but these algorithms are all very mission specific instead of generic and are not easily customizable[8].

2.5 Multi-Objective Optimization for Satellite Constellations

Often in engineering problems there is a desire to optimize more than one objective. This is called Multi-objective optimization (MOO). As shown above there are multiple uses for large satellite constellations and each use has different objectives to optimize for optimal mission results. Multi-objective optimization problems can be expressed as:

$$\text{Minimize } \mathbf{F}(\mathrm{x}) = [F_1(x), F_2(x), \ldots F_k(x)]^T \qquad \text{Eq. 1}$$

Subject to $g_j(x) \leq 0, j = 1, 2\ldots, \mathrm{m}$,

$h_l(x) = 0 \; l = 1, 2\ldots, \mathrm{e}$,

Where k represents the number of objective functions, m represents the number of inequality constraints and e represents the number of equality constraints. The vector of design or decision variables is represented as $x \in E^n$ where n represents the number of independent variables x_i. $\mathbf{F}(\mathrm{x})$ is a vector that represents the objective functions. In some situations, instead of minimizing $\mathbf{F}(\mathrm{x})$, the situation may call for maximizing instead [25]. For telecommunication constellations there can be many different decision variables such as maximizing the data amount transferred or minimizing the cost of the data transfer.

Multi-objective optimization was constructed from three main areas: economic equilibrium and welfare theories, game theories, and pure mathematics [29]. There are many different methods of MOO but many of them are incomplete in comprehensive coverage and algorithm presentation, and they are not often to easily applicable to engineering designs because they are either too specific or too general [25]. There have been many papers on proposed MOO algorithms such as

Marler et al. which suggested a weighted sums method, and Deb et al., which suggested a generic algorithm (GA) (this method does not add artificial constraints to the problem equation) [29][10]. Linkov et al. suggested using a max-min method which would avoid the worst-case scenario performance while considering minimal preference criteria. This method can only be used when the criteria are comparable to one another and therefore can be measured against each other on a common scale [4]. Shanian and Savado (2009) suggest ranking the decision variables by how similar they are to the ideal solution. To do so the ideal solution mut be known beforehand [35].

MOO can be categorized into three different classifications:

1) Priori articulations of preference – This classification needs to have the preferences specified and the preferences are the articulated by their importance to the different objectives being considered. Such methods include Weighted min-max method, Goal programming, Bounded objective function, Lexicographic method, Weighted product method, and Reference Point Method [29][10][25].
2) Posterior articulation of preferences – This classification needs to have an explicated approximation of the preference function. This means that the preferences are presumed to be embedded into the parameter set. Such methods include Generic Algorithms (GA), Physical programming method, and Normal constrain (NC) method [25].
3) No articulation of preferences – In many cases the optimal solution or best preference scenario must be explicitly known. This means the articulation of preferences is not required. Methods in this category are often simplified version of priori articulation methods such as Compromise function and Min-max method [25].

GAs are a popular choice for MOO problems. They can be modified to find multiple solutions in a single run. This is because GAs can search different regions of the solution space simultaneously. This is advantageous for discontinuous, multi-modal, and non-convex solutions spaces. Most GAs do not require the preferences to be prioritized, scaled, or weighted which is why GAs are such a popular choice. The following table shows the advantages and disadvantages to some commonly used GAs:

Table 1 GA comparisons

Algorithm	Fitness assignment	Diversity mechanism	Advantages	Disadvantages
VEGA [6]	Each subpopulation is evaluated with respect to a different objective	No	First MOGA Straightforward implementation	Tend coverage to the extreme of each objective
MOGA [5]	Pareto ranking	Fitness sharing by niching	Simple extension of single objective GA	Usually slow convergence
WBGA [[8]]	Weighted average of normalized objectives	Niching Predefined weights	Simple extenuation of single objective GA	Difficulties in nonconvex objective function space
NPGA [14]	No fitness assignment, tournament selection	Niche count as tiebreaker in tournament section	Very simple selection process with tournament selection	Problems related to niche size parameter. Extra parameter for tournament selection
RWGA [36]	Weighted average of normalized objectives	Randomly assigned weights	Efficient and easy implementation	Difficult in nonconvex function space
PESA [[40]	No fitness assignment	No fitness assignment	Easy to implement Computationally efficient	Performance depends on cell sizes Prior information needed about objective space
PAES [10]	Pareto dominance is used to replace a parent if	Pareto dominance is used to replace	Random mutation hill climbing strategy Easy to implement	Not a population-based approach

	offspring dominates	a parent if offspring dominates		Performance depends on cell sizes
NSGA [[42]	Ranking based on non-domination sorting	Fitness sharing by niching	Fast Convergence	Problems related to niche size parameters
SPEA [3]	Raking based on the external archive of non-dominated solutions	Clustering to truncate external population	Well tested No parameter for clustering	Complex clustering algorithm
SPEA-2 [23]	Strength of dominators	Density based on the k-th nearest neighbor	Improved SPEA Make sure extreme points are preserved	Computationally expensive fitness and density calculations
RDGA [[37]	The problem reduced to bi-objective problem with solution rank and density as objectives	Forbidden region cell-based density	Dynamic cell update Robust with respected to the number of objectives	More difficult to implement than others
DMOEA [19]	Cell-based ranking	Adaptive cell-based density	Includes efficient techniques to update cell densities Adaptive approaches to set GA parameters	More difficult to implement than others

Note: The information in the table is a summary from Konak, Abdullah & Coit, David &Smith, Alice. (2006). Multi-objective Optimization using Genetic Algorithms: A Tutorial. Reliability Engineering & System Safety. 91. 992-1007. 10.1016/j.ress.2005.11.018.

Another category of algorithms is the greedy algorithm. In a greedy algorithm a decision is made at each step for the optimal solution. Once the algorithm moves on to the next step the previous options are no longer considered. Therefore, greedy algorithms do not always find the optimal solutions because they fail to consider the whole problem when making decisions and instead focus only on the information of the current step. For example, considered the tree below:

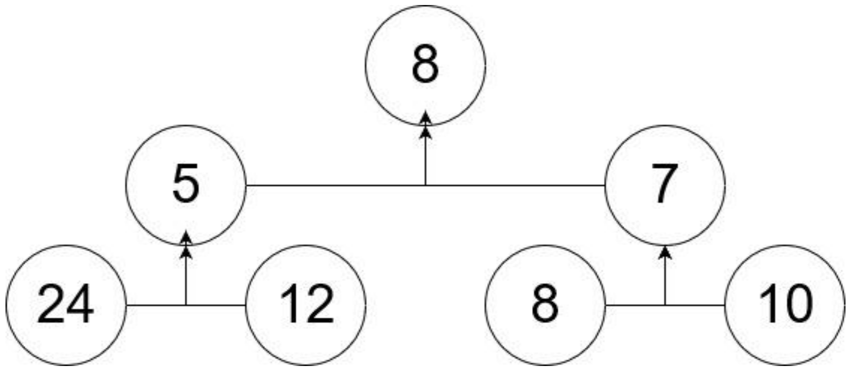

Figure 4 Decision Tree

A greedy algorithm would choose the right path (7) a then the right path (10) again for a total of 25. The actual optimal solution would be taking the left path (5) and then the left path (24) again for a total of 37. The algorithm discussed in this thesis finds all conflicts with the current request. Using the example of the tree above, this means that the algorithm would look at all options in each row of the tree. The solution is then calculated based on the cost function using the optimization algorithm. Furthermore, the methods of optimization discussed above are designed to find an optimal solution for a single mission type with limited constraints and the constellation sizes considered for most of the publications are smaller than the forecasted constellation sizes to be launched in the next decade.

Kim and Weck claim the weighted optimization is the, "most widely used method for multi-objective optimization" [24]. It is often used in structural optimization such as a four-bar space truss by Koski and Silvennoien [27] and for maximizing critical buckling sheer stress and minimizing deflection by Kassaiman et all. [22]. Proos et all [34] used weighted sums optimization in two-dimensional plane stress too minimize compliance and maximizing the first mode of the natural frequency. While weighted sums optimization is used extensively it does have two faults. First is that weighted sums optimization does not always capture the Paetro optimal points for the points that lie on non-convex portions of the Pareto optimal curve, as stated by in [2], [7], [18], [26], [38] and [39]. The second fault is that with a consistent change of weights does not provide an even distribution of points in the Pareto optimal set. The main concern with weighted sums optimization is that the method does not provide a complete Pareto optimal set but does provide a single solution or a priori articulation of preferences [29].

Despite these faults weighted sums Marler and Arora states that it is still extensively used to provide multiple solutions when the weights are consistently varied and also to provide a single solution where the preferences are reflected by the selection of weights[30]. Marler and Arora have identified two broad classes of approaches taken when weights are assigned to the objective functions [30]. The classes were identified using surveys [13], [17], [19], [40]. The first approach is the rating method. The rating method has the decision maker assign weight values that represent the relative importance of the objective function. The second method is the categorization method which groups objective functions into categories of importance from low to high. The first method will be used in the algorithm studied in Chapter 3.

Chapter 3: Mission Scheduling and Optimization Algorithm

The Mission Scheduling and Optimization Algorithm was designed to produce the optimal schedule for constellations that utilize many satellites. The optimization algorithm is used to compute the optimal schedule for a variety of missions, including Earth observation, telecommunication, and disaster aid, by using the chosen design variables in a weighted-sums optimization to test the effectiveness of blending the design variables. The algorithm uses a "brute-force" method (also referred to as "generate and test approach" where a series of potential solutions are generated then systemically evaluated to check if each candidate solution meets the criteria; once they are ranked based on pre-determined criteria, the optimum solution is selected based on the ranking) for each time window within the schedule. To be customizable the algorithm asks for inputs that include the satellite's Two-Line Element (TLE) data, maximum storage, maximum battery power, and power drains. TLE data for satellites describes the satellite's orbit and identification. The algorithm has two modes: pre-scheduling and on-demand scheduling. Pre-scheduling is used when the algorithm produces an optimized schedule that assumes all targets (defined as ground resources that are not ground stations such as customer terminals and areas of interest for EO missions) have data to transfer at every access time. This mode of operation is more useful to Earth observation and science missions. The second mode operates in real time where customers make requests to transfer data and the schedule is re-calculated for every request. The algorithm is designed to operate with a delay tolerant network (DTN) and transfers are based on a store-and-forward data link service. Once the data are uplinked, they are stored onboard the satellite until they can be downlinked to ground stations based on the schedule created by the algorithm. All algorithms have been developed in MATLAB™ and consist of four main components: input, orbit propagation, schedule manager, and cost calculation. The output of the scheduling algorithm is a matrix that represents the schedule. This matrix format may be easily translated to satellite commands.

3.1 Input

The first step of the algorithm is to assign input variables. The algorithm was designed to accommodate various mission and constellations sizes. The algorithm is cable of handling a changing satellite constellation and ground network. Satellites, ground stations and targets may be added or removed from the schedule quickly and easily. To accomplish this, the algorithm requires an input file which contains the variable data. The input to the file is a text file, with the following information More information can be found in Appendix A.

Table 2 Input Parameters

Input	Description
Ground Stations	Latitude, longitude, and elevation
Target Locations	Latitude, longitude, and elevation
TLE	TLE data file location
Start and End Date and Time	Start date and End date of the schedule including time.
Satellite Constraints	Storage capacity in GB
Satellite Power Drains	Power consumption during data transfers.
Decision variable weights	Weights are from 0 to 1 and their sum must be equal to 1. There are 4 design variables: Transfer time, age of data, spatial coverage, and temporal coverage.

The number assigned to each satellite is based on the order in which the TLE data files are listed in the input file. The same numbering system is also applied to the ground stations and targets. Ground stations are the ground resources that can either store or relay the data to and from the satellites. Three types of targets are considered. The first is a (1) user terminal where its latitude, longitude and elevation would be of the user terminal’s location. The second is an (2) Earth observation target whose coordinates indicate the desired area of observation. Instead of a data transfer from the target location the satellite would complete a desired mission task using the onboard scientific payload for that mission. The third is (3) scientific ground instruments and therefore the coordinates indicate the location of the instrument. This is also a data transfer from the target location.

3.2 Orbit Propagation

The next step in the operation of the scheduling algorithm is to integrate the input file data with an SPG4 orbit propagator to calculate the satellite orbits and access times. The access times consists of two categories: ground access and target access. The target access times are treated as requests denoted as R1, R2 R3, … Rn) where n is the number of target access times across all satellites. The requests are stored together in chronological order. The information in each request is the start time, target ID number, and satellite ID number. All ground access times are stored together also in chronological order. The information in each ground access time is the start time and ground station ID number.

3.3 Schedule Manager

The schedule manager is the algorithm to store and update the schedule and process requests. There are two modes for the schedule manager. Mode 1 is used to create a schedule before the start time and is used when the targets are known before the scheduled start time. Mode 2 is used for on demand scheduling and therefore is continuously updating with each new request.

For Mode 1 it is assumed that all targets have data to transfer for all access times. Mode 1 accesses the request matrix and processes the requests in order. Starting with R1, the schedule manager then accesses the ground access matrix and finds all ground access times for the satellite involved with Rn after the end time of Rn. The schedule manager is then used to find and records all conflicts between Rn and the current schedule, and all ground access times for Rn and the current schedule. It is assumed that a target, satellite, and ground station has only one channel of communication and therefore cannot transmit and receive at the same time or transmit or receive multiple signals at a time. Therefore, a conflict is when there is more than one request that uses the same resources (satellite, target, or ground) at the same time.

For every ground access time, the schedule manager is used to create two temporary schedules to compare. The first schedule is the current schedule with no changes. The second schedule inputs the current request and a ground access time. If there are any conflicts with the second schedule, then the schedule manager removes the request in conflict and the requests corresponding ground access time and leaves the current request in the schedule.

Both schedules are then sent to the optimization algorithm which will be discussed in section 3.4. The optimization algorithm is then used to calculate the optimal schedule and passes it back to the schedule manager. If the second schedule is chosen, then the schedule manager updates the request structure and current schedule. The request structure keeps track of the satellite ID, target ID, and ground ID for each request in the current schedule as well as the start and end times for the target and ground access durations. This process is repeated for every request.

While the algorithm is currently set up for data transfer missions it is also possible to use it for Earth observation missions. The scheduling process remains the same but instead of using satellite commands to transfer data during target access times (request times), the commands are to operate the scientific payload for the Earth observation mission (camera, spectrometer, etc.).

Mode 2 works like Mode 1 except that instead of a list of requests passed to the schedule manager, there is only one request is passed in real time. These requests are submitted by the users. The target locations are therefore the users' terminals. The schedule is saved in the schedule manager after every request. The requests are handled in 24-hour periods. If a request is made but cannot be completed in the current 24-hour schedule it is saved and passed back to the schedule manager for the next 24-hour schedule. Mode 2 asks the operator the following 4 questions:

1. *Target number:* This question ascertains the ID of the ground target which is the user's terminal.
2. *Start time:* This question ascertains the time that the request is made. The user inputs when they want the transfer to begin the following day. The algorithm then finds the nearest communication times available to meet the request
3. *Size of the file*: This question allows the algorithm to schedule multiple communication passes if the file is not transferred completely on the first pass.
4. *Urgency of the request:* This question allows the algorithm to schedule using a priority protocol if desired. An urgency of 1 takes top priority and 3 the lowest priority.

3.4 Optimization and Constraints

Before the temporary schedules are passed to the cost function, the schedules undergo a constraint-check. The constraints considered are power and storage limits. The algorithm assumes that the satellite recharges the battery using solar panels.

The constraints code is designed to take in the uplink and downlink schedule for a group of satellites, targets, and ground stations to make sure the satellites have the required power and storage to complete the scheduled tasks. This is accomplished by first taking the input schedule and breaking it into its components: uplink, downlink, and standby. Each component has a power requirement and storage requirement per minute of operation, which includes the solar panels that passively recharge the battery. The power and storage requirements are tracked per minute to integrate smoothly with the scheduling code. The power and storage rates for each operation can be changed and swapped for different modes or different satellites as they are imported in by the software. Once all the tasks are known they are split into individual rows for each task showing when it is in operation and when it is not. These are then summed together over the schedule period to get the overall change in storage and power. The constraints considered are as follows:

- Payload Power Consumption
- Payload Storage Consumption
- Uplink Power Consumption
- Uplink Storage Consumption
- Downlink Power Consumption
- Downlink Storage Consumption
- Data Processing Power Consumption
- Data Processing Storage Consumption
- Hibernation Power Consumption
- Hibernation Storage Consumption
- Power Regenerated per day.

The software requires the initial condition of the battery and storage as it uses the change in power and storage to find the power and storage across the entire schedule. The software creates graphs of the power and storage drain as well as the overall power and storage as shown in Figure 5-7 below. If the power or storage ever drops below a minimum value that is selected by the user, the schedule is declared invalid and outputs when and why the schedule failed. The following graphs illustrate an example of the constraints.

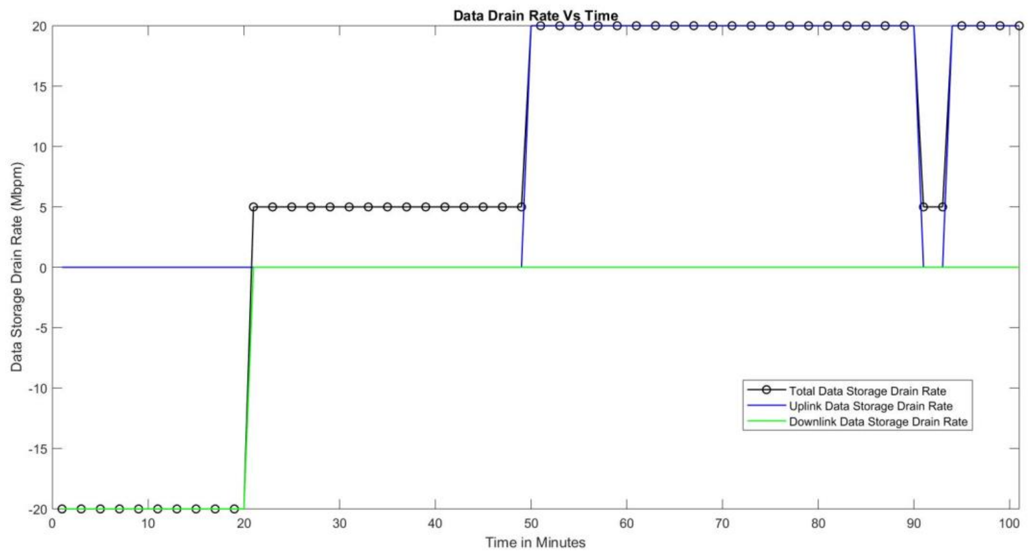

Figure 5 Data Drain Rate vs Time for a test case.

In Figure 5, the data drain rate starts with a 20 Mbpm (megabyte per minute) downlink and 0 Mbpm uplink. The data drain rate remains that way until t = 20 minutes where the data drain rate changes to 0 Mbpm downlink and 5 Mbpm uplink . At t = 50 minutes the data drain rate for uplink changes to 20 Mbpm while downlink drain rate stays at 0 Mbpm. The data drain rate for both uplink and downlink stay at 20 Mbpm and 0 Mbpm until t=90 minutes where the uplink data drain rate changes to 5 Mbpm. The total storage drain rate is the same as the uplink or downlink drain taking place at any time across the 90-minute window. This is because it is assumed that the satellite cannot transmit and receive at the same time.

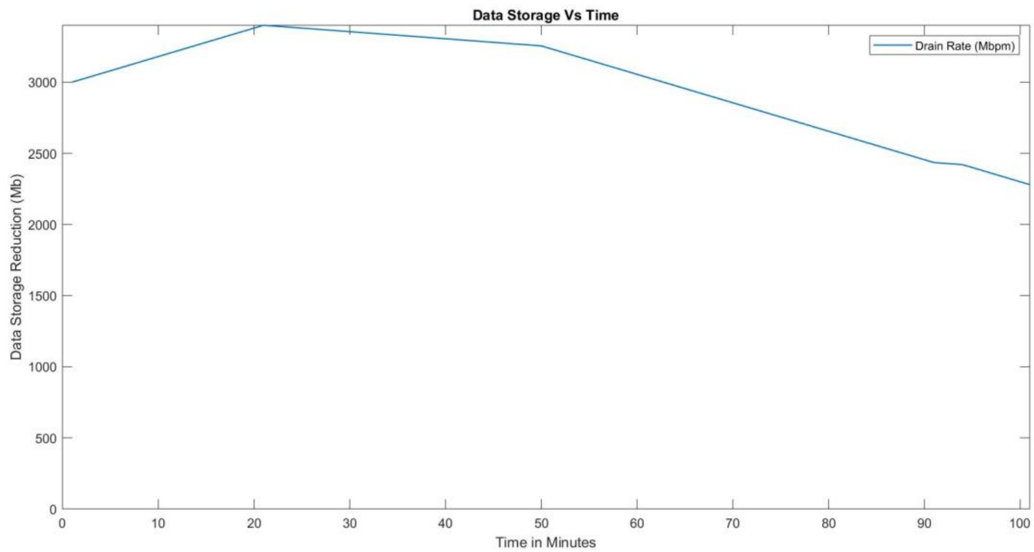

Figure 6 Data Storage vs Time for a test case.

In Figure 6, the constraints shown are 0 data stored on board and 3500 MB stored on board. At t = 0 minutes the satellite has 3000 MB stored on board; it gains data stored until the maximum of 3500 is reached. It then declines for the rest of the test till it reaches approximately 2300 MB. This means that between minute 0 and approximately minute 22 there is at least one uplink of data to the satellite. There are three distinct slopes that represent the downlink of data which represents at least 3 downlinks of data to the ground.

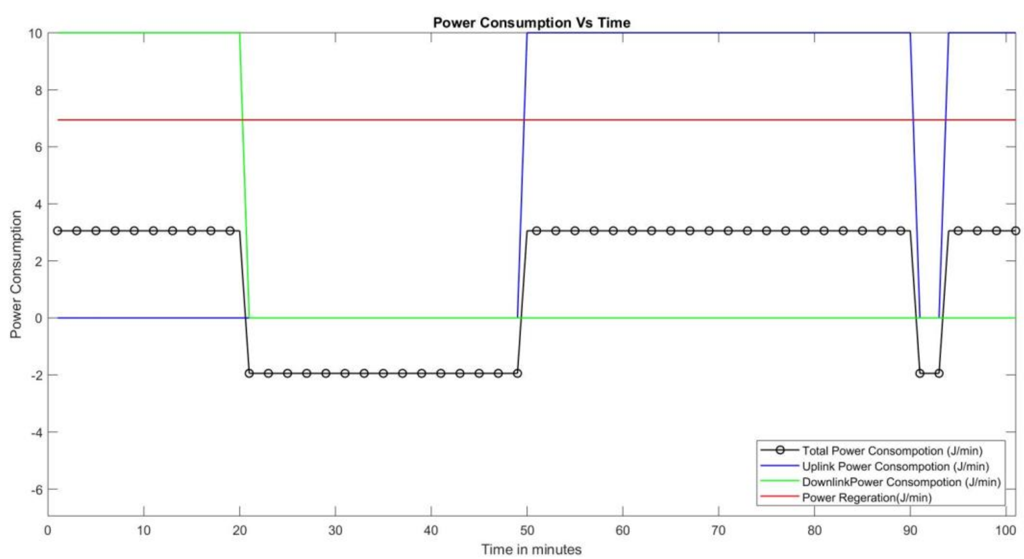

Figure 7 Power Consumption vs Time for a test case.

In Figure 7 the red line is the power regeneration in J/min, the black line and circles represents the total power consumption in J/min. The green line is the power drained from downlinking and the blue is the power drain from up linking. There is a constant power regeneration of 7 J/min as shown by the red line. From minute 0 to minute 20 the satellite is downlinking data and using 10 J/min of power as seen by the green line. The blue line representing power used during uplink is at 0. The total power consumption is then 10 J/min – 7J/min which is 3 J/min as shown the black line with the circles. From minute 20 to minute 50 the power used during downlink goes to zero and the power used during uplink stays at 0. The total power consumption is -2 J/min which is the power regenerate minus the power consumption for when the satellite is idle. This means that the satellite is gain power at 2 J/min from minute 20 to minute 50. From minute 50 to minute 90 the satellite is regenerating power at 7 J/min, and using 10 J/min to uplink data, this leads to a total use of 3 J/min.

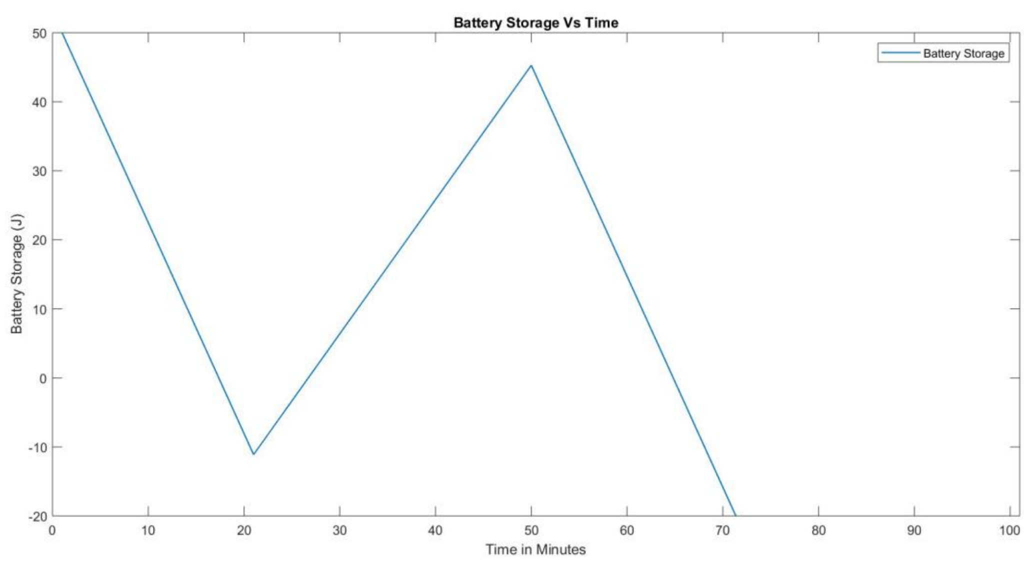

Figure 8 Battery Storage vs Time for a test case.

In Figure 8, the battery storage is depicted. Figure 8 shows that the test starts at 50 J and the decreases to -10 J at t = 20 minutes. This would mean that the test failed since you cannot have negative battery storage.

The algorithm is designed to be adaptable to different mission objectives such as data transfer, Earth observation or disaster relief aid. It is also designed to be flexible and allow for a change in ground or space resources and mission objectives.

The weighted sums method was chosen for optimization algorithm because of its flexibility. Different types of mission require different weighting of different decision variables. For example, a disaster relief mission is to minimize the time between uplinks and downlinks and an Earth observation mission requires images taken throughout the day. With the weighted sums method, it is possible for the user to set the weights on the preferences depending on the desired outcomes.

The objective function of the weighted sums method can be expressed as,

$$U(x) = \sum_{i=1}^{k} w_i F_i(x) \qquad \text{Eq. 2}$$

Where w_i is the weight assigned to the design variable $F_i(x)$

k is the number of objective functions or design variables.

$f_i(x)$ is the objective function.

The weight sums method can be expressed as

$$\text{Maximize } U = \sum_{i=1}^{k} w_i F_i(x) \qquad \text{Eq. 3}$$

And is subject to feasible constraints. [31]

Four design variables were chosen as schedule characteristics to optimize based on the mission's objectives. The design variables are as follows:

Transfer time –total transfer time in minutes across all satellites that data is uplinked from target to satellite and downlinked from satellite to ground. If the mission was an earth observation mission instead of data transfer mission, then the transfer time would also include to length of time a satellite has over a target. This design variable is used to increase or express the importance of the total data transferred during the schedule.

Age of data –number of minutes between the end of the uplink and the beginning of the downlink. It is measured separately for each request and then averaged across all requests. This variable is used to increase the importance of scheduling the data so that the newest information is transferred.

Spatial coverage – This is the variable that designs the schedule so that all targets are visited a uniform number of times. This variable is more suited to Earth observation missions that include a mission objective that involves multiple Earth targets.

Temporal coverage – this is the variable that designs the schedule so that a target is visited uniformly across the duration of the schedule. This is again a variable more suited to Earth observation mission where information about a target is needed at different time stamps of the schedule.

These design variables were chosen because they best represent the objectives of Earth Observation and Science Missions, telecommunication missions, and disaster aid missions. In Earth Observation missions the spatial and temporal coverage are important to the mission's success. A target may need to be observed at the same time every day to observe long term changes or at varying periods during the same day to observe short term changes. For meteorology missions the spatial and temporal coverage is again important to the mission's success as well as minimized age of data so that the user is using the most current data available. For telecommunication missions the amount of data should be maximized while the age of data remains within a chosen time constraint. Disaster aid missions will want the newest data available and an even spatial coverage across the target area.

The cost function is defined for a weighted sum optimization as shown below:

$$\sum_{i=1}^{k} w_i C_i \qquad \text{Eq. 4}$$

Where $k$is the number of decision variables, w_i is the weight assigned to that decision variable and C_i is the design variable scalar quantity. The design variable scalar quantities are calculated as follows:

<u>Link Access:</u>

$$C_1 = \sum_{1}^{n}\sum_{1}^{m} \min\left(\frac{Uplink\ Access}{Max\ Possible\ Uplink\ Access}, \frac{Downlink\ Access}{Max\ Possible\ Downlink\ Access}\right) \qquad \text{Eq. 5}$$

Where n is the number of satellites and m is the number of targets. A value of 1 is the best option. The max possible data up and down is across all satellites and targets.

For the design variables to be optimized successfully, the design variables must be of similar order of magnitude. Therefore, the sum of data uplinked and downlinked is divided by the max data up/down calculated for that schedule. The max data value does not solve schedule conflicts and there for the max data value is not possible unless there are no conflicts calculated during the orbit propagation. Calculating the design variable amount of data using this method will bring all order of magnitude values for the scalar quantity of the design variable called amount of data to be between 0 and 1.

Age of Data:

$$C_2 = max\left[1 - \frac{[\sum_1^n \sum_1^m (Minutes\ between\ Data\ Uplink\ and\ Data\ Downlink)]}{1440 * n * m}\right] \qquad \text{Eq. 6}$$

Where n is the number of satellites and m is the number of targets. The max age of data is chosen from the list of age of data for each transmission. The max age is then divided by the length of the schedule (1 day) in minutes and subtracted from 1 so that the scalar value of Age of data is in the same order of magnitude as the other design variables and so that 1 is the best possible value

Spatial Coverage:

$$Visits = [V_{1,} \dots V_n]$$

$$RT = number\ of\ requests$$

$$C_3 = \left(1 - \left(\frac{var(Visits)}{sum(Visits) - var(Visits)}\right)\right) * \left(\frac{sum(Visits)}{size(RT)}\right) \qquad \text{Eq. 7}$$

Where $[V_{1,} \dots V_n]$ are the number of times each target is visited by any satellite, n is the number of targets and RT is the number of requests across all targets by all satellites. The Spatial Coverage design variable includes the variance because the goal of this design variable is to calculate a schedule that approaches uniform spatial coverage. The variance is divided by the sum of visits subtracted from the var again to bind the value between 0 and 1. It is then subtracted from one to get a maximum value instead of minimum value. It is multiplied by the second term because it is desired that the number of visits be considered as well. For example, if the Visits vector is [1,1,1,1] and if the second term was not included then the algorithm would never choose [1,1,1,2] or any variation of the vector because the variance would be larger.

Temporal Coverage:

$$Time\ Visits\ per\ Target_T = [TV_{1,} \ldots TV_n]$$

$$RT = number\ of\ requests$$

$$C_4 = \left(1 - \left(\frac{var([TV_1,\ldots TV_n])}{sum([TV_1,\ldots TV_n]) - var([TV_1,\ldots TV_n])}\right)\right) * \left(\frac{sum([TV_1,\ldots TV_n])}{size(RT)}\right) \qquad \text{Eq. 8}$$

Where $[TV_{1,} \ldots TV_n]$ are the number of times Target_T is visited during that time slot n and n is the number of time slots (4 slots of 6 hours each therefore n is 1 to 4) and RT is the number of requests across all targets by all satellites. Unlike Spatial coverage which considers all satellites and targets, temporal coverage only considers that target that is in the request.

The algorithm is a time-windowing optimization method. This means that for each request and ground pair the schedule duration is from the start time of the schedule to the end time of ground access time for the request, this is the time window to be optimized. The result is a matrix that describes, by the minute, when each satellite is in contact with targets and grounds based on the chosen weights for the weighted sums optimization that best suit the mission's objectives. The next step is then, based on the matrix schedule, to send the corresponding command and time stamps to each satellite in the constellation. The command structure will depend on the chosen method of satellite control software.

Implementing a weighted sum optimization method is not unique, but the use of data latency, spatial coverage, and temporal coverage in a weight sums optimization with data and energy constraints is unique. The algorithm uses the pre-defined weights, target locations, ground locations, and satellite constraints to create the data set that is then passed onto the cost function. The cost function evaluates every scheduling option for every time window. The different options are constructed from the different schedules resulting from conflict resolution. The cost function is as follows:

$$U_i = w_1 C_1 + w_2 C_2 + w_3 C_3 + w_4 C_4 \qquad \text{Eq. 9}$$

Where w_1 to w_4 are the predefined weights, C_1 to C_4 are the values calculated for the designed variables as described before and U_i is a solution in the solution space. The solutions space is all scheduling options for the current time window that pass the constraints check. The time window is the time duration from the start of the schedule to the end time stamp of the current request being analyzed. The cost function solution space is the evaluated to find the maximum value as shown below:

$$\mathbf{U} = Max(U_i) \qquad \text{Eq. 10}$$

Where **U** is the max value for the whole schedule from start time to the end time stamp of the current request. The current accepted schedules **U** value is then compared to the newly calculated **U** for the current request. If the current value is larger, then the schedule remains the same and the request is added to the end of the redo que. If the newly calculated value is larger, then the schedule is changed to the schedule that is associated with the that **U**.

A flow chart for the algorithm is shown below:

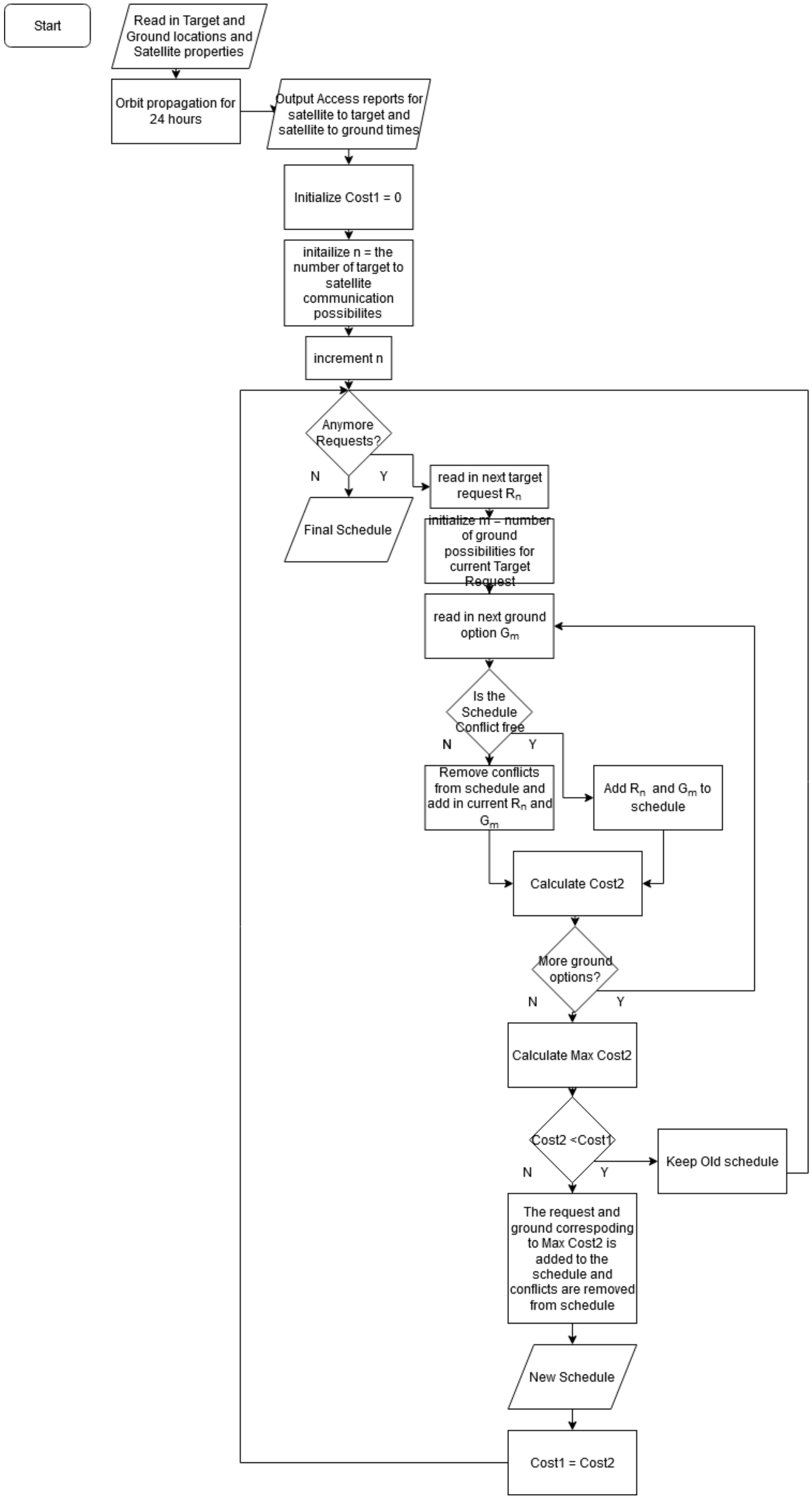

Figure 9 Flow chart for algorithm

A flow chart representing the cost function is shown below:

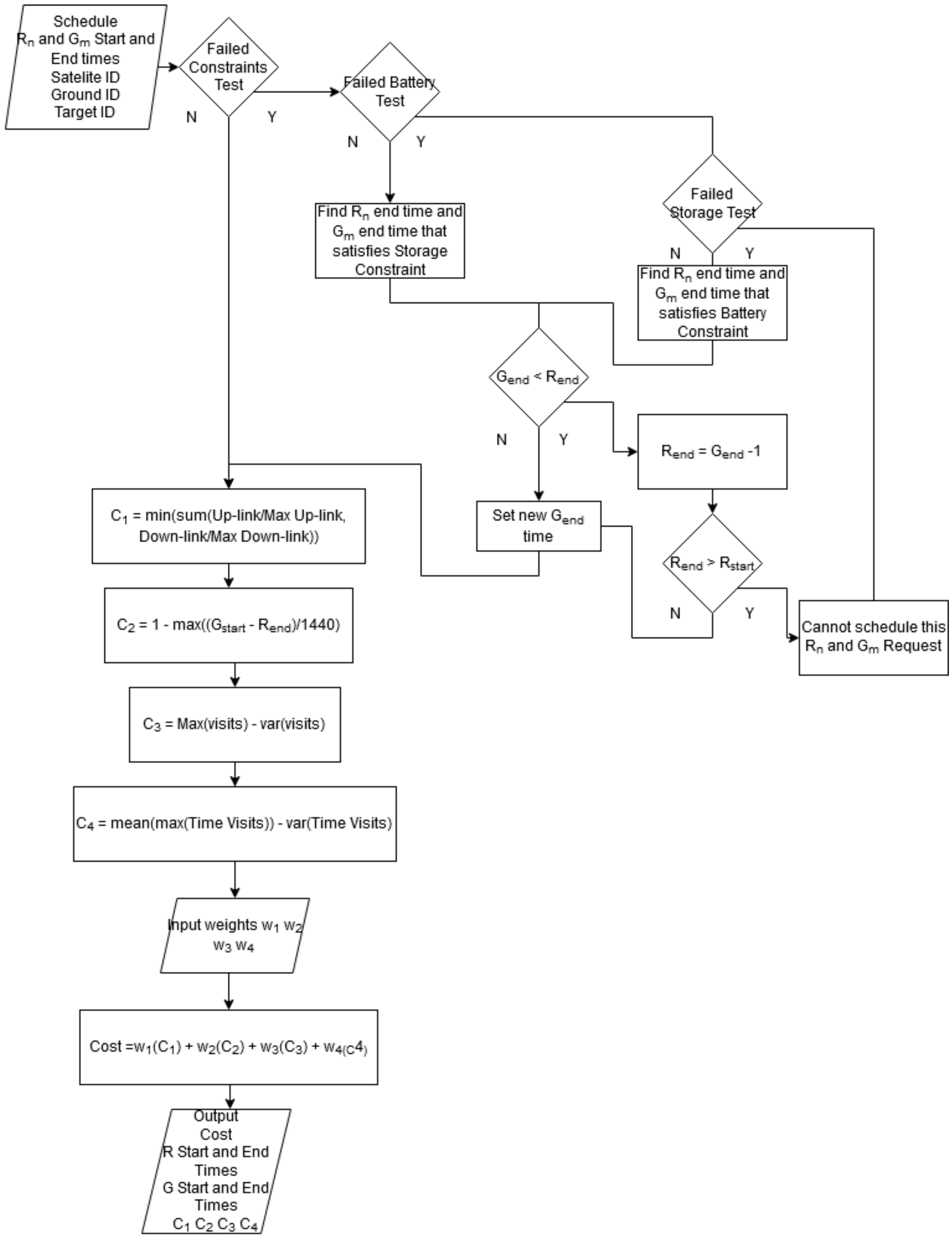

Figure 10 Flow chart for cost function

Chapter 4 Algorithm Validation

The algorithm has been designed for large constellation and non-real-time data collection that is then sent to limited ground resources. The algorithm can manage constellations that have either global coverage or specific target region coverage, with low data latency. To verify that the algorithm is successful, the basic functionality is tested with five simulated test cases. The basic functionality show that the algorithm produces different results when the weights are significantly changed. The second set of tests is to examine more complex scenarios that are based on real-life missions.

There are 4 design cases as follows:

- Test Case 1 – Algorithm verification with five scenarios
- Test Case 2 – Large constellation that maximizes link access
- Test Case 3 – EO mission with uniform coverage
- Test Case 4 – Disaster Aid rapid data transfer that optimizes age of data.
- Test Case 5 - Mode 2 algorithm verification

Test case 1 is to verify the algorithm's functionality. It is a simple case of 5 target and 5 ground stations with a constellation of 10 satellites. In this test case, there are 5 scenarios that aim to demonstrate that the algorithm produces different schedules when varying weights are assigned to the design values. In the first scenario the weight is distributed evenly between the design values. The following four scenarios are where the weight is assigned to one of the design values and the rest have a weight of zero. Test case 2 is to validate the algorithm used for a larger constellation that has a data transfer mission. There are 100 satellites in this constellation. The goal of this test case is to show that a significant amount of the data that is represented as requests are transferred from target to ground. Test case 3 is to validate the algorithm in EO missions. The goal of this test case is to demonstrate that the proposed algorithm can produce a schedule with as near to equal spatial coverage and temporal coverage and large data transfer amounts for each target. Test case 4 is to validate the algorithm for disaster relief mission. The goal of this test case is to show that the algorithm can create a schedule where data is transferred from target to ground quickly and that the amount of data that is able to be transferred is significant enough to encompass all the data needed by relief workers to be kept up to date on the situation. Test Case 5 is to test Mode 2 which takes in requests in real time and schedules them for the next day. This test is for future work since, at present, the algorithm is not able to be used in real-time.

4.1 Test Case # 1

Test case # 1 is to demonstrate a data transfer mission and to demonstrate the effect each design variable has on the schedule and other design variables. It is assumed all targets always have data to transfer. The first test case involves the first 10 satellites of the Iridium next satellite constellation, 5 ground stations, and 5 targets with the following coordinates:

Table 3 Test Case 1 Ground Resource Locations

ID #	Ground (Lat, Lon, Elevation)	Target (Lat, Lon, Elevation)
1	64.86, -147.85, 0	43.7735, -79.5019, 200
2	26.73, -82.03, 3	49.2827, -123.1207, 100
3	67.88, 21.07, 341	53.5444, -113.4909, 645
4	29.00, -81.00, 0	50.4452, -104.6189, 577
5	42, 13.55, 652	49.8951, -97.1384, 239

The above ground stations are modeled after the stations located around the world and are currently in use. The targets are the locations of capital cities across Canada. The Iridium constellation was chosen because it is a currently operating telecommunication constellation. The orbits and ground resources are shown below.

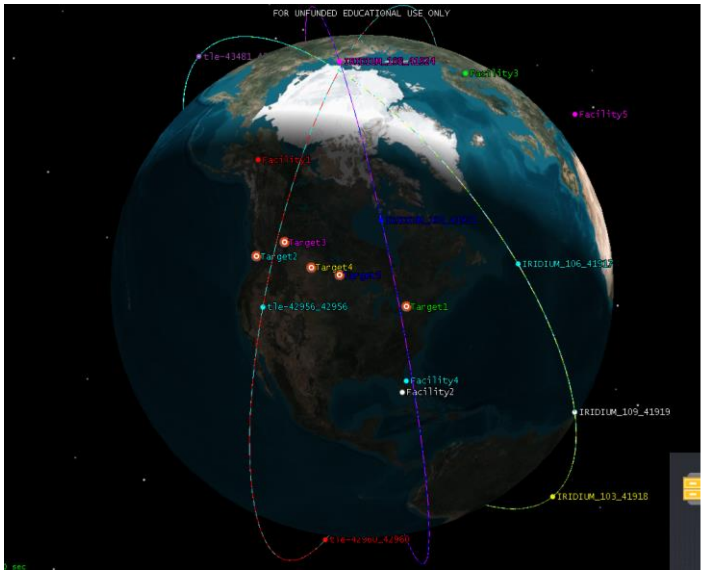

Figure 11 Test Case 1 Orbits

There are 3 satellites in the orbit furthers left and in the middle. The orbit farthest right has 4 satellites.

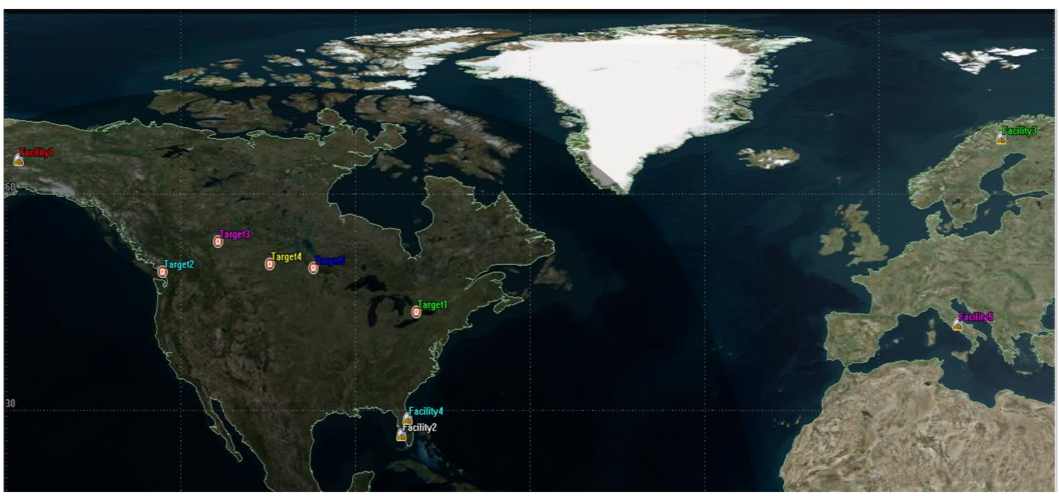

Figure 12 2D map of ground resources

The following table illustrates the remaining input variables:

Table 4 Test Case 1 Input variables and constraints

Start and End Date and Time	May 1, 2019 to May 2, 2019
Satellite Constraints	Initial Storage Capacity – 1000 GB Initial Battery Capacity – 100%
Satellite Power and Storage Drains	Payload Power Consumption – 1% Payload Storage Consumption – no payload Uplink Power Consumption – 1% Uplink Storage Consumption – 1 GB/min Downlink Power Consumption – 1% Downlink Storage Consumption – 1 GB/min Data Processing Power Consumption – 1% Data Processing Storage Consumption -1 GB/min Hibernation Power Consumption – 0% Hibernation Storage Consumption – 0 GB Power Regenerated a day – 100%

There are five scenarios used for this test case. The weights for the weighted sums optimization algorithm for each scenario are as follows:

Table 5 Test Case 1 Scenario Weighting

Scenario #	C_1Transfer Time weight	C_2 Age of Data weight	C_3 Spatial Coverage weight	C_4 Temporal Coverage weight
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1
5	0.25	0.25	0.25	0.25

The different scenarios were chosen to illustrate the impact the weight assigned to the design variables C_1 to C_4 can have on the final schedule. To validate the use of the weighted sums optimization, each scenario is tested to optimize the design variable assigned the weight of 1; thus, the resulting schedules should reflect this. If the schedules are similar, then it can be assumed that a weighted sums optimization is superfluous and that the chosen design variables have little effect on the final schedule.

4.2 Test Case # 2

For test case #2, the constellation consists of 100 satellites, 10 grounds and 8 targets whose locations are shown in table 3 below:

Table 6 Test Case 3 Ground resource locations

ID #	Ground (Lat, Lon, Elevation)	Target (Lat, Lon, Elevation)
1	64.86, -147.85, 0	43.7735, -79.5019, 200
2	26.73, -82.03, 3	49.2827, -123.1207, 100
3	67.88, 21.07, 341	53.5444, -113.4909, 645
4	29.00, -81.00, 0	50.4452, -104.6189, 577
5	42, 13.55, 652	49.8951, -97.1384, 239
6	-25.64, 28.08, 1288	46.8139 -71.2080 98
7	68.80 -133.5 51	45.965 -66.6463 17
8	-77.81 166.69 183	45.4215 -75.6972 70
9	64.80 -147.45 145	
10	-63.32 -57.9 26	

This test case represents a telecommunication constellation mission. A 3D and 2D representation of the test case is shown below:

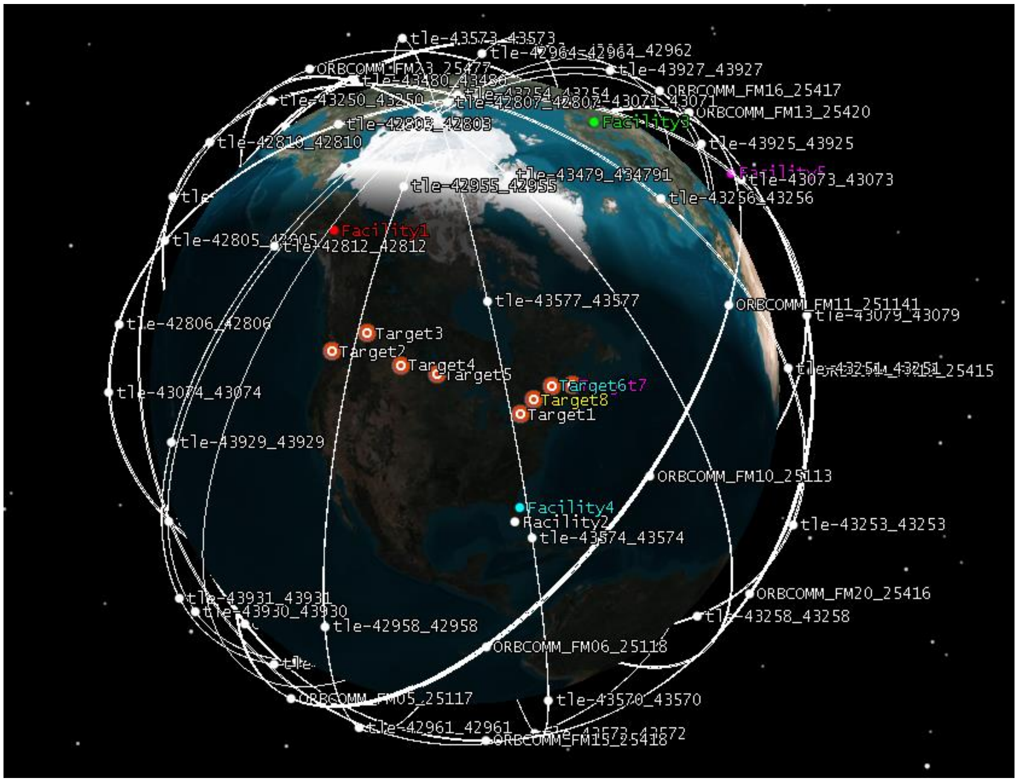

Figure 13 3D representation of space and ground resources for Test Case 2

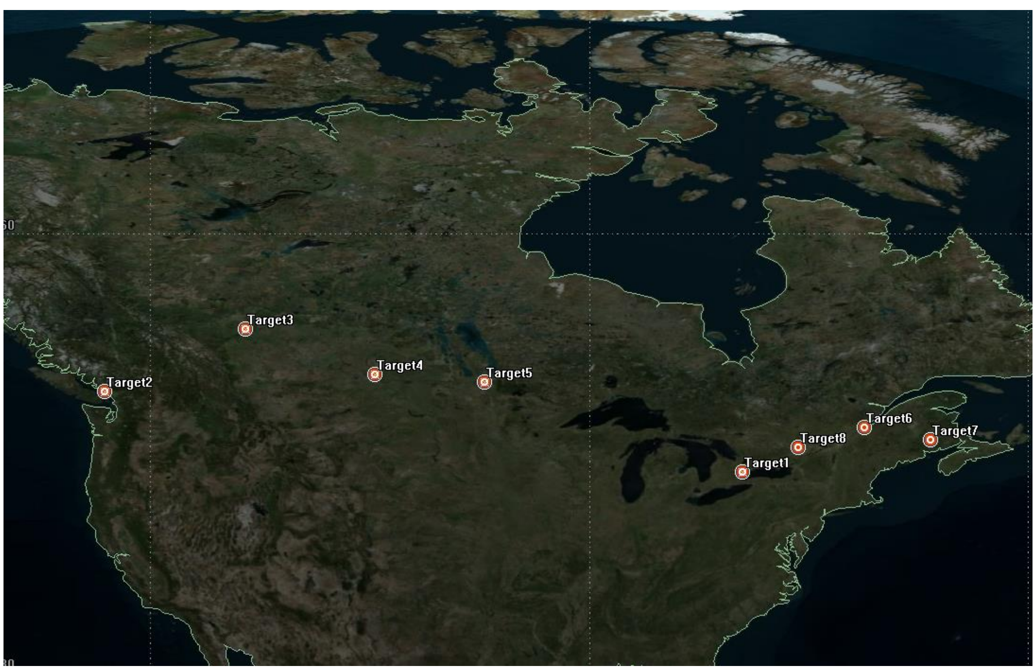

Figure 14 2D representation of ground resources for Test Case 2

Test case #2 simulates a real-world data transfer mission with limited targets to make calculation and verification efficient. It differs from Test Case #1 with more satellites involved in the test case. There is a limited number of satellites used in Test Case #1 so that the functionality is easier to perceive. In Test Case #2, it is assumed the mission is to download as much data as possible from the targets. While the number of satellites in the constellation has been vastly increased, the number of ground stations are not. Ground resources are often more limited compared to the space resources when dealing with large constellations. This test case could be for a data transferring service such as the service provided by Kepler Communications. For this Test Case maximum, the weight is placed on the amount of data transfer time to optimize the amount of data transferred. It is assumed that all satellites upload and download at the same rate. This is an extreme case, as a telecommunication constellation that has multiple users may also be interested in spatial coverage to ensure that all users receive similar services. There may also be a time constraint for the data to be transferred that would require some weight to be assigned to the age of the data. The extreme case was chosen to illustrate the differences between schedules produced for the different mission objectives. This test case illustrates the results for only one constellation. Changing the orbit of the satellites in the constellation could vastly change the results. This test case also has a limited number of targets. Further tests may include larger target numbers.

.

4.3 Test Case # 3

For test case #3, an Earth Observation (EO) mission is considered. This test case simulates a possible EO mission where it is desired that the spatial and temporal coverage are as uniform as possible to compare and study changing phenomenon such as air pollution over time.

For this mission it is assumed that the spatial and temporal coverage should be as uniform as possible to track the changes throughout the day and across all targets. The temporal coverage illustrates the short-term changes while the spatial coverage provides data for long-term changes. It is also assumed that the maximum amount of data while considering the desired spatial and temporal coverage is a key to mission success. Therefore, temporal coverage, spatial coverage and transfer time will each receive an equal weight for the weighted-sums optimization. For easier comparison, the power and storage drains are the same as in Test Case #1. The targets and grounds are as follows:

Table 7 Test Case 3 Ground resource locations

ID #	Ground (Lat, Lon, Elevation)	Target (Lat, Lon, Elevation)
1	64.86, -147.85, 0	43.7735, -79.5019, 200
2	26.73, -82.03, 3	49.2827, -123.1207, 100
3	67.88, 21.07, 341	53.5444, -113.4909, 645
4	29.00, -81.00, 0	50.4452, -104.6189, 577
5	42, 13.55, 652	49.8951, -97.1384, 239
6	-25.64, 28.08, 1288	46.8139 -71.2080 98
7	68.80 -133.5 51	45.965 -66.6463 17
8	-77.81 166.69 183	45.4215 -75.6972 70
9	64.80 -147.45 145	
10	-63.32 -57.9 26	

The satellites used for this test case are five of the BEESAT CubeSats currently in orbit. This satellite family was chosen since it is a CubeSat constellation already in orbit and would better reflect the test case mission parameters. This test case uses targets in Canada only. The ground locations are either ground stations in current use of stations that are in development.

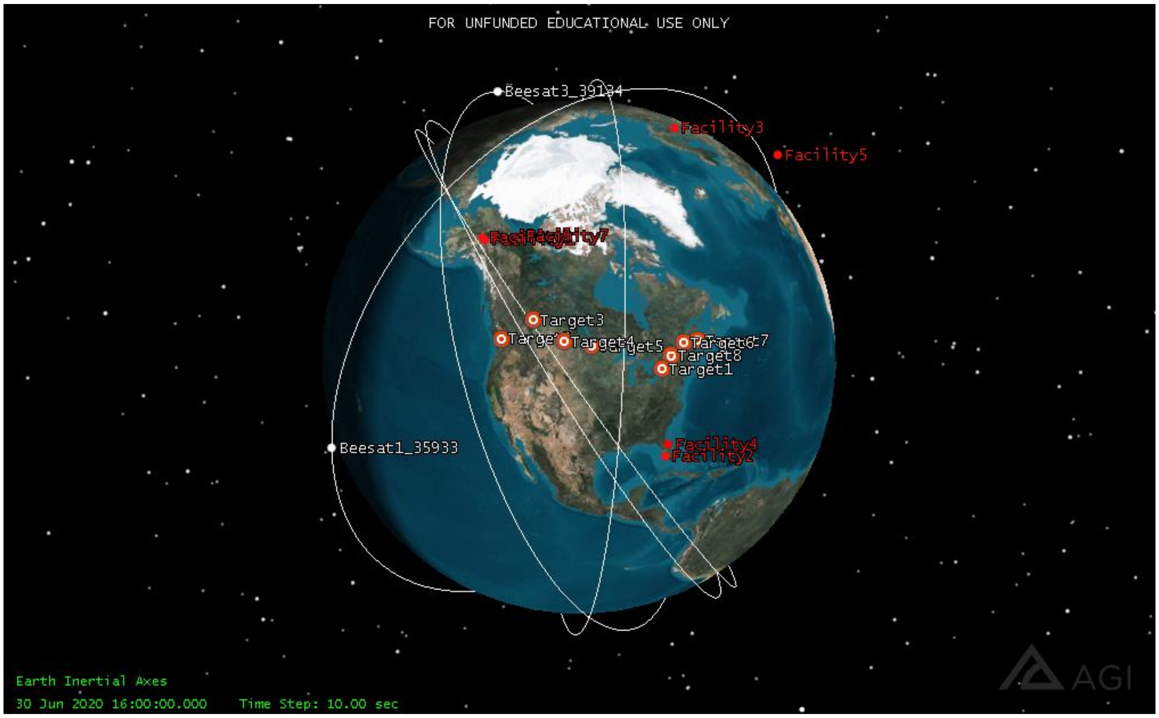

Figure 15 3D representation of Test Case 3 and 4 satellite constellation and ground resources

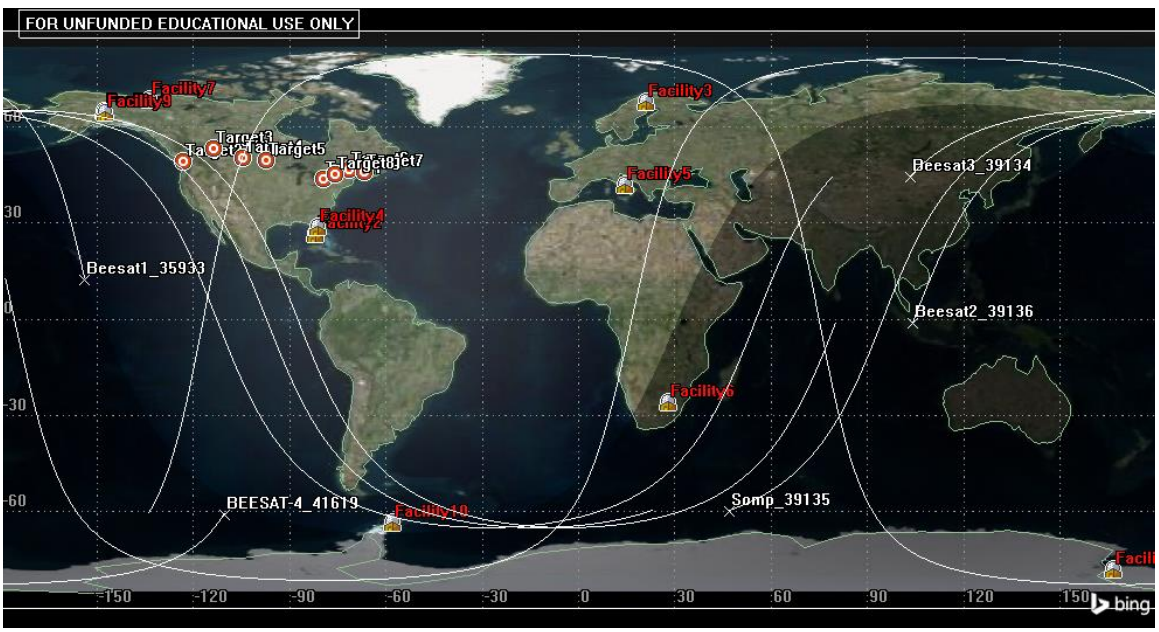

Figure 16 2D representation of ground resources for Test Case3 and 4

.

4.4 Test case # 4

This test case is designed with disaster aid relief mission in mind. It is assumed that there is a natural disaster taking place where the targets and located and that the Earth observation satellites from test case #3 have been re-purposed to aid in monitoring the natural disaster. This test case simulates the possible re-tasking of space and ground resources to aid in natural disaster relief and monitoring as proposed by ESA. In Test Case #4, age of data and amount of data are the most critical variables to optimize. To best aid disaster relief workers, the latest data is needed and therefore the age of data is minimized. The more data downlinked, the better prepared relief agencies can be and therefore the amount of data is also maximized. The weight for age of data in the test case is 75% because acquiring the newest data is assumed to be the most important objective, and 25% is assigned for amount of data. The current algorithm does not yet operate in real-time, therefore this test case is for future reference.

4.5 Test Case # 5

Test Case #5 is testing in Mode 2 while the previous 4 test cases have been in Mode 1. In Mode 2 the requests are submitted by a user and scheduled for the next day. For this case it is assumed that there are multiple users making requests to use a commercial company's space and ground resources to transfer data. The users provide three key information required for Mode 2. The ground locations, target locations and initial inputs are the same as in Test Case # 1. There were two design variables chosen to the variables: amount of data (transfer time) with a weight of 75%, and urgency with a weight of 25%. 50 requests were formulated by using a random function to answer the 4 questions and the resulting requests are shown in Appendix B. In this test case, it is assumed that all requests will use the same weight.

Chapter 5 Results

The results and analysis of all 5 Test Cases are presented below. For each Test Case the key design variables are presented graphically at each target location. For certain Test Cases tabular data is also provided to support the analysis. A common depiction for each graph has been used as follows:

 - Transfer time is represented as a solid color bar , and the transfer time scale is in total minutes per target per 24hr. A large amount of transfer time is desirable.

 - Age of data is represented as horizontal stripes patterned bar and the age of data scale is in average minutes between uplink and downlink per target per 24 hr. A low age of data is desirable.

 - Spatial coverage is represented as a checkered patterned bar, and the spatial coverage scale is in total visits per target per 24hr. A large number visits and/or uniformity in visits is desirable.

 - Temporal coverage is represented as a diagonal striped patterned bar, and the temporal coverage scale is in total visits per time slot per target per 24 hr. the 24hr day is broken up into 4 time slots. Many visits in each time slot for each target and/or uniformity across time slots for each target is desirable.

5.1 Test Case 1

Test Case 1 is a data transfer mission where the weights assigned to each design variable are varied to test the functionality of blending using weighted sums. There are 10 satellites, 5 grounds and 5 targets. There are 5 scenarios that have been analyzed in detail to test functionality of the design variables. A further 25 were then generated to produce a surface plot of the variation of design variables as a function of the assigned weights.

The table below shows the details for the 5 scenarios analyzed in detail:

Table 7 Scenario Weights

Scenario #	C_1Transfer Time weight	C_2 Age of Data weight	C_3 Spatial Coverage weight	C_4Temporal Coverage weight
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1
5	0.25	0.25	0.25	0.25

The table below shows the weights assigned to the design variables for the additional 25 scenarios:

Table 8 Additional Scenario weights

Scenario	W1	W2	W3	W4
1	0.295096	0.328081	0.045995	0.330828
2	0.406589	0.062716	0.179066	0.351629
3	0.313875	0.316295	0.051666	0.318164
4	0.401377	0.203537	0.335588	0.059498
5	0.295096	0.328081	0.045995	0.330828
6	0.406589	0.062716	0.179066	0.351629
7	0.313875	0.316295	0.051666	0.318164
8	0.401377	0.203537	0.335588	0.059498
9	0.136528	0.296432	0.256444	0.310596
10	0.264991	0.014431	0.343141	0.377436
11	0.263911	0.29463	0.28895	0.152509
12	0.418958	0.109416	0.451279	0.020346
13	0.222663	0.037125	0.0781	0.662112
14	0.348007	0.15882	0.475921	0.017252
15	0.184267	0.16025	0.321508	0.333975

16	0.105665	0.276932	0.251952	0.365451
17	0.293153	0.311882	0.11407	0.280894
18	0.456492	0.113313	0.082921	0.347275
19	0.295096	0.328081	0.045995	0.330828
20	0.406589	0.062716	0.179066	0.351629
21	0.313875	0.316295	0.051666	0.318164
22	0.401377	0.203537	0.335588	0.059498
23	0.136528	0.296432	0.256444	0.310596
24	0.264991	0.014431	0.343141	0.377436
25	0.263911	0.29463	0.28895	0.152509

Each design variable is now studied across all scenarios starting with transfer time.

5.1.1 Transfer Time

The transfer time performance for Test Case 1 is illustrated in the diagram and table below. Figure 17 illustrates the transfer time for each scenario and target.

Table 9 Scenario Results

Scenario	C_1 Transfer Time (min)*	C_2 Age of Data (min)**	C_3 Spatial (visits)***	C_4 Temporal (visits)****
1	659	325	[17,24,25,15,10]	[17,25,5,5]
2	118	1	[1,2,6,3,3]	[7,0,1,2]
3	645	57	[21,22,22,22,21]	[20,30,9,5]
4	451	44	[13,22,10,12,10]	[10,20,10,8]
5	599	40		

*Total minutes of data transfer across all satellites from targets to ground stations.

** Average (over 24 hours) time between uplink and downlink cross all satellites and targets.

*** Total scheduled visits across all satellites and targets over 24hr.

**** Total scheduled visits across all satellites and targets in each 6-hour window of the 24-hr. day.

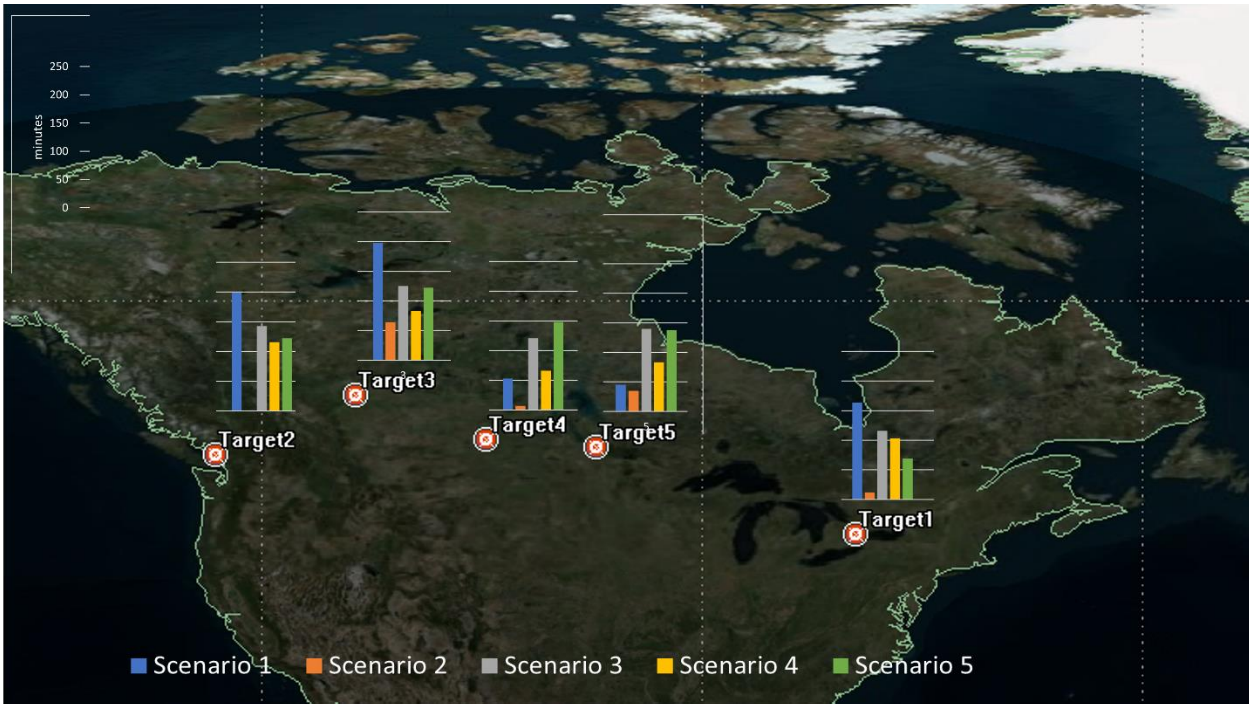

Figure 17 Transfer Time for all scenarios across all targets

For Test Case 1 the transfer time for each target is shown in figure 17 for each of the 5 scenarios. The blue bar represents scenario 1 which optimizes for transfer time and we can see that for target 1,2, and 3 it has the largest value. For target 4 and 5 scenario 1 has a lower bar than scenario 3,4 and 5. However, when totaled across all targets the blue bars for scenario 1 have the largest transfer time amount. It is 9.1% larger than scenario 5 which has the second largest amount of transfer time across all targets.

A more uniform transfer time across all targets can be seen in the grey bars, which represent scenario 3. This is expected since spatial coverage was optimized in scenario 3. The orange bars represent scenario 2 where age of data was optimized and therefore the orange bars are always the smallest since if the time between uplink and downlink is too large then the request would not be scheduled. The yellow bars, which represent scenario 4 where temporal coverage is optimized, show a more uniform transfer time across all targets compared to scenario

1, and 2 but not compared to scenario 3 or 5. It can be inferred that the satellite configuration is not as uniform in temporal coverage as it is in spatial coverage. The green bars represent scenario 5 where all design variables were given equal weight for the weighted sums optimization.

The following plot shows how the cost varies for scenario 1, where transfer time was optimized, as events are scheduled over the 24hr window.

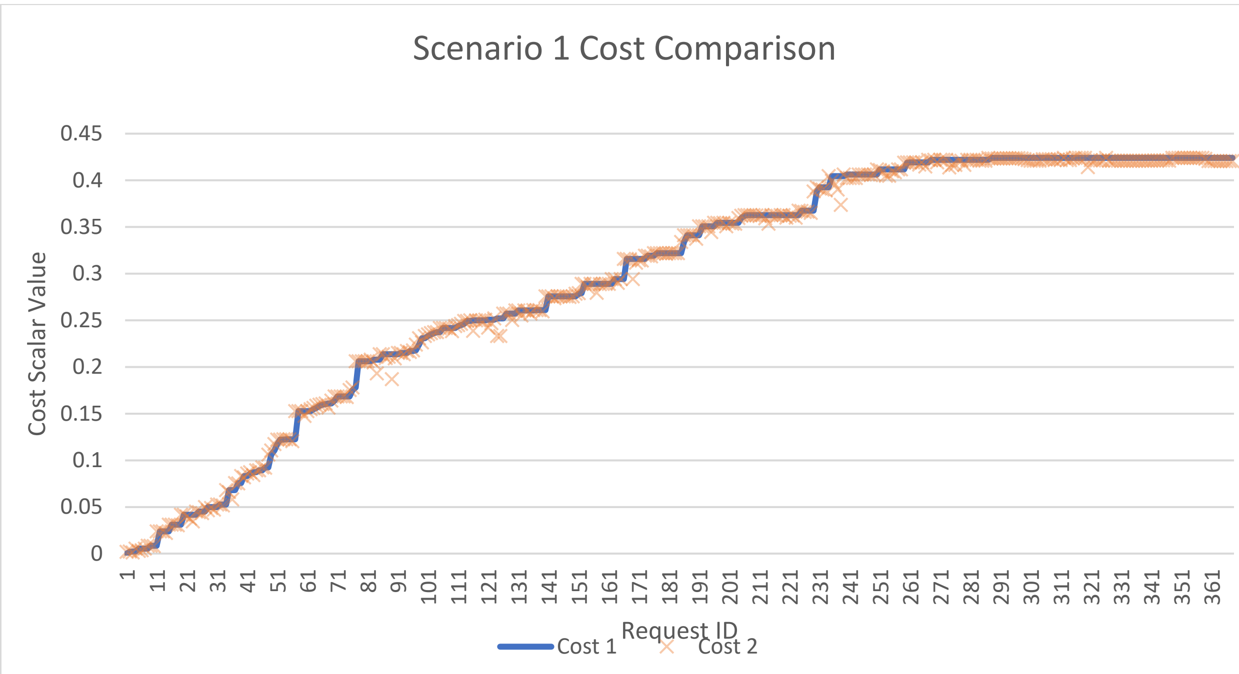

Figure 18 Cost graph for Scenario 1. Cost 1 is the cost of the approved schedule and Cost 2 cost of the next considered schedule. It can be seen that if the Cost 2 value is lower than the current Cost 1 value, the approved schedule doesn't change.

Figure 18 shows how the brute force algorithm arrives at the optimal schedule for the scenario when transfer time is optimized. The blue line represents the cost of the approved schedule, the orange x's is the cost of the schedule in question with the current request added. The x-axis shows each of 361 events that were proposed to be scheduled. The y-axis shows the transfer time cost function for the scheduled events. When the blue line remains flat over a step-in request ID this means that scheduling the given request did not increase the transfer time or scheduling the request would decrease the transfer time and therefor would not be scheduled. The figure shows the overall trend increasing and then plateauing out this means that at a certain point no additional requests improve transfer time. This is because beyond a certain point in the day there is no longer access points for downlinking the data to the ground and therefore uplinks are not scheduled either.

5.1.2 Age of Data

The age of data performances for test case 1 are illustrated in the diagrams below. Figure 19 illustrates the age of data for each scenario and target.

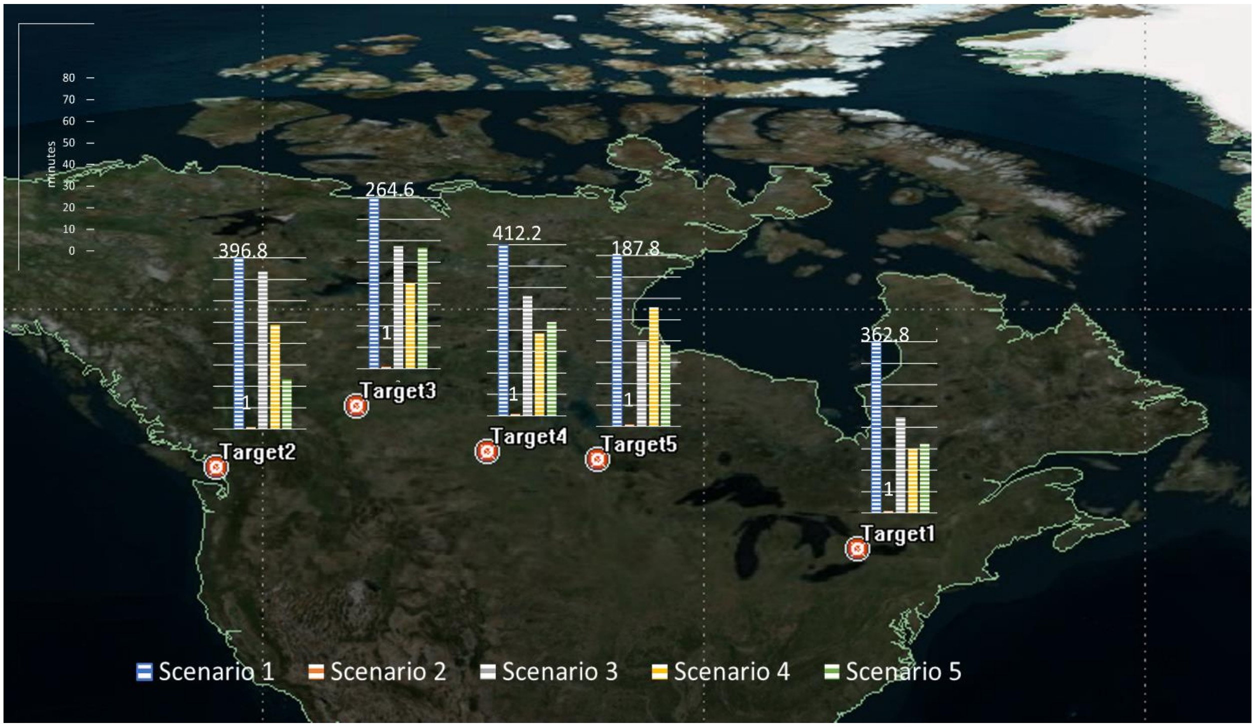

Figure 19 Age of Data for all scenarios across all Targets

The blue bar represents scenario 1 and has the highest age of data, as discussed previously. The orange bars representing scenario 2, where age of data was optimized, have the lowest age of data, much smaller than all the other scenarios. In fact, scenario 2 ensures all collected data are transmitted at their nearest downlink, resulting in ages of data of approximately one minute. Scenario 3 correspondingly has an age of data almost 60 times that of scenario 2 and scenario 4 almost 45 times as large. Scenario 5 is around 40-times worse than Scenario 2, making it second-best overall. In contrast, scenario 1 is almost 10 times worse than scenario 2, showing the very large variation in schedule possible for age of data by adjusting the parameter weights.

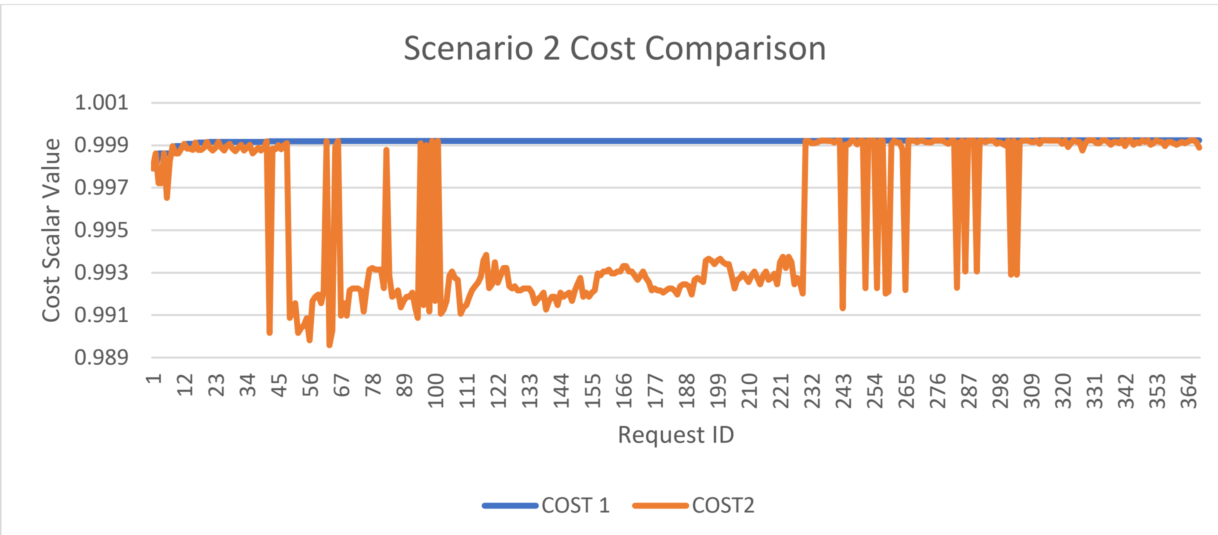

Figure 20 Cost graph for Scenario 2. Cost 1 is the value of the approved schedule and Cost 2 is the value for next considered

Figure 20 shows the cost graph for scenario 2 where age of data was optimized. The blue line represents the cost of the approved schedule, the orange line is the cost of the schedule in question with the current request added. For Test Case 1 the scenario 2 cost comparison does not show significant improvement over the majority of the scheduling window. This is due to low number of requests where the time between uplink and downlink is small. They are clustered around morning and evening passes because the target and ground are accessible in the same orbit. For Test Case 1 Scenario 2 the schedule requests saturate when there is not a more robust choice of ground access. In a scenario with a more diverse set of targets and grounds this cost function is expected to have a greater variation.

5.1.3 Spatial Coverage

The spatial coverage performances for test case 1 are illustrated in the diagrams below. Figure 21 illustrates the spatial coverage for each scenario and target.

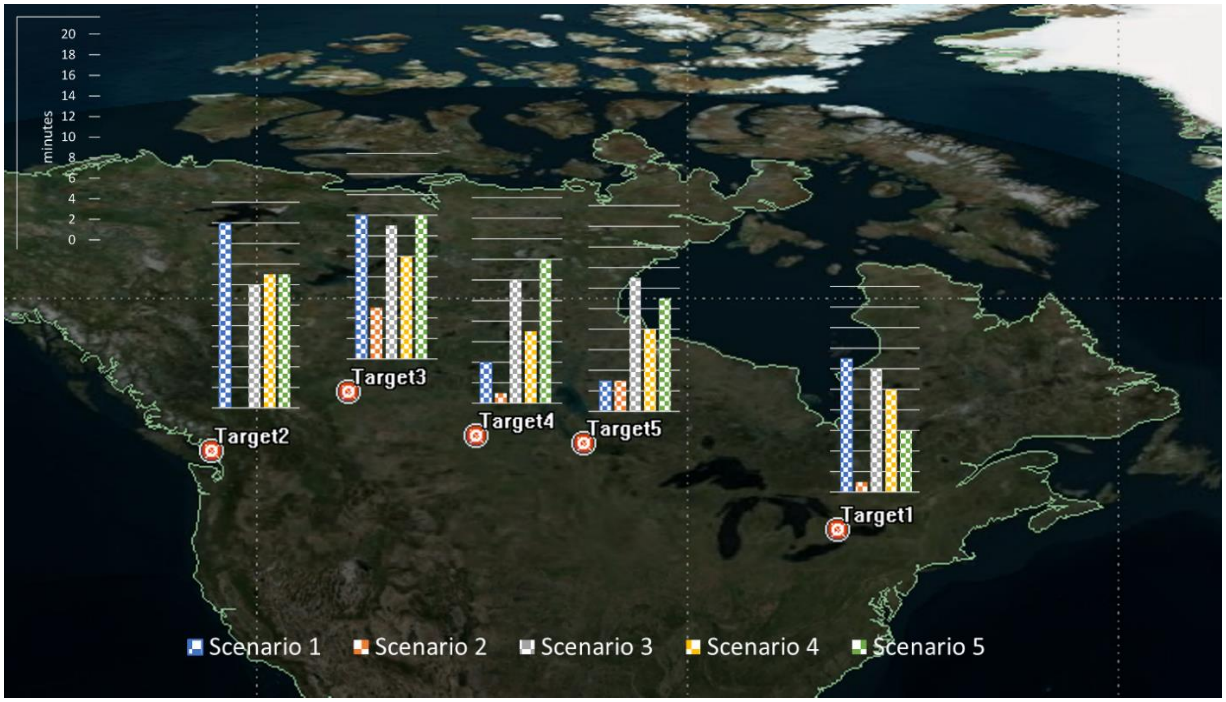

Figure 21 Spatial Coverage for all scenarios across all Targets

The spatial coverage for all scenarios is shown in the figure above. Figure 21shows that the grey bars representing scenario 3 has the most uniform coverage across all targets. This is to be expected since scenario 3 optimized spatial coverage. It also shows that scenario 5 has the most uniform coverage after scenario 3.

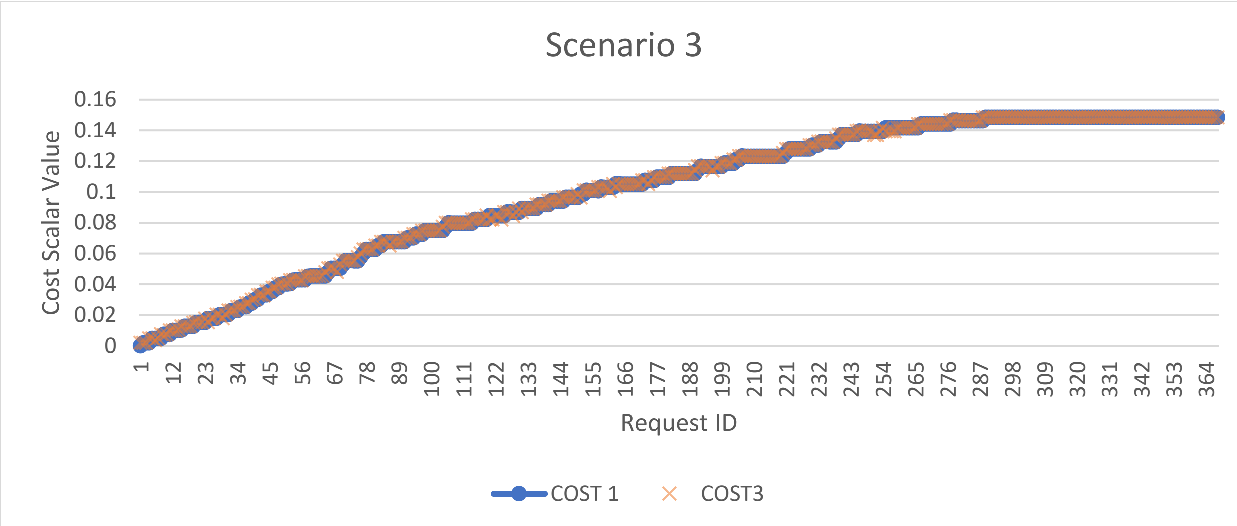

Figure 22 Cost graph for Scenario 3. Cost 1 is the value for the approved schedule. Cost 2 is the value for the next schedule considered.

Figure 22 shows the cost graph for scenario 3 where spatial coverage was optimized. The blue line represents the cost of the approved schedule, the orange line is the cost of the schedule in question with the current request added. This cost graph is similar to the cost graph for scenario 1. There is an ascending trend until it reaches a plateau where new requests are no longer added due to conflicts.

5.1.4 Temporal Coverage

The temporal coverage performances for test case 1 are illustrated in the diagrams below. Figure 23 illustrates the temporal coverage for each scenario and target.

Figure 23 Temporal Coverage for all Scenarios

It should be noted that the plots shown in Figure 23 do not follow the same colour scheme as other plots in this section. Red indicates accesses between 0h and 6h, yellow from 6h-12h, green from 12h-18h and blue from 18h-24h. Scenario 4 targeted the best uniformity in temporal coverage across all targets. It can be seen that all but 2 targets are seen in each time window, which is the same for Scenario 3 (spatial coverage uniformity), but Scenario 3 has less uniform numbers of passes per time slot. All other scenarios have at least one target that is not seen in at least two of the daily time slots, with Scenario 2 completely bypassing target 2 and only collecting in one time slot from targets 1, 3 and 4.

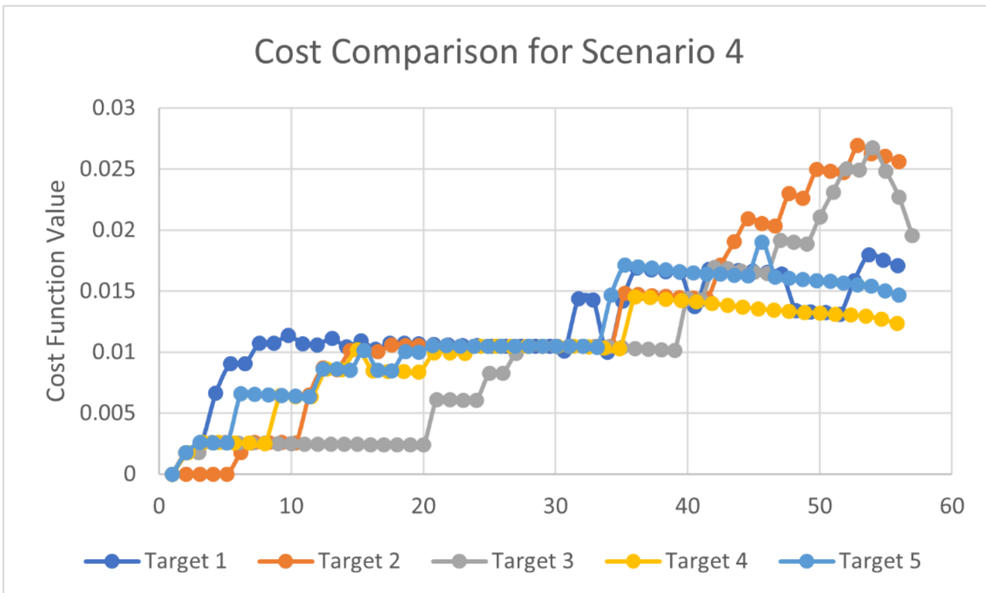

Figure 24 Cost Comparison for Scenario 4 Cost values

Figure 24 shows the cost graph for scenario 4 where temporal coverage was optimized. Temporal coverage is calculated by target while the other 3 design variables are calculated across all targets. This means that a request from satellite 1 and Target A might require a conflict resolution with satellite 1 and target B. If the request between satellite 1 and Target A produces a better scheduled, then the conflict resolution would require that the request scheduled between satellite 1 and target B be erased so that the request between satellite 1 and target A can be scheduled. The next time the cost function is run on target 1 the cost function would be different than anticipated due to this conflict resolution. This explains the more scattered trend of the cost comparison graph for scenario 4. Overall, the trend for each target is increasing with each request and the absolute value for cost functions for each target are similar. While target 3 had a higher cost value at the 52end step, target 1 only increases when target 3 loses some access as we see in the 55th step. Therefore, the drop in cost function value at the end for each target is due to the algorithm selecting the schedule that produces the highest but also closest cost function value for each target.

Scenario 5 blends the cost functions of the design variables equally. Because scenario 5 includes time optimization, it is useful to compare cost functions across targets, as discussed for scenario 4 above.

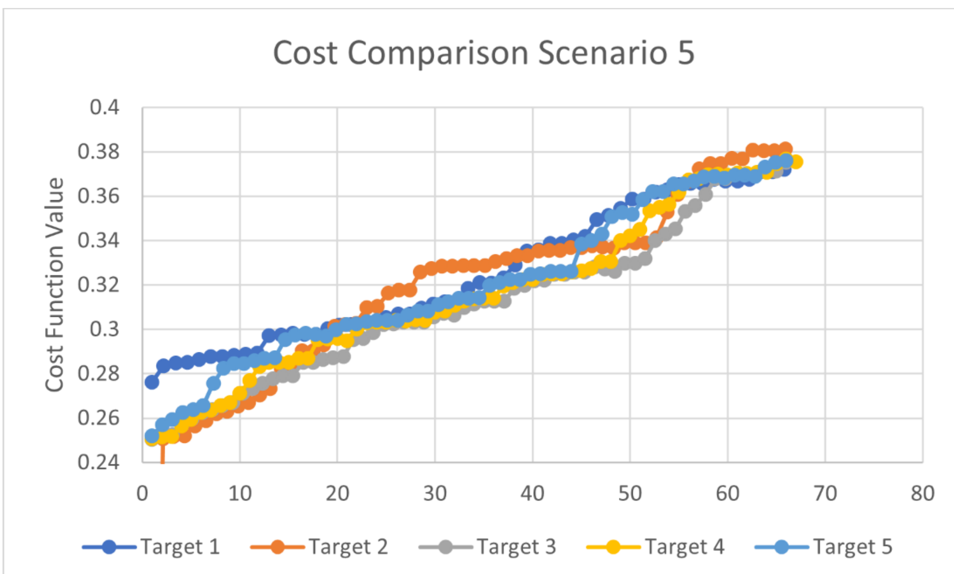

Figure 25 Cost Comparison for Scenario 5

Figure 25 shows the cost graph for scenario 5. The trend for all targets is increasing, the values for each target remain close, and there is no plateau at the end of the graph. It can be inferred that the schedule could take on more requests.

The results of test case 1 have shown how varying the weighting of each of design variable does result in a significantly different resulting schedule. The variation in transfer time between schedules is a minimum of blank for scenario 2 and a maximum of blank for scenario 1. The variation in age of data is a minimum of bank in scenario 2 and a maximum of blank in scenario 1. The variation in uniformity of spatial coverage has been shown to vary between having all targets seen approximately equal amounts to having some targets not being seen at all. The variation in uniformity of temporal coverage has been shown to vary between having all time slots seen approximately equal amounts for all targets to having multiple targets not seen in a given time slot.

Having demonstrated the utility of targeting each design variable for different mission objectives and having looked at a simple equal blending of design variable cost functions, a further 25 variations of weighting of cost function were also looked at, to better understand the shape of cost function variation by blended weight.

By blending the design variables with different weights, it is possible to study how the schedule design space varies for scenario 1. The following figures illustrates the variation as a surface plot.

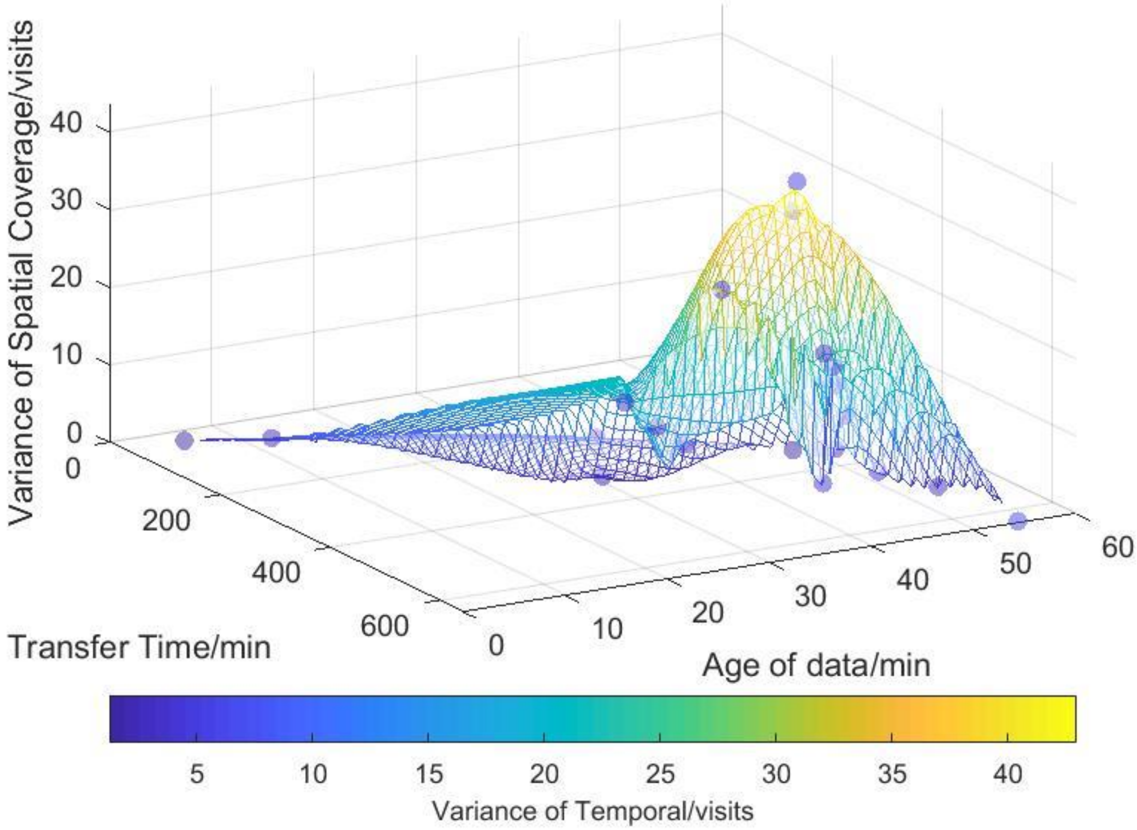

Figure 26 Variation of design variables as a function of each other

Figure 26 shows the variation of design variables as a function of each other. The Transfer time is on the x–axis, Age of data is on the y-axis, the variance of the spatial coverage (across all targets) is on the z-axis, and the colour bar represents the variance of the temporal coverage. As can be seen, the spatial and temporal coverage have optimal range of transfer time and age of data where they have the best results. Too little transfer time does not result in a uniform spatial or temporal coverage and too much transfer time has the same effect. Transfer time is correlate with age of data (a large transfer time causes a large age of data as well).

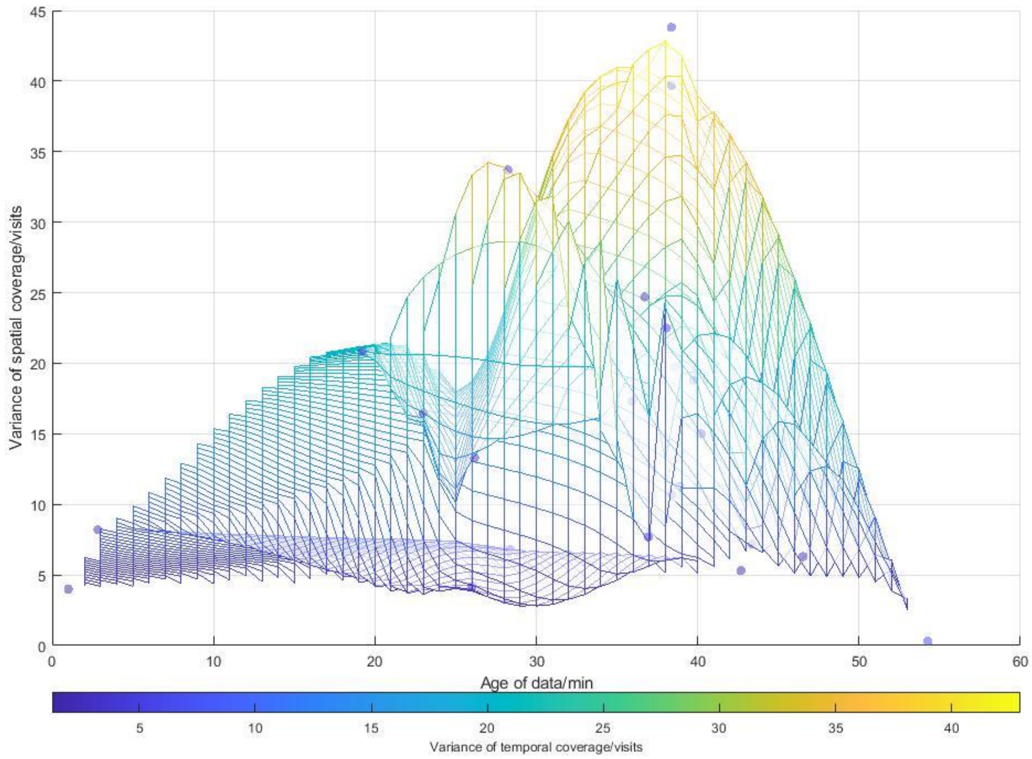

Figure 27 Variation of design variables along the x-axis

In figure 27 we are viewing along the x-axis, it can be seen that there is a strong correlation of the variation of the vertical axis (Variance of spatial coverage) and the colour axis (variance of temporal data). It has been seen in earlier data that temporal and spatial coverage are coupled.

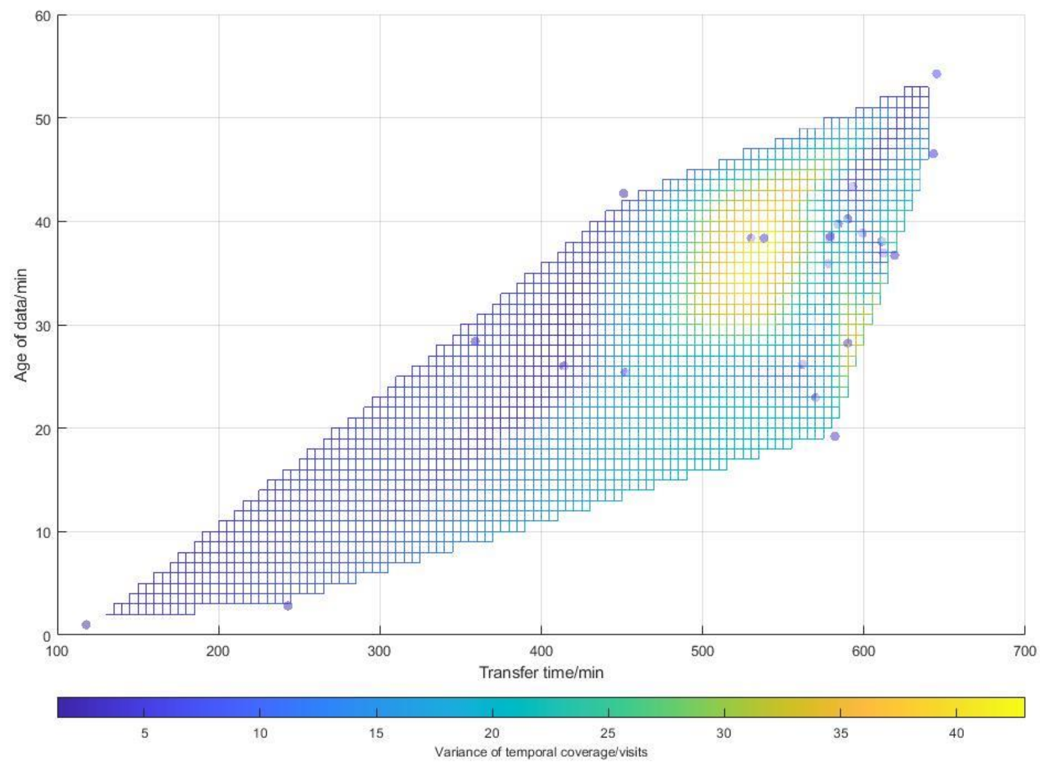

Figure 28 Variation of design variables along the z-axis

Figure 28 is viewing along the z-axis or from above. It can be seen that as transfer time increase so does age of data.

Knowing the form of this surface for a given scenario, we can select weights that will place the schedule anywhere on this surface.

5.2 Test Case 2

Test Case #2 is an example of a possible telecommunications constellation CubeSat mission. Simulation for this test case takes 4 to 6 hours to run the most time-consuming part of the algorithm is the sgp4 which is the orbit propagation section, due to the number of satellites being simulated. The results are summarized in Table 10:

Table 10 Test Case 2 Results

Target ID	Transfer Time (min)	Age of Data (min)	Spatial Coverage (visits)	Temporal Coverage (visits)
1	624	254.9623	53	[15,14,16,9]
2	663	325.8571	56	[21,15,13,10]
3	634	289.4615	52	[16,15,16,8]
4	760	304.4107	56	[18,16,16,8]
5	537	266	48	[14,11,15,9]
6	659	246.386	57	[20,17,13,8]
7	725	281.6515	66	[20,20,18,20]
8	503	255.6857	35	[7,10,12,6]

The total transfer time for Test Case 2 is 5105 minutes with an average age of data is 278 minutes. There are an average of 53 visits to a target. Target 8 is the target visited the least at 35 visits and target 7 the most at 66 visits. The last time slot in temporal coverage is the least visited for all targets expect target 7. Target 7 also has the most uniform temporal coverage.

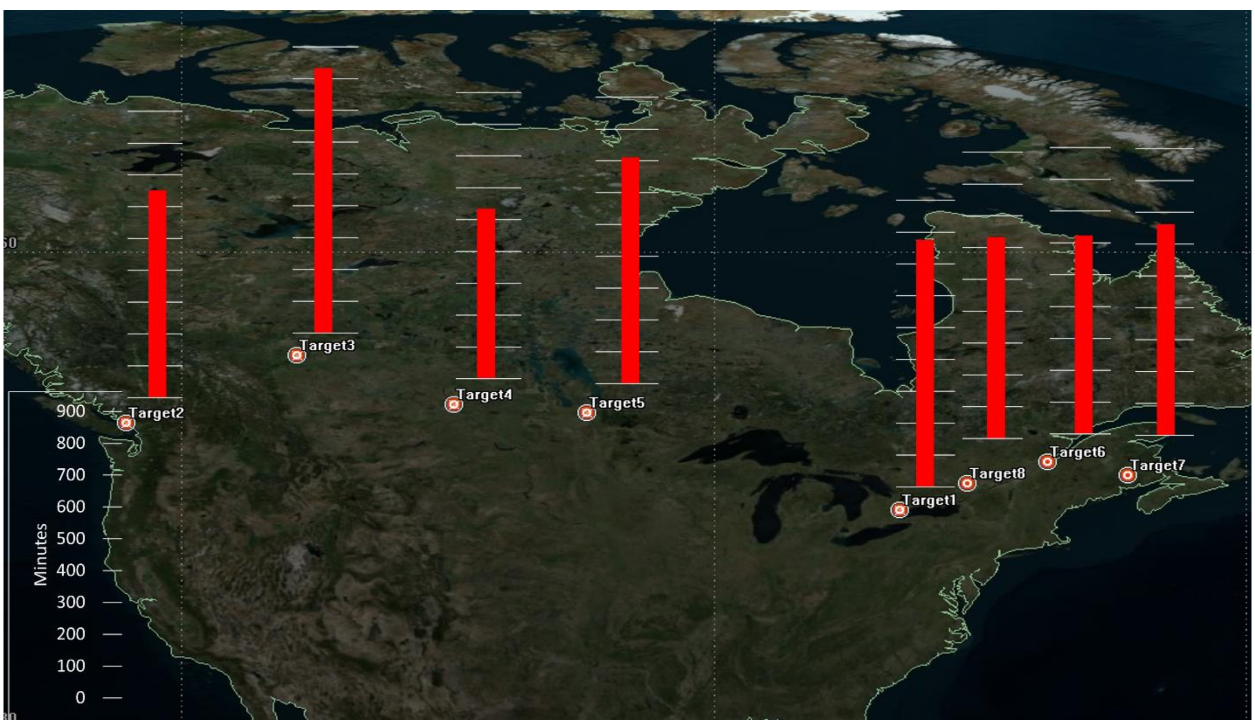

Figure 29 Transfer Time for Test Case 2

If each target was to be monitored 24 hours a day, there would be 1440 minutes of transfer time per target for a total of 11520 minutes across all 8 targets. The scheduling algorithm achieves approximately 44 % of this number. This can be explained when studying the ratio of satellites seen by targets and satellites seen by ground stations. In a typical minute of the schedule a target can be seen by 4 satellites and a ground station can be seen by 8. There are more ground stations than targets (10 vs 8) therefore, if 100% of the ground station time were used the maximum target time that could be collected would be

$$\frac{((4\,Satellites) * (8\,Targets))}{(8\,Satellites * 10\,Grounds)} = \sim 44\ \% \qquad \textit{Eq. 110}$$

Therefore, the algorithm is scheduling as much transfer time as the constellation configuration allows.

5.3 Test Case 3

Test Case #3 is an example of an Earth Observation mission, with the results summarized in Table 11.

Table 11 Test Case 4 Results

Satellite ID #	Transfer Time (min)	Age of Data (min)	Spatial Coverage	Temporal Coverage
1	59	129.4	[0,4,1,0,2,0,0,0]	[3,2,1,1]
2	62	9.1	[1,0,0,0,1,1,2,2]	[2,4,1,0]
3	62	8.4	[0,0,0,0,0,2,4,1]	[3,3,1,0]
4	62	88.7	[2,1,1,1,2,1,0,1]	[3,1,3,2]
5	55	106.6	[1,0,1,1,0,1,1,3]	[0,4,3,1]
Total/Average	300	68.5	[4,5,3,2,5,5,7,7]	[11,14,9,4]

As seen in Table 10 above the temporal and spatial coverage has an average of 4.75 visits per target. The largest deviation is being 2.75 visits. Temporal coverage has less uniformity than spatial coverage with the largest deviation being 10. The number of visits in the first two time slots is 25 while there are only 13 visits in the last two time slots. If the Earth Observation mission needs more uniformity in temporal coverage than the weight for temporal coverage needs to be increased. The following figures illustrate the results.

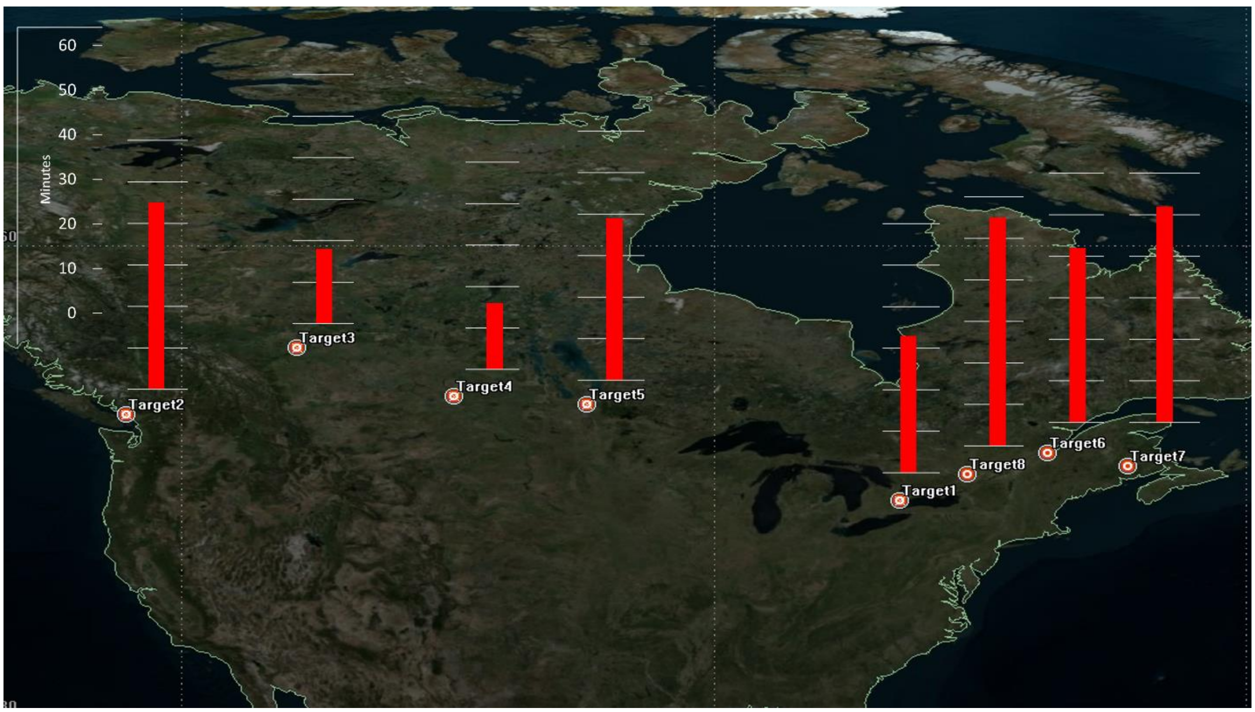

Figure 30 Transfer Time for Test Case 3

Figure 30 shows the transfer time for test case 3. Whilst transfer time has previously been considered as the time available for uploading data from a target that can still be downlinked in a subsequent ground station pass within the 24h window, it can also be considered as the time a payload (optical imaging camera for example) has available to collect data *over* a target region and still downlink that data within the 24h window. Transfer time was given 1/3 of the weight along with temporal and spatial coverage. There are only 5 satellites for test case 3 so the transfer time is less than the transfer times seen in test case 1 and 2. All targets are seen with target 3 and target 4 having the least amount of transfer time and target 8 and 7 having the most amount of transfer time. While the variation in transfer time is a factor of 2-3 between targets, all targets have access to at least one downlink and uplink in the 24hr window.

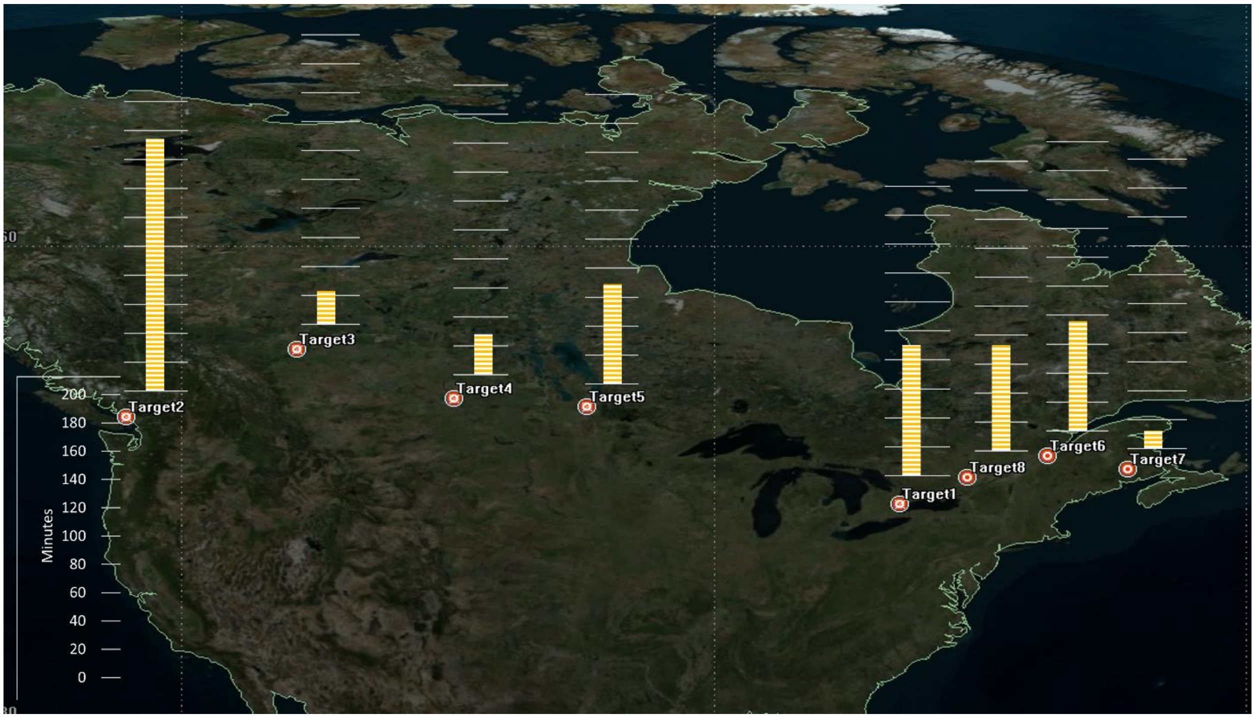

Figure 31 Age of Data for Test Case 3

Figure 31 shows the age of data for test case 3. Age of data was not optimized in test case 3. Figure 31 shows that targets 7 has the lowest age of data and target 2 has highest age of data. The average age of data 68.5 minutes. The larger age of data from target 2 is consistent with figure 17 for transfer time; western targets have less access to ground therefore, while target 2 had x amount of transfer time the age of data suffered as a consequence.

Figure 32 Spatial Coverage for Test Case 3

Figure 32 shows the spatial coverage for test Case 3. Spatial coverage was given 1/3 of the weight. All targets were seen and targets in the east have more visits then targets in the west. Target 4 has the least amount with 2 visits and target 7 and target 8 have the highest amount with 7 visits each. While target 4 only had approximately 1/3 of the visits of eastern targets, all targets were seen more than once in the 24hr period.

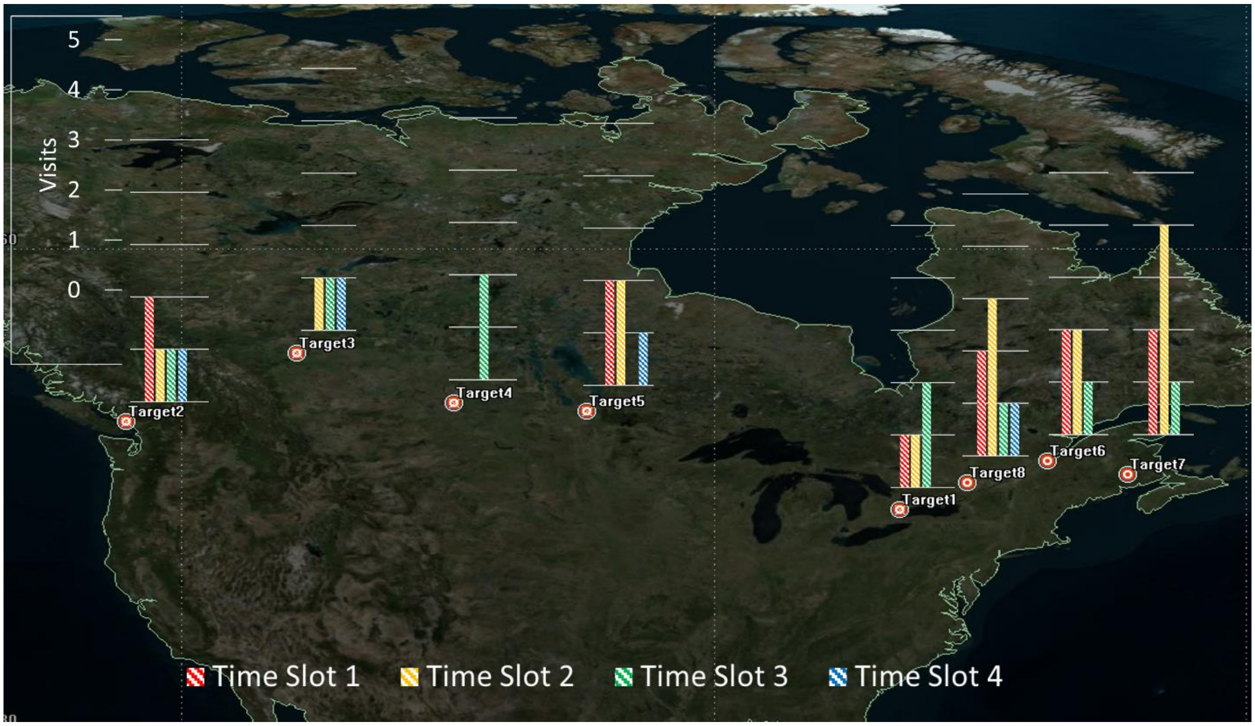

Figure 33 Temporal Coverage for Test Case 3

Figure 33 shows the temporal coverage for test case 3. Temporal coverage was given 1/3 of the weight. Target 2 and target 8 have visits in all 4 time slots while target 3,5,1,6 and 7 have visits in 3 out of the 4 time slots and target 4 has visits in only 1 of the 4 time slots. This figure re-enforces the previous plots that showed limitations to western targets with target 4 most severely impacted. All other targets had access during 3 of the 4 time slots.

Test Case 3 was an Earth Observation mission that optimized transfer time, temporal coverage, and spatial coverage. Each design variable was given a weight of 1/3. Age of data was not optimized in this Test Case since Earth observation missions do not always need the data to be rapidly refreshed. Whilst exceptions such as Meteorological Earth observation missions exist, the scenario provides relevant performance data for several types of Earth Observation mission.

5.4 Test Case 4

Test case 4 is an example of re-tasking existing satellites for disaster aid. The results are listed in Table 12.

Table 12 Test Case 4 Results

Satellite ID #	Transfer Time (min)	Age of Data (min)	Spatial Coverage	Temporal Coverage
1	77	11.2857	[0,1,1,0,0,1,4,0]	[1,1,2,3]
2	62	3.166667	[1,0,1,0,1,1,2,0]	[2,3,1,0]
3	40	2.75	[0,0,0,0,0,4,0,0]	[2,2,0,0]
4	34	1	[0,0,0,0,0,4,0,0]	[3,0,1,0]
5	31	1.5	[2,0,0,0,0,0,1,1]	[0,0,1,3]
Total/ Average	244	3.9	[3,1,2,0,1,10,7,1]	[8,6,5,6]

Table 12 shows that the average age of data is 3.9 minutes, and the transfer time is an acceptable value. For this case 75% of the weight was assigned to age of data and 25% of the weight was assigned to transfer time. Depending on the type of natural disaster and the aid relief requirements, variations of weight distribution would be applicable. If it were not acceptable increasing the age of data's weight for the cost weighted-sums optimization would produce a more acceptable schedule. The spatial coverage and temporal coverage are not uniform; but all targets are visited which would provide a good coverage of data. The results show that the algorithm could be used in a disaster for monitoring and to aid emergency services.

5.5 Test Case 5

Test cast #5 is test Mode 2 of the algorithm and the scenario is assumed to be for a commercial company that transfers user data from the user terminal to any ground station which then uploads it to a database for the users to access. It is a delay tolerant transfer platform. The results are summarized in the table below:

Table 13 Test Case 5 Results

Satellite	C_1 Transfer Time(min)	C_2 Age of Data(min)	C_3 Spatial [T1…T5]	C_4 Temporal [T1…T5]
1	82	495	[3,0,2,1,0]	[2,1,2,1]
2	74	60	[0,1,2,2,1]	[1,3,0,2]
3	82	183	[1,1,2,1,1]	[1,3,0,2]
4	46	64	[2,0,1,0,1]	[1,2,0,1]
5	61	185	[0,2,1,1,1]	[1,3,0,1]
6	48	266	[0,0,1,0,3]	[1,1,0,2]
7	72	287	[1,3,0,1,0]	[1,1,1,2]
8	58	116	[1,1,1,2,0]	[0,1,1,3]
9	19	43	[0,0,0,1,1]	[0,0,0,2]
10	54	160	[1,0,0,1,2]	[0,1,1,2]
Total	596	186	[9,8,10,10,10]	[8,16,5,18]

47 out of 50 of the results were scheduled. One of the requests had a start time of minute 1440 which is at the end of the 24-hour scheduling window and therefore could not be scheduled, the other two requests did not fit into the schedule. This means that the two requests had conflicts at all transfer times and that the other request had a higher urgency or had more data to transfer. Age of data was not a design variable with weight for this test case and therefore the average age of data for each satellite has a large discrepancy, with the largest value being 495 minutes. For this test case there was no restrictions on age of data and therefore the average age of data of 186 across all satellites is acceptable. The spatial and temporal coverage was affected by the start time and target ID of each request. The spatial coverage is dependent on the requests entered and therefore was not assigned a weight for this test case. The temporal coverage was affected by both the requests entered and the temporal coverage of the possible transfer times and therefore the large difference between temporal time slots is acceptable.

Chapter 6 Results Summary

Having run through the five test cases it is seen that the scheduling algorithm produces a constellation schedule that targets and optimizes a combination of the chosen design variables. Test Case 1 has given the space to explore how the scheduler reacts to different weights. The two schedules that produce the most data through put are scenarios 1 and 5. In scenario 1 all the weight is on transfer time and in scenario 5, ¼ of the weight is on transfer time. A constellation operated would gain 9.1% more throughput by focusing only on transfer time, which for missions with a large data collection and where age of data is less critical to mission success, this is a large increase in data throughput. The average age of data increases from 40 min to 272 from scenario 5 to scenario 1. This increase would suggest that an operator may choose to trade the 9.1 % amount of additional transfer time against a much higher average age of data.

The two schedules with the smallest average age of data are for scenario 2 and 5. For applications that require real time data transfer having the full weight on age of data ensures that the data is transferred in the same pass as it was uploaded (average age of data of 1 minute). Balancing the objectives in scenario 5 results with the data being downlinked with in the same orbit as when it was uploaded (average age of data of 40 min). The real time objective of scenario 2 limits data transfer to only 20% of scenario 5's transfer time.

Scenario 3 provides the most uniform access to all targets with the second highest transfer time (645 minutes) and an age of data (57 minutes) that is not significantly higher than scenario 5 (40 minutes). For an operator servicing all targets uniformly scenario 3 performs well on all metrics other than temporal coverage which has a high variation across time slots. Scenario 4 trades transfer time for more uniformity in temporal access. By analyzing a further 24 values for design variable weights, it was possible to generate a 3D surface plot with a colour bar for the 4th variable, to illustrate how the design variables vary as a function of each other for scenario 1. Adjusting the weights allows for there to be a schedule produced that corresponds to any point of that surface. The surface will be different for any scenario. With this algorithm an operator can generate the surface and select the schedule from the points on that surface.

In the mission specific test cases the algorithm provides an appropriate scheduling output of the given mission objectives. In Test Case #2, which is optimizing the transfer time for a large constellation, the average downlink time for a 100-satellite constellation is about an hour per satellite per day for satellites in an orbit just below 800 km to ten ground stations. This is approximately equivalent to daily access time of any one satellite to one ground station even though there is a ten to 1 ratio of satellite to ground station.

In Test Case #3, which is an EO mission, a blend of transfer time, spatial and temporal coverage are shown, with the goal of uniformity of coverage in space and time, but with quantity of data also prioritized. In this scenario, the most Easterly Target 7 is prioritized over the nearby targets 6 and 8 and target 4 is losing out to target 5 similarly, due to the relative placement of targets and ground stations, however all targets are relatively uniformly covered. This scenario could illustrate the potential for future work to better optimize ground station locations using this algorithm.

In Test Case #4, which is the disaster relief re-tasking mission, the average age of data is 20 minutes with target 2 and 3 being prioritized and the first temporal time shift also being prioritized. In this mission 25% of the weight was given to transfer time. Since the age of data is 20 minutes it could be advisable to re-assign the 25% either to age of data or spatial coverage to better optimize the schedule for the mission. In this test case there were 10 targets across Canada used in the scenario. It might be more realistic for a disaster recovery situation to have the targets centered in a smaller area, which could change the performance of the algorithm.

In Test Case #5, 47 out of 50 requests were scheduled. One request started at minute 1440, which is the end of the schedule, and therefore could not be scheduled. Two others could not be scheduled due to conflicts and would be saved to be first scheduled the next day. This scenario was intended to illustrate the algorithm flow for real-time requests, which has been demonstrated successfully, and for the given requests and target/downlink locations, the scenario was not saturated. Future work will look at real-time request prioritization in the case of higher saturation of requests.

It has been shown that the scheduling algorithm is able to generate varying schedules depending on the weights assigned to design variables. The variation is the schedules is significant to constellation operators. Having a scheduling algorithm that allows the operator to re-prioritize at a moments notice allows the operator to address emergency response situations, segment failures in the system, and support future growth.

Chapter 7 Conclusion

SmallSats have historically been used by research organizations and universities to test theories and new technologies. The industry is now experiencing a trend of increased nanosatellite implementation by commercial companies as well as the continued use by universities, research organizations, and EO missions. This trend comes from the size and cost reduction of COTS parts and a trend towards swarm technology. Instead of using one large satellite to complete a mission objective a constellation of smaller satellites will be used. These smaller satellites can allow for larger coverage without increases in budget and allow for multiple objectives to be accomplished at the same time.

Scheduling is an important problem for many missions, but its importance increases for satellite constellation missions. This is because the amount of access options and conflicts increases with the number of satellites using the same ground resources (ground stations and targets). Traditional methods of an operator deciding the outcome of every conflict individually is no longer feasible. Scheduling becomes key to constellation mission automation. Satellite constellation scheduling can be considered a Multi-Objective Optimization problem. Other papers discuss the optimization of one or two design variables but the algorithm in this thesis considered a range of missions and therefore, allows the algorithm to be considered a more generic constellation planning tool rather than an optimization method to solve a specific problem.

In the test cases in chapter 4 it is discussed what types of missions the algorithm has been designed to handle. These include earth observation missions, telecommunication mission specifically for a delay tolerant network, and disaster aid. In chapter 5 the results from the tests cases in chapter 4 are summarized and it is shown that the algorithm produces optimized schedules for the different types of missions mentioned above. The algorithm has demonstrated that an operator that is focused on transfer time can achieve an improvement of 9.1% over a mixed operation where all are weighted equally. Similarly, a mission objective of minimizing age of data can produce a schedule that transfers data in near real time at the cost of a loss of 80% of transfer time. Operators seeking more uniformity in temporal coverage or spatial coverage can achieve this with corresponding degradation in the other design variables. The operator's objective can change at any point during the mission to support contingency situations allowing them to target specific service commitments to the users.

In this thesis the goal was to design an algorithm that can optimize the schedule for a larger constellation of satellites with limited ground stations that focused on data latency and target coverage. The goal was to produce an optimization method using a weight sums optimization technique to blend the different goals (design variables). The thesis has demonstrated the algorithm's success using test case designed to simulate possible missions because the larger CubeSat missions that the algorithm was designed for are not in orbit yet. The thesis has verified the algorithm's success to differentiation between goals, by varying the weights assigned to the designed variables, and to blend the goals, by assigning equal weight to all design variables in a test case. This algorithm is unique, since the constellations it is designed to aid are currently in being designed and researched now. Its design was inspired but the work of others, but the algorithm can be used for a wider range of missions due to the weight-sum optimization's ability to blend design values and the algorithm includes the satellites data and energy constraints.

Such an approach is not ideal for *all* constellation missions. It would not be appropriate for mission data routing for real time communications, nor for other types of data routing-based optimization. For constellations with much more downlink capacity than uplink the weight of objectives becomes unnecessary and for missions with more complex onboard constraints the onboard constraints may drive the schedule more than the objectives.

Chapter 8 Future Work

The algorithm presented above has some limitations that should be addressed in future work. The algorithm currently does not allow for varying uplink and downlink rates and assumes all space craft and ground stations are identical. The algorithm presented in this thesis is meant to help implement an autonomous constellation mission but cannot currently be implemented in real-time. Future work would entail designed the algorithm to be implement in real-time and therefore be designed to be implemented by the satellites in orbit instead of on the ground. This would also mean the further testing would be needed to show that every satellite is optimizing its own tasking by sharing a minimal set of information with the other satellites. In addition, should include satellite mode, slew maneuvers, available resources, and contingency cases. Future work, for Mode 1, would be to take the completed schedule and break it down into the necessary satellite commands needed to accomplish the schedule. The commands would include satellite slew maneuvers so that the satellite is always nadir pointing, payload operation, data transmissions, schedule uploading transmissions and satellite status updates. For Mode 2 it would be the same as Mode 1 plus working towards onboard computation for same day scheduling in real time.

Currently the algorithm uses brute force and therefore, the algorithm computation time has not been optimized. A large amount of the computation time is the orbit propagation. The part of the algorithm should be examined to reduce the computation time. If there was a larger number of targets the scheduling time would blow up. Future work should include a study on ways to decrease the number of requests that go through the brute force method and a simplification of conflict resolution.

The algorithm could also have additional uses that could be studied. The algorithm could be used as a tool to select optimal locations for ground and space resources but would needs some restructuring to do so. The algorithm could also be used to identify the right weights for a given set of objectives or be used to identify weights so that an inputted minimum value for the design variables is reached. The algorithm was designed to test the effectiveness of blending the chosen design variables using a weighted sums optimization algorithm and found that blending produced schedules that were unique to the weights chosen. The algorithm should now be designed to find the optical solution based on mission goals.

Chapter 9APPENDIX

9.1 A

The following shows the format of the input file:

[Ground 1 Lat, Ground 1 Lon, Ground 1 Elev., Ground2 2 Lat...]
[Target 1 Lat, Target 1 Lon, Target 1 Elev., Target 2 Lat...]
TLESAT (Location of TLE files)
Example
C:\Users\admin\Desktop\Testingsept\testingnew\testing2\testing\OrbpropLoop1/100.tle
.
.
.
TLEEND
STARTDATE
DD-MON-YYY 00:00:00
ENDDATE
DD-MON-YYYY 00:00:00
WEIGHTS
W1
W2
W3
W4
POWERDRAINS
Payload Power Consumption value
Payload Storage Consumption value
Uplink Power Consumption value
Uplink Storage Consumption value
Downlink Power Consumption value
Downlink Storage Consumption value
Data Processing Power Consumption value
Data Processing Storage Consumption value
Hibernation Power Consumption value
Hibernation Storage Consumption value
Power Regenerated a day value

9.2 B

Q1 – Target ID

Q2 – Start time (in seconds)

Q3 – File size (in Gb)

Q4 - Urgency

Q1	Q2	Q3	Q4
2	17	0.1	1
3	25	0.6	1
3	45	0.2	3
3	64	0.6	1
1	90	0.2	3
5	153	1	3
4	183	0.5	2
3	202	0.1	2
2	232	0.5	3
3	348	1	1
3	370	0.4	2
4	375	0.7	1
3	459	1	2
3	462	0.2	2
5	469	0.7	3
5	502	0.3	2
4	519	0.3	2
1	522	0.3	2
4	566	0.2	3
4	649	1	1
1	663	0.7	2
2	684	0.3	3
2	694	0.3	1
1	710	0.2	2
4	729	0.4	3
1	789	0.9	3
1	883	0.6	3
4	904	0.2	3
3	925	0.3	3
5	935	0.7	2
2	949	0.7	1
4	1017	0.3	1
5	1021	0.7	2
4	1047	0.7	2
2	1078	0.2	2
5	1109	0.8	2
2	1125	0.9	3
3	1143	0.6	2
1	1150	0.5	1
4	1189	0.1	2
2	1216	0.7	2
1	1241	0.1	1

5	1267	0.9	3
3	1296	0.8	3
1	1343	0.8	2
2	1346	0.1	2
5	1351	0.5	2
3	1428	0.7	2
3	1437	0.6	1
3	1440	0.3	3

Chapter 10 References

[1] Amato, Christopher; Konidaris, George D. and Kaelbling, Leslie P., Planning with Macro-Actions in Decentralized POMDPs. In Proceedings of the Workshop on Planning and Robotics (PlanRob) at the Twenty-Fourth International Conference on Automated Planning and Scheduling (ICAPS-14), pages 1273-1280, Portsmouth, NH, 2014.

[2] Athan TW, Papalambros PY (1996) A note on weighted criteria meth-ods for compromise solutions in multi-objective optimization.Eng Optim 27:155–176

[3] Bonnet, Jonathan & Gleizes, Marie-Pierre & Kaddoum, Elsy & Rainjonneau, Serge & Flandin, Grégory. (2015). Multi-satellite Mission Planning Using a Self-Adaptive Multi-Agent System. Ninth IEEE International Conference on Self-Adaptive and Self-Organizing Systems (SASO'15)}. Self-Adaptive and Self-Organizing Systems (SASO), 2015 IEEE 9th International Conference on. 11-20. 10.1109/SASO.2015.9.

[4] Burleigh, Scott C.; De Cola, Tomaso; Morosi, Simone; Jayousi, Sara; Cianca, Ernestina; and Fuchs, Christian, "From Connectivity to Advanced Internet Services: A Comprehensive Review of Small Satellites Communications and Networks," Wireless Communications and Mobile Computing, vol. 2019, Article ID 6243505, 17 pages, 2019. https://doi.org/10.1155/2019/6243505

[5] Canadian Space Agency. (2020). *International satellites supporting disaster management*. Retrieved from Government of Canada website https://www.asc-csa.gc.ca/eng/satellites/disasters.asp.

[6] Castaing, J. (2014). Scheduling Downloads for Multi-Satellite, Multi-Ground Station Missions.

[7] Chen W, Wiecek MM, Zhang J (1999) Quality utility—a compromise programming approach to robust design. J Mech Des 121:179–187

[8] Chien, S.; Johnston, M.; Frank, J.; Giuliano, M.; Kavelaars, A.; Lenzen, C.; and Policella, N. 2012. A generalized timeline representation, services, and interface for automating space mission operations. In Proceedings of the 12th International Conference on Space Operations, SpaceOps AIAA.

[9] Choi, Han Lim; Brunet, Luc; and How, Jonathan P., Consensus-Based Decentralized Auctions for Robust Task Allocation. IEEE Transactions on Robotics, 25:912-926, 2009.

[10] Curiel, Alex Da Silva; Boland, Lee; Cooksley, John; Bekhti, Mohammed; Stephens, Paul; Sun, Wei and Sweeting, Martin. First results from the disaster monitoring constellation (DMC). Acta Astronautica, 56(1-2):261-271, 2005.

[11] Damiani, Sylvain; Verfaillie, Gerard and Charmeau, Marie-Claire. An Earth Watching Satellite Constellation: How to Manage a Team of Watching Agents with Limited Communications. In Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems - AAMAS '05, pages 455-462, New York, NY, 2005. ACM.

[12] Deb, K. (2001). Nonlinear Goal programming using Multi-objective Genetic Algorithms. Journal of the Operational Research Society, vol. 52, No. 3, Pg. 291- 302

[13] Eckenrode RT (1965) Weighting multiple criteria. Manage Sci 12:180–192

[14] European Space Agency. *Background on Genso*. Retrieved from https://www.esa.int/Education/Background_on_GENSO#.XibMEDtKi_s.link.

[15] Gombolay, Matthew; Wilcox, Ronald; and Shah, Julie. Fast Scheduling of Multi-Robot Teams with Temporospatial Constraints. In Robotics: Science and systems (RSS), pages 24-28, Berlin, Germany, 2013. RSS

[16] Herold, Thomas M.; Abramson, Mark R.; Kahn, Alexander C. ;. Kolitz, Stephan E ; and Balakrishnan, Hamsa. Asynchronous, Distributed Optimization for the Coordinated Planning of Air and Space Assets. In AIAA Infotech at Aerospace 2010, Atlanta, Georgia, 2010

[17] Hobbs BF (1980) A comparison of weighting methods in power plantsiting. Decis Sci 11:725–737

[18] Huang C-H, Galuski J, Bloebaum CL (2007) Multi-objective Paretoconcurrent subspace optimization for multidisciplinary design.AIAA J 45:1894–1906

[19] Hwang C-L, Yoon K (1981) Multiple attribute decision making, meth-ods and applications: a state-of-the-art survey. In: Beckmann M,Kunzi HP (eds) Lecture notes in economics and mathematical systems, no 186. Springer, Berlin

[20] Iacopino, Claudio & Harrison, S & Brewer, Andy. (2017). Current and Future Challenges in Mission Planning Systems for Commercial Earth Observation Constellations.

[21] Jones, Nicola. (2014). Mini satellites prove their scientific power. Nature. 508. 300-1. 10.1038/508300a.

[22] Kassaimah SA, Mohamed AM, Kolkailah FA (1995) Bi-criteria opti-mum design of laminated plates under uniform load and shear.In: Proceedings of the 27th international SAMPLE technical conference (held in Albuquerque, NM), 27, pp 731–737

[23] Kennedy, Andrew Kitrell. (2018). Planning and scheduling for earth-observation small satellite constellations. Massachusetts Institute of Technology Department of Aeronautics and Astronautics.

[24] Kim, I., de Weck, O. Adaptive weighted-sum method for bi-objective optimization: Pareto front generation. *Struct Multidisc Optim* **29,** 149–158 (2005). https://doi.org/10.1007/s00158-004-0465-1

[25] Konak, Abdullah & Coit, David & Smith, Alice. (2006). Multi-objective Optimization using Genetic Algorithms: A Tutorial. Reliability Engineering & System Safety. 91. 992-1007. 10.1016/j.ress.2005.11.018.

[26] Koski J (1985) Defectiveness of weighting method in multicrite-rion optimization of structures. Commun Appl Numer Methods1:333–337

[27] Koski J, Silvennoinen R (1987) Norm methods and partial weightingin multicriterion optimization of structures. Int J Numer MethodsEng 24:1101–112

[28] Lal, Bhavya & Blanco, Elena & Behrens, Jonathan & Corbin, Benjamin & Green, Ellen & Picard, Alyssa & Balakrishnan, Asha. (2017). Global Trends in Small Satellites IDA Paper P-8638. IDA Science & Technology Policy Institute, Washington D.C...

[29] Marler R. T., Arora J. S. (2004). Survey of multi-objective optimization methods for engineering. Struct. Multidisciplinary Optimization 26, 369-395.

[30] Marler, R. Timothy, and Jasbir S. Arora. "The Weighted Sum Method for Multi-Objective Optimization: New Insights." *Structural and Multidisciplinary Optimization*, vol. 41, no. 6, 2009, pp. 853–62. *Crossref*, doi:10.1007/s00158-009-0460-7.

[31] Odu, Godwin. (2013). Review of Multi-criteria Optimization Methods – Theory and Applications. IOSR Journal of Engineering. 3. 01-14. 10.9790/3021-031020114.

[32] Sachdeva, Shagun. “Small sat Growth on Shaky Ground, According to NSR Analysis Report”. NSR Northern Sky Research. February 19, 2019.

[33] Parham, J Brent; Zosuls, Aleks; Walsh, Brian and Semeter, Joshua. Multipoint Measurements of the Aurora with a CubeSat Swarm. In Proceedings of the 30th Annual AIAA/USU Conference on Small Satellites, SSC16-P4-14, 2016.

[34] Proos KA, Steven GP, Querin OM, Xie YM (2001) Multicriterionevolutionary structural optimization using the weighted and theglobal criterion methods. AIAA J 39:2006–2012

[35] Shanian A., Savadogo O., (2009). A methodological concept for material selection of highly sensitive components based on multiple criteria decision analysis. Expert Systems with Application. An International Journal Archive, volume 36, Issue 2, pg. 1362-1370 Part 1, Published by Pergamon Press, Inc. Tarrytown, NY, USA.

[36] Surka, D. (2003). Multiple Agent-Based Autonomy for Satellite Constellations. Artificial Intelligence.

[37] Spangelo, Sara and Cutler, James. Analytical Modeling Framework and Applications for Space Communication Networks. Journal of Aerospace Information Systems, 10(10):452-466, 201.

[38] Stadler W (1995) Caveats and boons of multicriteria optimization.Microcomput Civ Eng 10:291–299

[39] Stadler W, Dauer JP (1992) Multicriteria optimization in engineering:a tutorial and survey. In: Kamat MP (ed) Structural optimiza-tion: status and promise. American Institute of Aeronautics andAstronautics, Washington, DC

[40] Voogd H (1983) Multicriteria evaluation for urban and regional plan-ning. Pion, London

[41] Wang, David and Williams, Brian C. Burton: A Divide and Conquer Temporal Planner. In MIT Computer Science and Artificial Intelligence Laboratory, TR2014-027, Cambridge, MA, 2014. MIT.

[42] Wang, Yu; Sheng, Min; Zhuang, Weihua; Zhang, Shan; Zhang, Ning; Liu, Runzi and Li, Jiandong. Multi-Resource Coordinate Scheduling for Earth Observation in Space Information Networks. IEEE Journal on Selected Areas in Communications, 36(2):268-279, 2018.

[43] Weiss, G. (Ed.), Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence, MIT Press, Cambridge, MA, 1999.

[44] Wu, Guohua; Wang, Huilin; Li, Haifeng; Pedrycz, Witold; Qiu, Dishan; Ma, Manhao and Liu, Jin. An adaptive Simulated Annealing-based satellite observation scheduling method combined with a dynamic task clustering strategy. page 23, 2014.

45 [24] Zhou, Di; Sheng, Min; Wang, Xijun; Xu, Chao; Liu, Runzi and Li, Jiandong. Mission Aware Contact Plan Design in Resource-limited Small Satellite Networks. IEEE Transactions on Communications, 65(6):2451-2466, 2017.