8-2023

Mission Planning Techniques for Cooperative LEO Spacecraft Constellations

Skylar A. Cox
Utah State University

MISSION PLANNING TECHNIQUES FOR COOPERATIVE LEO SPACECRAFT CONSTELLATIONS

by

Skylar A. Cox

A dissertation submitted in partial fulfillment
of the requirements for the degree

of

DOCTOR OF PHILOSOPHY

in

Electrical Engineering

Approved:

Greg Droge, Ph.D. Major Professor	David Geller, Ph.D. Committee Member
Pat Patterson, Ph.D. Committee Member	Don Thompson, Ph.D. Committee Member
Burak Sarsilmaz, Ph.D. Committee Member	D. Richard Cutler, Ph.D. Vice Provost of Graduate Studies

UTAH STATE UNIVERSITY
Logan, Utah

2023

ABSTRACT

Mission Planning Techniques for Cooperative LEO Spacecraft Constellations

by

Skylar A. Cox, Doctor of Philosophy

Utah State University, 2023

Major Professor: Greg Droge, Ph.D.
Department: Electrical and Computer Engineering

Sustained interest in spacecraft constellations continues to drive advancement in sophistication, capability, and mission complexity. As these systems proliferate, they require increased levels of automation for orchestrating the achievement of their mission objectives. This research develops a constellation planning capability for distributed operations supporting an on-orbit servicing (OOS) mission as well as cooperative centralized planning for proliferated Earth sensing missions. The techniques presented provide a tractable approach for formulating and solving in-schedule and cross-schedule dependent problems for cooperative systems within an extended time environment. OOS cooperation is accomplished via a modified consensus-based approach while the Earth sensing constellation coordination is performed using a network flow formulation that solves the coupled data collection and distribution problem. These methods are used to simultaneously obtain a plan for each individual satellite within the constellation while attempting to optimize the performance of the collective constellation.

(129 pages)

PUBLIC ABSTRACT

Mission Planning Techniques for Cooperative LEO Spacecraft Constellations

Skylar A. Cox

This research develops a mission planning approach that allows different systems to cooperate in accomplishing a single mission goal. Using the techniques described allows satellites to cooperate in efficiently maneuvering, or collecting images of Earth and transmitting the collected data to users on the ground. The individual resources onboard each satellite, like fuel, memory capacity and pointing agility, are used in a manner that ensures the goals and objectives of the mission are realized in a feasible way. A mission plan can be generated for each satellite within the cooperating group that collectively optimize the mission objectives from a global viewpoint. The unique methods and framework presented for planning the spacecraft operations are flexible and can be applied to a variety of decision making processes where prior decisions impact later decision options. This contribution to the satellite constellation mission planning field, thus has greater applicability to the wider decision problem discipline.

Dedicated to Ashley, Kyla, Carsten, and Ember.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

A saying within one of the most capable and renowned forces of the U.S. military is "Individuals play the game, but teams beat the odds," indicating that a team is a more formidable entity than an individual and able to more effectively respond to challenges faced during a mission [1]. While an individual may be highly trained and motivated, a team will often accomplish more due to its ability to handle more complex and challenging task environments [2]. This concept is intuitive when applied to human teams, but it can be extended to coordinated teams of unmanned systems as well. The methods employed to plan operations for unmanned systems are varied with diverse applications and objectives, often requiring sophisticated approaches. The aim of this research is to provide a robust and extensible methodology for planning the operations of multi-agent, unmanned systems in an intuitive manner. In particular, the research focuses on optimally planning flight operations for a coordinated constellation of satellites flying in low Earth orbit (LEO).

Currently within the space flight community, there is a renewed interest in large spacecraft constellations owing to the very recent decrease in cost to access space [3]. The bulk of this decrease is due to breakthroughs in launch vehicle cost with those improvements having profound impacts on relaxing some of the previous performance requirements of satellite components. This relaxation has resulted in significantly less expensive spacecraft [4] and drawn a renewed interest in space, resulting in heavy investment into the space business ecosystem. Over $177 B of equity was invested into more than 1300 companies between 2011 and 2021 alone [5]. This significant amount of capital has fueled additional entries into the sector and introduced a variety of options for new satellite constellations. While many of the individual satellites within these constellations are shrinking in mass, volume, and cost [6], the constellations continue to grow in numbers, capability, and sophistication, requiring increased levels of automation [7] for orchestrating the achievement of their mission

objectives [8] [9].

As new mega-constellations are designed and deployed, several new challenges are realized [10] [11]. Some of these challenges deal specifically with general orbital configuration design [12] [13], while others are more concerned with the orchestration of the individual satellites within these mega-constellations. As the number of satellites within a constellation increases, the operational planning problem complexity compounds due to the spatial and temporal dependencies present within the constellation's operational environment, and the limited resources onboard each vehicle [14]. This orchestration of multi-satellite constellations in fulfilling a larger holistic mission is the purpose and focus of this research. The unique challenges presented by current and future constellation systems make this problem a unique and valuable opportunity for research today. The research presented herein addresses two specific missions, that of planning multi-agent rendezvous and proximity operations (RPO) for on-orbit servicing (OOS) of a constellation of resident space objects (RSOs), and the more common mission of Earth-sensing and its associated payload and data transfer tasks.

The RSO serving constellation planning problem requires the planning system to consider current and future available resources for both the servicing agents as well as the RSOs. The planning approach must evaluate the needs of the RSO to be serviced against the available capabilities and resources of the servicing agents and must do this over relatively long periods of time (i.e., years) into the future since every decision made by the servicing agents has a cascading effect across both constellations.

The primary objective of the constellation Earth-sensing mission type is to generate an efficient, operationally-viable schedule that maximizes the collection and delivery of priority Earth-sensing data, and fits within all specified constraints of the mission. Planning such operations for large satellite constellations becomes a tremendous challenge when attempting to maximize the global utility of this mission while considering the individual capabilities and available resources of each satellite within the constellation.

The research presented herein addresses the needs of these missions and builds upon

existing planning approaches while extending them in novel ways. For example, for the RSO servicing constellation problem we develop a planning and scheduling methodology for a constellation of RPO-capable vehicles tasked with visiting and servicing a larger number of RSOs. Event scheduling is accomplished by leveraging a decentralized task allocation algorithm developed within the robotics and unmanned aerial vehicle community and augmenting it with a long-term orbital model. We develop a novel utility function that combines the score of a rendezvous event, driven by servicing urgency and preferred conditions, with the associated cost calculated relative to the required vs. available resources. These contributions yield a highly-capable scheduling framework for decentralized task allocation and address the long-term RSO constellation servicing problem.

Our contributions to the Earth-sensing constellation planning problem include a graph-based formulation augmented with network flow theory that is capable of generating an operational schedule for every satellite within the constellation, while considering the mission performance from a global perspective. This graphical formulation enables a simultaneous search for each satellite's path through fulfillment of Earth-sensing and data transfer tasks and, with the application of network flow and resource constraints, ensures coordination across the constellation while guiding the solver to a mission-realistic schedule solution. Our formulation blends task generation, and fulfillment, within an operationally-constrained environment and is a significant contribution to the overall constellation planning literature.

Prior to a more detailed explanation of these contributions and their implementation, a background of constellation missions and existing planning techniques is presented in the next chapters.

The remainder of this document is organized to first provide the reader with background on the planning problem in Chapter 2, followed by a review of existing approaches and literature in Chapter 3. Chapters 4, 5, and 6 provide the in-depth technical content explaining the constellation planning techniques devised in this research to accomplish the OOS, Earth sensing, and crosslink-enabled Earth sensing missions, respectively. Following

those chapters, a conclusion summarizes the contributions in Chapter 7. Finally, Appendices A and B are provided to substantiate the methods described.

CHAPTER 2

BACKGROUND

The purpose of this chapter is to familiarize the reader with several generalities about the research, what foundations it builds on, and the approaches that exist within the field. Subsection 2.1 deals with the research considerations of decisions problems and helps classify the problem types, considered in this research, in terms of complexity and dependency. Subsection 2.2 provides an explanation of two common planning architectures, when they apply, and identifies some of their benefits and limitations. Finally, subsection 2.3 outlines the basic problem formulation and solution methods for mixed integer linear programming as it applies to this research.

2.1 Problem Classification

Planning operations for a cooperative constellation of satellites falls within a specific field of research often referred to as multi-robot task allocation (MRTA). The term 'robot' in this case refers to each individual element, or agent, within the system for which operations are being planned. Within the research presented, 'agent' and 'robot' are often used synonymously and simply refer to the satellite, or actor, within the system that is performing the allocated task.

As part of a review of the MRTA literature, a formal taxonomy must first be introduced to properly categorize the types of problems within the space, and the associated dependencies. Doing so allows researches the ability to use existing algorithms for different problems as long as the problem types fall within the same classification of a defined taxonomy. An adaptable taxonomy was developed by Gerkey and Matarić that described different types of tasks and the agents, or robots, capable for performing those tasks [15]. Three axes of differentiation are considered: whether the robot can perform a single task (ST) or multiple-tasks (MT) at a time, whether the individual tasks can be performed by

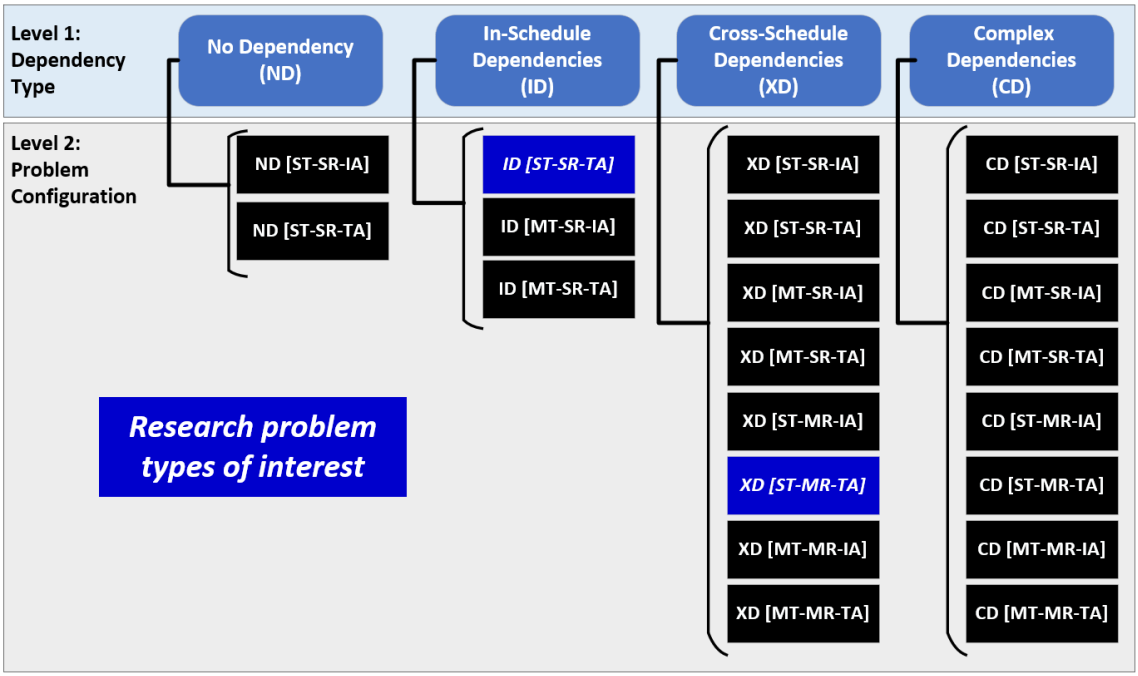


Fig. 2.1: Multi-robot task allocation problem taxonomy. The satellite constellation problems researched are highlighted in dark blue.

a single robot (SR) or require multiple robots (MR), and whether the robot is capable of accepting only an instantaneous allocation (IA) or can be scheduled over an extended period of time (TA). Korsah et al. extended this concept by accounting for dependencies that can exist between the tasks or the agent schedules. These include in-schedule dependency (ID) where tasks may require a specified sequence for fulfillment, cross-schedule dependency (XD) where agents must coordinate schedules, and complex dependency (CD) where task fulfillment utility depends on the schedules of other agents in the system in a manner that is determined by the particular task decomposition chosen [16]. A visual representation of this taxonomy is provided in Figure 2.1.

The research presented in this document provides a tractable methodology for intuitively formulating and solving a global constellation planning problem falling into the ID [ST-SR-TA] and XD [ST-MR-TA] classifications.

The resident space object (RSO) servicing mission is ID [ST-SR-TA] as each servicing agent is capable of performing a single servicing task at a time (ST), each servicing task

requires a single robot to perform that task (SR), the tasks are allocated over an extended period of time (TA), and task decisions have a cascading impact to later tasks, making them in-schedule dependent (ID).

Similarly, the base Earth-sensing mission is ID [ST-SR-TA] due to each satellite, or robot, only being capable of fulfilling a single task at a time (ST), each task requiring only a single robot to fulfill (SR), and the tasks being planned over an extended period of time (e.g., many orbit revolutions) (TA). This mission also has in-schedule dependencies since a decision to perform a particular task impacts the feasibility of subsequent task selections (ID). Finally, the base Earth-sensing mission will be extended into the cross-schedule (XD) dependency domain by requiring coupled interaction of satellites. This is accomplished by introducing the concept of crosslinking data sets between satellites with the intent of more efficiently using resources across the constellation. Crosslinking data requires a satellite pair to coordinate operations in fulfilling this task type and thus results in a multi-robot task (MR). The addition of a cross-schedule dependent task that requires multiple robots to fulfill, dramatically increases the complexity of the planning problem.

This taxonomy will be referenced throughout the remainder of this document to formally classify previous research contributions and the particular types of problems this research addresses.

2.2 Centralized and Decentralized Planning Approaches

With the taxonomy defined, the next step is to understand the different planning paradigms within the MRTA literature and operational space that still plan operations within a global context. These paradigms dictate how the planning is conducted and how the tasks and resources are allocated to agents operating within the space. The two primary planning approaches considered in this research are centralized and decentralized approaches [17]. Centralized planning is generally conducted by one empowered entity that has a complete global picture of the environment and the states of all the agents within that environment. This entity receives information and updates the global planning approach to respond and coordinate the tasks and resources assigned to each agent across the fleet.

Centralized planning is highly effective when the trajectories of the agents can be easily planned or can be predicted prior to the start of each planning period and the tasks are relatively predictable in time and space [17]. However, this approach can be prone to misinformation, delays, or outages which can result in degraded performance [18]. The centralized planning approach is generally a reliable method for planning a constellation of cooperative satellites. Within this paradigm, satellite ephemerides are known precisely along with the locations and availability of both Earth-sensing opportunities and data transfer windows. This information allows a centralized planner to properly orchestrate task fulfillment opportunities across the constellation and generate a flight schedule for each satellite within the constellation. The ability to generate an optimal mission plan for the entire constellation that respects individual satellite resources, makes the centralized approach a very attractive solution for constellation mission planning.

Alternatively, decentralized planning is generally conducted by the agents themselves due to the operational requirements dictated by their environment. Environments that are uncertain, highly dynamic, or lacking the option for a centralized operational facility, often require a decentralized approach to properly orchestrate operations between disparate agents [19]. A highly dynamic, or uncertain, environment can preclude the effective use of a centralized planning system since it lacks a reliable global understanding of the state of the environment and the tasks requiring fulfillment. A good example of an environment requiring a decentralized planning approach is that of a building that has been damaged due to a natural disaster. The location of victims inside, if any, their injuries, and the accessible routes through the building are unknown and require investigation by agents before deciding on which tasks to fulfill (e.g., routes, victim treatment or extraction). A decentralized planning approach would allow agents to make decisions about routes and victim treatment or extraction based on their observations of both floor plan status as well as victim health, and then share that information with each other for proper coordination in a global planning sense.

A satellite-specific decentralized planner would operate in a similar way with each satellite receiving localized information about the operational environment and also observing events in real-time that may influence decisions. The collected information would include Earth-sensing opportunities, current weather conditions, available ground stations, and the activities being conducted by other satellites in the constellation. With that information, and knowledge about its onboard resources, a satellite could make informed decisions about which tasks to fulfill and when to do so. Those decisions would then be communicated to all other satellites in the constellation that are within range and thus likely improve the global performance of the constellation, as a whole. However, a significant factor precluding a globally optimal solution is the timeliness of the information being shared since one satellite's decisions and observations are not instantaneously known by all other satellites within the constellation.

In summary, several considerations must be made before selecting a preferred planning methodology. This research highlights two unique applications for both centralized and decentralized planning approaches. The primary purpose of both planning approaches is to provide a feasible and operationally-relevant schedule of operations for each satellite within the constellation that accomplishes the objectives of the overall mission.

2.3 General Mixed Integer Formulation

Within the proliferated constellation planning space, it is evident that a single satellites's individual optimal schedule will likely not be the same as its schedule when a global optimum is achieved across the entire constellation. The optimal use of a single satellite's resources in accomplishing a given mission likely results in a different schedule when a planner factors in all available resources across the constellation, the individual satellite trajectories, and available task fulfillment windows. This is due to the spatial and temporal arrangement of the individual satellites and efficient collaborative planning that uses the satellites as a team. A centralized planning approach plans an entire constellation and seeks the global optimal solution while factoring all satellites within the constellation, whereas the decentralized approach tends to focus on an individual satellite with simply the intent

to avoid duplication of task fulfillment but can often be limited by not having an accurate, global picture of the entire constellation state and task assignments.

As discussed, the complete set of individual satellites must be considered holistically to effectively plan when searching for a global optimum within the overall mission and constellation. This global optimum must be a quantifiable performance metric that can be evaluated based on decisions made by the planning algorithm, while considering the capabilities and resources of each satellite on an individual basis. This optimization problem is substantial and, at its most basic level, requires three key elements to make it tractable. These elements include 1.) an objective function, 2.) the applicable constraints, and 3.) the decision variables within the problem space [20]. A common way of capturing this information is with the use of an integer program. Within this research, the optimization is linear and thus allows for the formulation to be captured as a mixed integer linear program (MILP). This problem formulation is well-known in the optimization community and has benefited from a tremendous amount of resources in determining reliable methods for solving it. The general MILP formulation captures the three key elements mentioned above as [20]:

$$\begin{aligned} \max_{z} \; & u^T z \text{ (Objective Function)} \\ \text{s.t. } & Dz = b \text{ (Equality Constraint)} \\ & Az \leq c \text{ (Inequality Constraint)} \\ & z = \left[x^T y^T\right]^T \text{ (Integer and Continuous Variables)} \\ & x \in \{0,\, 1\} \text{ (Integrality Constraint)} \end{aligned} \quad . \tag{2.1}$$

The objective function captures the intent of the optimization problem and provides a quantitative measure for evaluating the performance of individual solutions based on selections made with the decision variable x and continuous variable y. The equality constraints set specific requirements for the resulting decisions, whereas the inequality constraints set bounds. The integrality constraint provides a binary option for the decision variable, x, to be either 0 or 1. This is the generalized form of the optimization problems to be discussed in this research. As referenced previously, there are many commercial options for solving

MILPs. One in particular is leveraged throughout the research and provided by Gurobi [21]. An overview of MILP solution techniques is available in Appendix B.

CHAPTER 3
LITERATURE REVIEW

This research is timely and increasingly relevant to the new constellations being developed within the spaceflight community and focuses on addressing two primary mission types. The first being the RSO constellation servicing problem for long-term planning of LEO RPO scheduling using ΔV maneuvers. The second being on more traditional constellation planning for mission operations of large scale Earth-sensing, or geospatial intelligence (GEOINT) LEO missions. Operations supporting GEOINT missions include tasks such as Earth-sensing, data downlink, and satellite-to-satellite data crosslink. The next portion of this document will review the relevant research contributions to both of these problems, starting with the RSO servicing planning problem which is followed by a thorough review of the state-of-the-art within the GEOINT constellation planning problem space.

3.1 On-Orbit Servicing Mission

As discussed, many new constellations are being designed and fielded. The mass proliferation of satellites in LEO, especially, will require more sophisticated operational capabilities [22] [23]. For example, the planned Starlink constellation will comprise several thousand satellites in LEO [24]. Such a constellation must use some autonomy to avoid placing unrealistic demands on operations staff. Maneuvering satellites within a larger constellation, or formation, is one key element of properly managing these large systems and a variety of planning methods have been proposed [25] [26] [27] [28], but they have generally focused on geosynchronous orbits or lacked a true constellation focus to the problem. The constellation servicing mission we have addressed deals with how to most effectively route a team of servicing agents to a constellation of RSOs requiring servicing. This mission concept is illustrated in Figure 3.1.

In analyzing multi-spacecraft OOS mission literature, research has typically focused

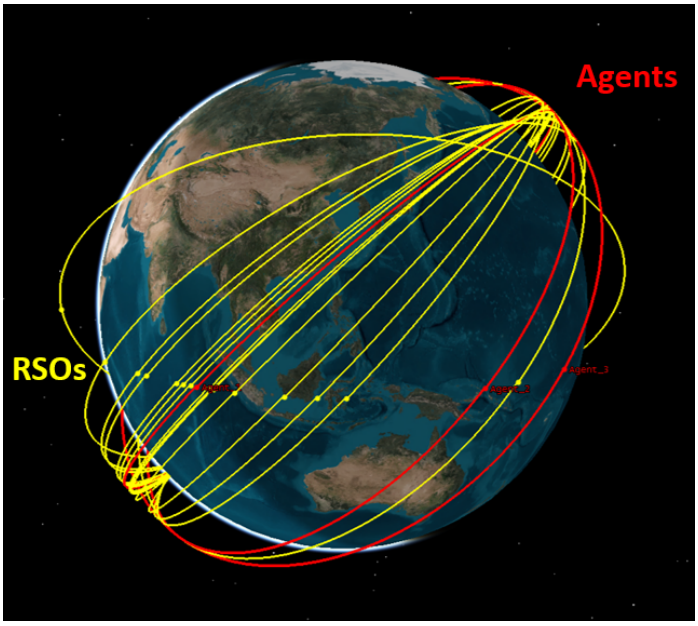


Fig. 3.1: LEO RSO servicing mission concept with a purpose of optimally routing servicing agents to a constellation of RSOs requiring servicing.

on fuel-optimal sequencing for a single agent vehicle, rather than a constellation of agents, servicing multiple RSOs. Shen and Tsiotras evaluate fuel optimal sequencing for visiting multiple vehicles in the same circular orbit, showing the best solution is to visit them in orbit-wise or counter-orbit-wise order [29]. Zhang et al. solve a multi-spacecraft refueling problem in a near-circular LEO orbit using a hybrid-encoding genetic algorithm [30]. Their analysis includes J2 perturbations and time-window constraints based on the orbital beta angle to optimize fuel usage and mean mission time. They also show how splitting the RSO's into two sub-groups to be visited by separate servicing vehicles affects the optimal solution. Optimal selection of the two sub-groups is left to future work. Zhao et al. expand on this work by analyzing a cooperative, multi-spacecraft refueling mission that includes constraints on the target vehicle's surplus propellant [31]. Verstraete et al. continue this line of operationally-relevant servicing by looking at a geostationary orbit servicing mission that accounts for the risk and value of accomplishing any given task. Their analysis shows that varying risk levels can significantly alter the mission objective function and the task sequencing to be implemented [32]. Zhang et al. evaluate the use of multiple servicing

vehicles to visit a set of target vehicles using a multi-objective quantum-behaved particle swarm optimization [33]. However, it is unclear on whether the approach can be extended to have each servicing vehicle visit multiple targets or a single target each. Liu and Yang present the debris removal from LEO problem and provide a multi-objective optimization formulation and a modified genetic algorithm to solve for the sequential transfer route, similar to the RPO problem of interest for our research [34].

The constellation OOS literature is limited to more simplistic problem formulations, primarily with the number of servicing vehicles and RSOs being serviced. Additionally, much of the literature is concerned with optimal fuel sequencing instead of feasible routes and maneuvers within a time-extended mission context. The research paper presented in Chapter 4 addresses these limitations by significantly extending the cooperative constellation use-case, for the ID [ST-SR-TA] problem type, by developing an initial planning and scheduling methodology for a constellation of RPO-capable vehicles tasked with visiting and servicing a larger number of RSOs in LEO. This research has been published by the author in [35] and [36].

3.2 GEOINT Mission

The purpose of a constellation planning system, within the GEOINT mission context, is to maximize prioritized Earth-sensing operations and data downlink for mission task fulfillment while respecting each satellite's capabilities and available resources. The satellite constellation and ground resources must be treated holistically to ensure that satellites do not duplicate efforts in imaging and that only a single satellite uses a particular ground terminal at any given time. The slew agility and onboard memory capacity of individual satellites must also be considered to ensure an executable and mission-relevant plan. Finally, while simple Earth-sensing tasks can be performed nearly instantaneously, the data transfer tasks require extended periods of time to fulfill.

3.2.1 GEOINT Mission without Crosslink

The base Earth sensing/GEOINT constellation planning problem plans Earth-sensing

operations and data transfers directly between a satellite and a ground antenna (no satellite-to-satellite data exchange is considered). This problem therefore falls within the ID [ST-SR-TA] category due to the presence of only in-schedule dependencies for each satellite, satellites only being able to perform a single task at a given time, tasks only requiring a single satellite to perform the task, and the allocation applying over an extended period of time. The primary objective of this mission planning problem is to generate an efficient operational schedule that fits within all specified constraints of the mission, including the resources of each satellite within the constellation. Several approaches have been presented to address a portion of the Earth imaging mission. A host of work has focused on arriving at sub-optimal solutions to the prioritized image collection problem [37] [38] [39] [40] [41] [42] [43].

Similarly, several researchers have investigated the downlink, or data relay, problem and provide robust solutions to scheduling the transfer of data [44] [45] [46] [47] [48] [49] [50] [51]. However, in both the prioritized image collecting maximization and data relay scheduling problems, the papers referenced only address a portion of the complete problem. The image collection and downlink problems are directly coupled and should be solved simultaneously to yield a reliable and accurate solution since optimal downlink requires complete knowledge of data collection, and optimal data collection requires knowledge about when data is being downlinked.

Some works have considered both the image collection and data relay problems within the same framework and provided various solutions to solve them [52] [53] [54]. However, Hu et al. dedicated most of their paper to evaluating a branch and price technique for finding a solution with only a short explanation of the problem itself [53]. Hu et al. simulated a LEO constellation of only 3 satellites [53] so scaling up to the anticipated size of future proliferated LEO constellations [55] is difficult to predict. Similarly, Peng et al. formulated their planning problem for a single satellite and neglected its application to a multi-agent cooperative system [54]. A key element of constellation planning requires that the planning system be capable of consistently generating operational schedules for large constellations while respecting all the constraints of the individual satellites.

The paper by Augenstein et al. addressed many of the requirements for a mission-relevant constellation planning technique and was particularly interesting. The combinatorial optimization problem addressed by Augenstein et al. is defined as a MILP with constraints and objectives applied globally [52]. Their MILP formulation can be solved to arrive at an optimal solution within the discretized space. However, to speed up solution generation time they separated the immediate coupling that exists between image collection and downlink opportunities. They accomplished this by creating a heuristic that estimates the amount of priority-weighted image data likely onboard the satellite as a function of time and the number of priority-weighted image collection opportunities a satellite would encounter during specific time periods. Augenstein et al. demonstrate that with these two pieces of information it is possible to reasonably allocate downlink time among the constellation of vehicles and then also schedule the additional image collection opportunities [52]. Fore a more thorough review of this paper, please reference Appendix A.

The formulation by Augenstein et al. provided a valid constellation planning method and also introduced various interesting, mission-specific constraints to guide the planning system's decisions to ensure the mission can be realistically conducted. However, the formulation does not provide a clear representation of each satellite's individual plan. The mathematical formulation simply organizes the problem in time for a single agent and does not explicitly provide a tractable method of determining the optimal schedule for all satellites collectively comprising the constellation. While it appears the formulation is used for operating a fleet of satellites, the literature lacks clear specifics on how this is accomplished in practice and thus requires a more complete formulation. The research presented in Chapter 5 of this document addresses these critical limitations while leveraging the contributions of Augenstein et al. with their graph-based approach. The research introduces an alternative solution method that relies on network flow theory to enable the planning of many satellites at once while respecting the mission constraints of both the constellation, as well as the individual vehicles that comprise it. The next section of this document introduces an augmentation of this problem by allowing direct cooperation between pairs of satellites

and thus extends into the XD [ST-MR-TA] problem space.

3.2.2 Crosslink-Enabled GEOINT Mission

The crosslink-enabled Earth sensing/GEOINT mission is a novel extension of the original problem in that it now requires the planning system to consider opportunities where satellites can cooperate with each other in exchanging data (i.e., cross-schedule dependencies (XD) from the taxonomy of Korsah [16]). This task is referred to as a crosslink operation and requires that one satellite transmit data while the other receives the data and both must be properly oriented relative to each other to execute the task. Performing this task allows the satellite constellation to potentially take advantage of resources across the constellation instead of relying solely on individual satellites for Earth-sensing, data storage, and eventual data downlink. Satellites are able to consider sharing data with other nearby satellites to take advantage of more effective downlink opportunities using the access opportunities of their partners in the constellation. Doing this may provide a specific satellite more opportunities for Earth-sensing operations that before may have conflicted with data downlink task fulfillment. Crosslinking has the potential to become a more common operation for small satellite systems and thus merits an effective, autonomous planning solution [56]. The crosslink-enabled GEOINT planning problem now moves beyond the basic GEOINT problem discussed in the previous section due to the additional cross-schedule dependency (XD) that arises during crosslink operations. The planning problem now considers both ID [ST-SR-TA] and XD [ST-MR-TA] categories within the GEOINT mission context.

Several significant contributions have been made to the cross-linked enabled GEOINT mission literature. For example, Zhou et al. present a problem formulation and solution approach that attempts to maximize network throughput while also considering the limitations imposed by onboard satellite energy and data storage constraints [57]. This formulation develops both an optimal and heuristic approach to data route planning using 6 satellites. The research provides unique insight into the modeling of the satellite power states in relation to network data routing. Their description provides an in-depth description on data routing using a continuous flow model with an understanding that Earth-sensing operations

are conducted independently [57].

Kondrateva et al. formulated a MILP with the purpose of maximizing data throughput within a LEO satellite network. They adapted a static network formulation to the dynamic LEO environment by discretizing the data networking problem into time windows when the LEO network can be modeled as a static entity and then solving the problem within each time window [58]. This formulation considers data routing in conjunction with link scheduling and models data collection on the LEO vehicles to be a continuous flow of collection. This continuous data generation assumption removes the need for planning discrete payload operations and focuses the formulation presented on addressing the data routing and link scheduling elements of the problem [58]. Their implementation was demonstrated within constellation of 18 satellites and proved the ability to directly calculate a complete data transmission schedule without relying on heuristics [58].

Wang et al. formulate this same problem as an event-driven time-extended graph with nodes representing a discrete location in time and edges representing an observation, transmission, or storage window [59]. These windows indicate what resources or operations are available as the path moves through the nodes and edges of the graph. The associated MILP formulation further specifies constraints to ensure only a single target is collected at each time step, only a single transmission occurs at each time step, the amount of data transmitted do not exceed the amount collected, and that data are only transferred after they are collected [59]. The formulation makes an assumption that each satellite within the constellation has unlimited data storage capacity, potentially complicating the applicability of the algorithm to a realistic mission. The formulation is demonstrated on a constellation of 6 satellites, and models data as a continuous flow.

Kennedy et al. formulated a mission planner with the intent of planning priority observations within specified revisit rates and also delivering low-latency observation to the ground via both crosslink and downlink operations within a constellation of satellites [60]. The planning approach by Kennedy et al. uses weightings for both Earth-sensing and communication activities and prioritizes early observation windows during planning [61].

This prioritization method applies a higher weight to the operation for the satellite first with the opportunity within the planning horizon. The formulation considers crosslink and downlink operations equally and considers data as a bulk quantity of homogeneous value [62]. The research provides a set of rules that approximate how ground resources could be allocated to satellites but may not be operationally viable in a realistic mission scenario [60].

In a subsequent publication, the original approach is updated by Kennedy et al. using a dual algorithm for planning both observation and communication operations across a crosslink-enabled constellation of satellites, dramatically reducing the latency of observation data delivery [63]. In that formulation, windows are set for performing operational activities and deconfliction is performed on any temporally-overlapping activities that are scheduled. The deconfliction appears to occur after activities are scheduled rather than as part of the planning process. The data are modeled as a homogeneous flow and allowed to be split across multiple communication windows, thus simplifying the optimal data routing problem. A fundamental and simplifying assumption in the formulation appears to restrict communication windows to begin planning only after an observation is scheduled rather than allowing a more holistic approach to emerge by considering observations, downlinks, and crosslinks together during the planning process [63].

Chen et al. investigated the XD [ST-MR-TA] problem space in a generic sense where robots were required to cooperate with each other to accomplish tasks but were also able to share resources via rendezvous [64]. The problem they addressed overlaps with the crosslink enabled GEOINT mission planning problem. In their formulation the robots were able to share resources across the fleet via rendezvous [64] whereas in the GEOINT application, satellites are able to share data storage resources when one satellite transmits and another receives data during an extended duration of exchange when the satellites are appropriately aligned in proximity, relative speed, and proper pointing. However, the problem addressed by Chen et al. required multiple robots for every task [64] whereas the crosslink-enabled GEOINT planning problem, presented herein, only requires satellites to cooperate with each

other during the actual exchange of data, or when resources are being shared.

The crosslink-enabled GEOINT formulation presented in Chapter 6 blends aspects of both the ID [ST-MR-TA] and XD [ST-MR-TA] categories and provides a more extensible formulation for cooperative LEO constellation planning. This is accomplished by considering both Earth-sensing and data transfer operations within the same framework to yield a realistic mission scheduling solution. A summary of the contributions of this research is now provided in the next section.

3.3 Research Contributions

The current constellation OOS literature lacks sophistication in many of the formulations and does not adequately address the appropriate size of future constellations. To address these limitations, the research described in Chapter 4 significantly extends the cooperative servicing constellation mission and provides a feasible planning and scheduling strategy. The formulation integrates an existing decentralized algorithm with the large scale constellation planning problem currently growing within the space community. The marriage of these two, results in a timely, operationally feasible solution with good performance. It also allows for dynamic re-planning due to the timeline of the resulting solution. Event costs such as fuel and time are also considered along with a novel utility function that can be used inside the planning framework for the management of a constellation of RPO-capable satellites for potentially sub-optimal, but operationally-relevant, scheduling solutions. The research contributions provide a decision aide in the short-term while simultaneously providing value to future solutions.

To address the existing ID [ST-SR-TA] GEOINT constellation planning method limitations, the research presented in this document augments the existing constellation planning literature and provides a powerful solution technique using a unique application of network flow theory. Utilizing the network flow approach, a highly-extensible and flexible method for specifying mission objectives and vehicle constraints is provided in Chapter 5. The solution for which results in optimal scheduling for individual satellites within a cooperative constellation context. Additional methods for improving the solve time are also evaluated

and discussed. The GEOINT mission research presented herein, directly confronts the challenges posed by the constellation planning problem and details a planning technique for this mission that enables the simultaneous coordination of both Earth-sensing and scheduling of ground stations for data downlink, across the cooperative constellation of space vehicles. Furthermore, this research provides a tractable method for intuitively formulating and solving a global constellation problem falling into the ID [ST-SR-TA] classification. The solution to the in-schedule dependency problem optimally coordinates multiple spacecraft to fulfill the base GEOINT mission.

A consistent limitation across much of the crosslink-enabled GEOINT satellite network (XD [ST-MR-TA] category) planning operations literature is a lack of consideration for both Earth sensing and data transfer operations within the same planning and scheduling environment. A constellation planning system should generate a good operations schedule for each vehicle within the constellation and make decisions for conducting certain operations in lieu of others (e.g., collecting Earth-sensing data instead of crosslinking data). The data collection and transfer tasks are directly coupled and thus require a holistic approach that properly addresses both operations. Another common assumption in the literature is that data are treated as a continuous flow rather than as discrete files. These limitations therefore demand a different approach to constellation planning operations that considers all spacecraft operations together to most effectively generate the desired globally optimal constellation schedule that respects all flight constraints imposed by the individual space vehicles that comprise it. To accomplish this goal, the research presented in Chapter 6 provides a tractable solution for planning Earth sensing activities, data storage, and data transfer operations across a constellation of crosslink-enabled space vehicles with the intent of maximizing mission utility. This dissertation concludes by summarizing the research contributions in Chapter 7.

CHAPTER 4

Resource-Constrained Constellation Scheduling for Rendezvous and Servicing Operations

Resource-Constrained Constellation Scheduling for Rendezvous and Servicing Operations

Skylar A. Cox* and Nathan B. Stastny†
Space Dynamics Laboratory, North Logan, Utah 84341
and
Greg N. Droge‡ and David K. Geller§
Utah State University, Logan, Utah 84322

https://doi.org/10.2514/1.G006153

This paper addresses the rendezvous and proximity operations constellation assignment problem by developing a responsive utility function for tasking a constellation of low Earth orbit satellites to several spacecraft servicing tasks. The paper develops the utility function that considers both score of servicing resident space object spacecraft in conjunction with the associated ΔV cost. A highly capable and operationally relevant task allocation method, called the consensus-based bundle algorithm, is leveraged for distributed processing and task allocation. This paper demonstrates that this methodology provides a robust technique for rendezvous and proximity operations scheduling within resource-constrained agent constellations.

Nomenclature

a	=	semimajor axis of the orbit, km
a^*	=	required semimajor axis of the transfer orbit, km
c	=	cost for an agent to perform resident space object servicing task
f	=	fractional parameter for an agent quantity
i	=	agent index
inc	=	orbit inclination, °
$J2$	=	Earth J2 coefficient
j	=	resident space object servicing task index
k	=	number of orbit revolutions associated with phasing maneuver
L	=	lifetime reference, years
N	=	number of agents or resident space objects in the simulation
R	=	radius, km
t	=	time since simulation epoch, s
T	=	total capacity
u	=	utility
v	=	score
W	=	score weight factor
x	=	decision assignment variable
β	=	beta angle of the resident space object orbit, °
ΔV	=	ΔV associated with the operation, m/s
ϵ	=	obliquity of the ecliptic, °
Λ	=	ecliptic longitude, °
λ	=	cost scale factor
μ	=	gravitational parameter of the Earth, km^3/s^2
τ	=	duration of event
Ω	=	right ascension of the ascending node of the orbit, °
$\dot{\Omega}$	=	drift rate of right ascension of the ascending node, °/s
ω	=	angular orbital rate, °/s

Subscripts

agent	=	agent indicator
life	=	lifetime
light	=	lighting conditions
phase	=	phasing orbit
RSO	=	resident space object servicing task indicator
req	=	required value
$\oplus$	=	Earth
$\odot$	=	sun

I. Introduction

MANAGEMENT of large satellite constellations is a unique and complex task requiring sophisticated planning tools and advanced constellation simulators. While today's constellations are primarily dedicated to ground imaging and communications, there is growing interest in developing teams of rendezvous and proximity operation (RPO) capable satellites with the ability to inspect and/or service other resident space objects (RSOs) [1,2]. This mission type is referred to as on-orbit servicing (OOS) and some vehicles are predicted to reach initial operational capability (IOC) by 2024 due to commercial and government investment [1]. Some systems currently under development include O. CUBED from Airbus [3] and SOUL from Busek [4], among others [1]. This paper has been written in anticipatory fashion in that the authors foresee OOS missions requiring multiple servicing vehicles in order to sustain the mission requirements of future constellations.

Planning the mission operations for a single RPO satellite is a very tractable problem, but as the number of satellites within the constellation scales, the problem dramatically increases in complexity due to a variety of spatial and temporal dependencies [5]. As the planning system considers tasking for an individual satellite, it must do so in terms of the performance of the aggregate whole. To schedule RSO tasks for a constellation of RSO satellites, this paper integrates an established distributed tasking algorithm with a novel utility function to balance the value of completing RSO servicing tasks with the fuel costs of rendezvous and the limited number of RSO satellites. The result is a mission-focused, operationally relevant, scheduling solution for conducting the conceptualized RSO-servicing mission.

Note that throughout this paper the term "servicing" will be used to reference a variety of possible operations, including, but not necessarily limited to, RSO inspection, vehicle maintenance, commodity replenishment, component upgrade, or repair. The specific operations and detailed assessment of resources required to conduct those operations are outside the scope of this paper. However, the value of

Presented as Paper 19-648 at the 2019 AAS/AIAA Astrodynamics Specialist Conference, Portland, ME, August 12–15, 2019; received 23 April 2021; revision received 1 February 2022; accepted for publication 2 February 2022; published online 15 March 2022.

*Principal Mission Systems Engineer, Strategic and Military Space, 1695 Research Park Way (Corresponding Author).
†Principal GNC Engineer, Strategic and Military Space, 1695 Research Park Way.
‡Assistant Professor, Electrical and Computer Engineering, EL 148.
§Professor, Mechanical and Aerospace Engineering, ENGR 419B.

conducting these operations is realized when extending the life of expensive operational space systems. Keeping a satellite on mission, by supplementing existing capability (e.g., refueling), avoids the potential dramatic increase in cost to design, build, test, launch, and replace that system on orbit. Mission life extension and/or mission performance enhancement are the primary objectives driving a solution to this scheduling problem.

Currently within the spaceflight industry, mission planning operations generally employ centralized planning systems with elements of the ground system at one facility or at geographically dispersed facilities [6]. The centralized method employed in planning is common due to the many benefits that it provides in a traditional mission setting. A centralized mission planner is able to maintain a single situational, coordinated, set of states for the whole constellation and make decisions based on that knowledge, thus avoiding conflict with other competing schedulers. Tasking operations are maintained and fulfilled within a central database and provide a global snapshot of the entire mission system [7]. This configuration reduces the complexity of each satellite within the constellation because they need to fulfill only the operations assigned to them and are not required to make decisions for themselves or coordinate those actions with other satellites within the constellation. This methodology simplifies the complexity of the simulation models and reduces the sophistication required to resolve conflicts [8].

While this concept is likely to endure for the immediate future, the mass proliferation of satellites in low Earth orbit (LEO) will require more sophisticated operational capabilities [9,10]. For example, the planned Starlink constellation will comprise several thousand satellites in LEO [11]. Such a constellation must use some autonomy to avoid placing unrealistic demands on operations staff. Maneuvering satellites within a larger constellation, or formation, is one key element of properly managing these large systems, and a variety of planning methods have been proposed [12–15].

In analyzing multispacecraft OOS missions, research has typically focused on fuel-optimal sequencing for a single agent vehicle servicing multiple RSOs. Shen and Tsiotras evaluate fuel-optimal sequencing for visiting multiple vehicles in the same circular orbit, showing that the best solution is to visit them in orbit-wise or counter-orbit-wise order [16]. Zhang et al. solve a multispacecraft refueling problem in a near-circular LEO orbit using a hybrid-encoding genetic algorithm [17]. Their analysis includes J2 perturbations and time-window constraints based on the orbital beta angle to optimize fuel usage and mean mission time. They also show how splitting the RSO's into two subgroups, to be visited by separate servicing vehicles, affects the optimal solution. Optimal selection of the two subgroups is left to future work. Zhao et al. expand on this work by analyzing a cooperative multispacecraft refueling mission that includes constraints on the target vehicle's surplus propellant [18]. Verstraete et al. continue this line of operationally relevant servicing by looking at a geostationary orbit (GEO) servicing mission that accounts for the risk and value of accomplishing any given task. Their analysis shows that varying risk levels can significantly alter the mission objective function and the task sequencing to be implemented [19]. Zhang et al. evaluate the use of multiple servicing vehicles to visit a set of target vehicles using a multi-objective quantum-behaved particle swarm optimization [20]. Although it is unclear on whether the approach can be extended to have each servicer visit multiple targets or a single target each. Liu and Yang present the debris removal from LEO problem and provide a multi-objective optimization formulation and a modified genetic algorithm to solve for the sequential transfer route, similar to the RPO problem evaluated in this paper [21].

This paper expands the cooperative constellation use-case by developing an initial planning and scheduling methodology for a constellation of RPO-capable vehicles tasked with visiting and servicing a larger number of RSOs. Event scheduling is accomplished using the consensus-based bundle algorithm (CBBA), a distributed task allocation algorithm developed within the robotics and unmanned aerial vehicle community [22]. To evaluate the utility of each task, a novel utility function is developed that combines the score of a rendezvous event with the associated cost. The value of a rendezvous and servicing event is based on ground identified priorities, event timeliness, lighting dependencies (e.g., beta angle), and the current state of the constellation relative to RSOs of interest. Event cost focuses on expended fuel where rendezvous sequences are based on J2 orbital dynamics and impulsive ΔV maneuvers.[¶] This paper integrates CBBA with the large-scale constellation planning problem currently growing within the space community. The marriage of these two results in a timely, operationally feasible solution with good performance. It also allows for dynamic replanning due to the timeline of the resulting solution.

Event costs such as fuel and time are also considered. Even simplistic constellation scheduling scenarios, with their associated dependencies in time and space, result in an non-deterministic polynomial-time (NP)-hard optimization problem, meaning that for a realistic number of satellites and tasks, finding the true optimal solution may not be operationally realistic. This paper develops a novel utility function that can be used inside a CBBA framework for the management of a constellation of RPO-capable satellites for potentially suboptimal, but operationally relevant, scheduling solutions.

We recognize the reality that spacecraft constellation mission planning will likely remain a centralized concept for the foreseeable future. However, the contributions of this paper could be seen as a decision aide in the short-term while simultaneously providing value to future solutions. The remainder of this paper will proceed with an explanation of the problem in Sec. II, followed by an explanation of the rendezvous model in Sec. III. Section IV then presents the example scenario, and Sec. V discusses the resulting solution generated using the previously described methods and models. The paper concludes in Sec. VI with a summary of the accomplishments.

II. Classifying the Problem

This task allocation problems falls into the broader category of an assignment problem where the goal is to assign tasks to agents [23]. Early research into solving the assignment problem focused on matching a single agent to a single task, with efficient optimal solutions discovered in the 1950s [24]. Efficient, optimal distributed solutions known as auction algorithms were later introduced by Bertsekas in 1981 [25]. Since then, a striking number of similar problems have been identified and methods developed to solve them [26].

To associate potential solutions with task allocation problems, a taxonomy was developed by Gerkey and Matarić [27] that describes different types of tasks and the robots/agents capable of performing those tasks. Three axes of differentiation were discussed: whether the robot can perform a single task (ST) or multiple tasks (MT) at a time, whether the individual tasks can be performed by a single robot (SR) or multiple robots (MR), and whether the robot is capable of accepting only an instantaneous allocation (IA) or can be scheduled over an extended period of time (TA). Korsah et al. [28] have further extended this concept by accounting for dependencies that can exist between the tasks and the robots executing them. The dependency types were classified as no dependency (ND) between either the tasks or the agent schedules; in-schedule dependency (ID), where tasks may require a specified sequence for fulfillment; cross-schedule dependency (XD), where agents must coordinate schedules; and complex dependency (CD), where task fulfillment utility depends on the schedules of other agents in the system in a manner that is determined by the particular task decomposition chosen. A note of particular importance, made by Gerkey and Matarić, is that any problem beyond the [ST-SR-IA] is strongly NP-hard and thus precludes enumerative solutions due to computational complexity and the associated run time [27]. Such problems can leverage approximate algorithms to find a feasible solution, although suboptimal, within some acceptable mission time frame.

In this paper, the assignment problem is that of tasking of multiple satellite agents to optimally rendezvous with a larger set of RSO spacecraft. Using the taxonomies defined above, this problem falls

[¶]The details of the particular servicing operation are not specified. Rather, the method of efficiently moving the agent satellites into proximity of the RSOs to conduct those operations is discussed.

into the ID category due to having dependencies on the other tasks an agent is performing. Furthermore, each vehicle is only capable of servicing a single object at a time, and that object only requires a single satellite to perform the task, thus placing it into the [ST-SR-TA] category. The full categorization is then ID [ST-SR-TA] that, while not being the most complex problem, is still known to be NP-hard [28].

A. Linear Program Formulation

The intent of the agent scheduling problem is to maximize the global utility of allocating RSO tasks to each agent within the servicing constellation, subject to the constraints that only a single agent is assigned to each RSO task, the total number of assignments does not exceed the total number of RSO servicing tasks, and that the total ΔV for each vehicle is not exceeded due to its assigned tasking. Our particular scheduling problem, servicing RSOs with a finite set of agents, can be mathematically modeled by the integer program below:

$$\max\left(\sum_{i=1}^{N_{\text{agents}}}\left(\sum_{j=1}^{N_{\text{RSOs}}} u_{ij}(t)x_{ij}\right)\right) \tag{1}$$

$$\sum_{j=1}^{N_{\text{RSOs}}} x_{ij} \leq 1 \tag{2}$$

$$\forall\ i \in \text{agents},\ \sum_{j=1}^{N_{\text{RSOs}}} x_{ij}\Delta V_{ij} \leq T_{\Delta V_i} \tag{3}$$

$$x_{ij} \in \{0, 1\} \tag{4}$$

where $x_{ij} = 1$ if agent i is assigned to task j, 0 if not, and the time-dependent variable $u_{ij}(t)$ is the utility associated with the assigning agent i to perform task j. ΔV_{ij} is the sum total ΔV for agent i to transfer from its starting orbit, rendezvous with the RSO, and perform the RSO servicing task j, while $T_{\Delta V_i}$ is the total ΔV capacity of agent i. Note that $u_{ij}(t)$ is a combination of both the task score and the cost associated with the ΔV necessary to maneuver to the RSO and perform the servicing task, and it changes based on when the orbital maneuvers are initiated. The maneuvering ΔV is computed based on the orbital geometry and configuration over time while the actual servicing ΔV is assumed fixed due to the variety of actions that may occur for a specific mission. Further description of the ΔV cost is provided in Sec. III. Equation (1) represents the global utility function based on the assignments made during the task allocation process. Equation (2) applies the constraint that only a single agent is assigned to each RSO servicing task, while Eq. (3) constrains each agent to the ΔV available onboard. This constraint is unique to the standard CBBA implementation but is required to simulate a realistic mission scenario. Equation (4) ensures that the assignment indicator is equal either to 1 for assigned or 0 for not assigned.

B. CBBA Task Allocation Method

CBBA is a multirobot task allocation algorithm designed to be applied in a distributed fashion among a team of robots and, while potentially suboptimal, guarantees at least 50% of optimality, meaning that the performance result will be at least half of the maximum objective value for a nonnegative reward scenario [22]. The distributed nature of the algorithm requires that robots communicate with each other to develop a conflict-free allocation of tasking and removes the need for a moderator. This is accomplished by blending an auction-based routine with a deconfliction algorithm to ensure convergence. In CBBA, these routines are formally named the bundle phase and the consensus or deconfliction phase. During the bundle phase, each agent greedily adds tasks to its bundle and attempts to maximize the score of performing those tasks.

Bundle construction occurs by each agent continuously adding tasks to its bundle until it is incapable of adding any new tasks. During this time, the bundle list is updated along with the associated path. The path is the sequence the agent will take to fulfill the list of tasks identified within the bundle and it cannot exceed the ΔV resources onboard. The CBBA scoring scheme inserts new tasks at the location (in the path) that incurs the largest score improvement. The general flow during bundle construction is that for each task currently not in the bundle/path, an agent computes the score for the task. The score is checked against the current winning bids list and if the score is higher, the task is added to the bundle. The next step is then to insert the task (just added to the bundle) into the path. However, this insertion process must not alter the current timing of tasks already in the path. Doing so would change the current score of the path. The problem is to then attempt to maximize the score of the tasks planned while ensuring feasibility of all tasks currently in the agent's bundle and path. An agent continues this process in conjunction with resolving conflicts among its teammates during the consensus phase of the algorithm.

The consensus phase attempts to resolve conflicts between competing agents to ensure a conflict-free allocation of tasks across the constellation by evaluating the utility of assignment. Utility is computed as the difference between the score of the task and the cost of performing it (i.e., score—cost). The utility function is the primary means used to determine which robot should be assigned each task and specific rules govern tie-break scenarios (e.g., most recent information wins). Because CBBA is a distributed algorithm, agents must maintain several pieces of information to determine which tasks each individual robot will perform. The consensus portion of the algorithm corresponds to robots sharing portions of that information in order to converge to a conflict-free solution across the team. Information such as auction winners, their bids, and the timestamp associated with that information is shared to ensure convergence [7].

Deconfliction begins once agents have bundles built. Agents must communicate their plans with their neighbors to ensure that one task is not being fulfilled by multiple agents. After an agent receives information from neighboring agents, including the list of winning bids, the list of winning agents, and the timestamp of the information, it can determine if it has been outbid for a specific task or if more timely information is available. If an agent is outbid for a given task, it releases that task and all other subsequent tasks due to dependencies in the utility function related to the path (i.e., the sequence of RSOs visited).

Several decision rules dictate the response of an agent upon receiving information from its neighbor. There are three actions that can occur, from the receiving agent's perspective, during the convergence phase of CBBA. These are the following:

1) Update: Information received from neighbor indicates a new winner or more timely/relevant information, so the agent updates its information accordingly.

2) Reset: Discrepancy exists between what receiver and sender believe, so both reset their information.

3) Leave: Agreement between sender/receiver or receiver information is believed over sender's, so the agent does not change its information.

A variety of situations arise during conflict resolution, and a fully populated table is available in [22] to explain each possible scenario. Using the rules established leads to a deconflicted plan across the constellation of agents.

C. CBBA in the RPO Mission

CBBA is applied to find a solution to the RSO-servicing problem by using a utility function that combines the ΔV cost and time-varying score of each task. These parameters guide the CBBA process in properly allocating tasks across the constellation and attempting to maximize the overall mission utility. Servicing time plays a central role in this process and must be used to determine not only cost and score, but also the feasibility of actually performing the specified task since each RSO task is constrained to be performed during a specified window. To provide clarity, Sec. II.C.1 will first discuss how the feasibility windows are established and then describe how the utility function applies to the scheduling process. Following

that, the actual bundle building and deconfliction phases of CBBA will be summarized in Sec. II.C.2. The ΔV cost design and rendezvous model are discussed in Sec. III, while the score is presented in Sec. IV.

1. *Feasibility Window and Utility*

Two fundamental elements required for utilizing CBBA in RSO mission operations are the feasibility window and the resulting utility of a task. A feasibility window represents the time span within which a servicing agent satellite can perform the necessary orbital maneuvers and the servicing operation for the RSO. The start and end times of the feasibility window are primarily driven by the RSO task window that identifies times when an RSO can be made available for servicing. Additional considerations must also be made based on the availability of the servicing agent. These considerations include the ΔV resources available onboard the vehicle and existing tasks the agent is planned to fulfill. The feasibility window start time is determined by assessing the ΔV cost and associated altitude required to arrive at the RSO at a given time. If an agent is able to arrive at an RSO at or before the RSO task window opens, the RSO task window start time is used as the earliest time bound of the feasibility window. However, if the earliest time in the RSO-task window is not feasible, due to an altitude below the mission limit or a ΔV greater than what is available onboard the agent satellite, the feasible window start time is delayed in time until a valid time is found. Note that transfer altitudes that are below the mission limit are considered infeasible due to atmospheric drag and the associated risk of atmospheric reentry.

A similar process is followed for establishing the end time of the feasibility window. If an agent can leave the previously serviced RSO at the end of the allowable servicing window and arrive at its next RSO servicing task on time, the end time of the feasible window becomes the end time of the RSO task servicing window minus the duration of time required to perform the task. Otherwise, the end of the feasibility window for the RSO must be set earlier in time such that the agent has time to perform the orbital maneuvers required to reach the next RSO and perform the servicing task. As long as the feasibility window end time occurs after the feasibility window start time, the new RSO task is added to the bundle and given further consideration in the scheduling process. A graphical example of this process is shown in Fig. 1. The task servicing windows are shown as gray boxes, while the duration required for completing the servicing tasks are marked as blue boxes within the servicing window. Within Fig. 1, representative transfers are shown as arrows. If transfer 1 can be completed in time to arrive at the start of the servicing window of task 1, then $t_{\text{start}-1}$ becomes the start time of the feasibility window for task 1. Otherwise, if transfer 2 represents the earliest arrival time at task 1, then $t_{\text{start}-2}$ becomes the start time for the feasibility window. Similarly, if transfer 3 can be completed in time to allow for task 2 to be completed before the available servicing window closes, then $t_{\text{end}-1}$ will be the end of the feasibility window for task 1. However, if transfer 4 is the soonest the agent can arrive on station for completing task 2, then the feasibility window will have to accommodate this by starting the transfer earlier (by the amount shown by the yellow box) and setting the end time of the task 1 feasibility window to be $t_{\text{end}-2}$. This same process is followed every time a potential task is considered during scheduling. Doing this results in a realistic schedule, and ensures that it can be fulfilled.

With the feasibility window established, the algorithm then selects the time that maximizes the score within the feasibility window times. The cost of servicing the RSO at that time is finally computed and combined with the cost to calculate the value of the utility function. The utility function considers the following time-varying mission parameters:

ΔV (cost): The amount of ΔV required to perform an RPO servicing task given a schedule. Agent vehicles have a finite ΔV, limiting the amount of orbital maneuvers, and thus the number of RSO servicing tasks that each can perform.

Lighting conditions (score): The beta angle required for the rendezvous and servicing operations. This parameter may be due to specific attitude constraints on the RSO and the docking face used by the agent for servicing.

Estimated remaining life of RSO requiring servicing (score): This quantity helps to act like an urgency multiplier in that if a vehicle is reaching end of life, the scoring function will drive the value high to encourage timely servicing from the agent constellation.

The aggregate utility is the weighted sum of the individual scores and cost penalty where different weightings can be chosen according to mission priorities. This task information will then be shared among the constellation agents to determine a realistic RPO assignment allocation using CBBA. The utility associated with a particular RSO task is computed as

$$u_{ij}(t) = \begin{cases} v_{\text{life}_j}(t) + v_{\text{light}_j}(t) - c_{\Delta V_{ij}}(t) & \text{if } t \in \{\text{Feasible Window}\} \\ 0 & \text{if } t \notin \{\text{Feasible Window}\} \end{cases} \tag{5}$$

where $u_{ij}(t)$ is the utility of assigning agent i to RSO task j, $v_{\text{life}_j}(t)$ is the score associated with the remaining life of the RSO, $v_{\text{light}_j}(t)$ is the score due to lighting conditions for the RSO, and $c_{\Delta V_{ij}}(t)$ is the cost due to the required ΔV for the particular task assignment. Once again, the cost is discussed in detail in Sec. III and the utility in Sec. IV. The next subsection will now provide a brief discussion of how CBBA operates within the RSO mission scheduling framework previously discussed.

2. *Bundle Building and Deconfliction*

CBBA employs two phases: bundle building and task deconfliction or consensus. During phase 1, bundle building, each agent greedily attempts to place the most valuable tasks on its schedule at the most opportune times. Ideally, agents would greedily assemble the tasks based upon the full utility of each task. However, since the ΔV cost will be a function of the entire RSO-task schedule for an agent, solely the RSO-task score is used for initially selecting the

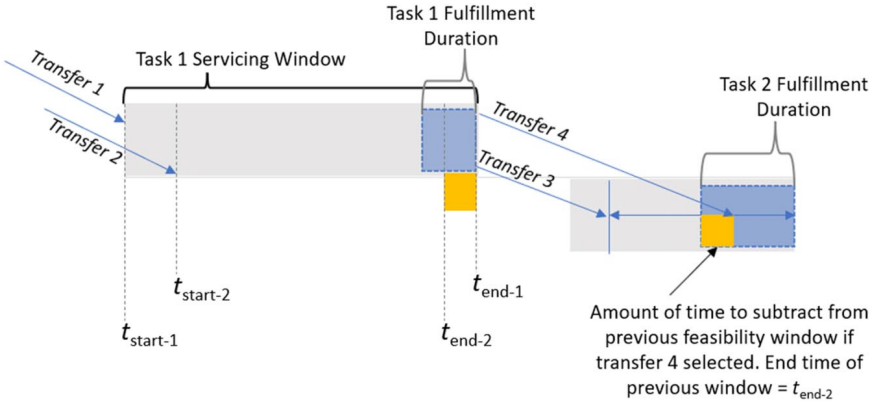


Fig. 1 Feasibility window considerations during scheduling.

servicing time during the bundle building process, while the full utility is updated each time a new task is inserted into, or removed from, the bundle. This resulting utility is used later during the consensus phase of CBBA. The bundling is accomplished by each agent cycling through the complete list of tasks and placing them individually on its schedule where it can assess the maximum score that each task generates. This process continues until the agent exhausts all of its ΔV resources or no additional tasks are available for bundling. To reiterate, the bundling process for each agent can be summarized by the following steps:

1) Find the task resulting in the highest score on agent i's schedule.
2) Place the highest scored task into agent i's bundle and schedule.
3) Recalculate ΔV_i with the new schedule.
4) If $\Delta V_i < T_{\Delta V_i}$, keep the task; otherwise, remove.
5) Repeat until no additional tasks are available or $T_{\Delta V_i}$ is exhausted.

Following the bundling process, agents enter the consensus phase by competing for each task using an auction method where each agent's bid is the utility they have computed during bundling. During deconfliction, each agent broadcasts information to its neighbors that informs them of each agent's bid, the current winning agent, and the timestamp associated with the information—all from the perspective of the broadcasting agent. With that shared information the constellation of agents can converge to a deconflicted task allocation that will be used for execution. Several rules govern how deconfliction takes place and how conflicts are addressed. These include basic rules such as a higher bid wins during auctioning but also additional rules about how to address disagreement using the most recent time-stamped information [22].

III. Cost Calculation and Rendezvous Model

The dynamics of the RSO-servicing problem are central to calculating the costs of performing the servicing tasks and determining the feasibility of performing the task within the allocated time frame of each task. The system dynamics are thus detailed in this section to inform the reader on the methods used in our model.

The rendezvous process can occur over a relatively large timeframe of months since we are relying on the Earth's J2 oblateness to drive orbital precession rates. The ΔV for completing a long-range rendezvous task is estimated through a four-step sequence that brings an active agent spacecraft into the same orbit as, and in proximity to, a passive RSO requiring servicing. The equations are shown here for explanatory purposes, and the interested reader can find them detailed further in the work of Vallado [29]. The rendezvous sequence is as follows and is performed for each agent with respect to each RSO:

Orbital node alignment (natural or forced): Accomplished via exploitation of the J2 Earth oblateness effect by raising or lowering the orbital semimajor axis.

Orbit altitude and inclination matching: Accomplished using plane change maneuvers and Hohmann transfers to match the RSO inclination and altitude, respectively.

Orbit phasing: Accomplished by performing a coplanar rendezvous phasing maneuver to match the true anomalies between the RSO and agent spacecraft.

Proximity operations: A fixed time (accounted as part of the task duration) and ΔV are assumed due to the diversity of tasks and uncertain conditions potentially encountered during operations.

This sequence of operations will be considered by the mission planner across the entire constellation to determine the best agent allocation for completing the servicing mission. The ΔV required to perform this sequence becomes the primary variable used in determining the cost of the servicing operation. That cost is combined with the score of the RSO servicing task to establish the utility of the task.

A. Orbital Node Alignment

Only near-circular orbits are considered in this paper. Thus, the drift or nodal regression rate is a function of the spacecraft's semimajor axis a, eccentricity e, and inclination inc [29]. For circular orbits, the equation simplifies to

$$\dot{\Omega} = \frac{-3}{2}\sqrt{\frac{\mu}{a^7}}R_\oplus^2 J_2 \cos(\text{inc}) \tag{6}$$

By assuming that these driving parameters do not change significantly over time, the right ascension of the ascending node (RAAN) of a spacecraft can be predicted at any future time as

$$\Omega_t = \Omega_0 + \dot{\Omega}_t(t - t_0) \tag{7}$$

The difference in RAAN between the agent and an RSO at any future time can be determined as

$$\Delta\Omega = \Omega_t^{\text{RSO}} - \Omega_t^{\text{agent}} = (\Omega_0^{\text{RSO}} - \Omega_0^{\text{agent}}) + (\dot{\Omega}_0^{\text{RSO}} - \dot{\Omega}_0^{\text{agent}})(t - t_0) \tag{8}$$

The two spacecraft's orbital nodes align when they have matching RAAN ($\Delta\Omega = 0$) at time t. If two spacecraft are separated in RAAN yet have similar altitude and inclination, the time required for the two nodes to align may be large or impractical, depending on the mission requirements, to meet the servicing objective. Under these circumstances, it is possible to adjust the drift rate of the agent spacecraft by increasing or decreasing the agent's orbital altitude. The desired drift rate to match nodes at a future time is calculated as

$$\dot{\Omega}^* = \dot{\Omega}_{\text{RSO}} + \frac{\Delta\Omega_0}{t} \tag{9}$$

Given a desired drift rate, the necessary semimajor axis a^* of the agent orbit can be calculated as

$$a^* = \left[\frac{9}{4}\frac{\mu}{(\dot{\Omega}^*)^2}R_\oplus^4 J_2^2 \cos^2(\text{inc})\right]^{1/7} \tag{10}$$

Altitude changes are accomplished using a two-maneuver Hohmann transfer where the initial and final maneuvers are calculated as

$$\Delta V_1 = \sqrt{\frac{\mu}{a_1}}\left\{\sqrt{\frac{2a_2}{a_1 + a_2}} - 1\right\} \tag{11}$$

$$\Delta V_2 = \sqrt{\frac{\mu}{a_2}}\left\{\sqrt{\frac{2a_1}{a_1 + a_2}} - 1\right\} \tag{12}$$

where a_1 and a_2 represent the semimajor axis of the initial and final circular orbits in increasing order [29].

B. Orbit Phasing

After matching orbital nodes, the agent continues the rendezvous sequence by matching the RSO's orbit. Altitude matching is accomplished by another set of Hohmann transfer maneuvers from the drift altitude a^* to the RSO's altitude. Inclination matching is achieved by a single ΔV maneuver applied perpendicular to the orbital plane of the circular orbit, and at the orbital node, to the magnitude calculated as

$$\Delta V_i = 2\sqrt{\frac{\mu}{a}}\sin\left(\frac{\Delta\text{inc}}{2}\right) \tag{13}$$

where Δinc is the difference in inclination between the two orbits [29].

C. Orbital Rendezvous

After matching the RSO's orbit (RAAN, inclination, and altitude), the agent vehicle must perform rendezvous maneuvers to match the RSO's true anomaly. This rendezvous operation is accomplished using a coplanar phasing maneuver for circular orbits. The phase angle θ between the RSO and agent spacecraft determines which

orbital maneuver is performed. The angle θ is measured starting from the RSO and moving toward the agent, and defined as positive in the direction of orbital motion and negative otherwise. If the RSO lags behind the agent satellite (positive θ), as shown in the left of Fig. 2, the agent raises apogee to allow the RSO to catch up, whereas, when the RSO leads the agent (negative θ), as shown in the right side of Fig. 2, the agent performs a maneuver to lower perigee and catch up to the RSO [29]. The phasing maneuver is performed to phase the satellites over a period of time that only induces a minor change in semimajor axis. This minor perturbation to RAAN, between the two vehicles, is neglected due to the long-term nature of the planning operations. The resulting phasing ΔV is the principle value of interest and provides a guiding factor for estimating the cost of the phasing operation. The ΔV required for this operation is determined by the following sequence of calculations:

$$\tau_{\text{phase}} = \frac{2\pi k_{\text{RSO}} + \theta}{\omega_{\text{agent}}} \tag{14}$$

$$a_{\text{Phase}} = \left(\mu\left(\frac{\tau_{\text{phase}}}{2\pi k_{\text{agent}}}\right)^2\right)^{1/3} \tag{15}$$

$$\Delta V_{\text{Phasing}} = 2\left|\sqrt{\frac{2\mu k_{\text{RSO}} + \theta}{\omega_{\text{agent}}} - \frac{\mu}{a_{\text{Phase}}}} - \sqrt{\frac{\mu}{a_{\text{agent}}}}\right| \tag{16}$$

The variable τ_{Phase} represents the time over which the phasing will occur, while a_{Phase} is the semimajor axis of the phasing orbit and a_{agent} the semimajor axis of the interceptor's orbit. The variables k_{RSO} and k_{agent} are the number of orbital revolutions (target and interceptor, respectively) that are used for completing the phasing operation and always equal to each other for this calculation in the algorithm; ω_{agent} is the angular orbit rate of the agent; $\Delta V_{\text{Phasing}}$ is then the complete ΔV for the phasing operation. The factor of 2, in its equation, is due to the fact that the initial ΔV required to generate the phasing orbit must be performed in equal magnitude but opposite direction to re-enter the initial circular orbit once the proper phasing is achieved.

D. Proximity Operations

Due to the diversity of actions that would be required for the individual proximity operations, this paper assumes a conservative, and fixed ΔV, of 25 m/s to complete for each RSO task with the duration being accounted for in the servicing task duration. This 25 m/s of ΔV is applied as part of the cost calculation routine when new RSO tasks are being considered for assignment. The primary intent of using the 25 m/s placeholder is to specifically indicate that ΔV resources have been reserved for these essential operations and are not neglected in the problem formulation.

IV. Scoring and Cost Functions

The score functions are independently computed and then combined for a time-varying score that spans the duration of the simulation. Both the lifetime and lighting values use an exponential function in the calculations to drive value high when lighting conditions meet requirements or when little operational life remains. Exponential functions were selected to dramatically peak at preferred times and differentiate the RSO task value significantly—greatly increasing the likelihood of being scheduled. The task score associated with lifetime is computed using the equation below:

$$v_{\text{life}} = W_{\text{life}} \cdot e^{\lambda_{\text{life}} \cdot f_{\text{life}}(t)} \tag{17}$$

where W_{life} is the life score weight and λ_{life} is the scale factor. The scale factor λ_{life} is the parameter in the function that allows a user flexibility in defining the shape of the remaining life scoring function. Each of the scoring functions uses the scaling factors in similar ways. The variable f_{life} is the fraction of life left for the particular RSO (between 0 and 1). The fraction of life left for the RSO is referenced relative to the design life and calculated by

$$f_{\text{life}}(t) = 1 - \left(\frac{L_p(t)}{L_d}\right) \tag{18}$$

where $L_p(t)$ is the amount of time passed since operations started for the particular RSO and L_d is the design life, or anticipated life, of the RSO vehicle. For example, if 5 years have elapsed from an anticipated vehicle life of 10 years, f_{life} would be equal to 0.5. Each RSO is assumed to have a finite lifetime represented by a vehicle capacity. This capacity may represent finite propellant or failing components that need to be serviced before failure. The parameters give a scheduling system flexibility in using this score to drive overall tasking as it pertains to boosting the score based on the remaining life of the vehicle. Some lifetime value functions, using example values for λ_{life}, are shown in Fig. 3. The lifetime plot indicates a full life as being equal to 1 and could vary significantly in actual length (e.g., months to tens of years) depending on both design processes as well as current state of a vehicle's consumables or component states.

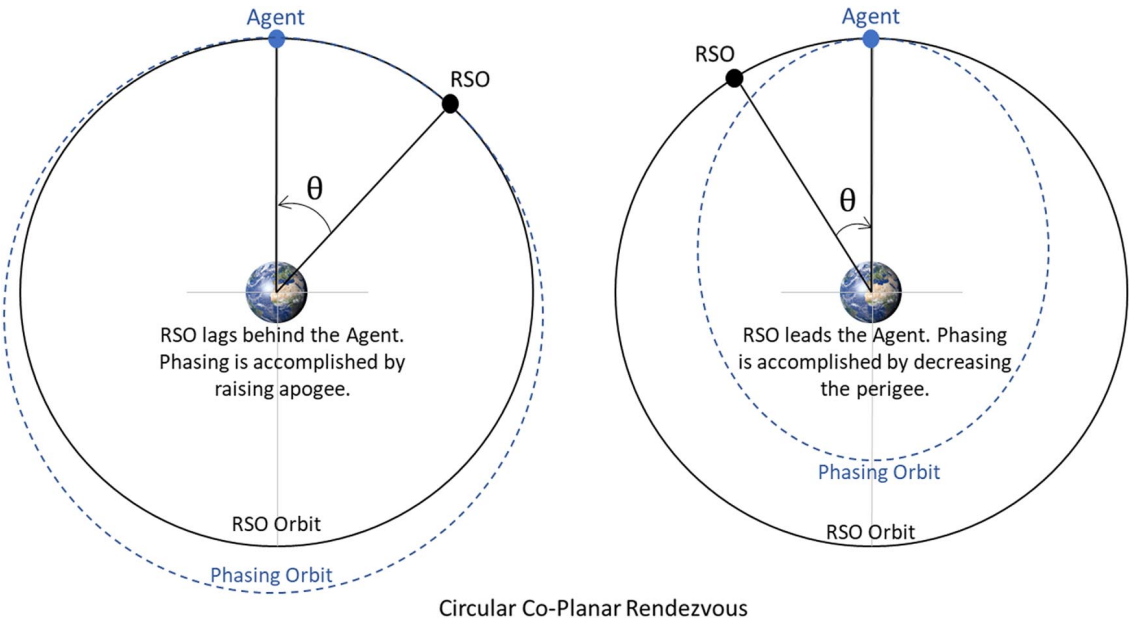


Fig. 2 Circular coplanar rendezvous maneuvers used for phasing the orbits between an agent and the RSO to be serviced.

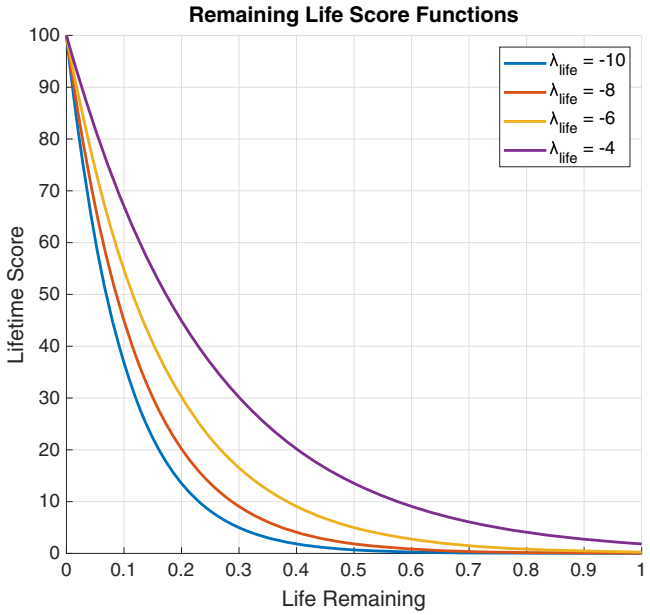


Fig. 3 Life score functions for various λ_{life} scale factors from full life to end of life.

The ability to perform proximity operations may be limited to certain lighting conditions due to either the agent or RSO's capability limitations (e.g., power or thermal requirements, or viewing geometry constraints). For analysis purposes, this constraint is modeled as the angle between the RSO's orbit plane and the sun vector, known as the beta angle [β in Eq. (20)], and calculated as

$$\sin(\beta) = \sin(\text{inc})\cos(\Omega)\cos(\Lambda_{\odot}) - \sin(\text{inc})\cos(\Omega)\sin(\Lambda_{\odot})\cos(\epsilon) + \cos(\text{inc})\sin(\Lambda_{\odot})\sin(\epsilon) \tag{19}$$

where $\lambda_{\odot}$ is the sun's ecliptic longitude and ϵ is the obliquity of the ecliptic [30]. The lighting conditions are quantified using a similar equation as the lifetime. The lighting score is calculated by

$$v_{\text{light}} = W_{\text{light}} \cdot e^{\left(\lambda_{\text{light}} \frac{|\beta_{\text{req}} - \beta(t))|}{180}\right)} \tag{20}$$

where W_{light} is the weight factor for the lighting, λ_{light} is the scale factor, β_{req} is the required beta angle for the operation, and $\beta(t)$ is the RSO orbit's beta angle in time. These β values are between -180 and $+180$ deg. Figure 4 illustrates several example lighting score functions using different magnitudes for the scale factor λ_{light}. The lighting score plot in Fig. 4 shows the score as only a function of beta angle but the beta angle is a strong function of time itself and dependent on the RSO vehicle's orbital parameters [reference Eq. (19)]. The lighting score functions shown are computed for an example required beta angle (β_{req}) of 20 deg.

The lighting and remaining life scores are then additively combined to generate a final score profile over the simulation timeframe. This score profile is then ultimately differenced with the ΔV cost profile to arrive at the utility function profile that is considered during planning.

The score assigned to each RSO task varies over the valid time window due to the lighting and lifetime contributions. The lighting score dominates due to its relatively short time constant when compared with the lifetime contribution. Again, the assigned validity window simulates some operational preference not captured by the lifetime or lighting calculation while the final score factors these items together before planning.

The cost associated with the ΔV takes a similar form to the score functions in that the cost gets greater as the remaining ΔV capacity of the agent approaches zero. This behavior was selected since the operations center may prefer to first use agent vehicles with large remaining capacity (less costly) while being more reserved with using agent vehicles that are almost out of fuel (more costly). The cost function, for agent i, is computed for each task j as follows:

$$c_{\Delta V_{ij}} = W_{\Delta V} \cdot e^{\lambda_{\Delta V} \cdot f_{\Delta V_i}} \tag{21}$$

where $W_{\Delta V}$ is the weighting factor for the cost, $\lambda_{\Delta V}$ is the scale factor, and $f_{\Delta V_i}$ is the remaining ΔV capacity for agent i (between 0 and 1). The variable $f_{\Delta V_i}$ is recalculated every time a new RSO task is considered for inclusion in agent i's task list. The cost is unique to the score function calculations in that it is recalculated based on the current task list of the agent. For example, if an agent is currently assigned a list of tasks and by adding another task, it would reduce the available ΔV capacity to 0, then the cost associated with the new task would be equal to the weighting factor. If the resulting cost were greater than the associated task value, the task would be removed from consideration in the current position in the task queue. Similarly, if any RSO task would consume more than the available ΔV, the task is considered infeasible and removed from consideration as well. Figure 5 illustrates some ΔV cost functions based on a weighting factor ($W_{\Delta V}$) of 100 and the example values for $\lambda_{\Delta V}$, as captured in the legend.

The score and cost approach outlined above gives an operator, or analyst, significant control of planning system behavior. The various parameters discussed allow for setting different sensitivities as well as overall magnitude for both cost and score. These parameters must be properly tuned and calibrated for the intended mission before deployment. As a simple example, Fig. 6 shows the resultant utility, as well as the associated cost and score, for an agent satellite with 800 m/s of ΔV capability onboard and the scoring and cost parameters as summarized in Table 1. The figure highlights the strong time dependency, and sinusoidal behavior, of the score. An example feasible window is overlaid on the figure for visualization purposes only. During normal planning the start and end of this window would be calculated as discussed in Sec. II.C.1. Note that the max score value (approximately 200 at week 40), within the valid time window, will be used as the task fulfillment time and the corresponding utility at that time will be used in the bidding competition portion of the CBBA.

The scoring and utility functions are composed of several parameters, as discussed previously in this document. The resultant utility values can be tuned for particular behaviors across the constellation of agents in order to yield the desired outcome. For example, if an agent constellation manager wanted to ensure only servicing very high value tasks, an operator could set the ΔV weighting factor $W_{\Delta V}$ high or vary $\lambda_{\Delta V}$ to a more impactful cost. These parameters can be used to drive costs relatively high, thus ensuring that only RSO servicing tasks with large values would be allowed for consideration during task allocation.

V. Simulation and Results

To create a varied operational scenario for this paper, where satellites have different altitudes, inclinations, and right ascension values, one agent seed satellite is specified and then other agents are created with random variations in those same orbital elements. A similar approach is taken when generating the RSOs, except that valid servicing time periods are established as well as task completion duration, remaining lifetime, and preferred beta angle. Running this randomizing routine generates a fleet of agents and RSOs with realistic orbital locations and task fulfillment parameters that are used to assess the score, and subsequent utility, of each RSO servicing task.

The lighting and remaining life scores are then fed into a scoring function that generates a time-based score function over the duration of the simulation. This time-varying function is then assigned to the particular RSO during its valid servicing time window.

The orbital configuration and scoring parameters specified in our mission scenario contained a diverse list of schedulable tasks and provided the CBBA with a variety of planning options. It also attempted to create a realistic planning situation for a constellation

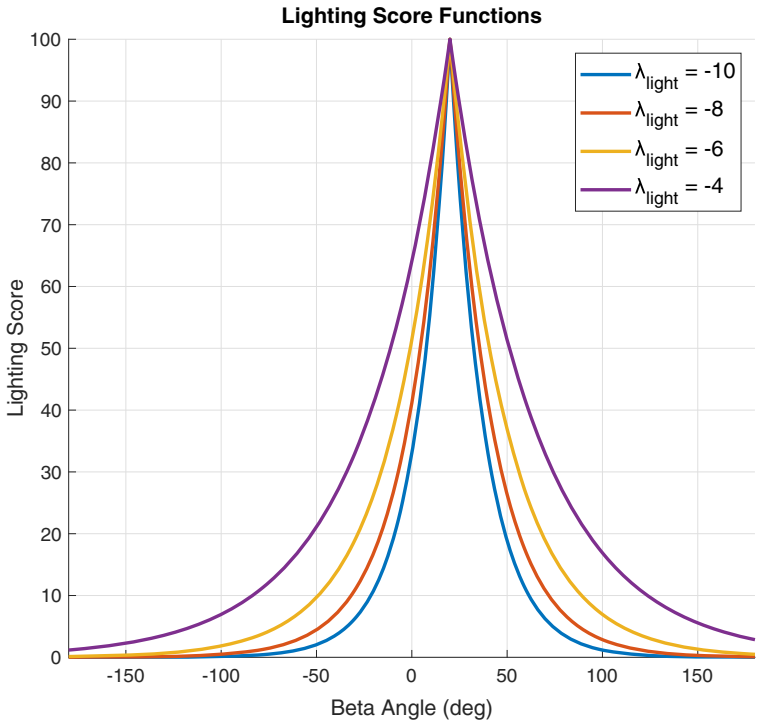


Fig. 4 Example lighting score functions using a weighting factor W_{light} of 100, a required beta angle β_{req} of 20°, and various example values for λ_{light}.

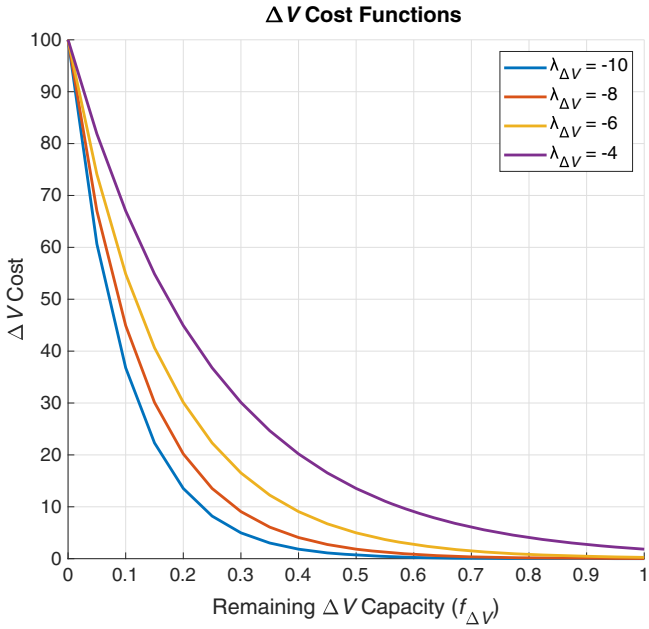


Fig. 5 Example ΔV cost functions using a weighting factor $W_{\Delta V}$ of 100 and different values for $\lambda_{\Delta V}$.

scheduling system. With the RSO tasks established and the scoring and cost parameters specified, it is then possible to feed that information into the RPO version of CBBA for bundle building and deconfliction operations.

An agent/RSO simulation was developed to exercise the RPO CBBA planning routine under specific orbital conditions. A test scenario was generated for the task scheduling problem using 3 active agents and 15 passive RSOs. The initial conditions for the agent satellites are summarized in Table 2, while the RSO orbital and task parameters considered during planning are shown in Table 3. Table 3 shows the start and end of the available RSO servicing window, the time required to fulfill the RSO servicing task, the required beta angle, and the remaining and design life for each RSO. Note that the orbital configuration selected for this scenario is similar to the configuration of the planned StarLink constellation. The range of altitudes of the constellation to be serviced in this paper is 505–577 km, while the orbital inclinations range between 52.7 and 54.0°. StarLink is currently planned to occupy an altitude of 550 km with a 53° inclination [11].

The weighting and scale factors used in this scenario match those contained in Table 1. The lighting and lifetime values were calculated based on the observations within the task fulfillment windows and then combined to yield the time-varying task score. The ΔV cost was computed based on the orbital configuration of the agent/RSO pair at each specific time considered by the scheduler.

A. Resource-Constrained Results

The CBBA method was applied to the 3-agent/15-RSO problem with individual agent ΔV capabilities as identified in Table 4. For this simulation, the minimum allowable altitude was set at 300 km and the time allowed for orbit true anomaly matching was capped at 1 week, while the ΔV required for every proximity operation was set at 25 m/s. The value of 25 m/s represents a conservative estimate based on prior operational mission experience. With these specifications, and available resources, the algorithm was able to assess utility of RSO servicing tasks within the available resources of each agent satellite. Table 4 also summarizes the ΔV consumed and remaining, following execution of the planned schedule. Note that agent 2 is left with only 38 m/s of ΔV at the end of the servicing operations. To avoid exhausting all ΔV resources or to ensure sufficient capacity at the end of an agent's mission, some ΔV capacity can be withheld from the scheduler for planning.

Out of the 15 RSO servicing tasks, only 11 were scheduled due to agent ΔV limitations. The resulting global utility score was 1715, while the total consumed ΔV was 2187 m/s. A summary of these results is presented in Table 5. The final schedule for this 3-agent/15-RSO planning scenario is represented graphically in Fig. 7, which illustrates the available RSO servicing windows using a gray box with the actual fulfillment time duration using a colored box within that window. The particular RSO being serviced is labeled on the y axis, while the x axis shows the week within the simulation time frame. Each agent's path is marked with a colored and dashed line to

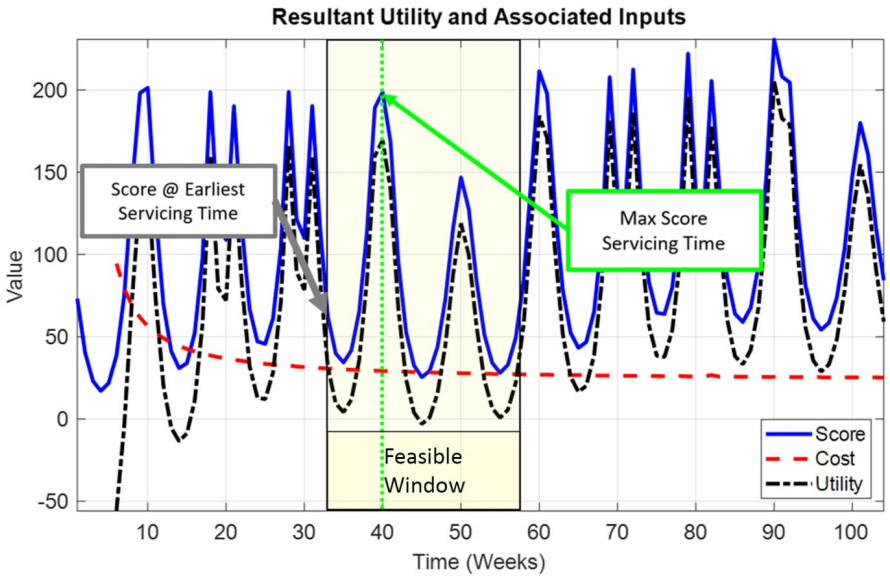


Fig. 6 Example utility, score, and cost plotted as a function of time overlaid with a representative feasible time window for visualization.

Table 1 Scoring and cost parameters used to establish an example utility profile

Parameter	Value
W_{life}	50
λ_{life}	−5
W_{light}	200
λ_{light}	−5
β_{req}	38.8°
$W_{\Delta V}$	100
$\lambda_{\Delta V}$	−2
$f_{\Delta V}$ @ Maneuver time	800 m/s

Table 2 Agent orbital configuration and available ΔV at the beginning of the planning scenario

Agent ID	a, km	inc, °	Ω, °	ΔV, m/s
1	6940.7	53.01	24.4	800
2	6947.8	53.04	35.8	900
3	6929.0	52.92	53.6	750

Table 3 RSO orbital configurations and planning parameters at the beginning of the planning scenario

RSO ID	a, km	inc, °	Ω, °	Start, wk	End, wk	Duration, wk	β_{req}, °	Remaining life
1	6910.0	52.92	1.41	36	49	3.7	37	0.73
2	6954.8	52.75	24.97	72	90	1	83	0.70
3	6939.1	52.90	−19.01	23	38	1.8	43	0.83
4	6927.2	52.94	69.42	97	103	1	8.9	0.10
5	6925.0	52.84	30.52	58	68	3.3	16	0.91
6	6922.0	53.18	57.45	11	15	2.2	45	0.49
7	6942.6	53.05	12.42	14	36	2.2	83	0.94
8	6940.7	53.08	3.05	76	96	2.9	36	0.55
9	6936.9	53.99	29.45	57	71	2	33	0.69
10	6932.7	52.92	−8.28	77	82	3.9	32	0.71
11	6925.9	52.93	39.96	61	65	3	31	0.86
12	6932.2	52.94	45.36	76	95	1.4	13	0.50
13	6905.1	53.06	−2.29	91	103	2.1	87	0.67
14	6882.5	52.84	42.46	70	85	2.1	73	0.23
15	6910.7	52.96	1.72	13	20	1.8	30	0.88

Table 4 ΔV propulsion resources, scheduled consumption, and resources remaining after task fulfillment

Agent	Available ΔV, m/s	Consumed ΔV, m/s	Remaining ΔV, m/s
1	800	651	149
2	900	862	38
3	775	675	100

Table 5 Mission performance summary for the 3-agent/15-RSO planning scenario

Quantity	Performance value
Total ΔV consumed	2187 m/s
Total RSO tasks scheduled	11
Total utility	1715
Utility/ΔV	0.78

indicate the path taken through the RSO servicing sequence. Similar information is summarized in Table 6.

B. Algorithm Scalability

Earlier in this paper, CBBA was highlighted as a mission-relevant planning algorithm. That statement was based on anticipated mission use cases (numbers of agents and RSOs). To further determine its relevancy, several scenarios were generated and analyzed using the RPO constellation planning algorithm discussed. The results indicate that the planning capability does scale and could support larger constellations, up to a point. Table 7 shows the computational solve time required to arrive at a deconflicted solution for the planning scenarios when running MATLAB 2020a on an Intel Xeon CPU E3-1270 processor @ 3.6 GHz with 32 GB of RAM. Table 8 shows the number of tasks allocated to the agent constellation that resulted for each of the scenarios listed. All scoring function parameters match those from the example scenario (Sec. V.A), and the orbital configurations were randomly generated using the same basic altitude, inclination, and RAAN variations as the example scenario. The timelines shown provide substantiating evidence that the CBBA could support relatively large numbers of agent satellites and RSO tasks. Extending the algorithm further to 25 agents and 125, 250, and 500 RSO tasks resulted in 4.2, 13.1, and 37.1 h of computation time for a converged solution, respectively.

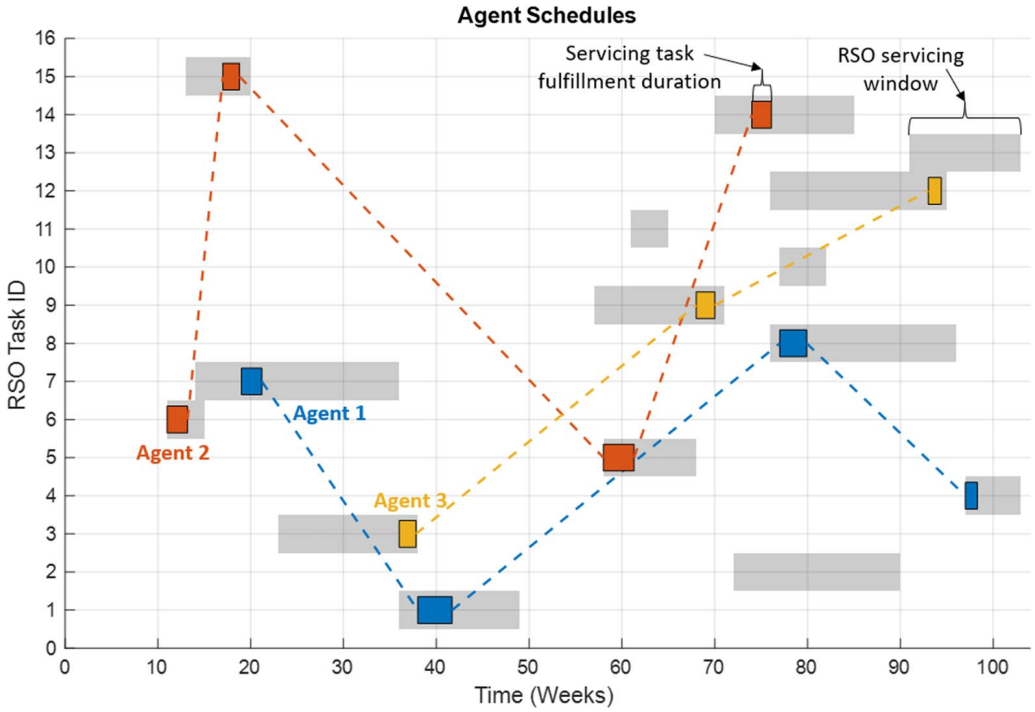


Fig. 7 Task assignments over the scenario timeline, with gray regions representing valid task windows and the highlighted boxes representing the period over which an agent performs the RSO servicing.

Table 6 Summary of the resulting schedule from the 3-agent/15-RSO planning scenario

Agent	RSO task	Start time, wk	Finish time, wk
1	7	19	21.2
	1	38	41.7
	8	77	79.9
	4	97	98.3
2	6	11	13.2
	15	17	18.8
	5	58	61.3
	14	74	76.1
3	3	36	37.8
	9	68	70
	12	93	94.4

Table 7 Computational time required for planning various numbers of agents satellites and RSO tasks

Computation time required, min						
		RSO tasks				
		20	40	60	80	100
Agent satellites	3	0.6	2.9	3.6	3.9	5.5
	6	1.7	6.9	10.9	13.2	24.4
	9	3.0	10.0	26.7	33.7	41.9
	12	2.9	15.0	32.2	50.8	72.7
	15	3.5	16.5	36.2	77.5	100.7
	18	4.3	17.0	43.1	83.2	127.3

Table 8 Number of RSO tasks fulfilled by number of agents satellites and RSO tasks

Total number of tasks allocated						
		RSO tasks				
		20	40	60	80	100
Agent satellites	3	11	18	16	15	14
	6	15	27	30	28	36
	9	20	32	46	44	47
	12	17	38	51	58	63
	15	20	37	51	70	75
	18	19	36	55	70	84

VI. Conclusions

This paper demonstrates the efficacy of using the established CBBA for tasking satellites, within a constellation, capable of rendezvousing with and servicing RSO vehicles. The tasking is done while considering the ΔV constraints of the servicing vehicles. While the CBBA results discussed in this paper have not been shown to be optimal, they were generated on a mission-realistic timeline that fits within operational scheduling requirements of satellite constellations. A scoring methodology was developed to balance the benefit of performing the servicing operation with the cost of performing rendezvous using a simplified J2 orbital dynamic model with relevant lighting and lifetime preferences, as well as individual agent ΔV constraints. Note that, while current technology is lacking and unable to support the ΔV levels this paper illustrates (up to 900 m/s), we do anticipate future systems carrying much more ΔV capacity and potentially reaching the levels highlighted.

Unlike previous papers, this work looks to solve the operationally relevant tasking of multiple servicing vehicles against a larger set of RSO vehicles to be serviced. Additional analysis shows that the algorithm can scale effectively for constellations of approximately 20 agents and 100 RSOs. The assignment problem has direct application in constellation scheduling and shows significant promise for further application in this burgeoning, and immediately relevant field.

References

[1] Davis, J. P., Mayberry, J. P., and Penn, J. P., "On-Orbit Servicing: Inspection Repair Refuel Upgrade and Assembly of Satellites in Space," The Aerospace Corporation Rept., 2019, https://aerospace.org/sites/default/files/2019-05/Davis-Mayberry-Penn_OOS_04242019.pdf.

[2] Stodgell, T., and Spencer, D., "Satellite Rendezvous Tours Using Multi-objective Evolutionary Optimization," *Astrodynamics 2007—Advances in the Astronautical Sciences, Proceedings of the AAS/AIAA Astrodynamics Specialist Conference*, Univelt, Inc., San Diego, CA, 2008, pp. 2069–2094.

[3] Martin, A.-S., "Legal Approach on the Dual-Use Nature of On-Orbit Servicing Programs," *On-Orbit Servicing: Next Generation of Space Activities*, Springer, Cham, Switzerland, 2020, pp. 1–11. https://doi.org/10.1007/978-3-030-51559-1_1

[4] Hruby, V. J., DeLuccia, C., and Williams, D., "Tethered Robot for Spacecraft Self Inspection and Servicing," *2018 AIAA SPACE and Astronautics Forum and Exposition*, AIAA Paper 2018-5342, 2018. https://doi.org/10.2514/6.2018-5342

[5] Ullman, J. D., "NP-Complete Scheduling Problems," *Journal of Computer and System sciences*, Vol. 10, No. 3, 1975, pp. 384–393. https://doi.org/10.1016/S0022-0000(75)80008-0

[6] Yost, B. D., Mayer, D. J., Burkhard, C. D., Weston, S. V., and Fishman, J. L., "Small Spacecraft Systems Virtual Institute's Federated Databases and State of the Art of Small Spacecraft Technology Report," Paper SSC18-IV-06, Utah State Univ., Logan, UT, 2018.

[7] Johnson, L., Ponda, S., Choi, H.-L., and How, J., "Improving the Efficiency of a Decentralized Tasking Algorithm for UAV Teams with Asynchronous Communications," *AIAA Guidance, Navigation, and Control Conference*, AIAA Paper 2010-8421, 2010. https://doi.org/10.2514/6.2010-8421

[8] Chien, S., Barrett, A., Estlin, T., and Rabideau, G., "A Comparison of Coordinated Planning Methods for Cooperating Rovers," *Proceedings of the Fourth International Conference on Autonomous Agents*, Assoc. for Computing Machinery, New York, 2000, pp. 100–101. https://doi.org/10.1145/336595.337057

[9] Frank, J., Jonsson, A., Morris, R., Smith, D. E., and Norvig, P., "Planning and Scheduling for Fleets of Earth Observing Satellites," *International Symposium on Artificial Intelligence, Robotics, Automation and Space*, Assoc. for Computing Machinery, New York, 2001.

[10] Ross, I. M., and D'Souza, C. N., "Hybrid Optimal Control Framework for Mission Planning," *Journal of Guidance, Control, and Dynamics*, Vol. 28, No. 4, 2005, pp. 686–697. https://doi.org/10.2514/1.8285

[11] Foust, J., "SpaceX's Space-Internet Woes: Despite Technical Glitches, the Company Plans to Launch the First of Nearly 12,000 Satellites in 2019," *IEEE Spectrum*, Vol. 56, No. 1, 2018, pp. 50–51. https://doi.org/10.1109/MSPEC.2019.8594798

[12] Campbell, M., "Planning Algorithm for Large Satellite Clusters," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, AIAA Paper 2002-4958, 2002. https://doi.org/10.2514/6.2002-4958

[13] Lee, S., and Hwang, I., "Hybrid High-and Low-Thrust Optimal Path Planning for Satellite Formation Flying," *AIAA Guidance, Navigation, and Control Conference*, AIAA Paper 2012-5046, 2012. https://doi.org/10.2514/6.2012-5046

[14] Mueller, J. B., and Larsson, R., "Collision Avoidance Maneuver Planning with Robust Optimization," *International ESA Conference on Guidance, Navigation and Control Systems*, Citeseer, 2008, http://support.psatellite.com/papers/ESAGNC2008_RobustAvoid.pdf.

[15] Sun, D., Zhou, F., and Zhou, J., "Trajectory Planning and Control for Satellite Formation Flying Maneuver of Leaving and Joining," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, AIAA Paper 2003-5588, 2003. https://doi.org/10.2514/6.2003-5588

[16] Shen, H., and Tsiotras, P., "Optimal Scheduling for Servicing Multiple Satellites in a Circular Constellation," *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, AIAA Paper 2002-4907, 2002. https://doi.org/10.2514/6.2002-4907

[17] Zhang, J., Parks, G. T., Luo, Y.-Z., and Tang, G.-J., "Multispacecraft Refueling Optimization Considering the J2 Perturbation and Window Constraints," *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 1, 2014, pp. 111–122. https://doi.org/10.2514/1.61812

[18] Zhao, Z., Zhang, J., Li, H.-Y., and Zhou, J.-Y., "LEO Cooperative Multi-Spacecraft Refueling Mission Optimization Considering J2 Perturbation and Target's Surplus Propellant Constraint," *Advances in Space Research*, Vol. 59, No. 1, 2017, pp. 252–262. https://doi.org/10.1016/j.asr.2016.10.005

[19] Verstraete, A. W., Anderson, D., St. Louis, N. M., and Hudson, J., "Geosynchronous Earth Orbit Robotic Servicer Mission Design," *Journal of Spacecraft and Rockets*, Vol. 55, No. 6, 2018, pp. 1444–1452. https://doi.org/10.2514/1.A33945

[20] Zhang, J., Zhang, Y., and Zhang, Q., "On-Orbit Servicing Mission Planning for Multi-Spacecraft Using CDPSO," *International Conference on Swarm Intelligence*, Springer, Cham, Switzerland, 2016, pp. 11–18. https://doi.org/10.1007/978-3-319-41009-8_2

[21] Liu, Y., and Yang, J., "A Multi-Objective Planning Method for Multi-Debris Active Removal Mission in LEO," *AIAA Guidance, Navigation, and Control Conference*, AIAA Paper 2017-1733, 2017.

[22] Choi, H.-L., Brunet, L., and How, J. P., "Consensus-Based Decentralized Auctions for Robust Task Allocation," *IEEE Transactions on Robotics*, Vol. 25, No. 4, 2009, pp. 912–926. https://doi.org/10.1109/TRO.2009.2022423

[23] Burkard, R., Dell'Amico, M., and Martello, S., *Assignment Problems*, Revised Reprint, Vol. 106, SIAM, Philadelphia, PA, 2012, pp. 73–89, Chap. 4. https://doi.org/10.1137/1.9781611972238

[24] Kuhn, H. W., "The Hungarian Method for the Assignment Problem," *Naval Research Logistics Quarterly*, Vol. 2, Nos. 1–2, 1955, pp. 83–97. https://doi.org/10.1002/nav.3800020109

[25] Bertsekas, D. P., "A New Algorithm for the Assignment Problem," *Mathematical Programming*, Vol. 21, No. 1, 1981, pp. 152–171. https://doi.org/10.1007/BF01584237

[26] Pentico, D. W., "Assignment Problems: A Golden Anniversary Survey," *European Journal of Operational Research*, Vol. 176, No. 2, 2007, pp. 774–793. https://doi.org/10.1016/j.ejor.2005.09.014

[27] Gerkey, B. P., and Matarić, M. J., "A Formal Analysis and Taxonomy of Task Allocation in Multi-Robot Systems," *International Journal of Robotics Research*, Vol. 23, No. 9, 2004, pp. 939–954. https://doi.org/10.1177/0278364904045564

[28] Korsah, G. A., Stentz, A., and Dias, M. B., "A Comprehensive Taxonomy for Multi-Robot Task Allocation," *International Journal of Robotics Research*, Vol. 32, No. 12, 2013, pp. 1495–1512. https://doi.org/10.1177/0278364913496484

[29] Vallado, D. A., *Fundamentals of Astrodynamics and Applications*, Vol. 12, Springer Science & Business Media, Dordrecht, The Netherlands, 2001, pp. 308–317.

[30] Fehse, W., *Automated Rendezvous and Docking of Spacecraft*, Vol. 16, Cambridge Univ. Press, Cambridge, England, U.K., 2003, pp. 126–132, Chap. 5. https://doi.org/10.1017/CBO9780511543388

CHAPTER 5

A Network Flow Approach for Constellation Planning

Cox *et al. Space Missn Plann Oper* 2022;1:1
DOI: 10.20517/smpo.2022.01

Space Mission Planning & Operations

Article **Open Access**

Check for updates

A network flow approach for constellation planning

Skylar A. Cox, Greg N. Droge, John H. Humble, Konnor D. Andrews

Electrical & Computer Engineering, Utah State University, Logan, UT 84322, USA.
Strategic & Military Space, Space Dynamics Laboratory, North Logan, UT 84341, USA.

Correspondence to: Skylar A. Cox, Electrical and Computer Engineering, Utah State University, 4120 Old Main Hill, Logan, UT 84322, USA. E-mail: skylar.cox@viasat.com; ORCID: 0000-0002-5930-0847

How to cite this article: Cox SA, Droge GN, Humble JH, Andrews KD. A network flow approach for constellation planning. *Space Missn Plann Oper* 2022;1:1. http://dx.doi.org/10.20517/smpo.2022.01

Received: 16 Feb 2022 **First Decision:** 15 Apr 2022 **Revised:** 28 Apr 2022 **Accepted:** 13 May 2022 **Published:** xx Jun 2022

Academic Editor: Madjid Tavana **Copy Editor:** Jia-Xin Zhang **Production Editor:** Jia-Xin Zhang

Abstract

A graph-based network flow approach to constellation planning is presented that respects spacecraft resources and mission constraints to arrive at a preliminary operational schedule. The method discretizes task fulfillment opportunity windows into graph nodes and then adds edges based on feasibility of transition between these opportunities, either by slew maneuvers or extended-time task fulfillment during downlink operations. Intelligent graph pruning is applied to improve computation performance and provide a versatile planning capability with a minimal impact on the resulting solution. A network flow formulation allows for a simultaneous search for spacecraft plans using a Mixed Integer Linear Program (MILP). The framework enables the definition of appropriate mission constraints to ensure viability and mission efficiency among the constellation of satellites. The planning technique is demonstrated for a mission of 100 satellites to illustrate its capability and performance within this proliferated regime.

Keywords: Constellation, mission planning, network flow.

1 INTRODUCTION

Satellite constellations have been conducting operations in space since the early 1960s when the CORONA mission became operational[1]. Since then, a number of additional constellations have joined this exclusive club. These include new Earth observing constellations such as Maxar's WorldView[2], NASA's A-train[3], and Planet Labs' Doves[4], to name just a few. While many of the individual satellites are shrinking in mass,

volume, and cost[5], the constellations continue to grow in numbers, capability, and sophistication, requiring increased levels of automation[6] to properly orchestrate in accomplishing the mission objectives[4,7]. As new mega-constellations are designed and deployed, several new challenges are realized[8,9]. Some of these challenges deal specifically with general geometric configuration and orbital design[10,11], with some research recommending methods to address the challenge[12]. Additionally, and more importantly to the content of this paper, is that as the number of satellites within a constellation increases, the operational planning problem complexity compounds due to the spatial and temporal dependencies present within the constellation's operational environment, and the limited resources onboard each vehicle[13].

The Earth imaging geospatial intelligence (GEOINT) mission, addressed in this paper, falls into the broader category of an assignment problem where the goal is to optimally assign tasks to agents[14]. It can be further classified into a particular group using a well-defined taxonomy developed by Gerkey and Mataric that describes different types of tasks and the agents capable of fulfilling those tasks[15]. The following three axes of differentiation are used for that classification:

1. The number of tasks a robot/agent can perform - a single task (ST) or multiple-tasks (MT) at a time.
2. The number of robots/agents required by a task in order to fulfill it - single robot (SR) tasks, or multi-robot (MR) tasks.
3. The time of allocation - instantaneous allocation (IA) or scheduling robots/agents over an extended period of time (TA).

Korsah *et al.* extended this concept by accounting for dependencies that often exist between the tasks and the robots/agents executing them[16]. These dependencies are the following:

1. No Dependency (ND) - no dependency exists between tasks or agents.
2. In-schedule Dependency (ID) - tasks fulfillment decisions impact other tasks fulfillment opportunities.
3. Cross-schedule Dependency (XD) - agents must coordinate schedules and collaborate directly for task fulfillment.
4. Complex Dependency (CD) - task fulfillment utility depends on the schedules of other agents in the system in a manner that is determined by the particular task decomposition chosen.

A note of particular importance, made by Gerkey and Mataric, is that any problem beyond the [ST-SR-IA] is strongly NP-hard and thus precludes enumerative solutions due to computational complexity and the associated run time[15]. In this paper, the assignment problem is that of tasking multiple satellite agents to optimally fulfill the GEOINT mission, which requires efficient image collection and data delivery to the ground. Using the taxonomies defined, this problem falls into the ID category due to having dependencies on the other tasks a particular agent is performing. Furthermore, each vehicle is only capable of performing a single task at a time, and that task only requires a single satellite to fulfill it, thus placing this problem into the [ST-SR-TA] category. The full categorization is then ID[ST-SR-TA] that, while not being the most complex problem, is still known to be NP-hard[16].

The GEOINT design reference mission (DRM) constellation scheduling problem is of special interest at this time to the space community and deserves additional research attention due to the complexity of formulating and efficiently solving it for mission-realistic scenarios. The primary objective of this mission type is to generate an efficient, operationally-viable schedule that fits within all specified constraints of the mission. Several approaches have been presented to address a portion of this DRM. A host of work has focused on arriving at feasible, and potentially optimal, solutions to the image collection problem[17–23]. Similarly, the data relay portion of the problem has also been investigated by several researchers and they provide robust solutions to scheduling the transfer of data[24–30]. However, in both the image collecting maximization and data relay scheduling problems, the papers mentioned address them independently. In reality, the image collection and downlink problems are directly coupled and should be solved simultaneously to yield a reliable and accurate

solution for the constellation.

Some works have considered both the image collection and data relay problems within the same framework and provided various solutions to solve them[31–33]. However, Hu *et al.* dedicated most of their paper to evaluating a branch and price technique for finding a solution instead of also providing an in-depth explanation of the problem itself[32]. Additionally, Hu *et al.* simulated a small LEO constellation of only 3 satellites and did not evaluate the solution methodology in the proliferated LEO context that the industry is beginning to witness today[34]. Similarly, Peng *et al.* only formulated their planning problem for a single satellite and neglected its application to a multi-agent cooperative system[33]. It is the opinion of the authors of this paper that a truly applicable solution to the GEOINT constellation scheduling problem must be capable of simultaneously planning operations for at least 100 Earth imaging satellites and delivering a robust solution for each of the individual spacecraft elements. The paper by Augenstein *et al.* addressed many of the requirements for a mission-relevant constellation planning technique and was of particular interest to the authors. Augenstein *et al.* formulated the problem for many cooperative agents and provided a solution to the coupled image collection and data downlink problem. However, they required a separation of the coupled problem and relied on heuristics in order to solve it within the required timeframe for their application. Doing so, enabled a rapid solve time but sacrificed a globally-optimal solution within the discretized space. To address these issues, the authors of this paper leveraged the work from Augenstein *et al.* and augmented it with a novel approach using network flow theory, which provided an alternative method for specifying mission objectives, vehicle constraints, and additional methods for improving the solve time. This work directly confronts the challenges posed by the constellation planning problem and presents an optimal planning technique for the GEOINT DRM that enables the simultaneous coordination of both image collection and scheduling of ground stations for data downlink, across the cooperative constellation of space vehicles.

The network flow technique, described in this paper, discretizes the possible task fulfillment opportunities for each satellite and represents available transitions between tasks in graphical form. This graphical formulation enables a simultaneous search for each satellite's path through fulfillment of image collection and downlink tasks. The application of network flow theory to this constellation problem is one of the central contributions made within this paper. Flow constraints allow for coordination across the constellation while resource constraints ensure a mission-realistic solution. The network flow formulation is used to create an operational schedule using a Mixed-Integer Linear Program (MILP) and can be used to solve the constellation planning problem in a mission-relevant way. Instantaneous tasks (e.g., image collection) are represented as nodes in the graph while extended-time duration task fulfillment opportunities (e.g., downlinking of data or slewing the satellite) are modeled as edges in the graph. Fundamental constraints such as lighting and on-board satellite memory are also considered. This blending of task generation, and fulfillment, within an operationally-constrained environment builds on previously introduced planning methods in the literature.

The remainder of the paper will proceed with a brief background on constellation planning methods and algorithms in Section 2. This is followed by a detailed decomposition of the problem into a graph-based representation in Section 3 and constructive development of a network flow optimization solution in Section 4. Finally, a mission scenario of 100 sun-synchronous imaging satellites is analyzed using the network flow approach to illustrate realistic mission performance in Section 6. The paper ends with concluding remarks in Section 7.

2 BACKGROUND AND RELATED WORK

As mentioned above, the purpose of a constellation planning system, within the DRM context, is to maximize prioritized image collection and data downlink for mission task fulfillment while respecting each satellite's capabilities and available resources. The satellite constellation and ground resources must be treated holistically to

ensure that satellites do not duplicate efforts in imaging and that only a single satellite uses a particular ground terminal at any given time for downlink of mission data. These problem elements are directly coupled since image data collection must occur prior to that data being sent to the ground via downlink. Additionally, the slew agility and onboard memory capacity of individual satellites must be considered to ensure an executable and mission-relevant plan. Finally, while image collection tasks can be performed nearly instantaneously, the downlink tasks require extended periods of time to fulfill.

Augenstein *et al.* discretize both downlink and image collection opportunity time windows to initialize the search space considered in arriving at a solution for this coupled problem[31]. While the solutions to the discretized problem are inherently sub-optimal to the granularity of the discretization, the discretization enables the use of a MILP formulation, making the combinatorial problem tractable. Task fulfillment instances allow the planner to evaluate specific decisions within a constrained and finite environment.

To increase applicability to realistic mission scenarios, several constraints are identified by Augenstein *et al.* The satellites can only perform a single task at a time, either image collection or downlink, and they are constrained with a finite slew rate, meaning that pointing maneuvers can only be conducted up to a specified rate. Satellites also have a fixed amount of onboard memory that can be consumed by image data storage[31]. Ground stations can only communicate with a single satellite at a time, requiring a deconflicted schedule for ensuring data delivery to the ground. An additional operational constraint is applied in that every 3 orbit revolutions, each satellite contacts the ground for at least 3 contiguous minutes. This helps maintain up-to-date ground knowledge about the health and safety of each vehicle within the constellation[31].

The combinatorial optimization problem addressed by Augenstein *et al.*, is defined as a MILP with constraints and objectives applied globally rather than using the original formulation of a DAG for each satellite. This MILP formulation can be solved to arrive at an optimal solution within the discretized space. However, the authors point out that the run-time performance of the formulated MILP is unacceptable to their mission timeline since they require multiple revisions to the plan based on immediate mission needs and potential late tasking. To address this problem they separate the immediate coupling that exists between image collection opportunities and downlink opportunities. Separating the imaging and downlink planning decouples the variables present in the MILP but requires a method to appropriately allocated downlink time based on estimated image collection and current onboard memory state of each vehicle. To accomplish this, they provide a heuristic that estimates the amount of priority-weighted image data likely onboard the satellite as a function of time and the number of priority-weighted image collection opportunities a satellite would encounter during specific time periods. The authors demonstrate that with these two pieces of information it is possible to reasonably allocate downlink time among the constellation of vehicles and then also schedule the additional image collection opportunities. This modification yields dramatically reduced solve times and enables the efficient, yet sub-optimal, scheduling of the TerraBella constellation of satellites[31].

The approach developed herein provides several contributions to the planning framework established by Augenstein *et al.* For example, the discretized task opportunities are organized into a directed acyclic graph (DAG) with network flow constraints used to allow for simultaneous search for each satellite's plan. Utilizing the DAG formulation, transition costs can be considered instead of solely scoring the imaging opportunities. Furthermore, the amount of data to be downlinked can directly be associated with the time spent along edges connecting downlink nodes rather than only scoring the instantaneous node as in Augenstein *et al.* By doing so, the edge transitions, and their associated costs, more effectively represent the actual operations to be conducted on orbit. This method is thus more realistic and is more intuitive for an operations team to understand.

Additionally, the flow formulation enables the definition of concise application of mission constraints to avoid duplication of task assignment and does not rely on heuristics to solve. The intent of this research is to formu-

Table 1. Notation used for Planning

Name	Description	Name	Description
Counting variables			
n_s	Number of satellites	n_v	Number of nodes / vertices
n_e	Number of edges	n_g	Number of task groups
n_t	Number of time steps		
Graph Representation			
V	Set of nodes or vertices	v_i	Node i, $v_i \in V$
E	Set of edges, $E \subset V \times V$	e_i	Edge i, $e_i \in E$
G	Graph, $\{V, E\}$	D	Graph incidence matrix
Adj	Graph adjacency matrix		
Task generation			
ENZ	East-North-Zenith frame centered at target	r_N^{sat}	Vector from target to satellite in ENZ
el_{sat}	Elevation angle of satellite wrt target	az_{sat}	Azimuth angle of satellite wrt target
γ_i	Desired pointing orientation of node v_i	θ_{ij}	Slew angle between nodes v_i and v_j
ω_{ij}	Slew rate between nodes v_i and v_j	el_{sun}	Elevation angle of sun wrt target
Variables of optimization			
x_i	Integer variable representing flow along edge e_i	x	Vector of all x_i, $x = \begin{bmatrix} x_1 & \dots & x_{n_v} \end{bmatrix}$
y_{lk}	Memory for satellite l at time k	y_l	Memory values for satellite l, $y_l = \begin{bmatrix} y_{l1}, \dots, y_{ln_t} \end{bmatrix}^T$
y	Vector of all y_i, $y = \begin{bmatrix} y_1^T, \dots, y_{n_s}^T \end{bmatrix}^T$	z	All optimization parameters, $z = \begin{bmatrix} x^T & y^T \end{bmatrix}^T$
Utility and scoring parameters			
u_i	Utility associated with optimization variable i	u	Vector of all u_i
s_i	Score of node v_i	β	Weighting associated with slew rate
Flow constraint parameters			
D	Incidence matrix	D_l	Incidence matrix for satellite l
d_{ij}	The i^{th} row and j^{th} column of D	d_i	The i^{th} row of D
b_k	k^{th} element of RHS of flow constraint	b	RHS of flow constraint with one source, one sink, and unit flow
Group constraint parameters			
$\mathcal{V}_i$	Nodes in group i, $V_i \subset V$	$\mathcal{E}_i$	Set of edges terminating at node in $\mathcal{V}_i$
$\mathcal{I}_i$	Set of edge indices in $\mathcal{E}_i$	A_g	Group constraint matrix
a_{ij}^g	The i^{th} row and j^{th} column of A_g	b_g	Group contraint RHS
Memory constraint parameters			
m_i	Marginal memory requirement for node i	V_k	Nodes associated with time k
I_k	Edge indices that terminate at a node in V_k	A_{data}^l	Data constraint matrix for satellite l
a_{kj}^l	The k^{th} row and j^{th} column of A_{data}^l	b_{data}^l	The RHS of data constraint for satellite l
y_{l0}	The initial data for satellite l	A_{data}	Aggregate data constraint matrix
b_{data}	Aggregate RHS vector for the data constraint		
Indices and index mappings			
σ_i	Index of the node where edge i terminates	η_{lk}	Mapping of y_{lk} to index within z
i, j, p	General indices into matrices and sets	l	Satellite index
k	Discrete time index		

late the problem in such a way that an optimal solution is possible to obtain within a mission-realistic timeline. While the MILP solver does use various heuristics during processing, mission-specific heuristics were not developed. Instead, the formulation discusses a graph pruning concept to simplify the search space without sacrificing optimality. As the DAG can become unwieldy with large numbers of edges, another contribution is the technique of trimming unnecessary edges with minimal effect on the resulting solution. A minor contribution is the explicit definition of image collection constraints such as allowable target collection geometries, target lighting, etc. A complete reference to the notation used in this paper is provided in Table 1.

3 GRAPH-BASED CONSTELLATION PLANNING

The network flow constellation planning approach taken herein can be broken into several distinct steps, as illustrated in Figure 1. It begins by computing the access windows of satellites to targets and ground station antennas. These access windows are defined by start and end times and can be discretized over that period of time. The specific instances in time become nodes within the graph. Nodes are then connected by edges if the transition is feasible, meaning that the required slew rate is below the satellite's allowable slew rate. Once the nodes and edges have been established, a utility is associated with each edge based upon the transition cost (e.g., slew rate) and the score (e.g., downlink value) or imaging task score at the node. The various constraints on task

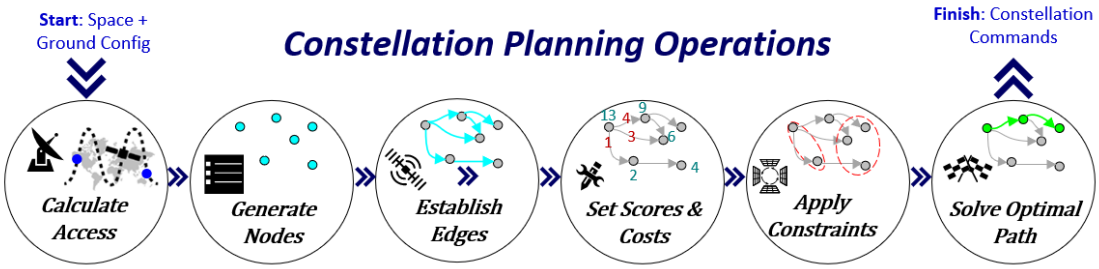


Figure 1. The constellation planning process from mission configuration through command generation.

Table 2. Definition of task constraints

Value	Description
lat/lon/alt	Latitude, longitude, and altitude of the target
min/max azimuth	The minimum and maximum azimuth angle from the target to the satellite
min/max elevation	The minimum and maximum elevation from the target to the satellite
min/max sun elevation	The minimum and maximum sun elevation angle from the target
start/end time	Temporal limits on the valid time window in which the satellite can perform the task

fulfillment and data collection are then associated with the edges. The final step is to solve for the paths through the graph which optimize the utility, subject to the operational constraints applied. The subsections that follow provide precise task constraint definition and greater detail on each of the steps taken during constellation planning.

3.1 Task definition

The satellites, within the constellation, will be commanded to perform two distinct types of tasks. The first is an imaging task and the second a downlink task. The imaging task corresponds to collecting an image of a particular target and the downlink task corresponds to downlinking image data to a ground station. Both the imaging and downlink tasks have temporal and spatial constraints dictating how and when the task can be performed. The imaging task may have an additional lighting constraint. Each of these constraints is summarized in Table 2. The spatial and lighting conditions can be transformed into temporal conditions for each satellite by considering the satellite trajectory, which defines the position of the satellite as a function of time. Each constraint is now discussed by assuming that a time value and satellite position are specified.

The temporal constraint satisfaction is determined by evaluating whether or not the opportunity time occurs within the allowable execution window. Temporal constraints allow the system to set limits on when a target is available for collection due to seasonal or weather concerns, or when a ground system outage occurs for a downlink antenna.

The spatial constraints depend upon the position of the satellite. These constraints are defined in terms of elevation and azimuth angles and use frames centered at the imaging target or ground antenna. Generally, the elevation angle constraints for ground antennas are set to ensure line-of-sight access to the satellite from the antenna, whereas the elevation constraints for an imaging target ensure high overhead imaging to ensure quality data. The azimuth angle constraint controls which side is available to the satellite and can be used to avoid target or antenna occlusion from neighboring buildings or nearby mountainous terrain.

The calculations for the elevation and azimuth angles are performed in the East-North-Zenith (ENZ) frame as depicted in Figure 2, and denoted as N in this paper. The relative position of the satellite with respect to the target in this frame is given as

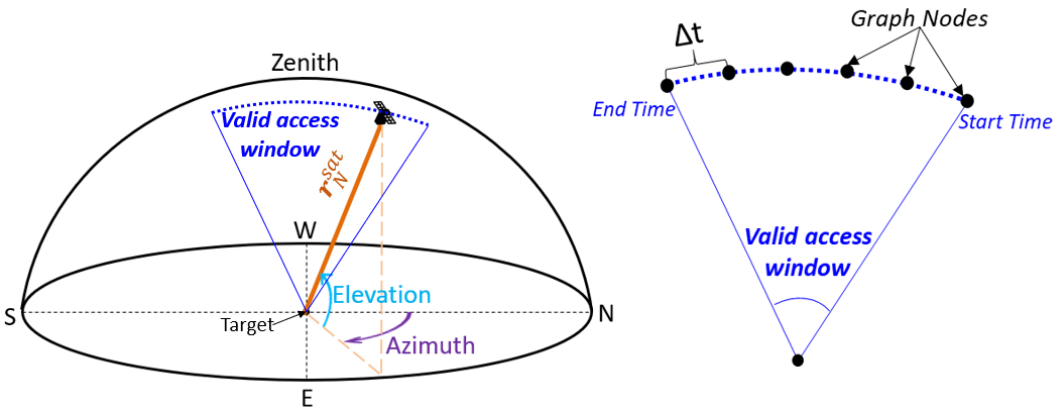


Figure 2. Depiction of a valid access window using elevation and azimuth constraints on the left and a discretized satellite trajectory through the window on the right.

$$\boldsymbol{r}_{\boldsymbol{N}}^{\boldsymbol{sat}} = \begin{bmatrix} r_{xn}^{sat} & r_{yn}^{sat} & r_{zn}^{sat} \end{bmatrix}^T. \tag{1}$$

The elevation angle of the satellite relative to the target is calculated as

$$el_{sat} = \arcsin\left(\frac{r_{zn}^{sat}}{\left\|\boldsymbol{r}_{\boldsymbol{N}}^{\boldsymbol{sat}}\right\|}\right). \tag{2}$$

The azimuth angle, of the satellite relative to the target, is calculated in a similar manner as described by

$$az_{sat} = \arctan2\left(r_{yn}^{sat}, r_{xn}^{sat}\right). \tag{3}$$

In addition to the temporal and spatial constraints, the imaging task may also include lighting constraints. For example, the simulation in Section 6 constrains all targets to be valid only during the daytime (i.e., sun elevation angle between 0° and 90°). The calculation of the sun elevation angle is done exactly as before by replacing the satellite location with the apparent location of the sun, as in

$$el_{sun} = \arcsin\left(\frac{r_{zn}^{sun}}{\left\|\boldsymbol{r}_{\boldsymbol{N}}^{\boldsymbol{sun}}\right\|}\right). \tag{4}$$

The time in question is considered valid for a particular satellite if the time is within the temporal bounds and the calculated elevation and azimuth angles from the target to the satellite, as well as from the target to the sun, are within the angular bounds at the given time. Given the valid access times, access windows for each task are found using root finding techniques to determine the start and end time of each window.

3.2 Graph node generation

The access window of each task forms an essential element to the creation of graph nodes. There are four types of nodes added to the graph. A starting and ending node for each satellite is added to the graph corresponding to the satellite position at the start and end time of the planning horizon. Backbone nodes are added (discussed

in Section 5 as means to maintain connectivity of the graph when pruning edges). The fourth node type corresponds to a task (image collect start time or ground station opportunity start time discretized as shown in right-hand side of Figure 2). The nodes are denoted as $v_i, i = 1, ..., n_v$ where n_v is the total number of nodes in the graph. Each node corresponds to a specific satellite at a specific point in time, either performing a task or in standby.

The task nodes are created from a discretization of the access window for each task with each point of discretization becoming a node in the graph. A more accurate graph can be created with a finer sampling of the valid time windows, but at the cost of increasing n_v, which adversely affects the graph complexity. Figure 2 illustrates a valid access window for a particular target and satellite flight path. The time window discretization is overlaid on the valid access window to arrive at the initial set of nodes for that particular target and satellite.

An orientation vector, γ_i where i is the node index, is associated with each node except the ending node. For the starting node, γ_i corresponds to the pointing vector of the camera at the starting time. For imaging and downlink nodes, γ_i corresponds to the vector pointing from the satellite to the target or ground antenna at that point in time while the backbone nodes assume a nadir orientation, and the final graph node is not assigned a value for γ_i. Not associating an orientation with the final node effectively allows any node to be the final imaging task assigned for the satellite.

Also associated with node i is a score, $s_i \geq 0$. As each node is associated with a particular satellite at a particular point in time, the score for a particular task can vary based upon the satellite performing the task and the time at which the task is being performed. Establishing viable transitions between these nodes is discussed in the next section.

3.3 Graph edge creation

With the nodes established, it is now possible to evaluate the feasibility of the satellite to transition between them via an edge. Associated with each potential edge is a slew angle that corresponds to the change in orientation between nodes. The required slew rate is determined by calculating the required angular change divided bye allowable time between nodes. An edge is only added to the graph if the slew rate is below the allowable slew rate, defined by the agility of the particular satellite.

Given two pointing vectors, γ_i and γ_j corresponding to nodes v_i and v_j, the angle between them, θ_{ij}, can be calculated using the relation between the cosine and dot product:

$$\gamma_i^T \gamma_j = ||\gamma_i|| \cdot ||\gamma_j|| \cos(\theta_{ij} \quad \text{or} \quad \theta_{ij} = \cos^{-1}\left(\frac{\gamma_i^T \gamma_j}{||\gamma_i|| \cdot ||\gamma_j||}\right).$$

Using t_i and t_j to denote the times that correspond to nodes v_i and v_j, the nominal slew rate is calculated as

$$\omega_{ij} = \frac{\theta_{ij}}{t_j - t_i}.$$

A transition between nodes is deemed feasible if the nominal slew rate is below the maximum allowed slew rate for a particular satellite. If feasible, the edge is added to the graph. As no orientation angle is associated with the final node, an edge is allowed between all nodes and the final node. This effectively allows any node to transition to the final task guaranteeing at least one valid solution through the graph.

3.4 Graph scoring and costs

When performing a graph search, the optimization engine attempts to traverse the graph in a way that maximizes utility for the mission. Utility is defined as the score of performing a task minus the cost, or penalty,

required to do so. The globally-optimal plan considers the mission holistically and selects a path through the graph for each satellite that accomplishes this. Each satellite is not necessarily optimal but the complete mission plan is.

For imaging tasks, the score is the parameter used for assessing priority of the collection and acts as a reward for arriving at the end node of the edge (i.e., the worth of the task performed at the end node). The notation σ_i represent the node index at the end of each edge i, denoted e_i. For example, if e_i connects nodes j and f, then $\sigma_i = f$. The notation s_i is used to represent the score of an edge. Imaging tasks scores are contained on the edge entering the imaging node.

For downlink tasks, the score for task i is calculated as the duration of the edge (d_i) multiplied by the data rate (r) and the arriving edge score (s_{σ_i}), divided by the on-board memory size of the satellite (p), or

$$s_i = \frac{d_i \cdot r \cdot s_{\sigma_i}}{p}. \tag{5}$$

For edges that end at a downlink task but do not come from a downlink task for the same ground station the score is 0, $s_i = 0$. This is because the graph must enter a downlink start node before realizing the downlink score when traversing the subsequent edges.

The cost, or penalty, associated with each edge of the graph is the weighted slew rate necessary to perform the maneuver between neighboring task opportunities (e.g., from node h to node i), with a higher required slew rate resulting in a higher penalty. The utility at node i, when transitioning from node h, is given as

$$u_i = s_i - \beta \cdot w_{hi}. \tag{6}$$

where the slew cost weighting factor is represented as β. The utilities are combined into a single vector as $u = \begin{bmatrix} u_1 & u_2 & \dots & u_{n_e} \end{bmatrix}^T$, where n_e is the total number of edges.

3.5 The graph representation

The graph G is now formally defined and contains set of nodes (or vertices), V, and edges, $E \subset V \times V$, i.e. $G = \{V, E\}$. Given two nodes in the graph, $v_1, v_2 \in V$, an edge from v_1 to v_2 implies that the pair (v_1, v_2) is in the edge set, E, i.e. $(v_1, v_2) \in E$. An example graph is shown in Figure 3 with four nodes and five edges. Recall that each node corresponds to an instant in time and applies to a particular satellite. Thus, given a constellation of n_s satellites, G consists of n_s distinct subgraphs.

The planning will make use of the graph's incidence matrix, D, as defined in [35]. Note that the definition of incidence matrix often includes a positive one in both nonzero rows of each column. The incidence matrix as defined herein is also referred to as the coefficient matrix [36]. Given $|V| = n_v$ and $|E| = n_e$, where $|\cdot|$ denotes the cardinality of a set, the incidence matrix is of dimension $n_v \times n_e$. Each column of D is used to represent an edge. Expressing the i^{th} row of the j^{th} column as d_{ij}, the incidence matrix can be expressed element-wise as follows:

$$D = [d_{ij}], \text{ where } d_{ij} = \begin{cases} 1 & \text{Edge } j \text{ originates at node } i \\ -1 & \text{Edge } j \text{ ends at node } i \\ 0 & \text{Otherwise} \end{cases} \tag{7}$$

The signed elements of the incidence matrix allow for quick evaluation of the number of incoming and outgoing edges for each node. This will be useful later to generate continuous paths through the graph. One additional

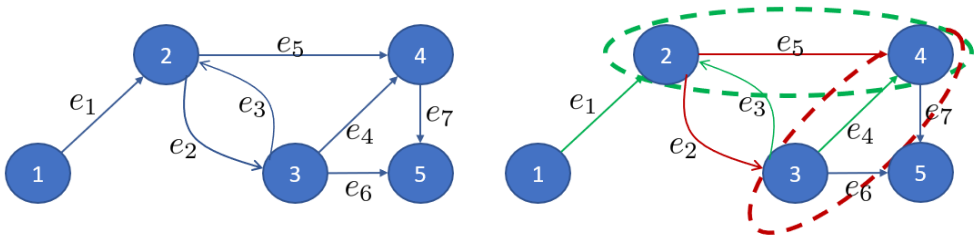


Figure 3. A simple graph is presented with numbered circles as nodes and labeled edges. The right image illustrates two group constraints, one around nodes v_2 and v_4 and one around nodes v_3 and v_4.

matrix that will be used for evaluating the density of the graph is the directed adjacency matrix. The adjacency matrix, denoted Adj, is an $n_v \times n_v$ matrix where row i column j will equal 1 if an edge begins at node v_i and ends at node v_j, with a value of 0 otherwise.

Simple Example of the Incidence and Adjacency Matrices:
For example, the incidence and adjacency matrices for the graph in Figure 3 are written as

$$D = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & -1 & 0 & 1 & 0 & 0 \\ 0 & -1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 & -1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & -1 & -1 \end{bmatrix}, (Adj)^1 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{8}$$

Column j of D corresponds to edge e_j in Figure 3 with the positive element corresponding to the originating node. Row i of the adjacency matrix shows the nodes to which node i can directly connect. With nodes and edges connected and properly quantified for utility, the flow through the graph can now be investigated.

4 A NETWORK FLOW APPROACH FOR CONSTELLATION PLANNING

Our attention now turns to formulating a network-flow based optimization problem for constellation planning. The constellation planning problem that includes image acquisition and downlink scheduling will be solved by finding a set of paths within the DAG with the maximal utility (score-penalty), subject to the mission constraints of non-overlapping task assignments and the limitations imposed by each satellite. These include on-board memory and slew agility. Iterative, dynamic programming-based techniques such as Dijkstra and its many derivatives, e.g.,[37], could be used to find a solution to the unconstrained graph search problem for a single satellite. However, a batched solution using a MILP formulation will be used as it can readily incorporate the constraints required for imaging and be used to simultaneous plan for all satellites. This section constructively develops the MILP problem. First, a network flow approach is presented for graph search. A group constraint is then added to ensure that only one of the many nodes corresponding to an imaging task will form part of the solution. A satellite memory constraint is then added to ensure proper downlinking of information to respect onboard data storage capacities.

4.1 A Network flow approach to graph search:

A mixed integer formulation of the graph search is developed by using a binary decision variable to represent each edge in the graph, denoted as $x_i \in \{0\ 1\}$ for $i = 1, ..., n_e$ (recall that $n_e = |E|$). A value $x_i = 1$ indicates that edge i is part of the path and a value of 0 indicates that edge i is not. To select one path over another, the utility u_i discussed in Section 3.4 is associated with each edge. The total utility of the selected paths can be written as the summation of the individual utilities, $u_1x_1 + u_2x_2 + ... + u_{n_e}x_{n_e} = u^Tx$.

To ensure that the choice of x_i forms continuous paths from each satellite's starting node to each satellite's

ending node, a network flow conservation constraint is employed. This constraint requires that the start node have a single exit path, the end node has only one entry path, and all other nodes, between the start and end node, have both a single entry and single exit to conserve flow. Imagine a simplistic network of pipes carrying water. Each node represents a connection point of various pipes and each edge represents the pipes. Special nodes called "sources" can provide water while others called "sinks" can store or consume the water. All other nodes are intermediary and must simply pass out whatever water comes into the node. Returning to the path planning problem, the "source" nodes are where the satellites start and the "sink" nodes are the terminal nodes for each satellite. All intermediary nodes are decision points. See[36] Chapter 10 for a thorough review of network flow problems.

For the graph search problem, the network flow conservation constraint consists of the source providing a unit of flow, the sink accepting a unit of flow, and all other nodes having a balance of flow (i.e., if an incoming edge to the node is selected, exactly one outgoing edge must also be selected). Recall that row i of the incidence matrix will have a "1" in columns corresponding to edges that start at node i and a "−1" where edges terminate at node i. Writing d_i as the i^{th} row of the incidence matrix, $d_i x = b_i$ implies that there are b_i more edges originating at node i than terminating at node i (note, b_i can be negative). Thus, a source node must have $b_i = 1$, a sink node $b_i = -1$, and an intermediate node $b_i = 0$.

Assuming that no two satellites are collocated, each satellite will have a distinct access schedule to each imaging task. The aggregate graph nodes and edges could each be separated into n_s disjoint subsets, one for each satellite. Each disjoint subset has its own source and sink node. D_l is used to denote the incidence matrix corresponding to the access schedule of satellite l. Assuming that the source node for each satellite corresponds to the first row of D_l, the sink node corresponds to the final row, and an abuse of notation is used to write the edge variables corresponding to satellite l as x_l, the network flow constraint for satellite l can be represented as

$$D_l x_l = b_{ss}, b_{ss} = \begin{bmatrix} 1 & 0 & \dots & 0 & -1 \end{bmatrix}^T. \tag{9}$$

The aggregate network flow could be written by combining the disjoint components of the incidence matrix as follows.

$$Dx = b, D = \begin{bmatrix} D_1 & 0 & \dots \\ 0 & \ddots & \ddots \\ \dots & 0 & D_n \end{bmatrix}, b = \begin{bmatrix} b_{ss} \\ \vdots \\ b_{ss} \end{bmatrix} \in \mathbb{R}^{n_v}$$

The optimization problem can then be written to include this constraint as:

$$\max_x \; u^T x$$
$$\text{s.t. } Dx = b, x_i \in \{0, 1\}$$

Simple Example of the Network Flow Constraint:
Consider the simple example from Figure 3 with corresponding incidence and adjacency matrices in (8). The network flow constraint in (9) would be reduced to

$$Dx = \begin{bmatrix} x_1 \\ -x_1 + x_2 - x_3 + x_5 \\ -x_2 + x_3 + x_4 + x_6 \\ -x_4 - x_5 + x_7 \\ -x_6 - x_7 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ -1 \end{bmatrix}$$

A line-by-line overview of the constraint can help to understand the flow constraint. Line 1 indicates that e_1 must be used. This is also obvious from Figure 3 as it is the only edge coming out of the source. Line five

states that either e_6 be active or e_7, but not both, which is also readily apparent from the figure as these edges lead into the sink. Rows two, three, and four maintain the balance of incoming edges to outgoing edges and ensures the continuous flow through those nodes.

4.2 The group constraint

As each satellite has its own disjoint subgraph, each satellite could plan independently. However, multiple nodes in the graph exist for each task. A group constraint is added to provide the requisite coordination, ensuring that only one node for each task will be selected. Each group of nodes is collected into a set where $\mathcal{V}_i \subset V$ corresponds to group i and there are a total of n_g groups (i.e., $i = 1, ..., n_g$). The set of edges leading into group i is denoted as $\mathcal{E}_i$, where

$$\mathcal{E}_i = \{(v_k, v_j) \in E | v_j \in \mathcal{V}_i \text{ and } v_k \in V \setminus \mathcal{V}_i\}$$

Note that this edge subset may include multiple edges corresponding to a single satellite's disjoint edge set or edges from multiple satellite edge sets. The index set, $\mathcal{I}_i$, contains the edge indices corresponding to $\mathcal{E}_i$. The constraint for group i can be written in summation form as

$$\sum_{j \in \mathcal{I}_i} x_j \leq 1 \; \forall i = 1, ..., n_g. \tag{10}$$

Combined with the fact that $x_j \in \{0, 1\}$, (10) ensures that a maximum of one edge will enter the group.

$\mathcal{I}_i$ can be used to write the constraint in matrix form. The matrix A_g and vector b_g are used to represent the constraint as $A_g x \leq \mathbb{1}_{n_g}$, where $\mathbb{1}_{n_g}$ is a column vector of n_g ones. Allow a^g_{ij} to be the entry of A^g in the i^{th} row and j^{th} column. A^g can be expressed as

$$A^g = [a^g_{ij}], a^g_{ij} = \begin{cases} 1 & j \in \mathcal{I}_i \\ 0 & \text{otherwise} \end{cases}.$$

The problem, including the group formulation can now be written as

$$\begin{aligned} &\max_x \; u^T x \\ &\text{s.t. } Dx = b, A_g x \leq \mathbb{1}_{n_g}, x_i \in \{0, 1\} \end{aligned}, \tag{11}$$

Note that the group constraint only limits the number of selected incoming edges to be at most one. Due to the flow conservation constraint, $Dx = b$, the number of chosen edges proceeding from each group will also be at most one.

Simple Example of the Group Constraint:

We continue with the previous graph example describing the group constraint depicted in Figure 3. There are two group constraints depicted in green and red in the figure. The first is around nodes v_2 and v_4 with the incoming edge set $\mathcal{E}_1 = \{e_1, e_4\}$ and the index set $\mathcal{I}_1 = \{1, 4\}$. The second is formed with nodes v_3 and v_4 with $\mathcal{E}_2 = \{e_2, e_5\}$ and $\mathcal{I}_2 = \{2, 5\}$. The group constraints can be written as

$$\begin{aligned} x_1 + x_3 + x_4 &\leq 1 \\ x_2 + x_5 &\leq 1 \end{aligned} \text{ or } \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} x \leq \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

4.3 Data constraints

Nodes corresponding to an imaging or downlink task, will affect the data onboard the satellite. Let m_i represent the memory consumption at node i with $m_i > 0$ for imaging tasks, $m_i < 0$ for downlink tasks, and $m_i = 0$ for

all non-task nodes. The data onboard spacecraft l at time k is denoted as y_{lk}. The data vector over time for spacecraft l is denoted as y_l and all data vectors are combined in the vector y, i.e.,

$$y_l = \begin{bmatrix} y_{l1} & y_{l2} & \dots & y_{ln_t} \end{bmatrix}^T, y = \begin{bmatrix} y_1^T, y_2^T, ..., y_{n_s}^T \end{bmatrix}^T \in \mathbb{R}^{n_d}. \tag{12}$$

where n_t is the total number of time steps being considered, n_s is the number of spacecraft, and n_d is the total number of data variables.

The data at time k will depend upon the data at the previous time step and the edge taken to arrive at time k. The set $V_{lk} \subset V$ represents the nodes associated with spacecraft l at time k. As each node has an associated time value, two observations can be made about the structure of the graph:

1. The network flow constraint will restrict the number of edges entering V_{lk} to be at most one.
2. The edge taken to arrive at time k will come from a node in V_{lj} where $j \leq k$.

Denote the index set I_{lk} as the set of edge indices corresponding to edges that terminate at a node in V_{lk}. Recall that σ_i denotes the index of the node where edge i terminates (i.e., if $e_i = (v_j, v_p)$, then $\sigma_i = p$). The updated memory consumed at time k can then be written as

$$y_{l,k} = y_{l,k-1} + \sum_{j \in I_{lk}} x_j \cdot m_{\sigma_j}, \tag{13}$$

where $y_{l,0}$ is the initial data storage. Note that multiplying x_j by m_{σ_j} ensures that the contribution of node σ_j to the total data is only received when an edge is chosen that enters node σ_j.

The variables of optimization are now a combination of the edge variables and dynamic variables, $z = [x^T\ y^T] \in \mathbb{R}^{(n_e+n_d)}$. Another index mapping, η_{lk}, is used to map the data variables to their respective index in the optimization vector such that $z_{\eta_{lk}} = y_{lk}$.

The relation in (13) can then be written in matrix form for spacecraft l by defining the matrix $A^l_{data}\mathbb{R}^{n_t \times (n_e+n_d)}$ where

$$A^l_{data} \cdot z = b^l_{data}, b^l_{data} = \begin{bmatrix} -d_{l0} \\ 0 \\ \vdots \\ 0 \end{bmatrix}, A^l_{data} = [a^l_{kj}], \text{where } a^l_{kj} = \begin{cases} m_{\sigma_j} & j \in I_{lk} \\ -1 & j = \eta_{lk} \\ 1 & k > 1 \text{ and } j = \eta_{l,k-1} \\ 0 & \text{otherwise} \end{cases}$$

The aggregate data update constraint can then be written as

$$A_{data} \cdot z = b_{data} \text{ where } A_{data} = \begin{bmatrix} A^1_{data} \\ \vdots \\ A^n_{data} \end{bmatrix}, b_{data} = \begin{bmatrix} b^1_{data} \\ \vdots \\ b^n_{data} \end{bmatrix} \tag{14}$$

Note that each data element is assigned a zero utility (i.e., $u_{\eta_{l,j}} = 0\ \forall l, j$). The optimization problem with group and on-board memory constraints can then be written as

$$\begin{gathered} \max_z\ u^T z \\ \text{s.t. } Dx = b, A_g x \leq \mathbb{1}_{n_g}, A_{data} z = b_{data}, \\ x_i \in \{0, 1\}, y_i \geq 0\ \forall\ i, j \\ z = \begin{bmatrix} x^T y^T \end{bmatrix}^T \end{gathered} \tag{15}$$

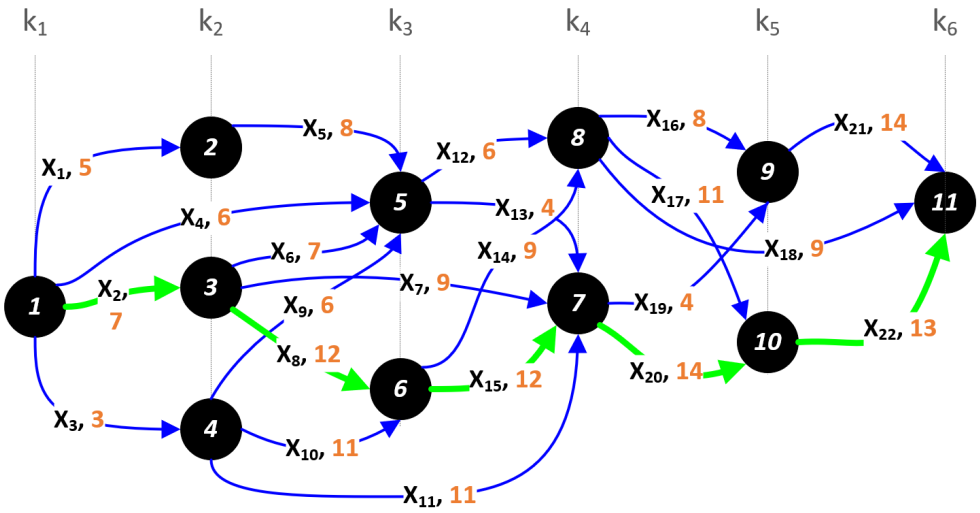


Figure 4. The optimal path through the graph when solving this problem using the network flow approach outlined. The path through the nodes is given as edges 2, 8, 15, 20, and 23.

D	EDGE																						
NODE	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	-1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	-1	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	-1	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	-1	-1	-1	0	0	-1	0	0	1	1	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	-1	0	-1	0	0	0	1	1	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	-1	0	0	0	-1	0	-1	0	-1	0	0	0	1	1	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	-1	0	-1	0	1	1	1	0	0	0	1	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	-1	0	1	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	-1	0	0	1
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	-1	-1	-1

Figure 5. The resulting D matrix from the example graph.

4.4 Example graph formulation

This section will now provide an example of the problem formulation for a single satellite performing imaging only (i.e., no downlink operations). A later example will address the coupled imaging and downlink scenario. The graph shown in Figure 4 will now be formulated using the network flow method outlined above. The edges are labeled with the letter 'x' and a numeric subscript to indicate the edge number. A numeric value is present with each edge to show the utility score associated with that edge. The grouping of nodes within each ellipse represent discrete time steps from k_1 to k_6 and show that only one node within each group can be visited within that time step since the satellites are modeled as satellites that can only perform one task at a given time. This application of grouping constraints is critical to the planning technique presented in this work and unique to the literature previously referenced. The resulting D matrix is shown in Figure 5.

Solving the problem, using the network flow implementation, results in the edge selection sequence of 2, 8, 15, 20, 23 as shown in Figure 4. The edges emphasized with green illustrate the optimal path through the graph when only considering imaging opportunities.

The network flow concept is now presented in the presence of imaging and downlink tasks within a memory-constrained satellite environment. Consider the same graph from above but now with nodes 5, 8, and 9 be-

A^1_{data}	EDGE																											
Time Step	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	d11	d12	d13	d14	d15	d16
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0
2	3	6	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	-1	0	0	0	0
3	0	0	0	0	0	0	0	10	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	1	-1	0	0	0
4	0	0	0	0	0	0	12	0	0	0	12	-11	12	0	12	0	0	0	0	0	0	0	0	0	1	-1	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-11	12	0	0	12	0	0	0	0	0	1	-1	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	-1

Figure 6. The A^1_{data} matrix representing the flow of memory through the graph.

Table 3. On-board memory consumption for each time step for the example graph presented

Time Step	k1	k2	k3	k4	k5	k6
Memory	0	6	16	16	5	5

coming downlink opportunities instead of imaging operations. To properly create the A_{data} matrix, we must specify the amount of memory associated with each node, or edge, in the graph.

The memory associated with each operation is shown overlaid on the graph as purple numbers in Figure 7. Note that memory associated with imaging operations are applied at the node but downlinking operations bookkeep memory on the edge between downlink nodes due to the time discretization applied in the problem. The actual amount of memory downlinked however, is the product of the length of the edge and the downlink rate. This ensures proper accounting for edge length since downlinking occurs over extended periods of time and imaging operations are modeled as instantaneous collections. Memory for downlinking is computed by multiplying the edge length (time) by the downlink rate of the spacecraft. Notice that the start and end nodes are assigned a memory value of 0 while the edges between nodes 5, 8, and 9 have all been allocated a memory amount of -11 to denote downlink to the ground terminal. The resulting A^1_{data} matrix (1 for a single satellite) is shown in Figure 6.

For this initial example, the memory limit is set to a value of 20. Note that this limit applies to every time step within the planning graph and ensures that the satellite does not collect more image data than it can save on-board. Solving this problem again with a memory limit set results in a new sequence through the graph nodes. This new sequence is shown in Figure 7. Note, that the new graph solution deviates from the original and passes through nodes 8 and 9 to reduce the amount of image data on-board the vehicle. The resulting data storage at each time step, from k_1 to k_6, is given in Table 3.

5 GRAPH PRUNING

Recall that an edge is created between two nodes if the slew rate to achieve that transition is below a threshold. Even if two targets have very different slew angles, if they are located far enough apart in time then an edge will exist. The number of edges for the graph can quickly become very large with increased planning horizons or numbers of imaging targets due to the physics of satellite motion. In reality the only edges that are naturally disallowed, in Section 3.3, are those that transition between nodes that are located very close in time. While beneficial for expressing possible transitions, the complexity of the graph (and thus the number of decision variables and constraints considered) can quickly become unwieldy.

For example, consider the active target tasks depicted in Figure 8. Given 2,814 targets, 41 of which are visible to at least one satellite, the number of edges for three satellites over a planning horizon of 94.616 minutes is 7,440. Shown in the figure is a depiction of the resulting adjacency matrix with cyan depicting a possible edge. Since the goal is to realize a large number of target assignments, the density is not actually needed. Taking long edges results in skipping over a large number of possible imaging targets, making it an undesirable path.

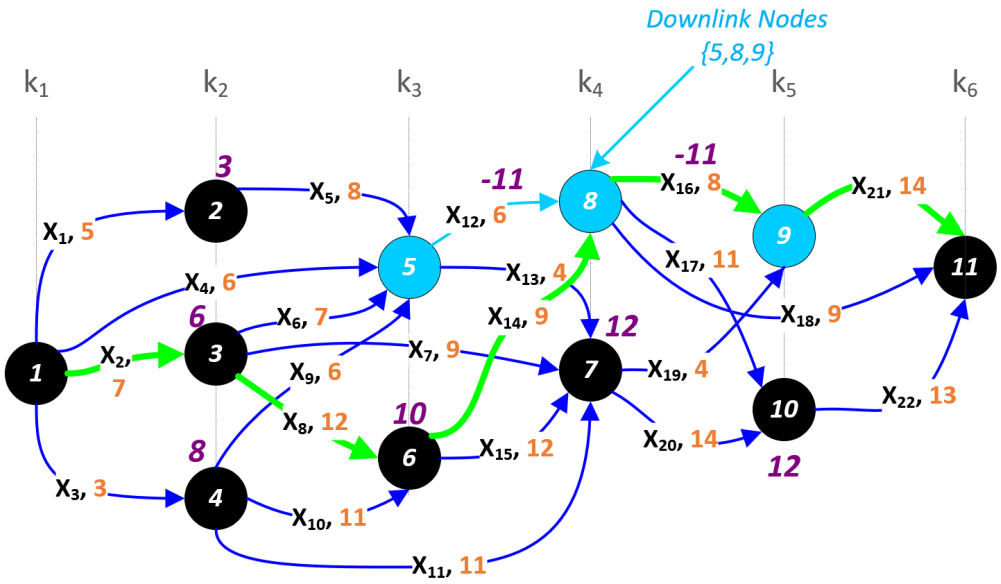


Figure 7. The path solution resulting when constraining the on-board memory to a value of 20. The path of edges is given as 2, 8, 14, 16, and 23.

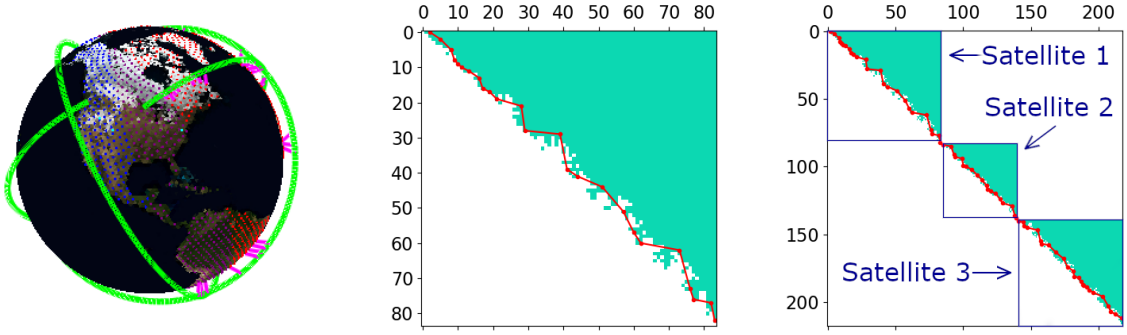


Figure 8. On the left is shown the imaging target deck, the middle shows the adjacency matrix assuming a single satellite, and the right image shows the adjacency graph for three satellites. The adjacency matrix is visualized using a cyan pixel for a 1 and white pixel for a 0. The nodes are sorted first by time and then by satellite. The red lines show the optimal path through the graph.

In fact, most optimal plans track closely to the diagonal, corresponding to satellites imaging as many targets as possible. An approach is now presented which can drastically reduce the graph complexity with little change in the resulting outcome.

A naive approach to reducing complexity is to eliminate all edges beyond a given maximum edge length. While it can significantly reduce the number of edges, it can also reduce the connectivity of the graph. A particular target may become disconnected from feasible paths from the start node to the end node. A group of nodes may be completely cutoff if the maximum edge length is too small. For example, lighting constraints and oceans can create large gaps in time between viable targets.

Instead of solely using a maximum edge length, a solution using "backbone" nodes has been developed. Similar to the start and end nodes, the backbone node has no associated task and has a zero score. Such nodes are

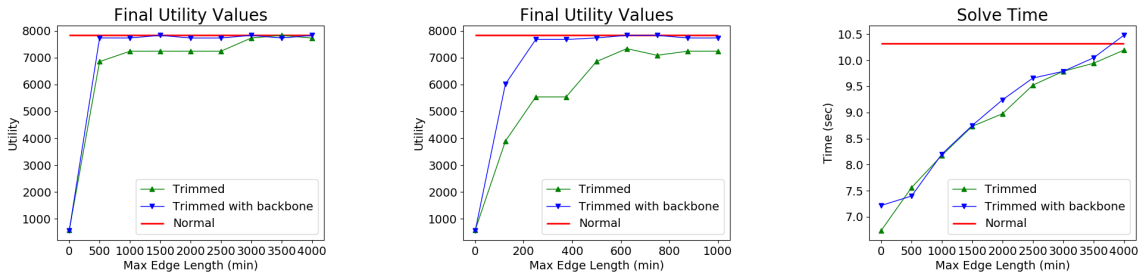


Figure 9. The resulting utility score and computation cost from the scenario in Figure 8 using three satellites. The left image shows the edge length vs utility for the untrimmed, naively trimmed, and backbone trimmed graphs. The middle image is a zoomed in view of the left image. The right image shows the resulting computation time for the trimming.

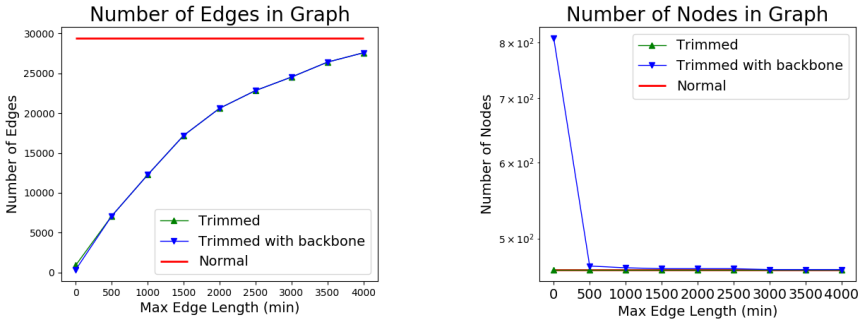


Figure 10. The number of edges and nodes vs the trim length.

placed between groups of nodes that would be disconnected due to the maximum edge length constraint. Nodes corresponding to each satellite are sorted in time and a linear search performed to find gaps larger than the maximum edge length. A backbone node is then placed in the temporal center of the gap and acts as a bridge between the separated groups. Edges are also added between the backbone nodes, when needed, to preserve connectivity. This method greatly reduces the number of edges, and thus the overall complexity of the graph.

Results for the backbone trimming approach using the example depicted in Figure 8 can be seen in Figures 9, 10, and 11. The horizontal axis in each figure shows the maximum edge length allowed. Each plot shows the results using an untrimmed graph, the naive trimming, and the backbone trimmed approach. Figure 9 shows that with a relatively small maximum edge length of 100, the same utility was achieved using the backbone node approach, but the naive trimming approach requires a length of roughly 3,000. The result is the ability to achieve nearly the same utility with a fraction of the computation time. Note that the number of edges, as depicted in Figure 10, has a strong correlation to the computation time. Also note that there is a small trade-off as the backbone approach does introduce additional nodes, becoming burdensome for very small maximum edge lengths. Figure 11 shows that the backbone approach achieves a similar reduction in graph complexity while maintaining a very similar plan to the untrimmed graph.

6 SIMULATION AND RESULTS

To fully examine the network flow constellation planning concept discussed throughout this paper, a realistic orbital flight scenario is analyzed in this section. The orbital scenario is first specified and then the results from planning are presented.

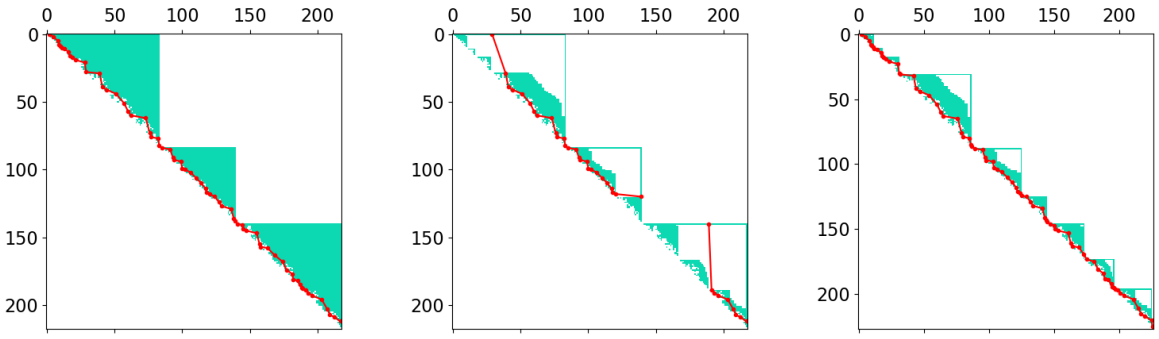


Figure 11. The result of trimming on the adjacency matrix of the graph. The left matrix shows the untrimmed graph. The middle matrix shows the trimmed graph without backbone nodes. The right matrix shows the trimmed graph with backbone nodes. The red lines on the plot show the planned path for each of the three satellites in this example. The max edge length for these graphs is set to 300.

Table 4. Seed satellite orbital elements for the flight scenario

Element name	Value
Semi-Major Axis	6878.14 km
Inclination	97.41°
RAAN	0.0°
Argument of Perigee	0.0°
Eccentricity	0.0
True Anomaly	0.0°

6.1 Orbital scenario

A 500 km sun-synchronous orbit was chosen to provide realistic access periods to both targets and ground stations within the view of the Earth imaging satellites. The epoch for this scenario is August 1, 2021 at 18:00:00 UTC with accompanying orbital elements specified in Table 4. This orbit represents the seed satellite of a Walker delta constellation of 100 satellites and 25 orbital planes with no relative phasing between neighboring planes. A Walker delta pattern constellation provides a simple, parameterized method for conveying the orbital geometry for a constellation of many satellites. These parameters specify the inclination, total number of satellites, number of orbital planes, and the relative phasing between satellites in neighboring planes. This phasing angle is the angle difference, in the direction of orbital motion, from the ascending node to the closest satellite, when a satellite in the next westerly plane is at its ascending node. Again, in this example the relative phasing is 0°. A Walker delta pattern constellation starts with a seed satellite and generates the entire constellation from its orbital information. For more information about Walker constellations, see[38]. The ground terminals supporting downlinking operations are located in Alaska, Antarctica, Australia, Hawaii, and Norway. A total of 2,814 imaging targets were considered during planning, with these targets being approximately evenly spaced across the Earth's land mass as depicted in Figure 12.

6.2 Results

The network flow constellation planning algorithm discussed throughout this paper was run on the mission scenario described previously to arrive at a constellation flight schedule. Several metrics were collected for each of the satellites to assess their performance and help evaluate the constellation planning routine. The metrics of particular interest in the DRM are satellite utility, number of targets collected, number of targets downlinked, and overall satellite memory state. These metrics are shown over the planning horizon timeline of approximately 3 hours (2 orbit revolutions) in Figure 13. The figure illustrates the performance that can be anticipated when using the network flow constellation planning routine presented. For context, Figure 14 also shows the satellite constellation with orbit trajectories, imaging events, and downlink operations illustrated.

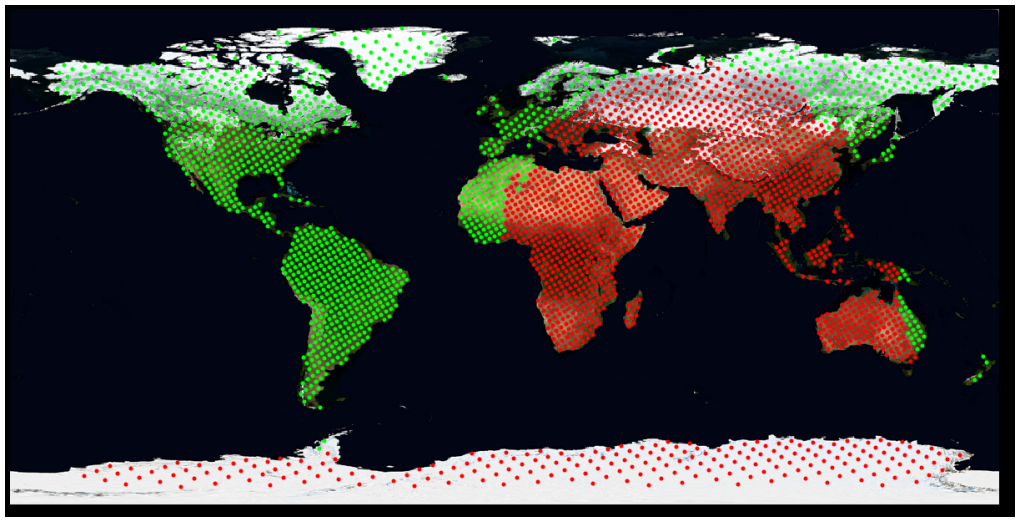


Figure 12. Imaging target distribution across the Earth's land mass area with targets accessible during the planning period marked in green. Targets marked in red do not meet the lighting constraints due to being in eclipse during the planning period.

The results shown in Figure 13 provide insight into specific satellite assignments with red dashed lines representing individual satellites while the blue line in each plot highlights the collective performance across the constellation. The utility score, number of targets collected, and number of targets downlinked trend in an upward fashion since these metrics are cumulative and continue to increase as the satellites perform the imaging and downlinking operations over the simulation period. The satellite memory state plot is more variable as the satellites balance on-board memory capacity with new imaging collection opportunities and downlink operations. The memory plots show some satellites reaching a plateau in consumed on-board memory, indicating that imaging or downlink opportunities were either not selected or unavailable during that time, while other satellites show both imaging and downlinking operations occurring. The simulation was initiated with all satellites having all memory available on-board. Due to this, the global blue line trends upward for this simulation as memory across the constellation is consumed with imaging operations across the fleet.

Planning the imaging and downlink operations for 100 satellites over a multi-revolution period on a desktop PC is a challenging endeavor, yet the network flow technique proved to be capable. This implementation shows the ability to arrive at an optimal plan for a large constellation of 100 satellites while not exceeding the individual satellite's resources. Furthermore, the plots show a relative global consistency for the constellation while also identifying unique trends and behaviors for each satellite. This consistency is due to the fairly homogeneous distribution of satellites relative to the imaging and downlink opportunities across the globe and lends confidence to the viability and consistency of the resulting solution.

The utility score and targets collected plots show a strong trend upward, meaning that imaging continues to progress throughout the planning period. Similarly, downlink is being optimized across the constellation, as illustrated by number of targets downlinked and onboard memory state plots, to ensure delivery of the data and prevent on-board resources from being exceeded. These results highlight the power behind the network flow method and support further research into this area. In particular, further research is warranted into additional methods for understanding the elapsed time from the collection of a specific image to the time that image information is delivered to the ground. The current implementation focuses on prioritized data collection and delivery but does not account for the value of quick delivery of image data. The latency metric is of particular interest to GEOINT missions due to the value in timely intelligence. Low-latency information allows planners to make time-critical decisions with greater confidence and is thus an important metric within the GEOINT community[39].

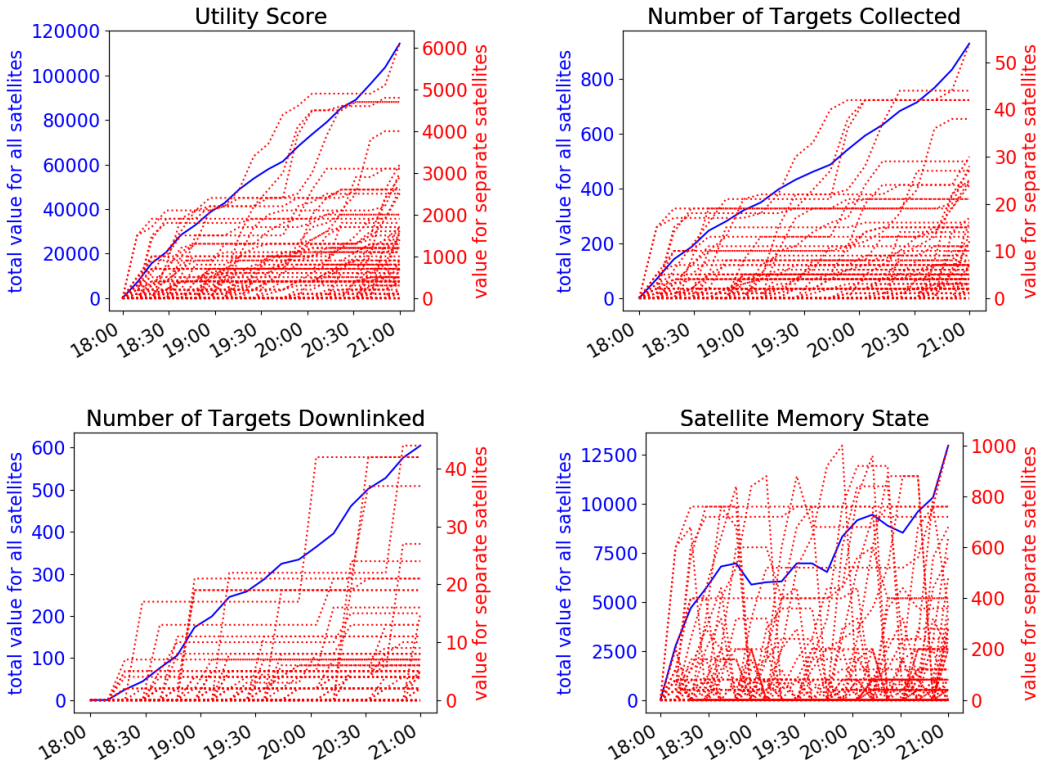


Figure 13. These figures show the results from a simulation with 100 satellites in 25 planes with 2814 ground targets spread over the surface of the earth and 5 ground stations in Alaska, Antarctica, Australia, Hawaii, and Norway. The left axis (in blue) shows the collective satellite values and the right axis (in red) shows the values for individual satellites

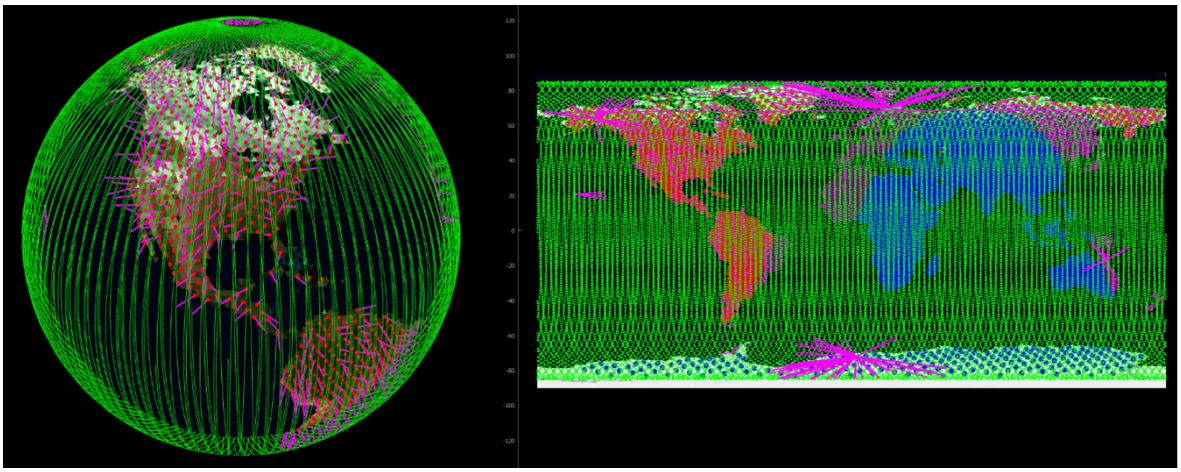


Figure 14. Global view of satellite planes and orbital configuration analyzed using 100 satellites in 25 planes with 5 ground terminals. In the left illustration, the red dots show where the satellite was located when an image was captured while the pink line illustrates the pointing at the time of collection. In the right illustration, the pink lines centered on the ground stations mark the downlink access periods. The targets colored in blue are those that were not considered during planning due to being in eclipse and not meeting the specified lighting constraints.

7 CONCLUSION

This paper has provided a detailed explanation of the constellation scheduling problem, within the context of the geospatial intelligence design reference mission, as well as a comprehensive overview of the graph method

and network flow approach to arrive at a globally-optimal solution within the discretized problem space. A realistic constellation planning scenario was presented for 100 satellites, prosecuting over 2,800 targets, while accessing up to 5 ground station terminals. The results show tractability and trending behaviors of increasing utility by collecting image data and downlinking it to the ground, while keeping within the onboard resources of each satellite within the constellation. This paper has also shown the dramatic improvement in the time required to solve the geospatial intelligence constellation planning problem once graph pruning has been applied. This improvement is the first step in making the algorithm run fast enough to support feasible solution generation in an operational environment. Furthermore, the presented methodology illustrates how a variety of specific spacecraft operational constraints can be created using network flow theory. The problem formulation and solving methods described in this paper have far-reaching implications and provide a robust strategy for planning coordinated operations across a fleet of cooperative systems in a time-extended mission context. Future research will leverage the network flow capabilities and extend the problem space by introducing cross-schedule dependencies for direct interaction between satellites and support crosslinking of data between these space systems. This capability will allow satellites to share data between them, thus freeing up onboard resources, while also potentially improving the global utility performance of the constellation, as a whole.

8 DECLARATIONS

8.1 Authors' contributions

Skylar A. Cox (Conceptualization, Methodology, Formal Analysis, Writing); Greg N. Droge (Conceptualization, Methodology, Formal Analysis, Writing); John H. Humble (Software Development, Mission Simulation); and Konnor D. Andrews (Software Development, Mission Simulation).

8.2 Availability of data and materials

Not applicable

8.3 Financial support and sponsorship

None.

8.4 Conflicts of interest

All authors declared that there are no conflicts of interest.

8.5 Ethical approval and consent to participate

Not applicable.

8.6 Consent for publication

Not applicable.

REFERENCES

1. Day D. Eye in the sky: the story of the CORONA spy satellites. Smithsonian Institution; 2015. Available from: https://link.springer.com/gp/book/9780792371489.
2. Scott WS, Anderson N, Rogers AQ. Design drivers for a viable commercial remote sensing space architecture 2020. In: 34th Annual Small Satellite Conference. Available from: https://digitalcommons.usu.edu/cgi/viewcontent.cgi?article=4709&context=smallsat.
3. Stephens GL, Vane DG, Boain RJ, et al. The CloudSat mission and the A-Train: a new dimension of space-based observations of clouds and precipitation. *Bull Amer Meteor Soc* 2002;83:1771-90. DOI
4. Kopacz JR, Herschitz R, Roney J. Small satellites an overview and assessment. *Acta Astronautica* 2020;170:93–105. DOI
5. Stock G, Fraire J, Hermanns H, et al. On the automation, optimization, and in-orbit validation of intelligent satellite constellation operations.

In: Small Satellite Conference; 2021. Available from: https://hal.archives-ouvertes.fr/hal-03494123/file/2021-SmallSat-AutomatedOperationsGomSpace.pdf.
6. Ben-Larbi MK, Pozo KF, Haylok T, et al. Towards the automated operations of large distributed satellite systems. Part 1: Review and paradigm shifts. *Advances in Space Research* 2021;67:3598-619. DOI
7. Cappaert J, Foston F, Heras PS, et al. Constellation Modelling, Performance Prediction and Operations Management for the Spire Constellation, 2021. Available from: https://digitalcommons.usu.edu/smallsat/2021/all2021/136/.
8. Lalbakhsh A, Pitcairn A, Mandal K, et al. Darkening Low-Earth Orbit Satellite Constellations: A Review. *IEEE Access* 2022;10:24383-94. DOI
9. Deng R, Di B, Zhang H, Kuang L, Song L. Ultra-dense LEO satellite constellations: how many LEO satellites do we need? *IEEE Trans Wireless Commun* 2021;20:4843-57. DOI
10. Gao Y, Wei C. Planning management exploration in the development of large-scale satellite constellation systems. In: International Conference on Intelligent Automation and Soft Computing. Springer; 2021. pp. 469–79. DOI
11. Guo S, Zhou W, Zhang J, Sun F, Yu D. Integrated constellation design and deployment method for a regional augmented navigation satellite system using piggyback launches. *Astrodynamics* 2021;5:49–60. DOI
12. Arnas D, Linares R. Uniform Satellite Constellation Reconfiguration. *Journal of Guidance, Control, and Dynamics* 2022:1–14. DOI
13. Ullman JD. NP-complete scheduling problems. *Journal of Computer and System Sciences* 1975;10:384–93. DOI
14. Burkard R, Dell'Amico M, Martello S. Assignment problems, revised reprint. vol. 106. Siam; 2012. DOI
15. Gerkey BP, Matarić MJ. A formal analysis and taxonomy of task allocation in multi-robot systems. *The International Journal of Robotics Research* 2004;23:939–54. DOI
16. Korsah GA, Stentz A, Dias MB. A comprehensive taxonomy for multi-robot task allocation. *The International Journal of Robotics Research* 2013;32:1495–512. DOI
17. Berger J, Lo N, Barkaoui M. QUEST – A new quadratic decision model for the multi-satellite scheduling problem. *Computers & Operations Research* 2020;115:104822. DOI
18. He L, de Weerdt M, Yorke-Smith N. Time/sequence-dependent scheduling: the design and evaluation of a general purpose tabu-based adaptive large neighbourhood search algorithm. *J Intell Manuf* 2020;31:1051–78. DOI
19. Mitrovic-Minic S, Thomson D, Berger J, Secker J. Collection planning and scheduling for multiple heterogeneous satellite missions: Survey, optimization problem, and mathematical programming formulation. In: Modeling and Optimization in Space Engineering. Springer; 2019. pp. 271–305. DOI
20. Nag S, Li AS, Ravindra V, et al. Autonomous scheduling of agile spacecraft constellations with delay tolerant networking for reactive imaging. *arXiv preprint arXiv:201009940* 2020. DOI
21. Sinha PK, Dutta A. Multi-satellite task allocation algorithm for earth observation. In: 2016 IEEE Region 10 Conference (TENCON). IEEE; 2016. pp. 403–8. DOI
22. Tangpattanakul P, Jozefowiez N, Lopez P. A multi-objective local search heuristic for scheduling Earth observations taken by an agile satellite. *European Journal of Operational Research* 2015;245:542–54. DOI
23. Yao F, Li J, Chen Y, Chu X, Zhao B. Task allocation strategies for cooperative task planning of multi-autonomous satellite constellation. *Advances in Space Research* 2019;63:1073–84. DOI
24. Deng B, Jiang C, Kuang L, et al. Two-phase task scheduling in data relay satellite systems. *IEEE Trans Veh Technol* 2017;67:1782–93. DOI
25. Gu X, Bai J, Zhang C, Gao H. Study on TT&C resources scheduling technique based on inter-satellite link. *Acta Astronautica* 2014;104:26–32. DOI
26. Karapetyan D, Mitrovic-Minic S, Malladi KT, Punnen AP. The satellite downlink scheduling problem: A case study of radarsat-2. In: Case Studies in Operations Research. Springer; 2015. pp. 497–516. DOI
27. Li J, Chen H, Jing N. A data transmission scheduling algorithm for rapid-response earth-observing operations. *Chinese Journal of Aeronautics* 2014;27:349–64. DOI
28. Song B, Yao F, Chen Y, Chen Y, Chen Y. A hybrid genetic algorithm for satellite image downlink scheduling problem. *Discrete Dynamics in Nature and Society* 2018;2018:1-11. DOI
29. Spangelo S, Cutler J, Gilson K, Cohn A. Optimization-based scheduling for the single-satellite, multi-ground station communication problem. *Computers & Operations Research* 2015;57:1-16. DOI
30. Zhao Wh, Zhao J, Zhao Sh, et al. Resources scheduling for data relay satellite with microwave and optical hybrid links based on improved niche genetic algorithm. *Optik* 2014;125:3370-5. DOI
31. Augenstein S, Estanislao A, Guere E, Blaes S. Optimal scheduling of a constellation of earth-imaging satellites, for maximal data throughput and efficient human management. In: Proceedings of the International Conference on Automated Planning and Scheduling. vol. 26; 2016. Available from: https://ojs.aaai.org/index.php/ICAPS/article/view/13784.
32. Hu X, Zhu W, An B, Jin P, Xia W. A branch and price algorithm for EOS constellation imaging and downloading integrated scheduling problem. *Computers and Operations Research* 2019;104:74–89. DOI
33. Peng S, Chen H, Li J, Jing N. Approximate path searching method for single-satellite observation and transmission task planning problem. *Mathematical Problems in Engineering* 2017;2017. DOI
34. McDowell JC. The low earth orbit satellite population and impacts of the SpaceX Starlink constellation. *The Astrophysical Journal Letters* 2020;892:L36. DOI
35. Mesbahi M, Egerstedt M. Graph theoretic methods in multiagent networks. vol. 33. Princeton University Press; 2010. DOI
36. Chen DS, Batson RG, Dang Y. Applied integer programming. Wiley, 2010. Available from: http://web.ist.utl.pt/~ist11038/acad/or/LP/

2010ChenBatsonDangApplIntProg.pdf.
37. LaValle SM. Planning algorithms. Cambridge university press; 2006. Available from: http://lavalle.pl/planning/.
38. Wertz JR. Orbit & Constellation Design & Management. Springer; 2009. Available from: https://link.springer.com/gp/book/9780792371489.
39. Dold J, Groopman J. The future of geospatial intelligence. *Geo-spatial Information Science* 2017;20:151–62. DOI

CHAPTER 6

A Mission Planning Approach for Crosslink-Enabled Earth Sensing Satellite Constellations

A Mission Planning Approach for Crosslink-Enabled Earth Sensing Satellite Constellations

Skylar A. Cox[a], John H. Humble[b], Greg N. Droge[a]

[a] *Utah State University, Department of Electrical & Computer Engineering, 4120 Old Main Hill, Logan, 84322, UT, U.S.A.*
[b] *Space Dynamics Laboratory, 1695 Research Park Way, North Logan, 84341, UT, U.S.A.*

Abstract

Current interest in spacecraft constellations for government and commercial uses continues to drive advancement in sophistication, capability, and mission complexity. As these systems proliferate, they require increased levels of automation to properly orchestrate in achieving the mission objectives. This paper develops a constellation planning capability for cross-link enabled Geospatial Intelligence missions. The planning for constellation coordination is performed using a two-level network flow approach, where the first level creates a communication plan, and the second layer determines the Earth sensing activities. A network flow method is used to simultaneously obtain a schedule for each unique satellite within the constellation and provides an optimal operational schedule for the entire cooperative system.

Keywords: Constellation, Crosslink
PACS: 0000, 1111
2000 MSC: 0000, 1111

1. Introduction

Currently, within the space flight community, there is a renewed interest in large spacecraft constellations owing to the very recent decrease in cost, to access space [1]. The bulk of this decrease is due to breakthroughs in launch vehicle cost, with those improvements having profound impacts on relaxing some of the previous performance requirements of satellite components. This relaxation has resulted in significantly less expensive spacecraft [2] and drawn a renewed interest in space, resulting in heavy investment into

the space business ecosystem and new options for constellations [3]. While many of the individual satellites within these constellations are shrinking in mass, volume, and cost [4], the constellations continue to grow in numbers, capability, and sophistication, requiring increased levels of automation [5] to properly orchestrate accomplishing the mission objectives [6] [7]. The aim of the research presented in this paper is to address the needs of these new constellations by providing a robust and extensible methodology for planning the operations for a networked constellation of low Earth orbit (LEO) satellites.

As the number of satellites within a constellation increases, the operational planning problem complexity compounds due to the spatial and temporal dependencies present within the constellation's operational environment and the limited resources onboard each vehicle [8]. Ensuring a feasible, flight-worthy plan for every satellite in the constellation is critical to sustained and autonomous mission operations. To address that critical need, the planning problem addressed by this paper spans the Earth sensing data collection, data storage, and ultimate data delivery to a ground-based user, all while ensuring the resources onboard each unique satellite are respected. Orchestrating these operations across a fleet of interconnected, or crosslinked, LEO satellites is the purpose and focus of this paper. Crosslinking enables satellites to potentially share information and resources across the constellation rather than relying solely on individual capabilities. Crosslinking does, however, increase the planning complexity due to the coordination requirement that one satellite must transmit while another receives. It also requires the satellites to maintain relative pointing to accomplish the task.

In facilitating the planning of a crosslink-enabled constellation, the research in this paper makes several contributions to the state of the art. The first is a unique graph creation method that trades graph complexity relative to optimality while always ensuring end-to-end operational connectivity and is an extension to the implementation in [9]. Next, the research provides a novel graph-based solution approach for planning crosslink operations between coordinated satellites. Third, a two-layer, graph-based heuristic is developed to inform selection of communication windows and efficiently arrive at an operational schedule for every satellite within the constellation. In summary, the research described herein provides a tractable solution technique for simultaneously considering scheduling of Earth sensing tasks, data storage, and data transfer operations across a constellation of crosslink-enabled space vehicles with the intent of maximizing mission utility across the con-

stellation.

The remainder of this paper will proceed with an in-depth review of previous constellation planning methods described in Section 2. This is followed by an overview of the problem-solution framework in Section 3 and its extension that supports crosslink planning in Section 4. A multi-layer framework is then introduced to improve computational performance in Section 5, followed by its application to a particular Earth sensing constellation mission in Section 6. The paper concludes with brief remarks in Section 7.

2. Related Work

Prior to reviewing the research contributions made within the constellation planning literature, it is imperative to properly classify the problems being addressed herein. The first is the base Geospatial Intelligence (GEOINT), or Earth sensing, planning problem, which plans Earth sensing and data downlink operations for each satellite within a constellation while attempting to optimize overall mission performance. The second problem is an extension of the first and deals with the crosslink-enabled Earth sensing mission in which satellites perform Earth sensing operations and are able to communicate not only with ground terminals (for downlink) but also with each other directly.

The Multi-Robot Task Allocation (MRTA) taxonomy defined by Gerkey and Matarić [10] and extended by Korsah [11] helps differentiate these problems by identifying the capabilities of the satellites, the requirements of the tasks, and the dependencies present within the problem space. The purpose of formally classifying the problems is to help find similarities and differences between relevant problem sets within the MRTA literature. The classification of the base Earth sensing and downlink problem is ID [ST-SR-TA], meaning that the agent satellites are capable of performing a single task (ST) at a time, tasks require only a single satellite to fulfill (SR), tasks are allocated over an extended period (TA), and in-schedule dependencies (ID) exist. The crosslink-enabled GEOINT mission extends this classification by adding in crosslink tasks that require multiple satellites (MR) to fulfill while also introducing cross-schedule (XD) dependencies between them. Therefore, the crosslink-enabled GEOINT mission can be classified as a combination of both ID [ST-SR-TA] and XD [ST-MR-TA]. With that formal classification addressed, the subsections that follow will review the literature applicable to each type and identify the unique contributions of this paper.

2.1. Earth Sensing and Downlink

The primary objective of the base GEOINT constellation planning problem classified as ID [ST-SR-TA] is to generate an efficient schedule of Earth sensing and downlink operations that fit within all specified mission constraints, including the resources of each satellite within the constellation. Several approaches have been presented to address a portion of this mission type. A host of work has focused on arriving at viable solutions to the prioritized Earth sensing or image collection problem [12] [13] [14] [15] [16] [17] [18]. Similarly, several researchers have investigated the data downlink problem and provided robust solutions to scheduling the transfer of data [19] [20] [21] [22] [23] [24] [25] [26]. However, in both the Earth sensing and data relay scheduling problems, the papers referenced only address part of the complete problem. The Earth sensing and downlink problems are directly coupled and, when solved simultaneously, can yield a reliable and more accurate solution. Otherwise, planning these operations independently will result in likely degradation of the solution since effective data transfer requires knowledge of data collection, and data collection requires knowledge of when data is to be transferred.

Some works have considered both the Earth sensing and data downlink within the same framework and provided various solutions to solve them [27] [28] [29]. Hu et al. simulated a LEO constellation of three satellites [28], while Peng et al. formulated the planning problem for a single satellite [29]. These contributions presented interesting methods for planning, but the extension to a proliferated LEO system is unclear [30]. A key element of constellation planning requires that the planning system be capable of consistently generating operationally feasible schedules for each individual satellite within a large constellation and make decisions to perform specific operations instead of others (e.g., downlink data instead of collect images). Augenstein et al. formulate the planning problem to address both Earth sensing and downlink operations simultaneously and also provide a novel heuristic approach to reduce the time required to arrive at a schedule solution. The intent of their formulation is to balance the data collected with the data downlinked and do so within a timeframe that meets the needs of an operational constellation [27]. Balancing the image data collection with data downlink prevents data from remaining onboard the satellite for extended periods and helps ensure efficient use of limited satellite resources. For a more in-depth review of that paper, see Appendix A.

Our previous work in [9] leverages the contributions of Augenstein et al.

and addresses one fundamental limitation of that approach. That limitation is the lack of clarity in how mission plans are generated for all satellites comprising the constellation while still optimizing operations across the fleet. The implementation in [9] provides an alternative solution method that relies on network flow theory to enable the planning of many satellites at once while respecting the mission constraints of both the constellation and the individual vehicles within it. The research in this paper extends that implementation with the additional capability of planning crosslink operations between satellites.

2.2. Crosslink-Enabled Earth Sensing and Downlink

This research extends the base Earth sensing problem into the cross-schedule dependency (XD) and multi-robot task (MR) domains due to the crosslink operational requirements mentioned earlier. The previously referenced papers of [23] and [25] do incorporate crosslinks in their formulations, but their research only addresses data transfer operations without the broader Earth sensing context in which this paper is interested. However, several other contributions have been made to the crosslink-enabled Earth sensing mission literature that are more applicable to this research. For example, Zhou et al. present a problem formulation and solution approach that attempts to maximize network throughput while also considering the limitations imposed by onboard satellite energy and data storage constraints [31]. Kondrateva et al. formulate a mixed-integer linear program (MILP) with the purpose of maximizing data throughput within a LEO satellite network and demonstrate it for a constellation of 18 satellites. This formulation considers data routing in conjunction with link scheduling and models data collection on the LEO vehicles to be a continuous flow of collection [32]. Wang et al. formulate this same problem as an event-driven, time-extended graph with nodes representing a discrete location in time and edges representing an observation, transmission, or storage window [33]. The associated MILP formulation further specifies constraints to ensure only a single target is collected at each time step, only a single transmission occurs at each time step, the amount of data transmitted does not exceed the amount collected, and that data are only transferred after they are collected [33]. Kennedy et al. formulated a mission planner with the intent of planning prioritized Earth sensing operations within specified revisit rates and also delivering low-latency observation to the ground via both crosslink and downlink operations within a constellation of satellites [34]. In a subsequent publication, the original

approach is updated by Kennedy et al. using a dual algorithm for planning both observation and communication operations across a crosslink-enabled constellation of satellites, dramatically reducing the latency of observation data delivery [35].

A consistent limitation across the crosslink-enabled, constellation planning literature is the lack of planning detailed Earth sensing and data transfer operations within the same framework. The value in using a constellation planning system is realized when it can generate a feasible schedule for each vehicle within the constellation and make decisions for conducting certain operations in lieu of others (e.g., perform Earth sensing instead of transferring data). This paper provides a response to these limitations and proposes a problem formulation and solution approach that considers all spacecraft operations collectively using a classical optimization framework. The crosslink-enabled GEOINT formulation presented in this paper addresses the current weaknesses in the literature by augmenting the original results of [9] and providing a more capable solution approach for addressing the challenges resulting from the addition of crosslink tasks.

3. Single-Layer Planning for Imaging and Downlink

This section provides a basic overview of the MILP formulation from previous work [9] coupled with important additions to the graph creation process that help manage complexity of the problem space. The methods highlighted help to lay the foundation of the baseline planning concept and then allow the extension into the crosslink domain. The graph creation method is first presented followed by the network flow MILP formulation. A complete summary of the variables used in the constellation planning formulation is provided in Table 1.

3.1. Downlink and Earth Sensing Graph Creation

The single-layer approach for graph creation is shown in Figure 1. The first four steps correspond to creating a directed acyclic graph (DAG), which forms the foundation for how the planning problem is formulated. It begins by computing the access windows of satellites to targets and ground station antennas. These access windows depend upon temporal and spatial constraints. The temporal constraints define time windows in which a task can be completed, such as seasonal conditions for targets and availability windows for ground antennas. The spatial constraints consider restrictions

Table 1: Notation used for planning.

Name	Description
Counting Variables	
n_s	Number of satellites
n_v	Number of nodes / vertices
n_e	Number of edges
n_g	Number of task groups
n_t	Number of time steps
n_{g-cl}	Number of crosslink groups
n_{g-v}	Number of visit groups
Graph Representation	
V	Set of nodes or vertices
v_i	Node i, $v_i \in V$
E	Set of edges, $E \subset V \times V$
e_i	Edge i, $e_i \in E$
G	Graph, $\{V, E\}$
K	Max number of task nodes a task node can connect to
N	Max number of nodes per group a task node can connect to
Variables of Optimization	
x_i	Integer variable representing flow along edge e_i
x	Vector of all x_i, $x = \begin{bmatrix} x_1 & ... & x_{n_v} \end{bmatrix}$
y_{lk}	Memory for satellite l at time k
$y_{l,max}$	The maximum memory capacity of satellite l
y	Vector of all y_i, $y = \begin{bmatrix} y_1^T, ..., y_{n_s}^T \end{bmatrix}^T$
z	All optimization parameters, $z = \begin{bmatrix} x^T & y^T \end{bmatrix}^T$
Utility and Scoring Parameters	
u_i	Utility associated with optimization variable i
u	Vector of all u_i
Flow Constraint Parameters	
D	Incidence matrix
b	RHS of flow constraint with one source, one sink, and unit flow
Group Constraint Parameters	
A_g	Group constraint matrix
b_g	Group constraint RHS
Memory Constraint Parameters	
A_{data}	Aggregate data constraint matrix
b_{data}	Aggregate RHS vector for the data constraint
Crosslink Group Constraint Parameters	
$\mathcal{I}_{i,d}^{g-cl}$	Index set of edges entering crosslink receive group i
$\mathcal{I}_{i,u}^{g-cl}$	Index set of edges entering crosslink send group i
A^{g-cl}	Crosslink group constraint matrix using the node formulation
Visit Group Constraint Parameters	
$\mathcal{I}_i^{g-v}$	Index set of edges entering visit group i
n_i^v	The maximum number of visits allowed to group i
n_v	Vector of all n_i^v. $n_v = \begin{bmatrix} n_1^V & \dots & n_{n_{g-v}}^v \end{bmatrix}^T$
A^{g-v}	Visit group constraint matrix
Indices and Index Mappings	
i, j	General indices into matrices and sets
l	Satellite index
k	Discrete time index

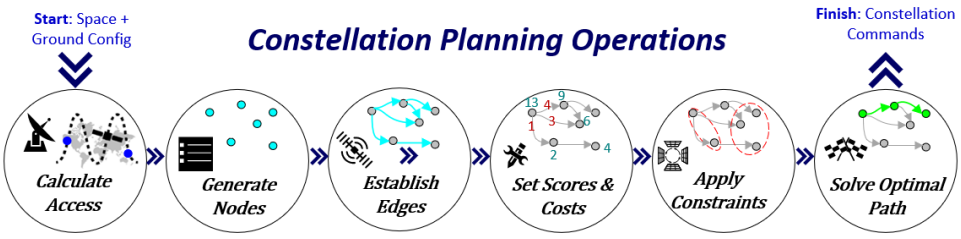


Figure 1: The constellation planning process from mission configuration through command generation.

on the elevation angle between the target and satellite for image quality, restriction on the azimuth angle to avoid obstructions, as well as lighting constraints (expressed as sun-elevation angle constraints). Given the satellite trajectory and assuming no orbital changes are considered by the planner, the spatial constraints are mapped to additional temporal constraints and combined to provide time windows in which each satellite may access each target or ground antenna.

The graph is then formulated directly from the access windows. The access windows are discretized, providing discrete points in time, or nodes, for which a particular satellite may access each target or ground terminal. Additional nodes, referred to as "backbone" nodes, are also added to ensure connectivity of the graph since long-duration edges are limited in the graph creation and can result in a disconnected graph.

The process for connecting the nodes with edges now discussed is distinct from [9] and helps manage the complexity of the generated graphs while also ensuring connectivity. The process is illustrated in Figure 2 and begins by binning task nodes within the time boundaries established by the backbone nodes. The edges are established by starting with the first backbone node (labeled node a in the figure) and connecting it to all task nodes within its window. Following this, each task node within the window is then connected to the backbone node that exists after the maximum slew time has elapsed. The maximum slew time is the time required by the satellite to perform a worst-case attitude maneuver, and applying the connection approach mentioned, ensures the satellite can reorient itself between nodes even under the most demanding circumstances. To illustrate this point, Figure 2 shows node i being connected to backbone node b, since the max slew time ends before node b. Conversely, task node j connects to backbone node c, since the max

slew time ends in the subsequent window. Note that if the max slew time matches the backbone node separation time, then each task node will be connected to the backbone node at the end of the subsequent window.

Other constraints are present within this graph building approach that restrict each node to connect to a maximum number of other nodes per group and a maximum number of total nodes. The purpose of these limits is to trade the complexity of the graph with the optimality of the future solution. In this context, a group is a collection of nodes pertaining to one access window for either a communication or Earth sensing operation. These constraints are specified by the parameters N and K. The value for N represents the maximum number of nodes within a particular group that any task node is allowed to connect to, while K represents the maximum number of total nodes a single task node can connect to. Allowing N and K to be infinite, results in a complex graph in which each node will connect to every other subsequent node if enough time exists to perform a transition between the task nodes. Conversely, setting N and K to very small numbers reduces the complexity of the graph by significantly reducing the number of connected nodes. Figure 2 illustrates this connectivity by showing the purple task node, labeled node r, connecting to two subsequent feasible nodes, one per group. This indicates that N is set to one (one node per group), and K is set to two (two total nodes). This graph connection process is followed for each node present within the planning horizon to connect nodes and node groups at the earliest opportunity, per the results in [9], and results in the final graph to be considered by the planner.

Once the nodes and edges have been determined, a score is placed on each node based on the value of performing the Earth sensing task. The node score is then assigned to the edge that connects to that node. A score is also assigned to each edge that represents a downlink operation. Conversely, a cost is assessed to the transition between nodes based upon the slew that must be performed. The score and cost of each edge is differenced and results in a utility denoted as u_i. As the utility defines an edge preference, an optimization problem can be formulated to choose which edges to take in an effort to maximize overall utility.

3.2. Network Flow MILP Formulation

Following establishment of the problem graph as explained in the previous subsection, it is now possible to convert it into a MILP that facilitates solving the optimal satellite paths through the graph. The variables of optimization

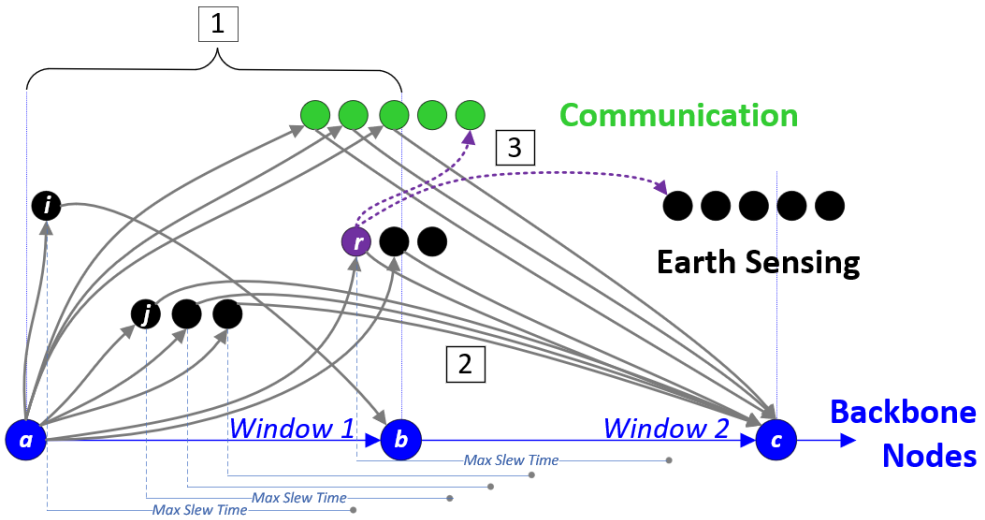


Figure 2: The method employed for connecting nodes with edges: 1.) Connect each backbone node to all the task nodes within the window. 2.) Connect each task node to the backbone node occurring after the max slew time has elapsed. 3.) Connect each task node to the feasible subsequent task nodes per specified limits.

include two types. The first is a binary variable for each edge, x_i, denoting whether or not the solution uses edge i. The vector x combines all x_i values. The second type of variable represents the data onboard each satellite for each time step. The data for satellite l at time k is denoted as $y_{l,k}$, with the aggregate vector of all data values represented as y.

Additionally, the problem conversion must properly account for the various constraints present within the optimization space. The general linear form of these constraints is presented herein with the detailed development of the constraints provided in [9]. These include a flow constraint that ensures movement through the graph from the start node to the end node. A coordination constraint is applied to avoid duplication of efforts by each satellite being planned such that only a single vehicle observes a particular ground target. This is necessary since each ground target observation opportunity starts as an access window, which is then discretized into potentially many opportunities (nodes). Finally, a data constraint is imposed on each satellite to ensure sufficient resources are available for capturing and storing the Earth sensing data in onboard memory. The MILP is formulated to maximize the mission utility u while being subject to the constraints mentioned.

This formulation is summarized in Equation (1) while a complete list of all planning variables is captured in Table 1.

$$\begin{aligned} \max_{z} \ & u^T z \\ \text{s.t. } & Dx = b \text{ (Flow)} \\ & A_g x \leq \mathbb{1}_{n_g} \text{ (Coordination)} \\ & A_{data} z = b_{data} \text{ (Data)} \\ & x_i \in \{0, 1\} \forall i \\ & 0 \leq y_{l,k} \leq y_{l,max} \ \forall \ l, k \\ & z = \left[x^T y^T\right]^T \end{aligned} \quad , \tag{1}$$

4. Augmenting the Single-Layer Formulation with Crosslinks

A solution to the MILP problem in Equation (1) provides a balance between Earth sensing and data downlinking, with the satellites having a limit to the amount of Earth sensing data that is allowed to stay onboard. However, it is often the case that a satellite will not have direct communication with a ground station for some time after the data have been collected. Thus, to facilitate the timely delivery of data, a crosslink capability between satellites is now introduced with the intent that it opens up additional avenues for data transport and delivery to the ground. This allows one satellite to communicate its onboard data to another, which can then transmit the data to a ground station with more favorable conditions (e.g., earlier availability for downlink access).

A specific example of this situation is illustrated in Figure 3. Satellite A is storing a significant amount of data, while Satellite B has significant storage available and will be flying through a data downlink window before Satellite A. Furthermore, Satellite A also has a high-priority Earth sensing window approaching. In this situation, it makes sense for Satellite A to crosslink data to Satellite B for an earlier downlink and allow Satellite A to prosecute the upcoming Earth sensing tasks. The following subsection formalizes this crosslink problem formulation and provides a simple example to solidify the concept.

4.1. Crosslink Formulation

A potential crosslink operation is represented in the graph through a combination of four nodes, two for each vehicle in the crosslink operation. Each

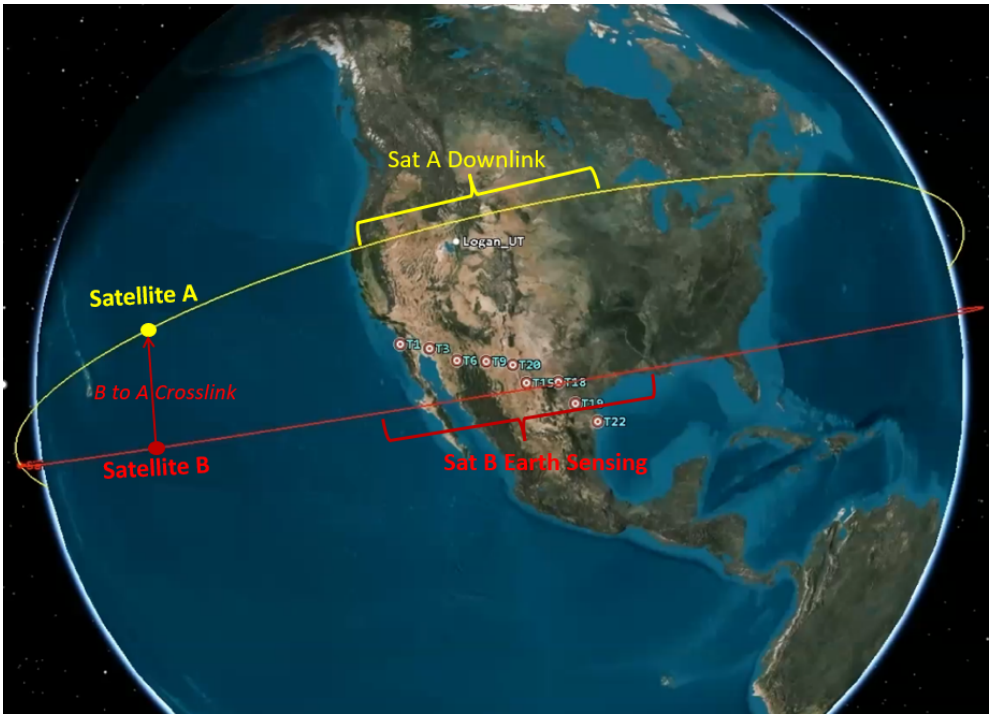
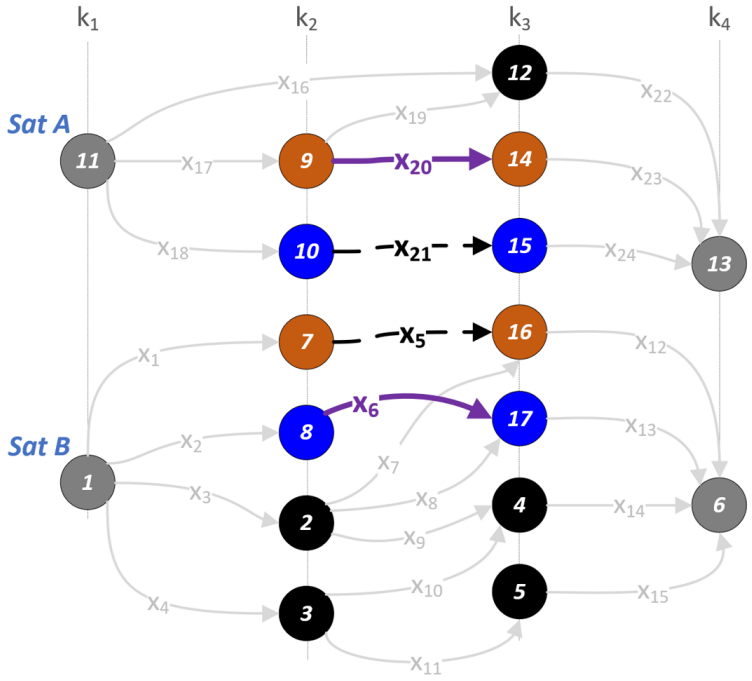


Figure 3: A scenario illustration for the crosslink concept. Satellite B sends data to Satellite A to free up storage resources and allow Satellite B to downlink data earlier.

node pair exists at a discrete instant in time and thus dictate a coordinated operation between satellites. During a crosslink operation, the two satellites will have the ability to transmit to, or receive from, the other satellite. While there are four nodes, the nodes must form part of the chosen path in one of two combinations: if a transmit node is selected from one satellite, then the corresponding receive node for the other satellite must be chosen and vice versa. Additionally, the transmit and receive nodes of a single satellite cannot be simultaneously chosen. Note that there is no need to add a constraint for the latter condition as both nodes occur at the same point in time, and the flow constraint will naturally disallow both nodes being chosen as part of the path.

Consider the two graphs shown in Figure 4. The black nodes represent Earth sensing tasks, while the blue and orange nodes represent transmit and receive crosslink tasks, respectively. Time is represented from k_1 to k_4. For the edges between time steps k_2 and k_3, edges 6 and 20 form a



Edge #	Memory
1	Initial B
2	Initial B
3	10
4	10
5	50
6	-50
7	0
8	0
9	10
10	10
11	20
12	0
13	0
14	0
15	0
16	10
17	Initial A
18	Initial A
19	10
20	50
21	-50
22	0
23	0
24	0

Figure 4: Grouping of crosslink nodes to ensure cooperation between Satellites A and B.

crosslink group, and edges 5 and 21 form another separate crosslink group. The dynamic memory variables for this graph are represented in tabular form on the right side of the figure. Notice that the edges arriving at the orange nodes increase the memory on a satellite, since it will receive the data from its neighbor, while the edges connecting to blue nodes are associated with a negative memory amount, since the satellite will transmit the data to its neighbor. This allows the memory to be properly conserved and accounted for across the constellation.

Crosslink will occur if one satellite's selected path through the graph traverses a transmit edge (existing between transmit nodes) and another satellite's path traverses the corresponding receive edge. Thus, the crosslink groups are formed as edge pairs, i.e., an edge between subsequent receive nodes and an edge between transmit nodes at the same time. The constraint matrix, where i is the row that identifies the edge coupling, and j is the

specific edge, is then constructed as

$$A^{g-cl} = [a_{ij}^{g-cl}] = \begin{cases} 1 & \text{Edge } j \text{ is receive for group } i \\ -1 & \text{Edge } j \text{ is transmit for group } i \\ 0 & \text{otherwise} \end{cases}.$$

The actual crosslink constraint is written as $A^{g-cl}x = 0$. Adding this single line to the original formulation shown in Equation (1) results in the MILP

$$\begin{aligned} & \max_{z} \ u^T z \\ \text{s.t. } & Dx = b \text{ (Flow)} \\ & A_g x \leq \mathbb{1}_{n_g} \text{ (Coordination)} \\ & A_{data} z = b_{data} \text{ (Data)} \\ & A^{g-cl} x = 0 \text{ (Crosslink)} \\ & x_i \in \{0, 1\} \forall i \\ & 0 \leq y_{l,k} \leq y_{l,max} \ \forall \ l, k \\ & z = \left[x^T y^T\right]^T \end{aligned} . \tag{2}$$

4.2. Crosslink Planning Example

A simplified example is now provided to illustrate the concept using the graph in Figure 4. As mentioned previously, edges 6 and 20 form a group, and edges 5 and 21 form another separate group. This grouping constraint ensures that the transmit and receive pairings occur together, as inherently required by the crosslink operation. These constraints are captured in matrix form as shown in Figure 5. The graphs are set up to allow for either Earth sensing or crosslink operations. Note that the initial memory values assigned to each spacecraft will have a direct impact on the paths chosen by the optimization routine due to the onboard memory availability. Two examples are now provided to illustrate the concept.

Example 1: Memory Space Available

For Example 1, consider the scenario where both Satellite A (initial node 11) and Satellite B (initial node 1) start the mission planning process with a completely empty onboard storage device that is capable of storing up to 200 GB. No memory is consumed yet when starting from the initial graph node

$A^{g\text{-}cl}$	EDGE																							
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Edges 20/6	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
Edges 21/5	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

Figure 5: Matrix form of crosslink constraint equations.

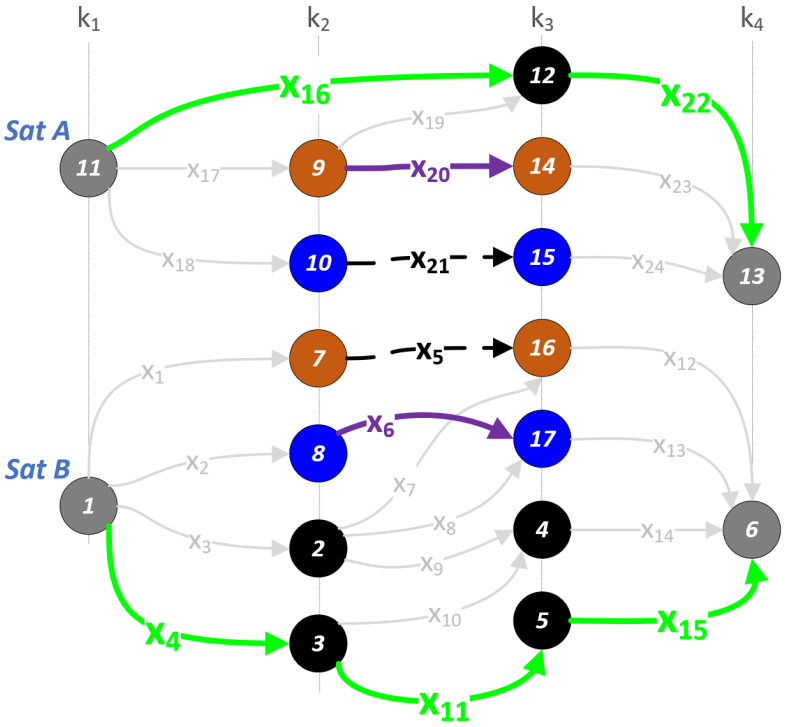


Figure 6: Example 1 Solution: Crosslink scenario results when sufficient onboard memory resources exist for imaging activities.

(1), and imaging opportunities are available for fulfillment as illustrated in Figure 4. Solving the optimal path through the graph results in the path highlighted in green in Figure 6. The corresponding stored data amounts, at each time step, are shown in Table 2. Note that the satellite operations match the selected path and never exceed the limit of 200 GB.

Example 2: Memory Space Unavailable

Now let us examine the scenario where the graph is still the same as the earlier scenario, but the onboard data recorder is initially at 80 GB

Table 2: Example 1 Results: Onboard memory status at each time step for Satellites A and B.

Time Step	Sat A Memory	Sat B Memory
k1	0 GB	0 GB
k2	0 GB	10 GB
k3	10 GB	30 GB
k4	10 GB	30 GB

Table 3: Example 2 Results: Onboard memory status at each time step for satellites A and B.

Time Step	Sat A Memory	Sat B Memory
k1	80 GB	200 GB
k2	80 GB	200 GB
k3	130 GB	150 GB
k4	130 GB	150 GB

on Satellite A and 200 GB on Satellite B. With these initial conditions, Satellite B is unable to follow the same route from the previous example due to the data storage constraints. To alleviate the situation, Satellite B must transmit data to Satellite A, and Satellite A must receive the data. This result is illustrated by solving the MILP and finding the path solution shown in Figure 7. The corresponding data storage at each time step is shown in Table 3. Note that the data constraint is respected, and that both satellites cooperate to ensure that the memory limits are not violated. This formulation and solution provides a means to solve the cross-schedule dependency problem when using crosslinking in the context of constellation planning.

5. Two-Layer Planning Approach

The previously described graph formulation presented in Equation (2) of Section 4.1, combined Earth sensing, data crosslink, and data downlink tasks together into one highly complex graph. To help alleviate the complexity of this all-encompassing graph, a two-layer approach is introduced that decouples the Earth sensing and communication operations yet reduces the potential detrimental impacts to the overall schedule.

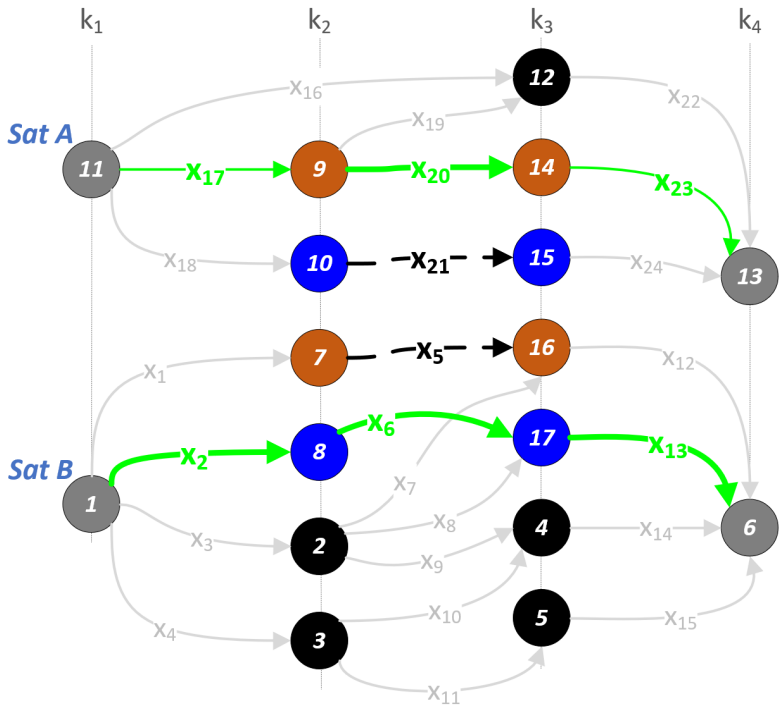


Figure 7: Example 2 Solution: Crosslink operation selected due to requiring additional onboard data storage resources for Satellite B.

Augenstein et al. used a similar decoupled heuristic approach that first solved for the downlink windows and then the individual data collection operations afterwards [27]. The downlink window calculation and assignment relied on estimating several heuristics for the opportunity costs and data potentially collected at each instant in time. With these heuristics, it was possible to assign downlink windows with a reasonable estimate of the impacts to the imaging assignment decisions. The result was a significant improvement in the time required to calculate an operationally valid satellite schedule [27]. Additional detail regarding the heuristics from Augenstein et al. is available in Appendix A.

The planning framework presented in the next subsection also uses a two-layer method for planning but does so using a graph-based approach for the entire constellation while yielding a unique operational schedule for each satellite within the fleet. This approach reduces complexity while still delivering a reliable planning solution for the constellation and does so in an

intuitive graphical context. In the first layer, Earth sensing activities are dramatically simplified into a line graph, that directly informs communication window selection. Then, during the second layer, the Earth sensing activities are planned in detail with the selected communications windows to yield a complete schedule of operations for all satellites within the constellation.

5.1. Earth Sensing Heuristic Graph

The basic premise of the two-layer planning approach is to decouple the problem components of communication window assignment from the Earth sensing operations assignment while minimizing degradation to the overall decision making. This is accomplished by first computing a heuristic about the likely Earth sensing operations each satellite could perform and then making informed decisions about when to schedule communication windows based on estimated sensing tasks, data storage capacity, and communication opportunities.

To determine when to schedule data transfer windows (both crosslink and downlink) the system must have a clear estimate of when and how much data will be collected during the Earth sensing activities to be performed. To address this need, all Earth sensing task fulfillment windows are discretized and converted into nodes in the same manner discussed in Section 3. Mission utility is assigned, coordination constraints are relaxed, memory constraints are removed, and then a longest path algorithm determines the selected route through the graph for each individual satellite. The resulting route is then discretized at a fixed time step into a line graph, which represents a reasonable estimate of the Earth sensing operations that could be performed.

5.2. Communication Window Selection

The heuristic Earth sensing line graph is then combined with the communication opportunity windows and results in a more complete representation of all potential mission tasks. The purpose of combining the graphs is to provide the planning system with an estimate of how mission utility is impacted by operational decisions. This opportunity cost is valuable when deciding between conflicting Earth sensing and communication operations and the proposed approach provides this insight while also reducing the complexity of the overall mission planning space. With this graph now in place, it is possible to apply the network flow approach and find a constellation-level solution for all communication windows within the planning horizon while

still having a reasonable estimate of the likely Earth sensing operations that will occur.

An example of a complete graph, for a single satellite, is depicted in Figure 8. Note that within this graph there are five node types, which are listed and described below:

1. Backbone Nodes: These nodes are placed in the graph to act as anchor points only. No tasks are performed on or between these nodes and thus are assigned zero utility and zero data.
2. Heuristic Data Collection Nodes: These nodes correspond to the satellite collecting data. Edges between subsequent data collection nodes will have positive utility and positive data acquired.
3. Downlink Nodes: These nodes correspond to the satellite downlinking data to a ground station. They will have positive utility and negative data acquired.
4. Crosslink-Transmit: These nodes correspond to the satellite transmitting data to another satellite. They will have non-positive utility and negative data acquired.
5. Crosslink-Receive: These nodes correspond to the satellite receiving data from another satellite. They will have non-positive utility and a positive data acquired.

To avoid oscillating between communication links and sensing operations, it may be advantageous to have satellites crosslink at most one time per crosslink opportunity and be limited in the number of times they can downlink in a downlink opportunity. Without this constraint, it would be possible for a satellite to select downlink tasks, transition to Earth sensing tasks, and then return to the downlink window. This may be undesirable in an operational environment. The constraints that help the system avoid this behavior are referred to as visit constraints, and they are now added to the formulation for operational viability. Let n_{g-v} be the number of groups with visit constraints. This number represents the total number of task groups n_g, including both sensing and communication groups. Allow $\mathcal{I}_i^{g-v}$ be the index set corresponding to incoming edges for the i^{th} group. Assume that group i should be visited a maximum of n_i^v times with

$$n^v = \begin{bmatrix} n_1^v \\ \vdots \\ n_{n_{g-v}}^v \end{bmatrix}.$$

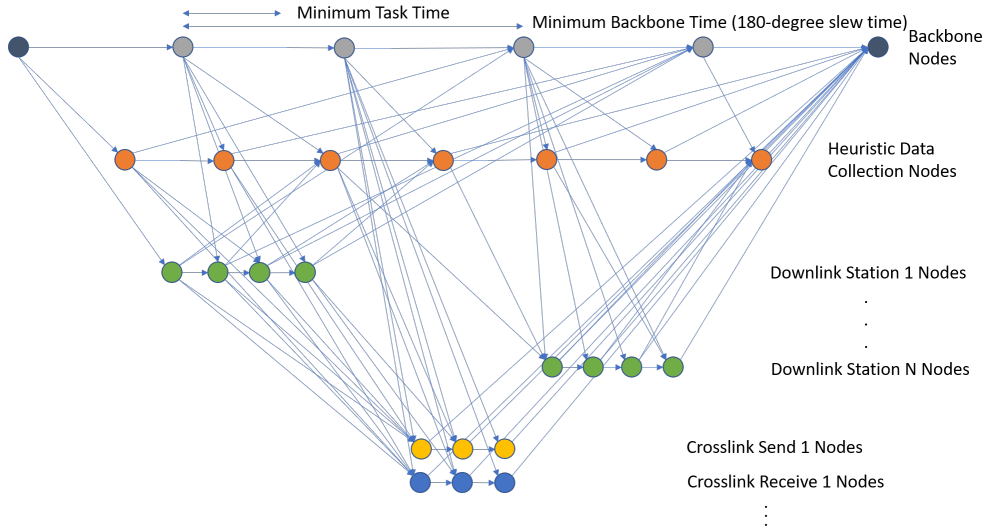


Figure 8: A representative flow for the communication layer in the two-layer network flow formulation.

The visit constraint matrix can be defined as

$$A^{g-v} = [a_{ij}^{g-v}] = \begin{cases} 1 & j \in \mathcal{I}_i^{g-v} \\ 0 & \text{otherwise} \end{cases},$$

with the visit constraint defined as $A^{g-v}x \leq n^v$. The full optimization problem then becomes

$$\begin{aligned} & \max_z \ u^T z \\ \text{s.t. } & Dx = b \text{ (Flow)} \\ & A_{data} z = b_{data} \text{ (Data)} \\ & A^{g-cl} x = 0 \text{ (Crosslink)} \\ & A^{g-v} x \leq n^v \text{ (Visit)} \\ & x_i \in \{0, 1\} \forall i \\ & 0 \leq y_{l,k} \leq y_{l,max} \ \forall \ l, k \\ & z = \left[x^T y^T\right]^T \end{aligned} . \tag{3}$$

Note the absence of $A_g x \leq \mathbb{1}_{n_g}$, from the original formulation specified in

Equation (2). Since the actual tasks have been abstracted within the groups, the visit constraint addresses the original coordination constraint as well.

5.3. Selection of Earth Sensing Tasks

The approach outlined in the previous sections for the communications layer will yield a plan for each satellite indicating when to communicate, either via downlink or crosslink. The next step is to determine when to perform Earth sensing tasks for each satellite while respecting the selected communication windows. The graph creation process now combines Earth sensing opportunity nodes with the previously-solved communication assignments and adds the edges where feasible transitions exist. Note that if an Earth sensing task node is unable to connect to the required communication tasks, it is discarded from the graph. Doing so reduces the overall complexity of the graph and ultimately speeds solution discovery. Solving the optimal path through the resulting graph, for each vehicle within the constellation, results in the final plan for all satellites and generates an operational schedule of both Earth sensing and data transfer operations. A visual summary of this multi-layer planning approach is provided in Figure 9, with Earth sensing tasks being shown as solid blue nodes and communication windows being shown as nodes marked with an X. For the sake of simplicity, backbone nodes are not shown.

6. Results

To evaluate the multi-layer planning approach to the crosslink-enabled mission concept, a specific scenario is now evaluated. The scenario constellation is comprised of two sets of satellites. The first set is made up of eight sun synchronous satellites orbiting at an altitude of 550 km and configured in a Walker Delta pattern. These satellites have the ability to collect and store up to 100 GB of sensing data and also transmit at a rate of 24 MB/sec. The Earth sensing targets are spread roughly evenly across the Americas and require 2 GB of memory storage space each. An additional set of eight equatorial satellites orbiting at an altitude of 2,000 km completes the constellation. These satellites are capable of storing up to 1 TB of data, transmit and receive at 24 MB/sec, and have excellent access opportunities to the ground terminals, which are located on land near the equator. A full list of these ground terminals is provided in Table 4. The described orbital configuration is shown graphically in Figure 10.

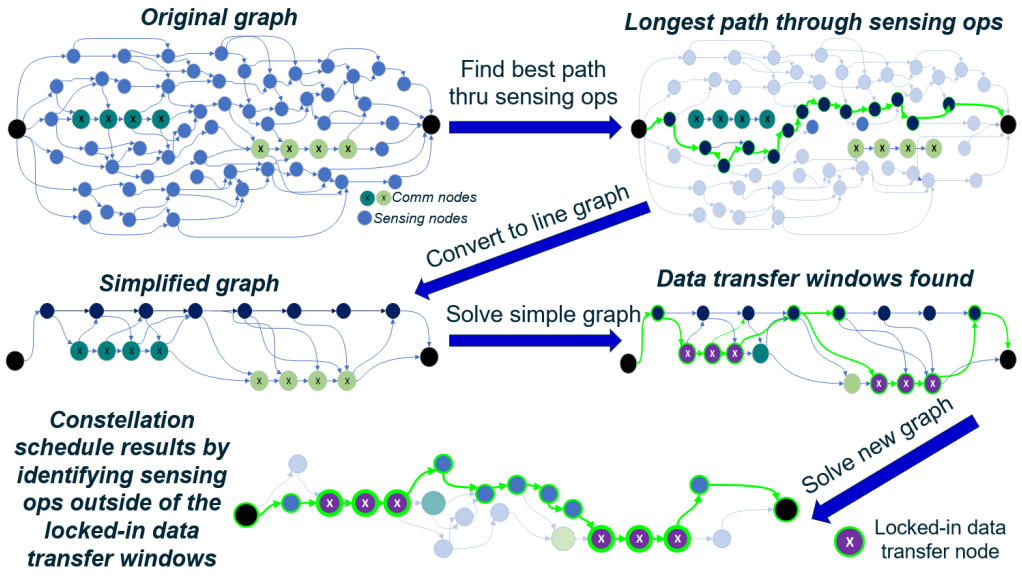


Figure 9: The graph-based constellation planning process using the two-layer approach.

Planning this scenario over five orbital periods of the sun synchronous satellites and using the single layer approach without crosslinks enabled, results in the plots shown in Figure 11. The figure contains four plots which, from top to bottom, show the time series of the memory state, data collected, data crosslinked, and data downlinked. Memory state refers to the amount of data onboard the satellites, data collected refers to the amount of data that results from performing Earth sensing activities, data crosslinked is the data sent or received between vehicles, and data downlinked is the amount of data

Table 4: Ground terminal locations for the flight scenario.

Ground Terminal Name	**Latitude, Longitude**
Libreville, Gabon	0.372°, 9.478°
Macapa, Brazil	0.0162°, -51.074°
Manokwari, Papua	-0.865°, 134.112°
Nairobi, Kenya	-1.300°, 36.935°
Pariaman, Sumatra	-0.566°, 100.114°
Quito, Ecuador	-0.187°, -78.518°
Santa Rosa, Ecuador	-0.619°, -90.445°

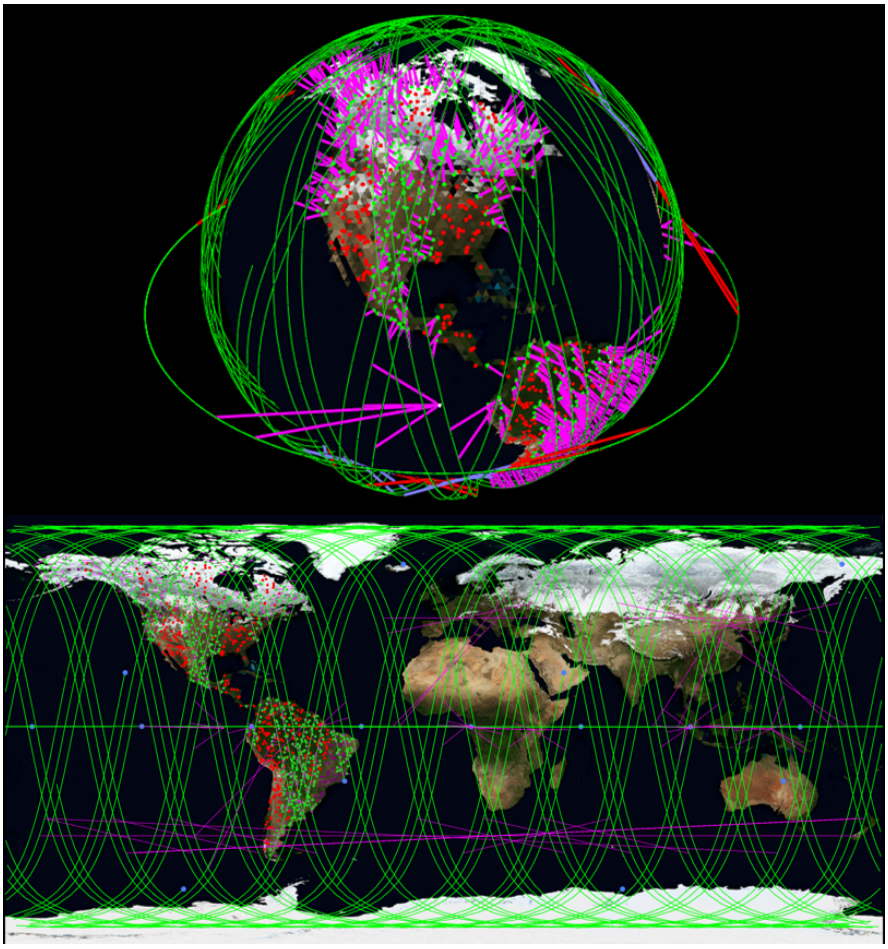


Figure 10: Orbital scenario for the constellation of eight sun synchronous Earth sensing satellites and eight equatorial communication satellites.

delivered to users on the ground. Individual satellite results are shown in blue dashed lines (right axis) while the full constellation results are plotted in red solid lines (left axis). Note that Figure 11 shows no crosslinking occurring since that functionality is not enabled. It is simply plotted here for continuity when comparing the next scenario. The key takeaway from these results is that this constellation is able to collect and downlink approximately 867 GB of data within the five orbital revolutions.

The next step is to illustrate the capability of adding the crosslink feature to the constellation and planning it using the two-layer approach discussed

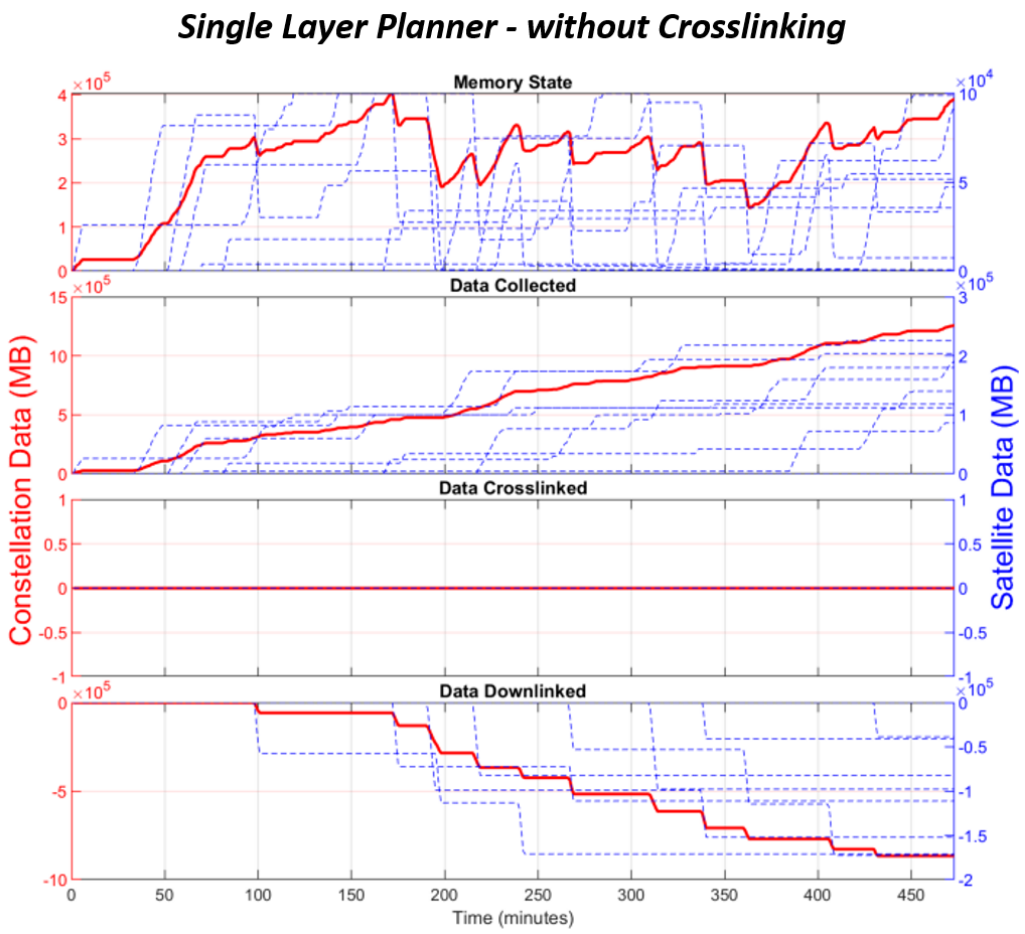


Figure 11: Results from the constellation planning simulation without crosslink available.

in Section 5. The results illustrated in Figure 12 show the improvement in data delivery when flying the crosslink-enabled constellation of Earth sensing and communication satellites. Note, that the figure shows variation in data crosslinked for individual satellites since some must transmit (decrease data) and some must receive (increase data). However, the summation across the constellation remains at 0 MB, since the crosslink data set is conserved. Data downlinked increased to 1086 GB, up by approximately 220 GB or 110 Earth sensing data sets, within only five orbital revolutions. Extrapolating this performance results in 661 GB more data delivered every day or up to 242 TB per year when utilizing the crosslink functionality. These results

provide substantial evidence that the capability could dramatically improve overall mission utility for Earth sensing missions.

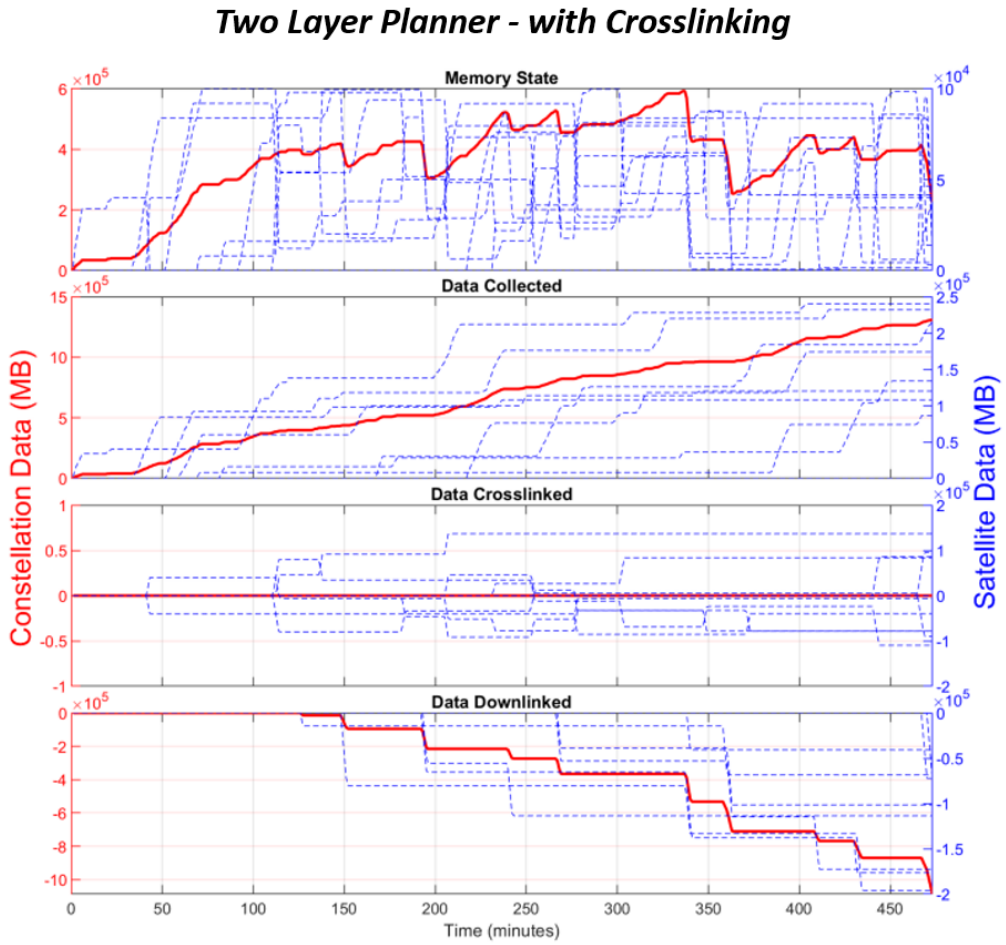


Figure 12: Results from the constellation planning simulation with crosslinks enabled.

The crosslink-enabled constellation was able to improve the amount of data delivered to the ground by providing more options for the Earth sensing satellites to offload data. Without crosslinking being available, the sun synchronous sensing satellite performance was downlink constrained. Even though the crosslinking operations consumed satellite operations time, it offered an effective way for the sensing satellites to free up onboard memory space for additional imaging, even though an equivalent amount of downlink time was still required for complete delivery of the data sets to the ground.

These results demonstrate the flexibility of the two-layer planner to accommodate both in-schedule and cross-schedule dependencies. Furthermore, the results illustrate the potential value of providing the ability to crosslink data between satellites for more effective delivery to users.

7. Conclusion

This paper has described the operational planning concept for a constellation of crosslink-enabled satellites and offered a solution for addressing the challenge of planning coordinated operations for a cross-schedule dependent system. The results discussed are promising and demonstrate the value in continued research using the two-level approach for proliferated LEO constellations. The problem formulation and solving methods described in this paper have important implications for cross-schedule dependent problems by providing a dependable strategy for planning operations across a system of crosslink-capable agents in a time-extended mission context.

Appendix A. In-Depth Review of Augenstein et al.

The formulation by Augenstein et al. addresses many of the requirements for a valid GEOINT constellation planning method and also introduces various interesting, mission-specific constraints to guide the planning system's decisions to ensure the mission can be realistically conducted. However, the formulation does not provide a clear representation of each satellite's individual plan. The mathematical formulation simply organizes the problem in time for a single agent and does not explicitly provide a tractable method of determining the optimal schedule for all satellites, collectively comprising the constellation. While it appears the formulation is used for operating a fleet of satellites, the literature lacks clear specifics on how this is accomplished in practice and thus requires a more complete explanation.

Augenstein et al. mathematically formulate the problem as:

$$\begin{aligned} \min_{x_i} & \sum_{i \in S_{dl}} \left(\tilde{c}_i x_i + \alpha d_i y_i \right) \\ \text{s.t. } & x_i \in \{0, 1\} \\ & y_i \in \mathbb{R} \\ & y_i \geq 0 \\ & \vdots \\ & y_i \geq y_{i-1} - \tilde{a}_i x_i + \tilde{b}_i \end{aligned} \quad , \tag{A.1}$$

where, within the objective function, $\tilde{c}_i$ represents a heuristic for the opportunity cost of a downlink task at time step i, x_i is the decision variable for task node selection, α provides a weighting factor on the value of the onboard data, d_i is the penalty associated with the data onboard at time step i, and finally y_i is the actual amount of data onboard at the same time step. The term S_{dl}, specifying the bound on the summation, represents only the set of downlink opportunities. The vertical dots represent additional constraints respecting slew agility, visiting the ground station periodically, and performing consecutive downlinks when possible. These constraints are specific to the mission being addressed by Augenstein et al. and do not directly impact this discussion. However, further detail is provided in [27], and a summary of the variables used by Augenstein et al. is provided in Table A.5. A high-level explanation of the terms in the objective function is as follows:

- $\sum_{i \in S_{dl}}$: The optimization is done purely for the downlink nodes (S_{dl})
- $\tilde{c}_i x_i$: The opportunity cost associated with selecting the i^{th} downlink node
- $\alpha d_i y_i$: Penalty on the data at node i

The final constraint describes the data as a function of the previous node and the choice of whether or not to collect data at the current node. The $\tilde{a}_i$ is somewhat more complicated than at face value, and what follows are details not present in [27]. To describe the full meaning of $\tilde{a}_i$, the data at time i can be written in two cases:

$$y_i = \begin{cases} y_{i-1} - \tilde{\alpha}_i & x_i = 1 \\ y_{i-1} + \tilde{b}_i & x_i = 0 \end{cases}, \; \tilde{\alpha}_i \geq 0, \; \tilde{b}_i \geq 0.$$

In words, y_i decreases by $\tilde{\alpha}_i$ if node i is chosen and increases by $\tilde{b}_i$ if node i is not chosen. Since x_i is binary, the data could be written as

$$y_i = y_{i-1} + (1 - x_i)\tilde{b}_i - \tilde{\alpha}_i x_i,$$

which, factoring out x_i, can be expressed as

$$y_i = y_{i-1} - (\tilde{b}_i + \tilde{\alpha}_i)x_i + \tilde{b}_i.$$

Thus, $\tilde{a}_i = \tilde{b}_i + \tilde{\alpha}_i$. So, $\tilde{a}_i$ is actually the sum of the data that could be obtained if node i were not chosen and the data that would be downlinked if it were. $\tilde{b}_i$ is the data that could be obtained if node i were not chosen. There are a few points that should be made on the formulation, which are summarized below as follows:

1. The opportunities are sorted in terms of time (i.e., x_1 occurs at a time less than or equal to x_2).
2. The formulation appears to be formulated for a single agent as y_i, for one agent should not be related to the value of another agent (i.e., you cannot simply order the y_i in time and relate y_i to y_{i-1} for another agent).
3. An inequality is used to express y_i instead of an equality. Thus, the amount of data onboard after a downlink opportunity is penalized, which forces the equality satisfaction at the optimal value (amount that can be downlinked within the window) if the data is positive. This also allows the visiting of a downlink node to utilize only a portion of the downlink capability. If there is less data onboard than could be downlinked, then the $y_i \geq 0$ constraint combined with the inequality constraint will keep the data at or above a zero threshold.

The values for $\tilde{c}_i$, $\tilde{a}_i$, and $\tilde{b}_i$ are not precisely known but are instead estimated using a heuristic.

References

[1] H. Jones, "The recent large reduction in space launch cost," in *48th International Conference on Environmental Systems*, 2018.

[2] J. R. Wertz and W. J. Larson, *Reducing space mission cost*. Microcosm Press Torrance, CA, 1996.

Table A.5: Notation used for Planning

Variables used by Augenstein et al. ([27])

Name	Description
x_i	Binary variable on whether or not to select node i
y_i	Amount of data on board (not a notion of agent)
d_i	Penalty on data at node i
α	Weighing on data
S_{dl}	Set of nodes corresponding to a downlink
$\tilde{c}_i$	Heuristic for the opportunity cost associated with downlink node i
$\tilde{a}_i$	Heuristic sum of possible data downlinked and the possible data obtained
$\tilde{b}_i$	Heuristic for the amount of data that could be obtained if node i not chosen

[3] S. Mukherjee and R. Jain, "Significant contribution of private investment in the hybrid nature of booming global space economy (2011–2021 q1)," in *72nd International Astronautical Congress (IAC)*, 10 2021.

[4] G. Stock, J. Fraire, H. Hermanns, E. Cruz, A. Isaacs, and Z. Imbrosh, "On the automation, optimization, and in-orbit validation of intelligent satellite constellation operations," in *Small Satellite Conference*, 2021.

[5] M. K. Ben-Larbi, K. F. Pozo, T. Haylok, M. Choi, B. Grzesik, A. Haas, D. Krupke, H. Konstanski, V. Schaus, S. P. Fekete, *et al.*, "Towards the automated operations of large distributed satellite systems. part 1: Review and paradigm shifts," *Advances in Space Research*, vol. 67, no. 11, pp. 3598–3619, 2021.

[6] J. R. Kopacz, R. Herschitz, and J. Roney, "Small satellites an overview and assessment," *Acta Astronautica*, vol. 170, pp. 93–105, 2020.

[7] J. Cappaert, F. Foston, P. S. Heras, B. King, N. Pascucci, J. Reilly, C. Brown, J. Pitzo, and M. Tallhamm, "Constellation modelling, performance prediction and operations management for the spire constellation," in *Small Satellite Conference*, 2021.

[8] J. D. Ullman, "Np-complete scheduling problems," *Journal of Computer and System sciences*, vol. 10, no. 3, pp. 384–393, 1975.

[9] S. A. Cox, G. N. Droge, J. H. Humble, and K. D. Andrews, "A network flow approach for constellation planning," *Space*

Mission Planning Operations, pp. 1–11, 2022. [Online]. Available: http://dx.doi.org/10.20517/smpo.2022.01

[10] B. P. Gerkey and M. J. Matarić, "A formal analysis and taxonomy of task allocation in multi-robot systems," *The International Journal of Robotics Research*, vol. 23, no. 9, pp. 939–954, 2004.

[11] G. A. Korsah, A. Stentz, and M. B. Dias, "A comprehensive taxonomy for multi-robot task allocation," *The International Journal of Robotics Research*, vol. 32, no. 12, pp. 1495–1512, 2013. [Online]. Available: http://dx.doi.org/10.1177/0278364913496484

[12] P. K. Sinha and A. Dutta, "Multi-satellite task allocation algorithm for earth observation," in *2016 IEEE Region 10 Conference (TENCON)*. IEEE, 2016, pp. 403–408.

[13] F. Yao, J. Li, Y. Chen, X. Chu, and B. Zhao, "Task allocation strategies for cooperative task planning of multi-autonomous satellite constellation," *Advances in Space Research*, vol. 63, no. 2, pp. 1073–1084, 2019.

[14] P. Tangpattanakul, N. Jozefowiez, and P. Lopez, "A multi-objective local search heuristic for scheduling earth observations taken by an agile satellite," *European Journal of Operational Research*, vol. 245, no. 2, pp. 542–554, 2015.

[15] S. Nag, A. S. Li, V. Ravindra, M. S. Net, K.-M. Cheung, R. Lammers, and B. Bledsoe, "Autonomous scheduling of agile spacecraft constellations with delay tolerant networking for reactive imaging," *arXiv preprint arXiv:2010.09940*, 2020.

[16] S. Mitrovic-Minic, D. Thomson, J. Berger, and J. Secker, "Collection planning and scheduling for multiple heterogeneous satellite missions: Survey, optimization problem, and mathematical programming formulation," in *Modeling and Optimization in Space Engineering*. Springer, 2019, pp. 271–305.

[17] J. Berger, N. Lo, and M. Barkaoui, "Quest – a new quadratic decision model for the multi-satellite scheduling problem," *Computers & Operations Research*, vol. 115, p. 104822, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0305054819302643

[18] L. He, M. de Weerdt, and N. Yorke-Smith, "Time/sequence-dependent scheduling: the design and evaluation of a general purpose tabu-based adaptive large neighbourhood search algorithm," *Journal of Intelligent Manufacturing*, vol. 31, no. 4, pp. 1051–1078, 2020. [Online]. Available: https://link.springer.com/article/10.1007/s10845-019-01518-4

[19] D. Karapetyan, S. Mitrovic-Minic, K. T. Malladi, and A. P. Punnen, "The satellite downlink scheduling problem: A case study of radarsat-2," in *Case Studies in Operations Research*. Springer, 2015, pp. 497–516.

[20] B. Deng, C. Jiang, L. Kuang, S. Guo, J. Lu, and S. Zhao, "Two-phase task scheduling in data relay satellite systems," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 2, pp. 1782–1793, 2017.

[21] B. Song, F. Yao, Y. Chen, Y. Chen, and Y. Chen, "A hybrid genetic algorithm for satellite image downlink scheduling problem," *Discrete Dynamics in Nature and Society*, vol. 2018, 2018.

[22] S. Spangelo, J. Cutler, K. Gilson, and A. Cohn, "Optimization-based scheduling for the single-satellite, multi-ground station communication problem," *Computers & Operations Research*, vol. 57, pp. 1–16, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0305054814002809

[23] X. Gu, J. Bai, C. Zhang, and H. Gao, "Study on tt&c resources scheduling technique based on inter-satellite link," *Acta Astronautica*, vol. 104, no. 1, pp. 26–32, 2014.

[24] J. Li, H. Chen, and N. Jing, "A data transmission scheduling algorithm for rapid-response earth-observing operations," *Chinese journal of Aeronautics*, vol. 27, no. 2, pp. 349–364, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1000936114000235

[25] W.-h. Zhao, J. Zhao, S.-h. Zhao, Y.-j. Li, X. Li, *et al.*, "Resources scheduling for data relay satellite with microwave and optical hybrid links based on improved niche genetic algorithm," *Optik*, vol. 125, no. 13, pp. 3370–3375, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0030402614001387

[26] J. Castaing, "Scheduling downloads for multi-satellite, multi-ground station missions," in *Small Satellite Conference*, 2014. [Online]. Available: https://digitalcommons.usu.edu/smallsat/2014/FJRStudentComp/4/

[27] S. Augenstein, A. Estanislao, E. Guere, and S. Blaes, "Optimal scheduling of a constellation of earth-imaging satellites, for maximal data throughput and efficient human management," in *Proceedings of the International Conference on Automated Planning and Scheduling*, vol. 26, no. 1, 2016.

[28] X. Hu, W. Zhu, B. An, P. Jin, and W. Xia, "A branch and price algorithm for eos constellation imaging and downloading integrated scheduling problem," *Computers and Operations Research*, vol. 104, pp. 74–89, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0305054818303113

[29] S. Peng, H. Chen, J. Li, and N. Jing, "Approximate path searching method for single-satellite observation and transmission task planning problem," *Mathematical Problems in Engineering*, vol. 2017, 2017.

[30] J. C. McDowell, "The low earth orbit satellite population and impacts of the spacex starlink constellation," *The Astrophysical Journal Letters*, vol. 892, no. 2, p. L36, 2020.

[31] D. Zhou, M. Sheng, X. Wang, C. Xu, R. Liu, and J. Li, "Mission aware contact plan design in resource-limited small satellite networks," *IEEE Transactions on Communications*, vol. 65, no. 6, pp. 2451–2466, 2017.

[32] O. Kondrateva, H. Döbler, H. Sparka, A. Freimann, B. Scheuermann, and K. Schilling, "Throughput-optimal joint routing and scheduling for low-earth-orbit satellite networks," in *2018 14th Annual Conference on Wireless On-Demand Network Systems and Services (WONS)*. IEEE, 2018, pp. 59–66.

[33] Y. Wang, M. Sheng, W. Zhuang, S. Zhang, N. Zhang, R. Liu, and J. Li, "Multi-resource coordinate scheduling for earth observation in space information networks," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 2, pp. 268–279, 2018.

[34] K. Cahoy and A. K. Kennedy, "Initial results from access: an autonomous cubesat constellation scheduling system for earth

observation," in *Small Satellite Conference*, 2017. [Online]. Available: https://digitalcommons.usu.edu/smallsat/2017/all2017/98/

[35] A. K. Kennedy, "Planning and scheduling for earth-observing small satellite constellations," Ph.D. dissertation, Massachusetts Institute of Technology, 2018.

CHAPTER 7

CONCLUSION

The primary objective of a constellation planning system is to efficiently use all system resources available to accomplish the specified mission. Research into this promising field is timely due to the tremendous government and industry interest in satellite constellations. Two specific constellation mission types were studied; that of long-range RPO, and GEOINT or Earth sensing. These two mission types were consistently studied in our research with the aim of developing new and tractable methods for addressing both. The accomplishments realized in our constellation planning and scheduling research span the following:

1. Mathematically formulated the long-range RPO constellation planning problem for LEO and a planning routine to solve the formulation using a distributed algorithm.
   (a) Augmented an existing routine with a J2 orbital model and a mission-relevant utility function to be maximized within a constrained framework.
   (b) Evaluated the formulation and solving method effectiveness in scheduling multiple agents in LEO to service existing RSOs in nearby orbits.
2. Mathematically formulated the GEOINT constellation planning problem and developed a methodology for solving it optimally using network flow theory, thus providing a tractable, realistic option for future constellation missions.
   (a) Modeled the extended time task allocation opportunities using a directed acyclic graph.
   (b) Applied new constraints to guide the constellation toward global optimality while respecting individual capabilities and resources for each satellite within the constellation.

(c) Evaluated the algorithm effectiveness in scheduling multiple agents in performing an Earth sensing mission with image collection and data downlink tasks only (ID[ST-SR-TA] problem type).

3. Mathematically formulated the Earth sensing constellation mission with the addition of cooperative crosslinking between satellite pairs to provide better connectivity and additional routes for managing data within the constellation.

   (a) Extended the ID[ST-SR-TA] formulation to enable cooperation between satellites and addressed the cross-schedule dependent (XD[ST-MR-TA]) planning problem.

   (b) Devised the necessary additional constraints, beyond the in-schedule dependencies, for enabling crosslink cooperation between constellation satellites.

   (c) Demonstrated how crosslink-enabled constellation satellites can cooperate in achieving mission goals.

The research successfully applied a crawl, walk, run approach by first modifying an existing multi-agent task allocation routine to address the RPO servicing mission, then developing a planning method for the ID [ST-SR-TA] Earth sensing constellation planning problem, and finally extending that formulation to address the cross-schedule dependency problem by incorporating crosslinks, specified as XD [ST-MR-TA]. The results of this research have fully demonstrated the utility and extensibility of the constellation planning methods developed and illustrated robust constellation planning methods with mission-realistic applicability. The contributions herein can be leveraged by commercial, civil, and military constellation systems interested in more effective cooperation.

While the GEOINT constellation planning methodology presented in this document focuses on a centralized implementation, future research may find a promising extension in a distributed method that supports better operations and survivability when limited interaction to each individual satellite might be infeasible or degraded. Such distributed planning methods would apply to the same systems as the centralized approach but enable more independent and autonomous operation.

REFERENCES

[1] S. Achor, *Big Potential: How Transforming the Pursuit of Success Raises Our Achievement, Happiness, and Well-Being*. Crown, 2018. [Online]. Available: https://books.google.com/books?id=TZgoDwAAQBAJ

[2] A. Almaatouq, M. Alsobay, M. Yin, and D. J. Watts, "Task complexity moderates group synergy," *Proceedings of the National Academy of Sciences*, vol. 118, no. 36, p. e2101062118, 2021.

[3] H. Jones, "The recent large reduction in space launch cost," in *48th International Conference on Environmental Systems Proceedings*. Environmental Systems, 2018.

[4] J. R. Wertz and W. J. Larson, *Reducing space mission cost*. Microcosm Press Torrance, CA, 1996.

[5] S. Mukherjee and R. Jain, "Significant contribution of private investment in the hybrid nature of booming global space economy (2011-2021 q1)," in *72nd International Astronautical Congress (IAC) Proceedings*, 10 2021.

[6] G. Stock, J. Fraire, H. Hermanns, E. Cruz, A. Isaacs, and Z. Imbrosh, "On the automation, optimization, and in-orbit validation of intelligent satellite constellation operations," in *Small Satellite Conference*, 2021.

[7] M. K. Ben-Larbi, K. F. Pozo, T. Haylok, M. Choi, B. Grzesik, A. Haas, D. Krupke, H. Konstanski, V. Schaus, S. P. Fekete *et al.*, "Towards the automated operations of large distributed satellite systems. part 1: Review and paradigm shifts," *Advances in Space Research*, vol. 67, no. 11, pp. 3598–3619, 2021.

[8] J. R. Kopacz, R. Herschitz, and J. Roney, "Small satellites an overview and assessment," *Acta Astronautica*, vol. 170, pp. 93–105, 2020.

[9] J. Cappaert, F. Foston, P. S. Heras, B. King, N. Pascucci, J. Reilly, C. Brown, J. Pitzo, and M. Tallhamm, "Constellation modelling, performance prediction and operations management for the spire constellation," in *Proceedings of the Small Satellite Conference, Mission Operations and Autonomy*, 2021.

[10] A. Lalbakhsh, A. Pitcairn, K. Mandal, M. Alibakhshikenari, K. P. Esselle, and S. Reisenfeld, "Darkening low-earth orbit satellite constellations: A review," *IEEE Access*, 2022.

[11] R. Deng, B. Di, H. Zhang, L. Kuang, and L. Song, "Ultra-dense leo satellite constellations: How many leo satellites do we need?" *IEEE Transactions on Wireless Communications*, vol. 20, no. 8, pp. 4843–4857, 2021.

[12] S. Guo, W. Zhou, J. Zhang, F. Sun, and D. Yu, "Integrated constellation design and deployment method for a regional augmented navigation satellite system using piggyback launches," *Astrodynamics*, vol. 5, no. 1, pp. 49–60, 2021.

[13] Y. Gao and C. Wei, "Planning management exploration in the development of large-scale satellite constellation systems," in *International Conference on Intelligent Automation and Soft Computing*. Springer, 2021, pp. 469–479.

[14] J. D. Ullman, "Np-complete scheduling problems," *Journal of Computer and System sciences*, vol. 10, no. 3, pp. 384–393, 1975.

[15] B. P. Gerkey and M. J. Matarić, "A formal analysis and taxonomy of task allocation in multi-robot systems," *The International Journal of Robotics Research*, vol. 23, no. 9, pp. 939–954, 2004.

[16] G. A. Korsah, A. Stentz, and M. B. Dias, "A comprehensive taxonomy for multi-robot task allocation," *The International Journal of Robotics Research*, vol. 32, no. 12, pp. 1495–1512, 2013. [Online]. Available: http://dx.doi.org/10.1177/0278364913496484

[17] L. B. Johnson, H.-L. Choi, and J. P. How, "The role of information assumptions in decentralized task allocation: A tutorial," *IEEE Control Systems Magazine*, vol. 36, no. 4, pp. 45–58, 2016.

[18] A. Sharma, D. Srinivasan, and D. S. Kumar, "A comparative analysis of centralized and decentralized multi-agent architecture for service restoration," in *2016 IEEE congress on evolutionary computation (CEC)*. IEEE, 2016, pp. 311–318.

[19] H.-L. Choi, L. Brunet, and J. P. How, "Consensus-based decentralized auctions for robust task allocation," *IEEE transactions on robotics*, vol. 25, no. 4, pp. 912–926, 2009.

[20] E. Chong and S. Zak, *An Introduction to Optimization*, ser. Wiley Series in Discrete Mathematics and Optimization. Wiley, 2013. [Online]. Available: https://books.google.com/books?id=iD5s0iKXHP8C

[21] "Mixed-integer programming (mip) - a primer on the basics," accessed: 2022-08-06. [Online]. Available: https://www.gurobi.com/resource/mip-basics/

[22] J. Frank, A. Jonsson, R. Morris, D. E. Smith, and P. Norvig, "Planning and scheduling for fleets of earth observing satellites," in *Proceeding of the 6th International Symposium on Artificial Intelligence and Robotics Automation in Space*, 2001.

[23] I. M. Ross and C. N. D'Souza, "Hybrid optimal control framework for mission planning," *Journal of Guidance, Control, and Dynamics*, vol. 28, no. 4, pp. 686–697, 2005. [Online]. Available: http://dx.doi.org/10.2514/1.8285

[24] J. Foust, "Spacex's space-internet woes: Despite technical glitches, the company plans to launch the first of nearly 12,000 satellites in 2019," *IEEE Spectrum*, vol. 56, no. 1, pp. 50–51, 2018. [Online]. Available: http://dx.doi.org/10.1109/MSPEC.2019.8594798

[25] M. Campbell, "Planning algorithm for large satellite clusters," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2002, p. 4958. [Online]. Available: http://dx.doi.org/10.2514/6.2002-4958

[26] S. Lee and I. Hwang, "Hybrid high-and low-thrust optimal path planning for satellite formation flying," in *AIAA Guidance, Navigation, and Control Conference*, 2012, p. 5046. [Online]. Available: http://dx.doi.org/10.2514/6.2012-5046

[27] J. B. Mueller and R. Larsson, "Collision avoidance maneuver planning with robust optimization," in *International ESA Conference on Guidance, Navigation and Control Systems, Tralee, County Kerry, Ireland*. Citeseer, 2008.

[28] D. Sun, F. Zhou, and J. Zhou, "Trajectory planning and control for satellite formation flying maneuver of leaving and joining," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2003, p. 5588. [Online]. Available: http://dx.doi.org/10.2514/6.2003-5588

[29] H. Shen and P. Tsiotras, "Optimal scheduling for servicing multiple satellites in a circular constellation," in *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, 2002, p. 4907. [Online]. Available: http://dx.doi.org/10.2514/6.2002-4907

[30] J. Zhang, G. T. Parks, Y.-z. Luo, and G.-j. Tang, "Multispacecraft refueling optimization considering the j 2 perturbation and window constraints," *Journal of Guidance, Control, and Dynamics*, vol. 37, no. 1, pp. 111–122, 2014. [Online]. Available: http://dx.doi.org/10.2514/1.61812

[31] Z. Zhao, J. Zhang, H.-y. Li, and J.-y. Zhou, "Leo cooperative multi-spacecraft refueling mission optimization considering j2 perturbation and target's surplus propellant constraint," *Advances in Space Research*, vol. 59, no. 1, pp. 252–262, 2017. [Online]. Available: http://dx.doi.org/10.1016/j.asr.2016.10.005

[32] A. W. Verstraete, D. Anderson, N. M. St. Louis, and J. Hudson, "Geosynchronous earth orbit robotic servicer mission design," *Journal of Spacecraft and Rockets*, vol. 55, no. 6, pp. 1444–1452, 2018. [Online]. Available: http://dx.doi.org/10.2514/1.A33945

[33] J. Zhang, Y. Zhang, and Q. Zhang, "On-orbit servicing mission planning for multi-spacecraft using cdpso," in *International Conference on Swarm Intelligence*. Springer, 2016, pp. 11–18. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-41009-8_2

[34] Y. Liu and J. Yang, "A multi-objective planning method for multi-debris active removal mission in leo," in *AIAA Guidance, Navigation, and Control Conference*, 2017, p. 1733.

[35] S. A. Cox, N. B. Stastny, G. N. Droge, and D. K. Geller, "Constellation planning methods for sequential spacecraft rendezvous using multi-agent scheduling," in *AAS/AIAA Astrodynamics Specialist Conference, 2019*. Univelt Inc., 2019, pp. 2959–2971.

[36] ——, "Resource-constrained constellation scheduling for rendezvous and servicing operations," *Journal of Guidance, Control, and Dynamics*, pp. 1–11, 2022. [Online]. Available: https://doi.org/10.2514/1.G006153

[37] P. K. Sinha and A. Dutta, "Multi-satellite task allocation algorithm for earth observation," in *2016 IEEE Region 10 Conference (TENCON)*. IEEE, 2016, pp. 403–408.

[38] F. Yao, J. Li, Y. Chen, X. Chu, and B. Zhao, "Task allocation strategies for cooperative task planning of multi-autonomous satellite constellation," *Advances in Space Research*, vol. 63, no. 2, pp. 1073–1084, 2019.

[39] P. Tangpattanakul, N. Jozefowiez, and P. Lopez, "A multi-objective local search heuristic for scheduling earth observations taken by an agile satellite," *European Journal of Operational Research*, vol. 245, no. 2, pp. 542–554, 2015.

[40] S. Nag, A. S. Li, V. Ravindra, M. S. Net, K.-M. Cheung, R. Lammers, and B. Bledsoe, "Autonomous scheduling of agile spacecraft constellations with delay tolerant networking for reactive imaging," *arXiv preprint arXiv:2010.09940*, 2020.

[41] S. Mitrovic-Minic, D. Thomson, J. Berger, and J. Secker, "Collection planning and scheduling for multiple heterogeneous satellite missions: Survey, optimization problem, and mathematical programming formulation," in *Modeling and Optimization in Space Engineering*. Springer, 2019, pp. 271–305.

[42] J. Berger, N. Lo, and M. Barkaoui, "Quest – a new quadratic decision model for the multi-satellite scheduling problem," *Computers Operations Research*, vol. 115, p. 104822, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0305054819302643

[43] L. He, M. de Weerdt, and N. Yorke-Smith, "Time/sequence-dependent scheduling: the design and evaluation of a general purpose tabu-based adaptive large neighbourhood search algorithm," *Journal of Intelligent Manufacturing*, vol. 31, no. 4, pp. 1051–1078, 2020. [Online]. Available: https://link.springer.com/article/10.1007/s10845-019-01518-4

[44] D. Karapetyan, S. Mitrovic-Minic, K. T. Malladi, and A. P. Punnen, "The satellite downlink scheduling problem: A case study of radarsat-2," in *Case Studies in Operations Research*. Springer, 2015, pp. 497–516.

[45] B. Deng, C. Jiang, L. Kuang, S. Guo, J. Lu, and S. Zhao, "Two-phase task scheduling in data relay satellite systems," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 2, pp. 1782–1793, 2017.

[46] B. Song, F. Yao, Y. Chen, Y. Chen, and Y. Chen, "A hybrid genetic algorithm for satellite image downlink scheduling problem," *Discrete Dynamics in Nature and Society*, vol. 2018, 2018.

[47] S. Spangelo, J. Cutler, K. Gilson, and A. Cohn, "Optimization-based scheduling for the single-satellite, multi-ground station communication problem," *Computers & Operations Research*, vol. 57, pp. 1–16, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0305054814002809

[48] X. Gu, J. Bai, C. Zhang, and H. Gao, "Study on tt&c resources scheduling technique based on inter-satellite link," *Acta Astronautica*, vol. 104, no. 1, pp. 26–32, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0094576514002501

[49] J. Li, H. Chen, and N. Jing, "A data transmission scheduling algorithm for rapid-response earth-observing operations," *Chinese journal of Aeronautics*, vol. 27, no. 2, pp. 349–364, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1000936114000235

[50] W.-h. Zhao, J. Zhao, S.-h. Zhao, Y.-j. Li, X. Li *et al.*, "Resources scheduling for data relay satellite with microwave and optical hybrid links based on improved niche genetic algorithm," *Optik*, vol. 125, no. 13, pp. 3370–3375, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0030402614001387

[51] J. Castaing, "Scheduling downloads for multi-satellite, multi-ground station missions," in *Proceedings of the Small Satellite Conference*, 2014. [Online]. Available: https://digitalcommons.usu.edu/smallsat/2014/FJRStudentComp/4/

[52] S. Augenstein, A. Estanislao, E. Guere, and S. Blaes, "Optimal scheduling of a constellation of earth-imaging satellites, for maximal data throughput and efficient human management," in *Proceedings of the International Conference on Automated Planning and Scheduling*, vol. 26, no. 1, 2016.

[53] X. Hu, W. Zhu, B. An, P. Jin, and W. Xia, "A branch and price algorithm for eos constellation imaging and downloading integrated scheduling problem," *Computers and Operations Research*, vol. 104, pp. 74–89, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0305054818303113

[54] S. Peng, H. Chen, J. Li, and N. Jing, "Approximate path searching method for single-satellite observation and transmission task planning problem," *Mathematical Problems in Engineering*, vol. 2017, 2017.

[55] J. C. McDowell, "The low earth orbit satellite population and impacts of the spacex starlink constellation," *The Astrophysical Journal Letters*, vol. 892, no. 2, p. L36, 2020.

[56] J. Hanson, A. G. Luna, R. DeRosee, K. Oyadomari, J. Wolfe, W. Attai, and C. Prical, "Nodes: A flight demonstration of networked spacecraft command and control," in *Proceedings of the Small Satellite Conference*, 2016. [Online]. Available: https://digitalcommons.usu.edu/smallsat/2016/S4LEOMis/4/

[57] D. Zhou, M. Sheng, X. Wang, C. Xu, R. Liu, and J. Li, "Mission aware contact plan design in resource-limited small satellite networks," *IEEE Transactions on Communications*, vol. 65, no. 6, pp. 2451–2466, 2017.

[58] O. Kondrateva, H. Döbler, H. Sparka, A. Freimann, B. Scheuermann, and K. Schilling, "Throughput-optimal joint routing and scheduling for low-earth-orbit satellite networks," in *2018 14th Annual Conference on Wireless On-Demand Network Systems and Services (WONS)*. IEEE, 2018, pp. 59–66.

[59] Y. Wang, M. Sheng, W. Zhuang, S. Zhang, N. Zhang, R. Liu, and J. Li, "Multi-resource coordinate scheduling for earth observation in space information networks," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 2, pp. 268–279, 2018.

[60] K. Cahoy and A. K. Kennedy, "Initial results from access: an autonomous cubesat constellation scheduling system for earth observation," in *Proceedings of the Small Satellite Conference*, 2017. [Online]. Available: https://digitalcommons.usu.edu/smallsat/2017/all2017/98/

[61] A. K. Kennedy and K. L. Cahoy, "Performance analysis of algorithms for coordination of earth observation by cubesat constellations," *Journal of Aerospace Information Systems*, vol. 14, no. 8, pp. 451–471, 2017.

[62] A. K. Kennedy, "Resource optimization algorithms for an automated coordinated cubesat constellation," Master's thesis, Massachusetts Institute of Technology, 2015.

[63] ——, "Planning and scheduling for earth-observing small satellite constellations," Ph.D. dissertation, Massachusetts Institute of Technology, 2018.

[64] Y. Chen, X. Mao, S. Yang, and Q. Wang, "Cost-efficient inter-robot delivery for resource-constrained and interdependent multi-robot schedules," *International Journal of Advanced Robotic Systems*, vol. 16, no. 1, p. 1729881419828049, 2019.

[65] D.-S. Chen, R. G. Batson, and Y. Dang, *Applied integer programming: modeling and solution*. Wiley, 2010.

[66] Q. Louveaux, "Lift-and-project inequalities," 2011.

[67] M. Lavrov, "Lecture notes in linear programming," October 2019.

APPENDICES

APPENDIX A

Review of Constellation Planning by Augenstein et. al

Due to the importance of the paper by Augenstein, it now receives a more in-depth review than the literature previously mentioned. Augenstein et al. formulated the planning problem and provided a solution to the coupled Earth-sensing and data downlink portions [52]. However, some limitations exist within their formulation and solving approach. Before evaluating these critically, a comprehensive explanation of their approach is now provided. The principal goal of the Augenstein paper is to balance data downlink with image data collection to prevent data from remaining onboard the satellites for extended periods of time. The basic concept underlying the planning technique relies on a graphical formulation with graph nodes representing imaging or downlink task fulfillment opportunities and edges between nodes representing feasible attitude slew transitions between those tasks [52]. The nodes are separated at discrete times and thus inherently introduce some sub-optimality but make the problem tractable. A key requirement for the formulation and solution technique is to generate a flight-worthy solution within a timeframe that supports the operational needs of the constellation. This timeline required a decoupling of the image collection and downlink portions of the problem. Decoupling these two elements improved run time but sacrificed some additional optimality. Two basic heuristics were used to evaluate imaging opportunities relative to downlink opportunities and therefore inform the selected operation on each space vehicle. These heuristics are applied in a 2-layer fashion with the first layer addressing downlink scheduling in a manner that attempt to most effectively work around the anticipated imaging tasks. This is done by encouraging downlinking to occur in such a way that prioritized Earth-sensing opportunities are realized while also decreasing the amount of image data that remains onboard the space vehicle.

The mathematical representation of this optimization problem is summarized below as:

$$\begin{aligned} \min_{x_i} & \sum_{i \in S_{dl}} \left(\tilde{c}_i x_i + \alpha d_i y_i \right) \\ \text{s.t. } & x_i \in \{0, 1\} \\ & y_i \in \\ & y_i \geq 0 \\ & \vdots \\ & y_i \geq y_{i-1} - \tilde{a}_i x_i + \tilde{b}_i \end{aligned} \quad , \tag{A.1}$$

where, within the objective function, $\tilde{c}_i$ represents a heuristic for the opportunity cost of a downlink task at time step i, x_i is the decision variable for task node selection, α provides a weighting factor on the value of the onboard data, d_i is the penalty associated with the data onboard at time step i, and finally y_i is the actual amount of data onboard at the same time step. The term S_{dl}, specifying the bound on the summation, represents only the set of downlink opportunities. The vertical dots represent additional constraints respecting slew agility, visiting the ground station periodically, and performing consecutive downlinks when possible. These constraints are specific to the mission being addressed by Augenstein et al. and do not directly impact this discussion. However, further detail about the variables used by Augenstein et al. is provided in Table A.1. A high level explanation of the terms in the objective function is as follows:

- $\sum_{i \in S_{dl}}$: The optimization is done purely for the downlink nodes (S_{dl})
- $\tilde{c}_i x_i$: The opportunity cost associated with selecting the i^{th} downlink node
- $\alpha d_i y_i$: Penalty on the data at node i

The final constraint describes the data as a function of the previous node and the choice of whether or not to collect data at the current node. The $\tilde{a}_i$ is somewhat more complicated

Table A.1: Notation used for Planning

Variables used by Augenstein et. al

Name	Description
x_i	Binary variable on whether or not to select node i
y_i	Amount of data on board (not a notion of agent)
d_i	Penalty on data at node i
α	Weighting on data
S_{dl}	Set of nodes corresponding to a downlink
$\tilde{c}_i$	Heuristic for the opportunity cost associated with downlink node i
$\tilde{a}_i$	Heuristic sum of possible data downlinked and the possible data obtained
$\tilde{b}_i$	Heuristic for the amount of data that could be obtained if node i not chosen

than at face value, and what follows are details not present in [52]. To describe the full meaning of $\tilde{a}_i$, the data at time i can be written in two cases:

$$y_i = \begin{cases} y_{i-1} - \tilde{\alpha}_i & x_i = 1 \\ y_{i-1} + \tilde{b}_i & x_i = 0 \end{cases}, \tilde{\alpha}_i \geq 0, \tilde{b}_i \geq 0.$$

In words, y_i decreases by $\tilde{\alpha}_i$ if node i is chosen and increases by $\tilde{b}_i$ if node i is not chosen. Since x_i is binary, the data could be written as

$$y_i = y_{i-1} + (1 - x_i)\tilde{b}_i - \tilde{\alpha}_i x_i,$$

which, factoring out x_i, can be expressed as

$$y_i = y_{i-1} - (\tilde{b}_i + \tilde{\alpha}_i)x_i + \tilde{b}_i.$$

Thus, $\tilde{a}_i = \tilde{b}_i + \tilde{\alpha}_i$. So, $\tilde{a}_i$ is actually the sum of the data that could be obtained if node i was not chosen and the data that would be downlinked if it were chosen. $\tilde{b}_i$ is the data that could be obtained if node i were not chosen. There are a few points that should be made on the formulation, which are summarized below as follows:

1. The opportunities are sorted in terms of time (i.e., x_1 occurs at a time less than or equal to x_2).

2. The formulation appears to be formulated for a single agent as y_i for one agent should

not be related to the value of another agent (i.e., you cannot simply order the y_i in time and relate y_i to y_{i-1} for another agent).

3. An inequality is used to express y_i instead of an equality. Thus, the amount of data onboard after a downlink opportunity is penalized, which forces the equality satisfaction at the optimal value (amount that can be downlinked within the window) if the data is positive. This also allows the visiting of a downlink node to utilize only a portion of the downlink capability. If there is less data onboard than could be downlinked, then the $y_i \geq 0$ constraint combined with the inequality constraint will keep the data at or above a zero threshold.

The values for $\tilde{c}_i$, $\tilde{a}_i$, and $\tilde{b}_i$ are not precisely known but are instead estimated using a heuristic. These heuristics are determined using dynamic programming which provides values for the opportunity cost, $\tilde{c}_i$, and the amount of data that could be acquired at each node, $\tilde{b}_i$ and were demonstrated to result in a tremendous improvement for an operationally-valid constellation schedule. The research in this document expands on the contributions from Augenstein et al. and introduces new methods to address some of the limitations.

APPENDIX B

Integer Program Solution Techniques

The Mixed-Integer Linear Program (MILP) formulation provides an effective method for modeling the optimization problems of interest to this research. The integrality of all decision variables within the optimization space allow direct evaluation of the overall objective function by selecting specific combinations of these variables. The approach of evaluating all possible permutations of the decision variables, meeting the specified constraints, is referred to as full or complete enumeration. While this approach can theoretically identify the optimal solution to the specified problem, actually doing so in practice is likely prohibitively expensive in terms of computation time. This is due to the fact that a problem with n integer variables with m values has a total of m^n possible results to evaluate [65]. Such a daunting task, thus demands a more efficient method for arriving at an optimal solution.

This appendix provides a brief explanation of some traditional methods and approaches for solving MILPs. These include the following:

1. Branch and Bound
2. Cutting Plane
3. Branch and Cut

Each of these methods tackles the problem in a slightly different way but each of them attempts to divide, reduce, or in some other way, simplify the problem being solved with the intent of arriving at the optimal solution in an efficient manner.

B.1 Branch and Bound

The branch and bound technique is a powerful method that sequentially breaks the original problem into smaller subproblems which are methodically solved, evaluated relative to each other, and potentially selected or eliminated based on that evaluation. This

approach provides a systematic method for progressively reducing the complexity of the optimization space to arrive at a final optimal solution. The problem is traditionally represented graphically as an inverted tree with a root node that branches into other nodes that refine the problem into smaller and smaller subproblems. Each node in this structure represents a subproblem with each subsequent branch representing a constraint. The process begins at the root node by removing the integrality constraints of the original problem and solving this relaxation to yield the optimal solution. While unlikely, if the initial solution to the relaxed problem meets all constraints of the original problem (including the integer constraints), the solution is the optimum for the problem. However, if the solution to the relaxed problem yields a fractional result in one ore more of the variables normally required to be integral, then the relaxed problem is branched into two nodes representing subproblems. This portion of the process is referred to as separation since it functionally separates the previous solution into two distinct areas for further consideration. This branching is performed in such a way as to reduce the complexity of the problem. For example, if the solution to the relaxation yields a value of 3.8 for a variable x, then branching on this variable would yield the following two constraints:

$$\begin{aligned} x &\leq 3 \\ x &\geq 4 \end{aligned} \tag{B.1}$$

This branching then provides two new subproblems (original relaxation problem with additional constraints) for evaluation and comparison to other existing branches and their results. Throughout this branch and bound process, the objective function is evaluated at each node and those meeting the integrality constraints become candidate solutions and set the lower bound. The incumbent solution is determined by the current best integer solution and is maintained throughout the process to provide one method of pruning or fathoming a given node. The upper bound of the search is determined by the best solution to the relaxed problem and can be used to provide an upper bound on the optimum while also informing

the proximity of the incumbent solution to this upper bound. The difference between the upper and lower bounds is referred to as the gap. Pruning, or fathoming, of a node can occur for any one of the following reasons:

1. Infeasibility - the branch lies outside of the space specified by the relaxed constraints.
2. Bound - the solution found is not better than the current incumbent solution. Note that for maximization problems, the upper bound is determined by the solution to the relaxed problem while the lower bound is determined by the best integer solution.
3. Candidacy - the solution meets all constraints including the integrality constraints so no further branching occurs.

Branch and Bound Example

To help the branch and bound explanation, the approach described above is now applied to the following problem, as an example:

$$\begin{aligned} \max \ & z = 2x + 3y \\ \text{s.t.} \ & x + 2y \leq 3 \\ & 4x + 5y \leq 10 \\ & x, y \in \{0,\ 1\} \end{aligned} \quad . \tag{B.2}$$

For visualization purposes, this problem is illustrated on the left in Figure B.1 with the constraints applied, and the objective function shown. The yellow area represents the feasible region while the black dots represent the integrality constraint for the problem. Relaxing the problem by removing the integrality constraint allows for a direct solve of the optimal point, shown on the right side of Figure B.1. The resulting solution of $(x, y) = \left(\frac{5}{3}, \frac{2}{3}\right)$ generates an objective value of $z = \frac{16}{3} \approx 5.33$ and then allows for subsequent branching. In this example, the variable $x = \frac{5}{3} \approx 1.67$ is first branched by applying additional constraints of $x \leq 1$ and $x \geq 2$, as illustrated in Figure B.2.

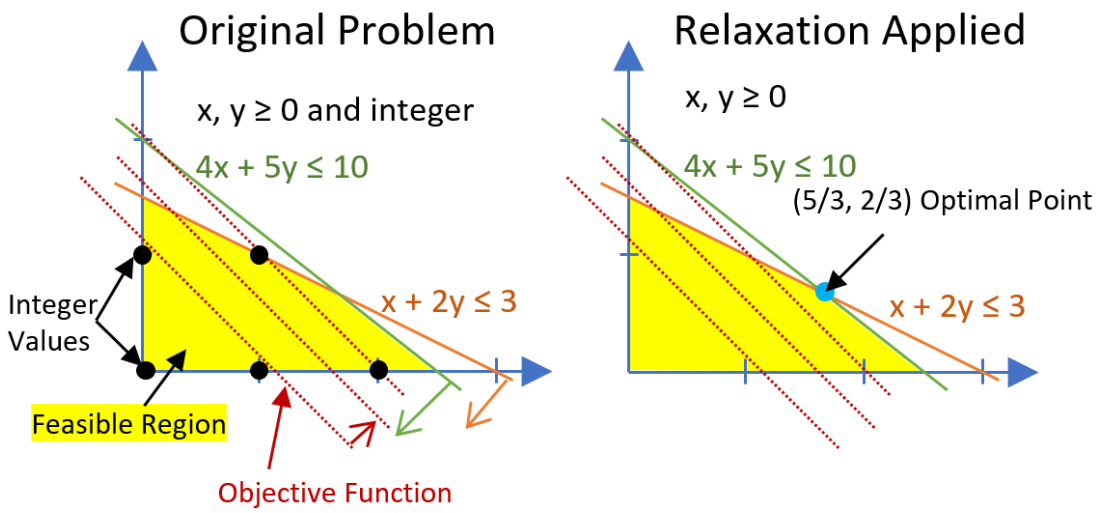


Fig. B.1: Plotted example branch and bound problem showing constraints, feasible region, and objective function upward trend. Left shows the original MILP while the right shows the relaxation and associated optimal point.

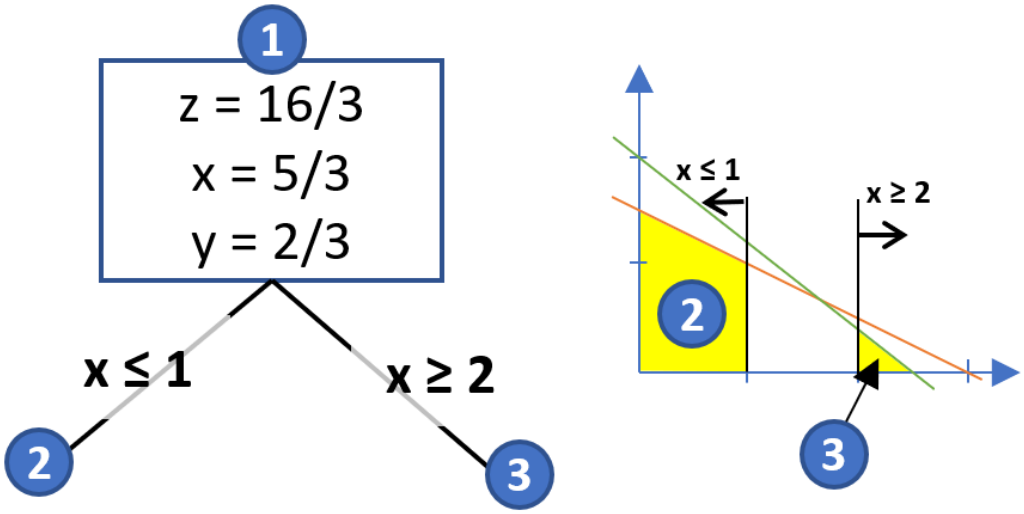


Fig. B.2: First branching applied on variable x.

The process continues by finding the solution after adding in the additional constraints specified by the branches. At node 2, this results in an objective value of $z = 5$ with $x = 1$ and $y = 1$. This solution meets all constraints including the integrality constraints, making it a candidate solution. Upon further inspection, this result is the optimal solution to the original integer problem. However for illustrative purposes, further branching continues for this example and is summarily illustrated in Figure B.3. At node 3, branching is applied to y since $x = 2$ (integer value) in that solution. That branching results in constraints being added to the problem with $y \geq 1$ (node 4) and $y \leq 0$ (node 5). The branch to node 4 results in an infeasible solution and is thus fathomed, or pruned. At node 5, y has two constraints present which force $y = 0$ which intersects the $x \geq 2$ constraint at point $(x, y) = (2, 0)$. This point results in an objective value of 4 and is therefore fathomed due to bound.

This example has directly illustrated the key concepts of branch and bound. To allow for ease of plotting, only two variables were present in the example. In many real-world applications there may be a large number of variables which precludes identifying a solution via plotting. However, the same approach illustrated can be applied to those problems with potential determination of an optimal solution that meets all specified constraints.

Branch and Bound Summary

In summary, the branch and bound technique is capable of efficiently breaking the original integer program into more refined and tractable subproblems that are easier to solve and evaluate based on a set of defined rules. These subproblems, when combined, result in the original problem but allow for more effective management of problem complexity. This technique is foundational to many of the approaches for solving integer linear programs.

B.2 Cutting Plane

Similar to the brand and bound approach from the previous subsection, the cutting plane method is used to reduce the feasible region of the relaxed ILP with the goal of improving the time required to identify the optimal solution to the original problem. A cutting plane is a constraint that is added to the original problem that separates a fractional

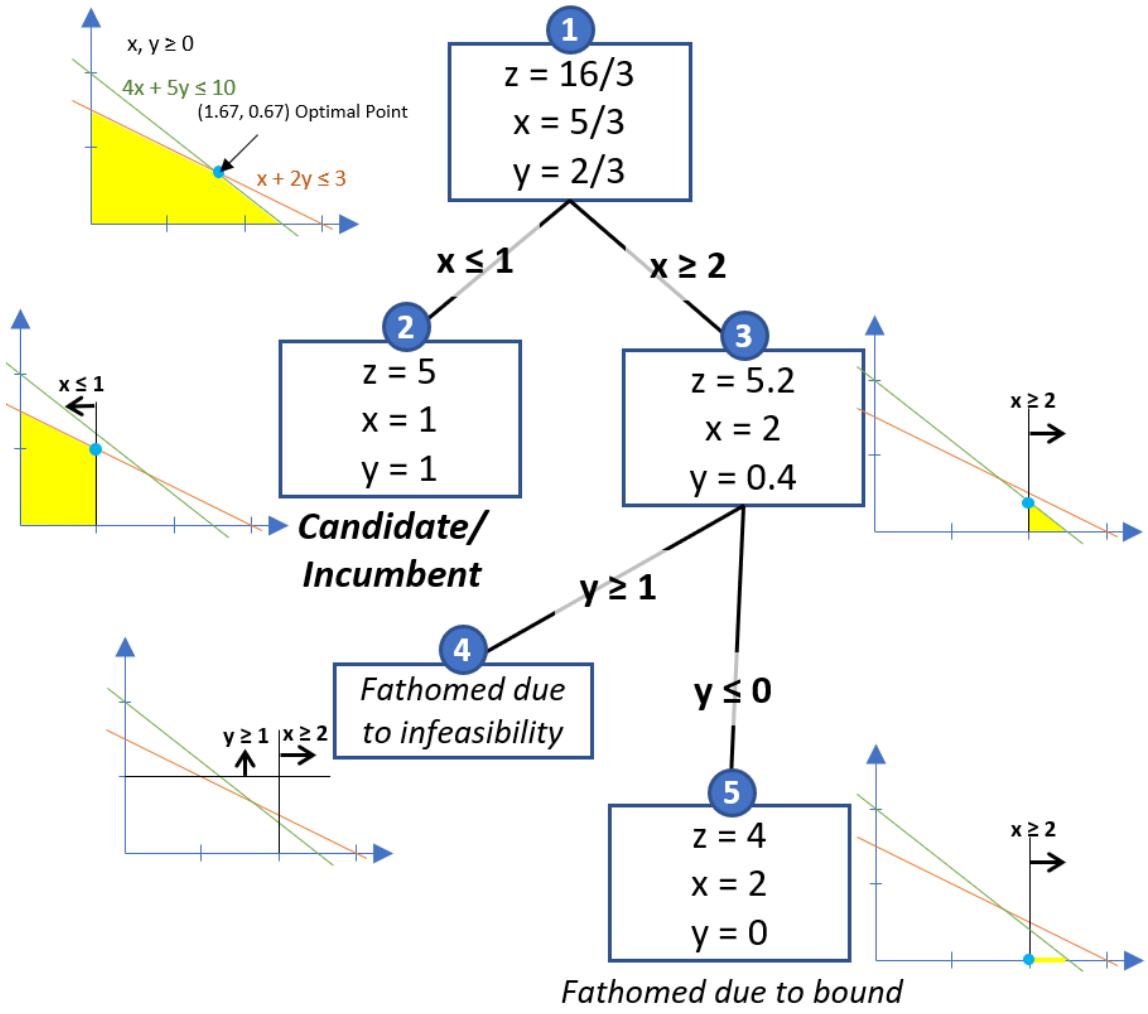


Fig. B.3: Visual summary of branch and bound approach for the example shown in Equation B.2.

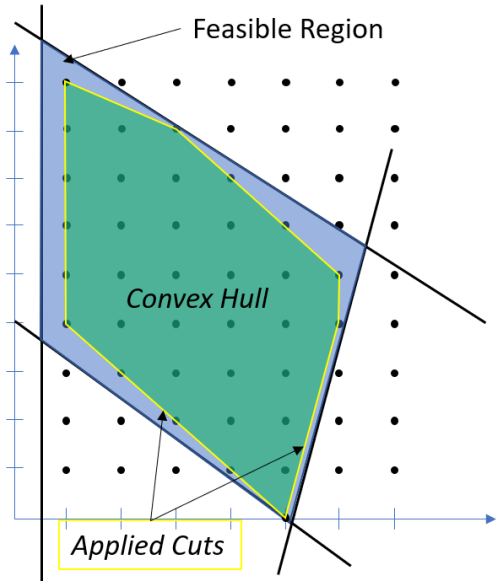


Fig. B.4: Illustration of the original feasible region (blue) defined by linear constraints, the applied cuts (yellow lines), and the resulting complex hull (green).

solution (resulting from a relaxation) from the integer candidates. This means that this cut will violate the relaxed LP but not any integer feasible solution. Conceptually, the cutting plane method sequentially removes sections of the feasible region from the ILP relaxation by applying cuts with the intent of making the optimal integer solution become an extreme point within the space. Doing so creates a scenario where the simplex algorithm can efficiently drive toward the optimal solution by evaluating the vertices created by applying the cuts. In the application of these cuts, no integer solutions are removed but the feasible region is reduced. It is possible to reduce the overall feasible region using cuts until only the feasible hull of the solution space remains. This concept is illustrated in Figure B.4, with the original feasible region in blue, the applied cuts in yellow, and the resulting convex hull highlighted in green. Reducing the feasible space to only the convex hull allows the simplex algorithm to explore the space more efficiently on its way to identifying the optimal integer solution.

The primary difference between the branch and bound and cutting plane methods is in how the constraints, or cuts, are generated. The foundational methods are the mixed and fractional cutting plane techniques developed by Gomory. These methods both belong to

the dual cutting plane class and provide robust approaches for reducing the feasible region and finding the optimal solution. The fractional cutting plane method requires that the starting integer program contain all integer coefficients. This ensures that all slack variables and added constraints are guaranteed to be integer and non-negative. The basic approach is outlined below:

1. Relax the linear program and find the optimal solution using the simplex method.
2. Select a source row from the optimal tableau of step 1 and derive a cut constraint.
3. Apply the cut constraint and resolve the optimization of the augmented linear program.
4. Check the integrality of the resulting solution. If the integrality constraints are met, the problem is solved. Otherwise return to step 2.

The approach mentioned covers the basics but there are several approaches that exist for generating cuts and some fundamental requirements about when to use them. Additional information regarding these methods and cut types is provided in the following subsection.

Cutting Techniques

Several cut generation techniques exist but all of them convert constraints into cuts that help reduce problem complexity. The main techniques used in generating cuts are the following:

1. Rounding Technique - often used in model pre-processing to generate stronger cuts.
2. Disjunction Technique - most common technique for generating cuts for problems with both integer and continuous variables. These cuts remove only fractional solutions with valid cuts and do not remove integer solutions.
3. Lifting Technique - most often performed on binary $(0, 1)$ problems to strengthen valid inequalities.

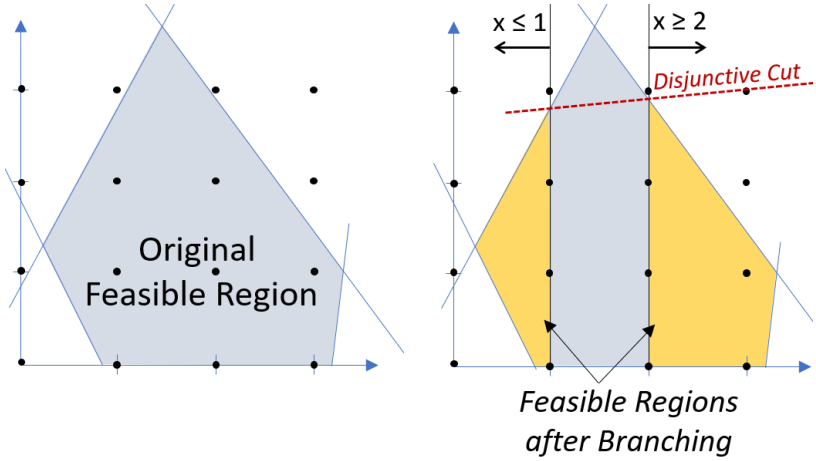


Fig. B.5: Illustration of a disjunctive cut that is feasible for the post-branching feasible regions but not the original relaxation of the integer program.

These cut generation methods will now be discussed in additional detail. For rounding techniques, the intent is to strengthen an existing constraint and is thus often used during pre-processing [65]. As a simple example, consider the case of a problem with a constraint specifying a fractional upper bound for a particular integer variable (e.g. $x \leq 5.7$). Since this variable is constrained to be an integer, it is possible to re-write the constraint to be $x \leq 5$. Similarly, for the case when a variable is constrained to be both an integer and a lower limit, the proper rounding can be performed such as $x \geq 7.3 \Rightarrow x \geq 8$. This cut generation method can also be applied to the case where a common divisor exists for a constraint of all integer coefficients. For example, the constraint $2x + 6y \leq 10$ can be divided by 2 to yield a stronger cut as $x + 3y \leq 5$.

The disjunctive cut generation technique is a common technique often used on problems with both integer and continuous variables. For a cut to be a disjunctive cut it must be valid for the feasible regions resulting after a branching operation of an integer variable, but not necessarily for the relaxation of the original integer program [65]. An example of a disjunctive cut is illustrated in Figure B.5. This cut technique is key to the branch and cut solution approach.

The intent of Lifting techniques is to strengthen valid inequalities within an integer program. This is accomplished by first lifting these constraints into an extended space

using multiplication and then projecting them back to the original space for application. The multiplication is performed on a variable and its complement. These techniques are most often used for binary integer programs and additional details can be found in [66]. Cut types are discussed in the next subsection.

Cut Types and Application

A variety of cuts exist and are used under specific circumstances. A brief list and summary of each of these main types is provided below:

- **Pure Integer Variable Problems**
  - ***Gomory Fractional Cut***: This cut is generated using the simplex tableau to find an optimal solution. If the solution results in fractional values for the basic variables, a cut can be generated by arbitrarily selecting a row from the tableau, writing it in equation form, separating the integer and non-negative parts, and finally extracting the non-basic variable equation for re-application. An example of this common cut is shown in Section B.2.
  - ***Chvatal-Gomory Cut***: This cut is very similar to the Gomory fractional cut and is generated by multiplying a scalar to a set of constraints, combining them, and then applying a floor function for maximization problems or a ceiling function for minimization problems. This combination of existing constraints can strengthen them.
  - ***Pure Integer Rounding***: This cut divides the constraint coefficients and RHS by a common integer value and then applies a floor or ceiling rounding for maximization and minimization problems, respectively. This method often results in relatively weak cuts but is often used in pre-processing.
  - ***Objective Integrality Cut***: This cut can be used when the coefficients on the LHS of the constraint equation are all integers. When this occurs the objective value may be set to an integer value via rounding (floor function for maximization

problems and ceiling function for minimization problems). This is allowed since the LHS coefficients are integers, and thus the RHS objective function must also be.

- **Mixed Integer Variable Problems**
  - ***Gomory Mixed Integer Cut***: This cut is used when some variables are integer while others are continuous. It is generated by solving the optimal tableau, sorting continuous variables into positive and negative coefficient groups, along with integer variables into those less than or equal to, and greater than the relative objective value. These terms receive specific coefficients based on these groups which are then combined for the mixed integer cut.
  - ***Mixed Integer Rounding (MIR)***: MIR cuts are generated by applying integer rounding on the coefficients of integer variables and the right-hand side of a constraint.
- **0-1 Binary Knapsack Set Problems**
  - ***Knapsack Cover***: A knapsack minimal cover is a subset of the variables of the inequality such that if all the subset variables were set to one, the knapsack constraint would be violated, but if any one subset variable were excluded, the constraint would be satisfied.
  - ***Lifted Knapsack Cover***: This is used to strengthen knapsack cover inequalities. Finding the appropriate lifting coefficients will improve the strength of the knapsack cover.
  - ***Generalized Upper Bound (GUB) Cover***: This constraint is formed by selecting variables from the specified constraints so that no variables are the same, adding them together, and forming a new constraint.
- **Binary Coefficients and Binary Variables Problems**

  - ***Binary Cuts***: This cut is often used for network and graph problems and relies on a maximum clique. A clique is a relationship among a group of binary variables such that at most one variable in the group can be positive in any integer feasible solution. A maximal clique can be converted into a constraint that often dominates the constraints set by the initial problem constraints and speeds the solution discovery.

- **Continuous Variables with Variable Upper Bounds Problems**

  - ***Flow Cover Cuts***: Applied to problems with variables having upper bounds from 0 and greater and that are associated with binary variables. Continuous variables are modeled as in-flows and out-flows to a node with the binary variables being whether flow is on or off. Similar to a knapsack constraint which can be used to generate a cover constraint.

Cut Example

To complete the explanation of the cutting plane approach, the Gomory fractional cutting method is now applied to the following problem (same as the branch and bound example in Section B.1):

$$\begin{aligned} \max \ & z = 2x + 3y \\ \text{s.t.} \ & x + 2y \leq 3 \\ & 4x + 5y \leq 10 \\ & x, y \in \{0,\ 1\} \end{aligned} \quad . \tag{B.3}$$

For visualization purposes, this problem is illustrated on the left in Figure B.6 with the constraints applied, and the objective function shown. The yellow area represents the feasible region while the black dots represent the integrality constraint for the problem [67].

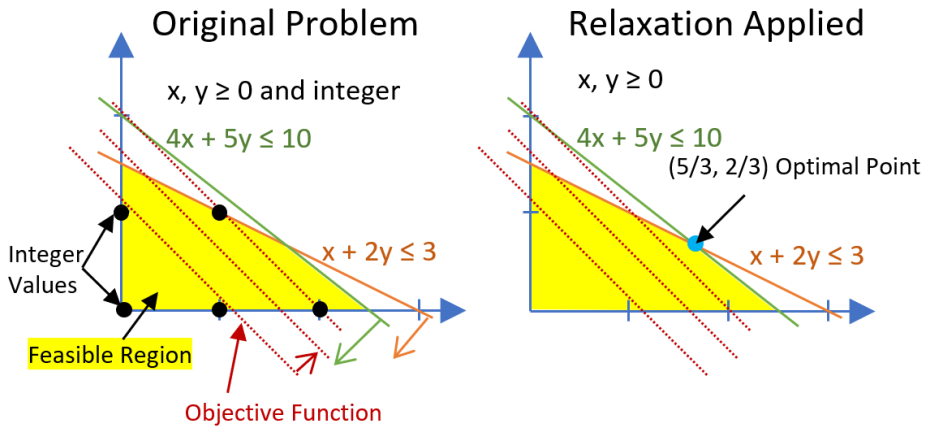


Fig. B.6: Plotted example problem showing constraints, feasible region, and objective function upward trend. Left shows the original ILP while the right shows the relaxation and associated initial optimal point.

Table B.1: Optimal tableau 1 for the relaxed ILP of Equation B.3.

	x	y	s_1	s_2	RHS
y	0	1	4/3	-1/3	2/3
x	1	0	-5/3	2/3	5/3
$-z$	0	0	-2/3	-1/3	-16/3

The solution search process begins by applying the simplex method to the relaxed problem and finding the optimal point. Performing the simplex method on the problem results in the tableau captured in Table B.1, with variables s_1 and s_2 being slack variables. Since both rows of the optimal tableau in B.1 have a fraction in the right hand side (RHS), an integer solution was not found. Thus, a cut is applied by selecting a row and writing it in equation form as:

$$y + \frac{4}{3}s_1 - \frac{1}{3}s_2 = \frac{2}{3} \tag{B.4}$$

In generating the cut, equation B.4 is separated into integer and non-negative parts. Since the slack variables both have fractional coefficients, an integer is generated for them which when combined with the resulting fractional portion, equal the original fractional coefficient. This results in the following equation:

$$y + s_1 - s_2 + \frac{1}{3}s_1 + \frac{2}{3}s_2 = \frac{2}{3} \tag{B.5}$$

Note that summing the coefficients of s_1 results in a value of $\frac{4}{3}$ while the summation of the s_2 coefficients results in a value of $\frac{-1}{3}$, which match the original coefficients in equation B.4. This is done to ensure integer values exist for every associated fractional value such that the cut separates them. Now we can separate the integer portion into its own equation as well as the non-negative portion. Doing so yields two equations as:

$$y + s_1 - s_2 + \leq 0 \tag{B.6}$$

$$\frac{1}{3}s_1 + \frac{2}{3}s_2 \geq \frac{2}{3} \tag{B.7}$$

Equation B.6 shows ≤ 0 because the lower integer (floor function) of $\frac{2}{3}$ is 0. Similarly, this requires that the fractional portion of the equation be $\geq \frac{2}{3}$. Now this cut can be applied to the problem but before doing so, it is possible to visualize the cut by expressing s_1 and s_2 in terms of x and y. This is possible by replacing the slack variables, s_1 and s_2, with their equivalents from the starting tableau. Performing that substitution yields the following:

$$y + (3 - x - 2y) - (10 - 4x - 5y) \leq 0 \Rightarrow 3x + 4y \leq 7 \tag{B.8}$$

Applying this new constraint results in a reduced feasible area without removing any of the integer solutions within that space. The impact of this new cut is illustrated in Figure B.7. The next step in the process is to add the cut to the simplex tableau and re-solve. The new augmented tableau is shown in Table B.2 with the final tableau shown in Table B.3. Since all values in the RHS column are integer, this represents the final solution of $(x, y) = (1, 1)$.

Cut Summary

In summary, the goal of the cutting plane method is to apply constraints that separate the fractional solution (from the MILP relaxation space) from all integer constraints, reduce

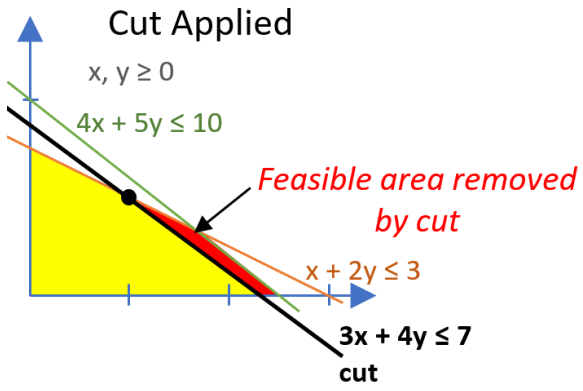


Fig. B.7: Resulting impact to the feasible region after applying the first cut.

Table B.2: Starting tableau resulting from the application of the first cut to the relaxed ILP of Equation B.3.

	x	y	s_1	s_2	s_3	RHS
y	0	1	4/3	-1/3	0	2/3
x	1	0	-5/3	2/3	0	5/3
s_3	0	0	-1/3	-2/3	1	-2/3
$-z$	0	0	-2/3	-1/3	0	-16/3

Table B.3: Final tableau resulting after optimizing based on the applied cut for the relaxed ILP of Equation B.3. Note that the sign of the slack variable coefficients indicate this is the optimal solution.

	x	y	s_1	s_2	s_3	RHS
y	0	1	3/2	0	-1/2	1
x	1	0	-2	0	1	1
s_2	0	0	1/2	1	-3/2	1
$-z$	0	0	-1/2	0	-1/2	-5

the feasible region, and generate additional vertices that can be tested for optimality using the simplex method. This process can be augmented with the branch and bound technique to generate a very efficient method for solving linear programs. This combination is briefly explained in the following subsection.

B.3 Branch and Cut

The branch and cut method blends the advantages of both the branch and bound and the cutting plane methods. Branch and bound allows a problem to be broken into distinct subproblems for solving and evaluation while the cutting plane method essentially tightens the relaxations that are applied to the problem to ease solving. The branch and bound method by itself is reliable but can be slow in arriving at a solution. The cutting plane method by itself has limited application except for very simple problems but it is fast in generating cuts. Combining the two approaches leverages the best aspects of each method and results in a powerful solution technique for solving integer linear programs. The basic approach is to relax the original integer program, solve it, determine valid cuts that reduce the feasible region, then apply those cuts. Once the generation and application of new cuts becomes less effective, execute a branching operation and repeat the process. This general procedure is followed until the optimal solution is identified. Additional details on the branch and bound and cutting methods are provided in Sections B.1 and B.2, respectively.

CURRICULUM VITAE

Skylar A. Cox

Published Journal Articles

- S. A. Cox, N. B. Stastny, G. N. Droge, and D. K. Geller, Resource-Constrained Constellation Scheduling for Rendezvous and Servicing Operations. *Journal of Guidance, Control, and Dynamics*, 1-11, 2022.

- S. A. Cox, G. N. Droge, J.H. Humble, K.D. Andrews, A network flow approach for constellation planning. *Space Mission Planning & Operations*, 2022;1:1.

Published Conference Papers

- S. Cox, J. Whitaker, J. Humble, and G. Droge, A Two-Layer Constellation Scheduling Approach for Imaging and Communication. *AAS/AIAA Astrodynamics Specialist Conference*, 2022.

- S. Cox, T. Smith, T. Jones, G. Droge, and A. Jones. Power-Optimal Slew Maneuvers in Support of Small Satellite Earth Imaging Missions. *AIAA/USU Small Satellite Conference*, 2020.

- S. A. Cox, N. B. Stastny, G. N. Droge, and D. K. Geller. Constellation planning methods for sequential spacecraft rendezvous using multi-agent scheduling. *AAS/AIAA Astrodynamics Specialist Conference*, 2019