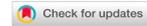




Review

Cox et al. *Space Missn Plann Oper* 2022;1:5-27
DOI: 10.20517/smpo.2022.01

Space Mission Planning & Operations
Open Access



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;1:5-27. <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:** 1 Sep 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,



© The Author(s) 2022. **Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License (<https://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, sharing, adaptation, distribution and reproduction in any medium or format, for any purpose, even commercially, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.



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 allocate 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
$e_{l_{sat}}$	Elevation angle of satellite wrt target	$a_{z_{sat}}$	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	$e_{l_{sun}}$	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 , $x = [x_1 \dots x_{n_v}]$
y_{lk}	Memory for satellite l at time k	y_l	Memory values for satellite l , $y_l = [y_{l1}, \dots, y_{lm}]^T$
y	Vector of all y_i , $y = [y_1^T, \dots, y_{n_s}^T]^T$	z	All optimization parameters, $z = [x^T \ y^T]^T$
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 constraint 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

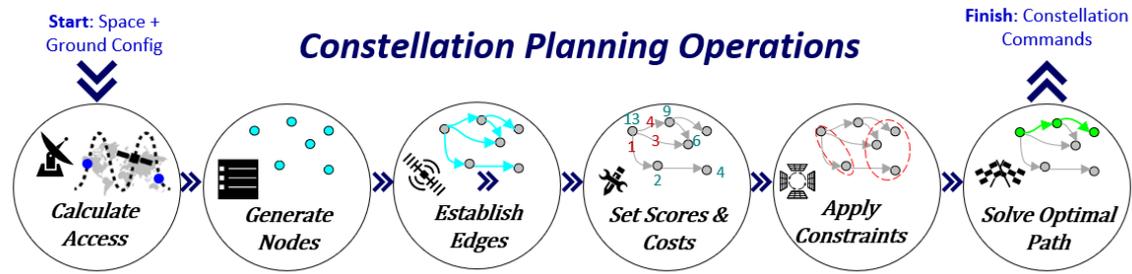


Figure 1. The constellation planning process from mission configuration through command generation.

Table 2. Definition of task constraints

Value	Description
<i>lat/lon/alt</i>	Latitude, longitude, and altitude of the target
<i>min/max azimuth</i>	The minimum and maximum azimuth angle from the target to the satellite
<i>min/max elevation</i>	The minimum and maximum elevation from the target to the satellite
<i>min/max sun elevation</i>	The minimum and maximum sun elevation angle from the target
<i>start/end time</i>	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

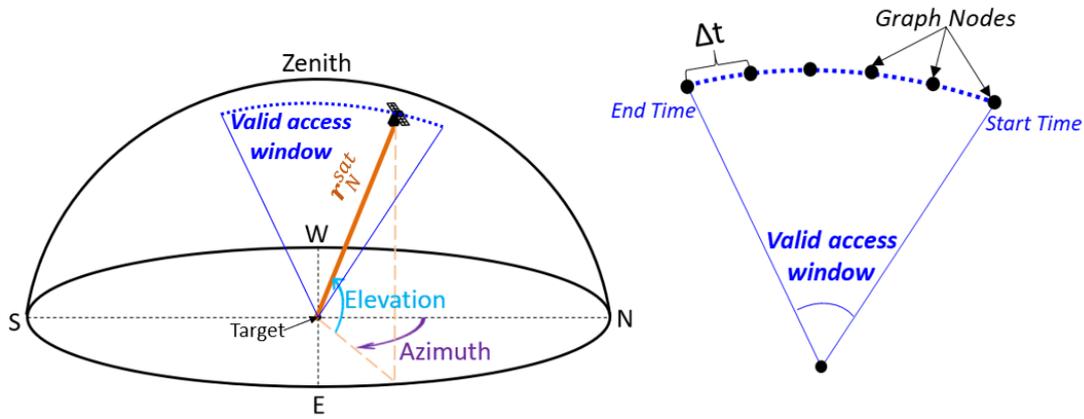


Figure 2. Depiction of a valid access window using elevation and azimuth constraints on the left and a discretized satellite trajectory through the window on the right.

$$r_N^{sat} = [r_{xn}^{sat} \quad r_{yn}^{sat} \quad r_{zn}^{sat}]^T . \tag{1}$$

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

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

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

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

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 $v_i, i = 1, \dots, n_v$ 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 by 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

$$s_i = \frac{d_i \cdot r \cdot s_{\sigma_i}}{p}. \quad (5)$$

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

$$u_i = s_i - \beta \cdot w_{hi}. \quad (6)$$

where the slew cost weighting factor is represented as β . The utilities are combined into a single vector as $u = [u_1 \ u_2 \ \dots \ u_{n_e}]^T$, 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:

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

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

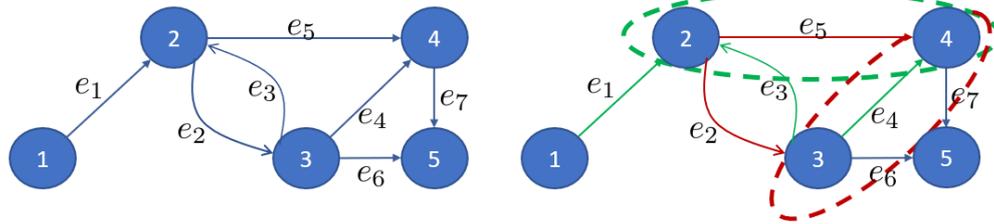


Figure 3. A simple graph is presented with numbered circles as nodes and labeled edges. The right image illustrates two group constraints, one around nodes v_2 and v_4 and one around nodes v_3 and v_4 .

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

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

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 simultaneously 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 $i = 1, \dots, n_e$ (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, $u_1x_1 + u_2x_2 + \dots + u_{n_e}x_{n_e} = u^T x$.

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

$$D_l x_l = b_{ss}, b_{ss} = [1 \ 0 \ \dots \ 0 \ -1]^T. \tag{9}$$

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:

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

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., $i = 1, \dots, n_g$). 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

$$\sum_{j \in \mathcal{I}_i} x_j \leq 1 \quad \forall i = 1, \dots, n_g. \quad (10)$$

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_{ij}^g to be the entry of A^g in the i^{th} row and j^{th} column. A^g can be expressed as

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

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

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

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

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

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.,

$$y_l = [y_{l1} \quad y_{l2} \quad \dots \quad y_{ln_t}]^T, y = [y_1^T, y_2^T, \dots, y_{n_s}^T]^T \in \mathbb{R}^{n_d}. \tag{12}$$

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 $A_{data}^l \in \mathbb{R}^{n_t \times (n_e+n_d)}$ where

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

The aggregate data update constraint can then be written as

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

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

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

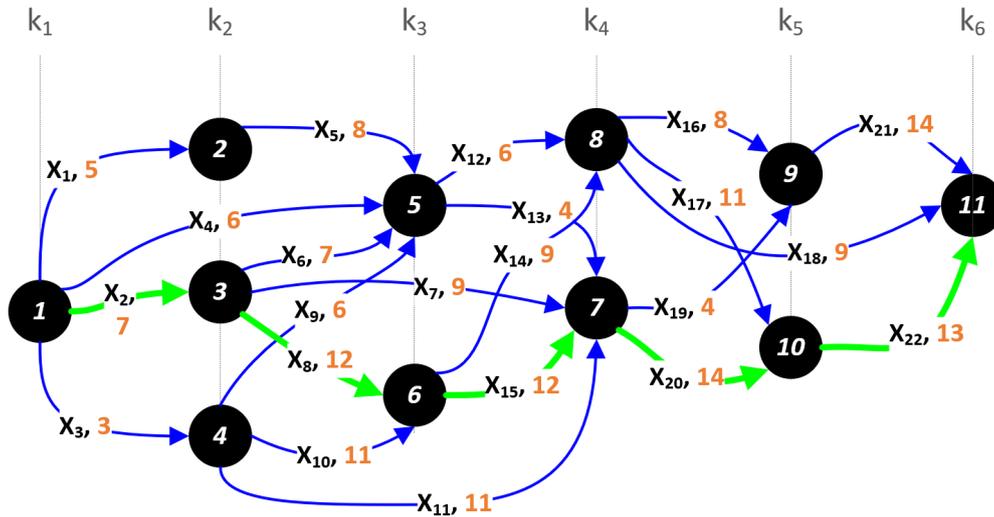


Figure 4. The optimal path through the graph when solving this problem using the network flow approach outlined. The path through the nodes is given as edges 2, 8, 15, 20, and 22.

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

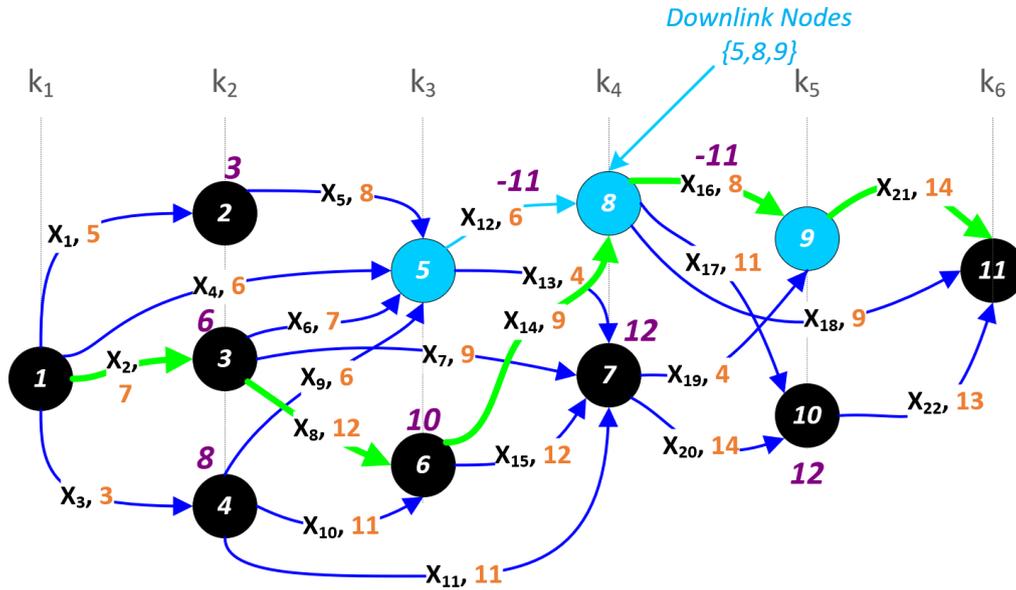


Figure 7. The path solution resulting when constraining the on-board memory to a value of 20. The path of edges is given as 2, 8, 14, 16, and 21.

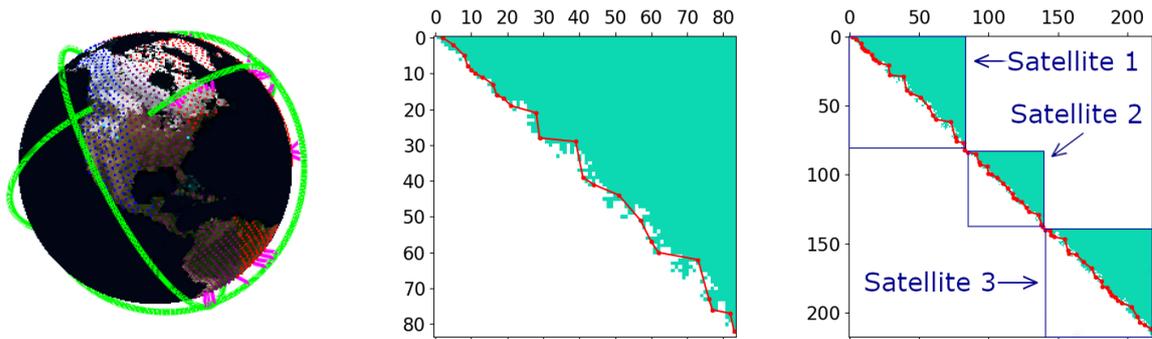


Figure 8. On the left is shown the imaging target deck, the middle shows the adjacency matrix assuming a single satellite, and the right image shows the adjacency graph for three satellites. The adjacency matrix is visualized using a cyan pixel for a 1 and white pixel for a 0. The nodes are sorted first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

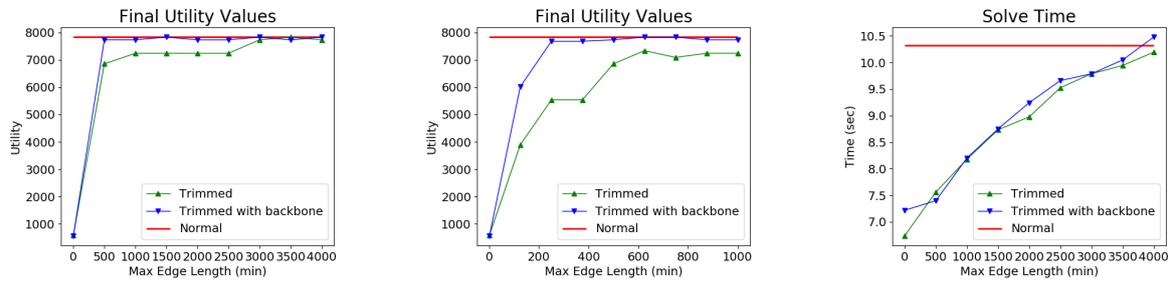


Figure 9. The resulting utility score and computation cost from the scenario in Figure 8 using three satellites. The left image shows the edge length vs utility for the untrimmed, naively trimmed, and backbone trimmed graphs. The middle image is a zoomed in view of the left image. The right image shows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

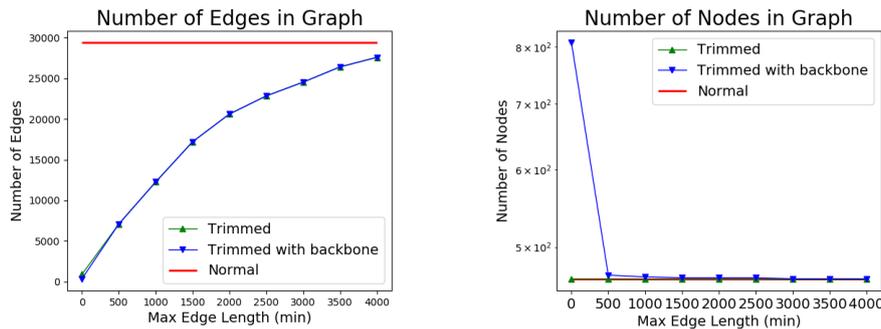


Figure 10. The number of edges and nodes vs the trim length.

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.

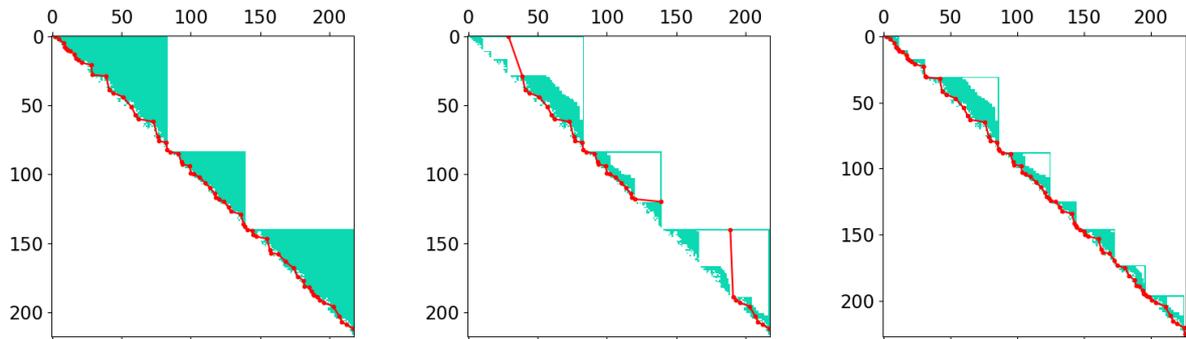


Figure 11. The result of trimming on the adjacency matrix of the graph. The left matrix shows the untrimmed graph. The middle matrix shows the trimmed graph without backbone nodes. The right matrix shows the trimmed graph with backbone nodes. The red lines on the plot show the planned path for each of 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.

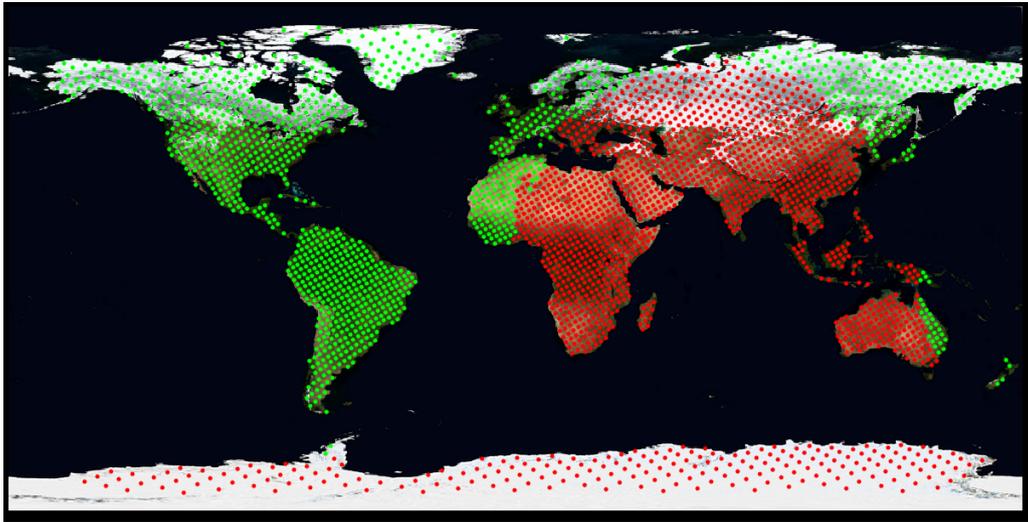


Figure 12. Imaging target distribution across the Earth's land mass area with targets accessible during the planning period marked in green. Targets marked in red do not meet the lighting constraints due to being in eclipse during the planning period.

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].

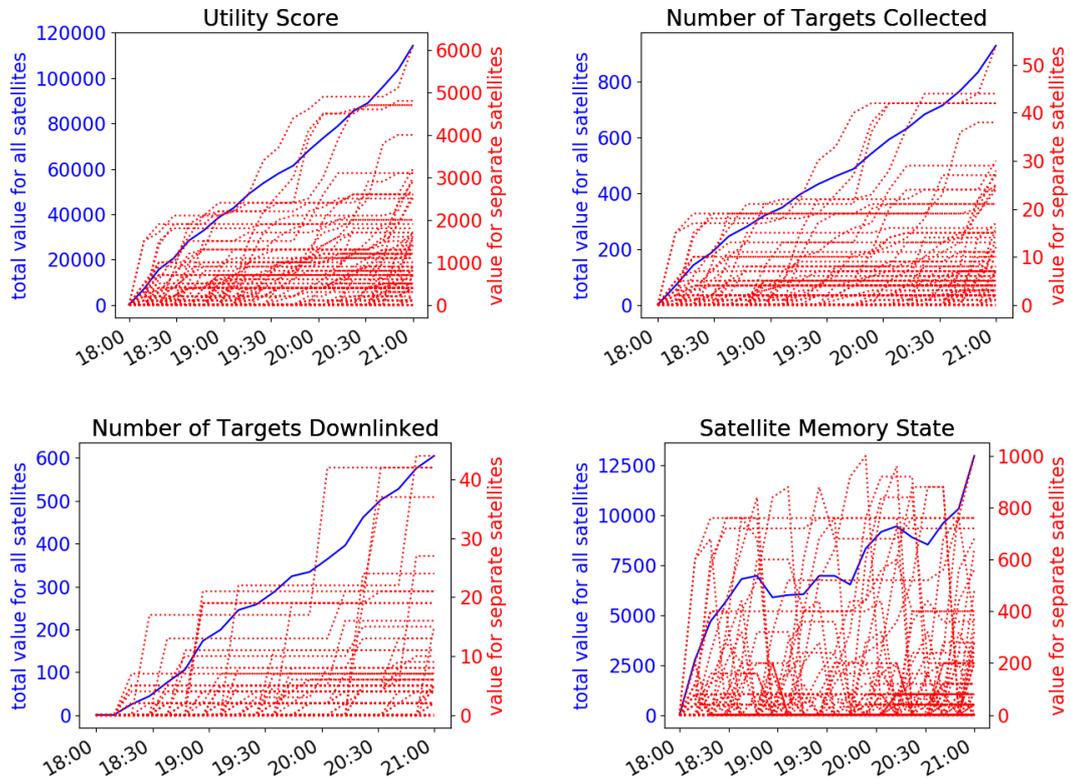


Figure 13. These figures show the results from a simulation with 100 satellites in 25 planes with 2814 ground targets spread over the surface of the earth and 5 ground stations in Alaska, Antarctica, Australia, Hawaii, and Norway. The left axis (in blue) shows the collective satellite values and the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

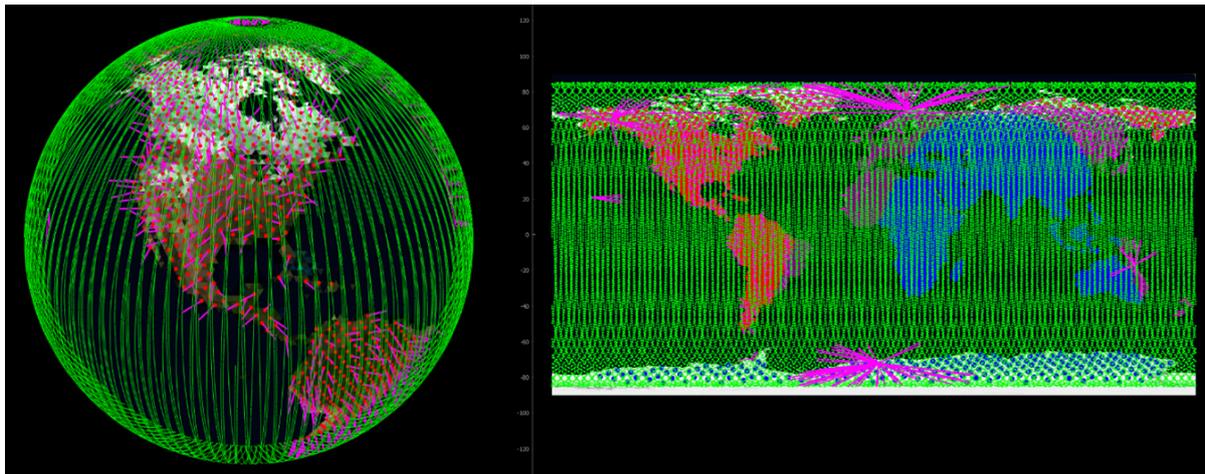


Figure 14. Global view of satellite planes and orbital configuration analyzed using 100 satellites in 25 planes with 5 ground terminals. In the left illustration, the red dots show where the satellite was located when an image was captured while the pink line illustrates the pointing at the time of collection. 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.

DECLARATIONS

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).

Availability of data and materials

Not applicable

Financial support and sponsorship

None.

Conflicts of interest

All authors declared that there are no conflicts of interest.

Ethical approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Copyright

© The Author(s) 2022.

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. Armas 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 Weerd 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, Jozefowicz 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