Optimal Multi-Satellite Planning for Collecting and Downlinking Data with Limited Storage, for Wildfire Danger Monitoring

Anonymous submission

Abstract

We present a novel Mixed Integer Linear Program formulation which produces optimal plans for a challenging climate science problem: wildfire danger monitoring. We are motivated by the need to solve a real-world mystery: What is the optimal plan for collecting wildfire-related sensor data using NASA's currently active CYGNSS satellite constellation?

Our planner generates coordinated plans for data collection and downlinking, for multiple satellites with limited storage capacity and energy constraints. Each potential observation target is associated with a science reward based on a wildfire probability index produced by the U.S. Geological Society. The planner maximizes the aggregate rewards collected for all observed targets on all satellites.

We present a novel interval-based abstraction which eliminates time-indexed variables and enables the first optimal solutions to this problem to be found. We compare optimal MILP results vs. suboptimal baselines produced using Monte Carlo Tree Search. We present experiments producing optimal 24-hr plans for a constellation of 8 satellites, which capture 99% of the $\sim$23,000 available targets.

Science Problem and Application

Wildfires are unplanned fires, which may result from human activity (campfires, arson, escaped prescribed burn projects), lightning, volcanic activity, etc. In the United States, from 2013 to 2022, there were an average of 61,410 wildfires annually and an average of 7.2 million acres impacted annually (Congressional Research Service, 2023). In addition to the immeasurable loss of life and property, the economic impact of wildfires depends on their location, size, duration, and severity and can be extensive. The average annual federal spending in the U.S. on fire suppression totaled $2.5 billion (in 2020 dollars) between 2016 and 2020 (Congressional Budget Office, 2022).

Remote sensing by satellites helps track pre/active/post fire conditions. During active fire conditions, early detection, and subsequent monitoring of progression of the fire front are possible at a global scale by satellites. For example, Fire Information for Resource Management System (FIRMS) distributes Near Real-Time active fire data from the Aqua and Terra satellites(NASA, 2023).

This paper is concerned with NASA's currently active CYGNSS mission (Ruf et al., 2018). It consists of 8 spacecraft, each carrying a GNSS Reflectometry (GNSS-R) instrument. The instrument collects Global Navigation Satellite System (GNSS) signals reflected off the Earth's surface, and geophysical properties such as soil-moisture are inferred from it. CYGNSS collects reflected signals from GNSS networks including GPS, Galileo, GLONASS, Beidou. Usually, the CYGNSS sensors are always kept ON in a low data volume gathering mode, where the information is clipped and compressed on-board (with loss). We are interested in a high data volume gathering mode called *RawIF*, where observations in raw format can downlinked without compression.

This work focuses on using multiple spacecraft for monitoring potential wildfires over the contiguous U.S. The term "fire danger" is used by the U.S. Geological Survey (USGS) to quantitatively define the risk of a wildfire. The likelihood of a wildfire occurrence can be evaluated from factors including weather, vegetation, soil moisture, human activity, which can then help with emergency response preparation or mitigation activities such as prescribed burns.

Background and Novelty

There is a lot of prior research on planning for Earth Observing Satellites, with many variations. Most of that work involves scheduling observations only (Frank et al., 2016), or downlinks only (Spangelo et al., 2015), or scheduling observations and downlinks as independent sub-problems (Cho et al., 2018). There is less work proposing an integrated mathematical model for simultaneously planning data collection and downlinking, and fewer still enforce limited data storage constraints. To our knowledge, those remaining papers in this subspace use synthetic, randomly generated targets such as (Xiao et al., 2019) and (Cho et al, 2018) as discussed in the Related Work section.

Our system starts with a novel set of inputs: Real orbits, real targets, real science rewards and real satellite operational constraints for data and energy capacities. We use CYGNSS' specific orbits and operating constraints (storage, energy), and real targets with associated science values based on a wildfire danger index produced by the U.S. Geological Survey (USGS).

Figure 1 shows the USGS Wildfire Large Fire Potential (WLFP) forecast for August 1, 2020. This heatmap was produced over a 1km grid over the contiguous U.S. and it drives the target values which the planner maximizes. There are

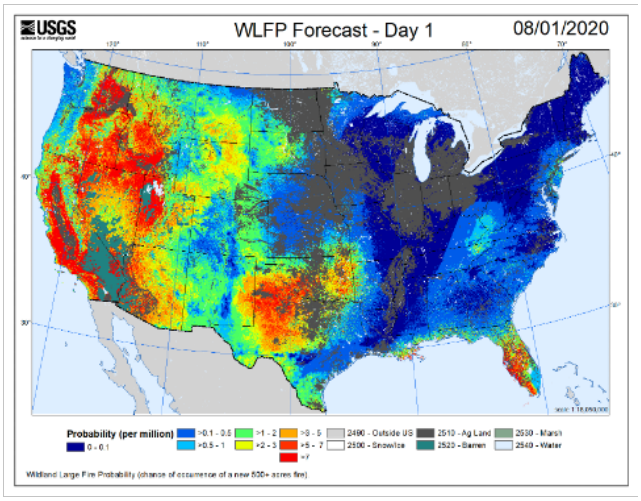

Figure 1: USGS Wildfire Large Fire Potential Forecast

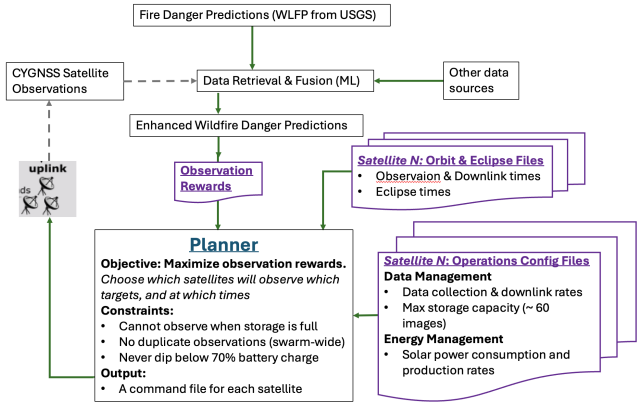

Figure 2: Planner Inputs and Outputs

$\sim$47k targets, and each one is assigned a WLFP value in the range 0 to $<$7. We remove grid-points associated with snow/ice, water, marsh, etc., where the WLFP value is invalid. The observation priorities are equal to the forecasted WLFP, i.e. we target locations of high predicted WLFP.

The large scale of this problem is a challenge. A 24-hour plan for 8 satellites involves $\sim$92,000 seconds when there is a binary choice between either collecting data or not, or between downlinking data or not, and there are $\sim$23,000 unique potential observation targets. Limited storage means each satellite can only store 60 images at a time. We are not aware of prior work with this combination of elements: Integrated planning for data collection and downlinking, for multiple satellites with limited data storage, with real world inputs, at the scale of our system.

We present a novel abstraction formulated and solved as a Mixed Integer Linear Program (MILP) to produce optimal, coordinated observation and downlink plans for the 8 CYGNSS satellites. This work aims to improve wildfire danger monitoring models. Key planner input includes a set of target visibility times. Each target has an associated science reward reflecting the probability of a large wildfire starting there. The planner aims to maximize the aggregate science reward for the whole constellation, by choosing to observe as many high-value targets as possible, without exceeding storage capacity which is limited to 60 seconds of raw data (60 images) at a time. When full, data must be downlinked to free up storage before collecting more images. There are too many targets to observe them all. Multi-satellite coordination is achieved by ensuring any target captured by more than one image is only counted once in the objective.

Motivation

A prior approach to this problem produced suboptimal solutions using Constraint Processing and Monte Carlo Tree Search (MCTS), with time-indexed decision variables at 1 second granularity (Levinson et al., 2024; 2022; 2021). In this paper we present an interval-based abstraction which enables optimal solutions to be found quickly using MILP.

The planner's job is to produce plans yielding the maximum science rewards, but we didn't know what the highest achievable score was. We didn't how many images or target targets could be covered. The primary motivation behind the work presented here was the need to find out what our application's maximum objective score is.

Note that for wildfire monitoring, we are looking for different targets than the nominal CYGNSS mission, so we have different optimization criteria than CYGNSS standard operations. CYGNSS was designed to track Cyclones not wildfires, so we are using their sensor data "off label" and have different requirements. In particular, we use only the RawIF sensor mode, which is not processed (compressed) on board and thus fills up storage faster than usual. This is why we have a 60 second (60 image) data collection limit which is not the case for standard CYGNSS operations.

Planner Model and Methods

Figure 2 shows the inputs and outputs for our centralized, ground-based planner. A key input is the U.S. Geological Survey (USGS) produced Wildland Fire Potential Index (WFPI)-based Large Fire Probability (WLFP) forecast (shown in Figure 1). WLFP is an estimate of the probability that a new fire will burn to more than 500 acres (202 ha) (U.S. Geological Survey, 2024). Other inputs include the observation and downlink opportunity times, and eclipse times for each satellite. These different inputs are integrated by the Data Retrieval and Fusion module which includes Machine Learning methods. Dashed arrows in Figure 2 (uploading and executing the plans) has not yet been implemented, but the concept of operations involves the plan being verified by the Mission Operations Center, then uploaded to the satellites. Data is downlinked and is utilized to improve fire danger monitoring (Ravindra et al., 2023).

The Earth Observation Simulator (EO-Sim) calculates observation opportunities and ground-station contacts (Ravindra et al, 2021). Satellite orbits are propagated (forecasted) using an SGP4 propagation model. The grid-points which could be observed with RawIF mode are calculated for each second. Three ground-stations are considered: Prioranet ground stations in (1) South Point, HI, USA, (2) Santiago, Chile and (3) Western Australia. Contact opportuni-

ties with these ground-stations are calculated to yield time-periods at which data downlink is possible.

Data Storage Model: Our data storage model is based on the CYGNSS satellite specs (Ruf et al., 2018). Each observation collects one "image" (corresponding to a 1 second observation), which may capture several targets. Storage capacity is 60 seconds of data (60 uncompressed RawIF observations), and it takes ∼20 minutes to empty a full buffer. More specifically, each observation (image) command fills up 96.22 megabits of storage, and each downlink command (second) frees up 4 megabits (about 5% of an image). If storage is full, it takes about 20 downlink commands (20 seconds) to free up enough space to collect another image. Onboard storage is a First In First Out (FIFO) buffer, and we cannot randomly access parts of the buffer to be cleared out.

Constraints: The planner enforces these constraints:

1. No observations allowed when storage (online data buffer) is full
2. No downlinks allowed when storage is empty
3. Greedy Downlink: Ensure downlink occurs whenever there is data in the buffer and a downlink opportunity is available
4. Never allow battery charge to dip below 55%
5. Target rewards are counted only once in the objective, even when observed by multiple satellites

Planning challenges: The key challenge is the huge search space, a sequence of$\sim 2^{92,000}$ binary choices. This is due to modeling multiple coordinated satellites with 1 second temporal resolution over large planning horizons (12 hours in our experiments). The small 1-second temporal resolution of the planning model is related to the high satellite speed (∼7km/s). Another challenge was developing an integrated model for data collection and downlinking, with limited data storage capacity. It takes 1 second to collect an image, and 20 seconds to downlink 1 image, and we can only store 60 images at a time. Another modeling challenge is that every image we collect contains multiple target locations. The planner can only *directly* choose images, not individual targets, but our objective is based on the targets, which introduces a challenge for coordinating the satellites when they collect different images with overlapping targets.

Figure 3 shows a key planner input, the *Choice File*, which defines the space of time-indexed command choices for all satellites. Each row defines a satellite's binary choice, for each second when the satellite has either a target viewing or downlink opportunity. The first row specifies : At time 28, satellite 2 has a binary choice between (a) observing image 25 (which contains targets 27 and 39), or (b) remaining idle. The second row says at time 42, satellite 4 may downlink data to a ground station in Hawaii, or remain idle. Finally, at time 120, satellite 3 may either observe image 302 (containing targets 39,50, and 51), or remain idle. Note that target 39 can be captured by either at time 28 by Satellite 2, or at time 120 by Satellite 3. The model ensures that each target is counted only once towards the objective, to provide multi-satellite coordination.

Time	Satellite	Choices	Image	Targets
28	2	Observe, Idle	25	27, 39
42	4	Downlink, Idle	--	Hawaii
120	3	Observe, Idle	302	39, 50, 51

Figure 3: The *Choice File* defines binary command choices

The duration of each command (observation or downlink) is 1 second, but we only consider times when there is a choice. With 8 satellites monitoring the contiguous US, and a 24-hour plan horizon, there are 92,283 binary choice points defined in the Choice File. This means the number of unique states in the search space of plans is $2^{92,283}$.

Objective: The planner's objective maximizes the sum of all target rewards in the plan:
maximize $\sum_t targetValue(t), \forall t \in \{$set of all targets in all images in all satellite plans$\}$.

Suboptimal Baselines using MCTS

Due to the large problem size, we initially produced approximate solutions with a mix of Constraint Processing and Monte Carlo Tree Search (MCTS). Detailed discussion of that work is outside the scope of this paper, please see (Levinson et al., 2024; 2022; 2021) for more details. In this paper, we refer to the MCTS results only to provide a baseline, suboptimal objective, to compare against our optimal solution which uses the same objective function (the sum of all observed target rewards). This enables us to finally learn how close our MCTS solutions were to optimal.

Optimal Solutions with an Abstract Model

Seeking an optimal solution, we developed an interval-based abstraction which reduces the problem size so much that we can now find optimal solutions quickly using MILP. The MCTS and MILP results use exactly the same *objective* calculation, so we can compare their objectives as apples to apples, even though the MCTS model is not (yet) using the same abstract model used for our MILP. The objective is maximizing the sum of all target rewards, regardless of which model is used to select the targets.

Our initial goal for MILP was to determine how close to optimal our MCTS was getting. Even if it took many days for MILP to solve, we wanted to know what optimal was. We have achieved that goal as presented in the Evaluation section below. Further, we are now achieving optimality so quickly, there is little motivation to rewrite the MCTS model to use the same data cycle abstraction as our MILP, because MCTS is unlikely to outperform these MILP results since it relies on randomness and many training cases.

The **key advantage** of this abstract model is the elimination of all time-indexed variables. Instead of decision variables indexed at 1 second granularity, we model temporal intervals called *data cycles*. Each cycle is a sequence of 2 phases (sub-intervals): First the *observation* phase fills up storage, followed by the *downlink* phase which frees up storage. Cycles are repeated for the duration of the plan horizon ($\sim$ 7 cycles per satellite in 24 hour plan). We extract these

Cycle	Observation Phase	Time Gap	Downlink Phase
1	Times: 4695 - 5410 (716 secs) Image IDs: 1-667 (668 images)	5411- 8277 (2867 secs)	Times: 8278-10423 (2146 secs) Available Downlink secs: 972
	Plan: Selected Images: 60 *Storage: Avail 100% - Used 100% (60 images) + Freed 81% (972 secs) = 81%*		
2	Times: 10683 - 11426 (744) Image IDs: 668-1369 (702)	11427-14437 (3011 secs)	Times: 14438 - 16363 (1926 s) Available Downlink secs: 360
	Plan: Selected Images: 48 *Storage: Avail 81% - Used 80% (48 images) + Freed 29% (348 secs) = 30%*		
3	Times: 16736 - 17129 (394) Image IDs: 1370-1669 (300)	17130 - 22276 (5146 secs)	Times: 22277 - 70133 (47857 s) Available Downlink secs: 4588
	Plan: Selected Images: 18 *Storage: Avail 30% - Used 30% (18 images) + Freed: 70% (844 secs) = 70 %*		
4	Times: 72640 - 72674 (35) Image IDs: 1670-1682 (13)	72675 - 75609 (2934 secs)	Times: 75610-76165 (556 secs) Available Downlink secs: 556
	Plan: Selected Images: 10 *Storage: Avail 70% - Used 17% (10 images) + Freed: 47% (556 secs) = 100 %*		
5	Times: 78383 - 78870 (488) Image IDs: 1683-2132 (450)	78871-81669 (2798 secs)	Times: 81670 - 83816 (2147 s) Available Downlink secs: 928
	Plan: Selected Images: 46 *Storage: Avail 100% - Used 77% (46 images) + Freed: 77% (920 secs) = 100%*		
6	Times: 84186 - 84866 (681) Image IDs: 2133-2782 (650)		*NO DOWNLINK PHASE* *(24-hourr horizon ends)*
	Plan: Selected Images: 60 *Storage: Avail 100% - Used 100% (60 images) + Freed: 0% (0 secs) = 0.0 %*		

Figure 4: Data Cycles for one satellite over 24 hours

data cycles during preprocessing from the *Choice File* (Figure 3). The cycles and phases within them vary in length. The last cycle in the 24-hr horizon may not include a downlink phase depending on where the satellite is in orbit when the 24-hour horizon ends.

Figure 4 shows the data cycles for one satellite over a 24 hour period. The first cycle starts at time 4695 (the first observation opportunity) and lasts 716 seconds through time 4510. There is a gap with no observation or downlink opportunities until the first downlink opportunity at time 8278. Obervation opportunities don't overlap with downlink opportunities because we are only observing targets in the CONtinental US (CONUS), and the three ground stations are not in CONUS. The downlink phase lasts 2146 seconds, but there are only 972 seconds when we could downlink data. The downlink opportunities within a phase may be discontinuous because the satellite may pass over multiple ground stations with gaps in between. There is also a gap between the end of each downlink phase before the next cycle starts, while the satellite travels back to CONUS.

All images ID's are assigned in sequence for each satellite in sequence. Figure 4 shows that this satellite owns image ID's 1-2782, with opportunities to collect images 1-667 during Cycle 1. Images 668-1369 may be collected during Cycle 2. Each cycle contains hundreds of imaging opportunities, but each satellite can only collect 60 images each cycle due to limited storage.

The plans produced for each cycle appear in the shaded boxes of Figure 4, showing the number of images collected, and the storage consumption and production on that cycle. Some cycles consume all storage or use all available downlink seconds, and some cycles don't. Note in cycle 3 there are 4588 downlink seconds available, but only 844 seconds are used, freeing only 70%. In contrast, 100% of the storage is freed in the next two cycles (4 and 5). The last cycle ends with full storage because the 24 hour horizon ends.

The model enforces constraints on the percentage of storage capacity available at the beginning and end of each cycle (not for every second within the cycle). It tracks aggregate storage consumption (filling up storage) and production (freeing up storage) at the temporal resolution of these data cycles. It does not track specific times or even the order that images are collected within a cycle (that is recovered during post-processing). It only counts the number of images collected in the observation phase of a cycle to ensure it doesn't exceed the available storage (which is left over from the prior cycle). It also ensures the number of downlink seconds per cycle does not exceed available downlink seconds per cycle. The model also tracks energy consumption and production to ensure it never dips below a minimum charge percentage (55%). This is never an issue because the solar panels keep the battery near full. Energy never dips below 98% of charge in our plans.

Mixed Integer Linear Program (MILP) Formulation
Our MILP formulation is shown below:

Parameters

S = A set of satellites, $s \in S$ is a satellite id
K_s = the set of cycles for sat s
I = the set of image opportunities for all satellites,
$i \in I$ is an image id, $1 \leq i \leq |I|$
T_i = the set of targets covered by image i
$J = \bigcup_{i=1}^{|I|} T_i$ = the set of all targets covered by all images
r_j = reward for target j, $j \in J$
I_j = images which cover target j
$I_{s,k}$ = images visible by sat s during cycle k
$dur_{s,k}$ = total # of seconds in cycle k on sat s, $1 \leq k \leq |K_s|$
$dnl_{s,k}$ = # of downlink secs *available* at end of cycle k on s
$eclipse_{s,k}$ = # of eclipse seconds in cycle k on sat s
Δs^- = Storage % *consumed* / observation
Δs^+ = Storage % *produced* / downlink second
Δe^+ = Energy % *produced* / second (except during eclipse)
Δe^- = Energy % *consumed* / second
Δe_d^- = Energy % *consumed* / second of downlinking
e_{min} = minimum energy % (min. state of charge)

Decision Variables

$x_i = 1 \Rightarrow$ image i is in the plan (binary), $\forall i : 1 \leq i \leq |I|$
$y_j = 1 \Rightarrow$ target j is in the plan (binary), $\forall j : 1 \leq j \leq |J|$
$sa_{s,k}$ = Storage % *available* for sat s on cycle k,
$0 \leq sa_{s,k} \leq 100$
$sc_{s,k}$ = Storage % *consumed* for sat s on cycle k,
$0 \leq sc_{s,k} \leq 100$
$sp_{s,k}$ = Storage % *produced* for sat s on cycle k,
$0 \leq sp_{s,k} \leq 100$
$e_{s,k}$ = Energy % *available* for sat s at *beginning* of cycle k,
after being capped at 100%, $0 \leq e_{s,k} \leq 100$
$e_{s,k}^{raw}$ = Energy *available* (%) for sat s at *end* of cycle k,
before being capped at 100%, $0 \leq e_{s,k}^{raw} \leq 100$
$e_{s,k}^{net}$ = net Energy change (%) for sat s *during* cycle k,
$-100 \leq e_{s,k}^{net} \leq 1000$

Objective: The objective is to maximize the sum of rewards for all observed targets:

Maximize $\sum_{j=1}^{|J|}$ r_j y_j

Constraints:

$$y_j \leq \sum_{i \in I_j} x_i, \quad \forall j \in J \quad (1)$$

$$sa_{s,1} = e_{s,1} = 100, \quad \forall s \in S \quad (2)$$

$$sa_{s,k} = sa_{s,k-1} - sc_{s,k-1} + sp_{s,k-1}, \forall s \in S, \forall k \in K_s \quad (3)$$

$$sc_{s,k} = \sum_{i=min(I_{s,k})}^{max(I_{s,k})} \Delta s^- x_i, \quad \forall s \in S, \forall k \in K_s \quad (4)$$

$$\mathrm{sc}_{s,k} \leq sa_{s,k}, \quad \forall s,k \quad (5)$$

$$sp_{s,k} \leq 100 - (sa_{s,k} - sc_{s,k}), \quad \forall s,k \quad (6)$$

$$sp_{s,k} \leq \Delta s^+ \ dnl_{s,k}, \quad \forall s,k \quad (7)$$

$$e_{min} \leq \ e_{s,k}, \quad \forall s,k \quad (8)$$

$$e_{s,k}^{net} = (dur_{s,k} - eclipse_{s,k})\Delta e^+ - (dur_{s,k}\Delta e^-) - (sp_{s,k}/\Delta s^+)\Delta e_d^-, \quad \forall s,k \quad (9)$$

$$e_{s,k}^{raw} = e_{s,k} + e_{s,k}^{net}, \quad \forall s,k \quad (10)$$

$$e_{s,k} = min(e_{s,k-1}^{raw}, 100), \quad \forall s,k \quad (11)$$

Constraints (1) enforce the requirement: If target j is in the plan, then at least one image containing j must be the plan. Constraints (2) set the initial conditions: all satellites start with 100% storage and energy available.

Storage Constraints: Constraints (3) ensure the storage available at the start of cycle $k =$ (the storage available at the start of prior cycle) $-$ (the storage consumed on prior cycle) $+$ (the storage produced prior cycle). Constraints (4) track the storage *consumed* by satellite s on cycle k. This leverages the fact that image ids are contiguous integers for each satellite and within each phase. This makes it easy to enumerate and limit the terms in this constraint to the subset of contiguously numbered images available for the given satellite in given cycle, without requiring images to be indexed by satellite. Constraints (5) ensure no cycle uses more storage than available. Constraints (6) ensure the plan never downlinks when storage is empty, and constraints (7) ensure the downlink duration fits within the available downlink time.

Energy Constraints: Constraints (8) ensure each satellite's energy never dips below the minimum energy threshold. Constraints (9) track the net energy change for satellite s *during* cycle k. Each satellite produces energy via solar panels (unless in eclipse) and simultaneously consumes energy during nominal operations. This constraint calculates the net change during the cycle. Downlinking data adds a small additional energy cost. This constraint says the net change = (the energy produced by solar panels) - (the energy consumed by nominal operations) - (the extra energy consumed while downlinking). The last term depends on how many seconds were spent downlinking in cycle k, but that is not a decision variable. Instead, the number of downlink seconds is calculated as a function of the decision variable $sp_{s,k}$, the storage produced during the cycle: $downlinkSecs = sp_{s,k}/\Delta s^+$. Constraints (10) calculate the available energy for sat s at the *end* of cycle k, *before* capping it at 100 %. Constraints (11) determine the available energy at the *beginning* of cycle k as a function of the energy available at the end of the $k-1$, *after capping* it at 100%.

The model addresses the issue that each image covers multiple targets (See Figure 3). Note the *objective* is driven by the *targets (not images)*, to maximize the sum of values for the observed targets (not images). However, the *constraints* are driven by the *images (not the targets)* because storage consumption is a function of the number of images taken (not the number of targets observed). Duplicate target observations contribute only a single reward to the objective, even if covered by multiple images in the plan.

Novel Knapsack variant: Our MILP formulation may be viewed as a novel *refillable* variant of the knapsack problem, where each satellite repeatedly fills, then (partially) empties its data storage "knapsack" several times throughout the 24-hr plan. The closest knapsack variant may be the "Online Removable Knapsack" (Iwama and Taketomi, 2002), where an online agent receives a stream of items it may choose to put in a knapsack, and it can remove items to make room for new ones. It assumes you don't know in advance which items will arrive or when they will arrive so it may have to remove low-value items when higher value items arrive later. We don't have that problem, because we know all targets and values in advance. In the removable knapsack problem, the removed items don't contribute to the objective, while in our case they do. The knapsack problem is typically not iterative, it fills up the knapsack once and it's done. Whereas our satellites carry their knapsacks all day and have opportunities to empty and refill along the way.

Our variant may be described as: Iterative, multiple, refillable semi-continuous knapsack. Its *iterative* because each satellite fills up its single knapsack multiple times per plan. Its *multiple* because there are multiple satellites, each with its own knapsack. *Continuous* means fractions of an item can be added to a knapsack. Our model *semi-continuous* because only whole images can be added to the knapsack but fractions of an image are removed from the knapsack.

Model abstractions and efficiencies: This model leverages several abstractions and efficiencies to facilitate model construction and solver performance, including:

Reduced number of variables and constraints: Replacing individual seconds with intervals, replacing some storage-related binary variables with continuous variables, and eliminating all downlink-related binary variables, significantly reduced the model size. Instead of tracking 95,653 seconds, we are now tracking only 49 data cycle intervals ($49 = 8$ satellites times $\sim$6 cycles/satellite). Additionally, using the variable $sp_{s,k}$ (storage produced) in constraints (9) avoided the need for tracking downlink seconds directly with additional decision variables and constraints.

The model does not include specific times when each image is collected. It only tracks the *number* of images collected and the *number* of downlink seconds per cycle, and the aggregate reward for the collected images.

Clearly the abstract model, with $\sim$47 intervals is much smaller than the 92,000 decision points in MCTS. This translates to half as many decision variables total, including new continuous variables which speeds up MILP linear relaxation. More specifically we traded 95,653 binary variables for 45,224 binary plus 147 continuous variables

Minimal Downlink Modeling: There is no modeling of specific seconds when downlinks occur when any particular image is downlinked. That information is filled in by post-

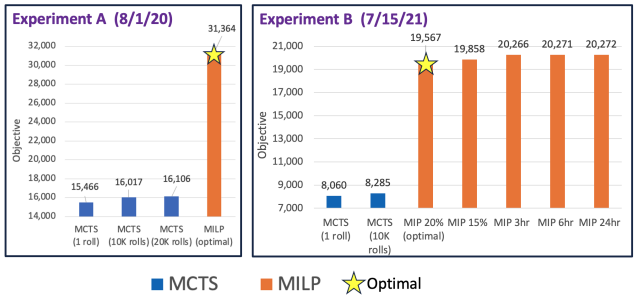

Figure 5: MCTS vs. MILP

processing. Downlinks are now modeled without any binary variables. We have replaced 73,425 binary decision variables for downlink commands, with 49 continuous decision variables for tracking the total download seconds required for each cycle. Continuous variables track only the number of planned downlink seconds per cycle, and constraints ensure it never exceeds the available downlink seconds.

Pre- and Post-processing: Much of the modeling in the original time-indexed model is now handled during preprocessing (before the model is constructed) and postprocessing (after the plan has been produced). During Preprocessing, the *Data Cycles* are extracted from the *Choice File*, and the image id's are assigned. Specific command times are plugged into the plan during post-processing. Each selected observation and downlink second in the plan is mapped to a specific time-indexed choice in the original *choice file* (Figure 3). When the planner does not use all of the downlink seconds available, then postprocessing assigns the specific seconds when downlinking occurs.

Image ID Partitioning: Partitioning image ids between satellites and between cycles facilitates tracking of the images selected for each cycle, without requiring any explicit tracking of image times within the cycle, and helps minimize the number of terms in constraints (4).

Evaluation

We evaluate our system with two data sets. Experiment A is the *high* fire danger case from 8/1/2020, one of the most active fire seasons on record (think "A" for "Active"). Figure 1 shows the USGS Wildfire Danger Predictions for that date, which are the basis for the target rewards in Experiment A. Experiment B is a *medium* fire danger case from 7/15/21. This year was less active and consequently the target rewards were generally lower in Experiment B. Figure 6 showd the total available rewards for Experiment A are ∼30K vs. ∼20K for Experiment B.

Experiment Environment: All experiments were run on a 2023 MacBook Pro, with an M3 Max chip, 36 GB Memory, using Python 3.8. Our MILP solver was Gurobi version 11.0.3. Our input data sets are available upon request.

We compare the optimal MILP solutions to our MCTS baseline solutions. They use different models but identical inputs and objective function calculation. Interestingly, the *approximate* solution uses an "exact model", modeling every second in the horizon in detail, while the *exact* solution

				Selected		Available			Plan Efficacy	
Test	Obj.	Gap %	Time	Images	Targets	Images	Targets	Total Rwd	Target %	Rwd %
1*: Experiment A	31,364*	0.00	72 min	2,063	22,366	23,912	22,563	31,374	0.991	0.9997
2 : Exp. B, 24-hr	20,272	0.08	T/O	2,075	22,177	24,153	23,275	20,366	0.953	0.9954
3 : Exp. B, 6-hr	20,271	0.09	T/O	2,042	22,166	24,153	23,275	20,366	0.952	0.9953
4 : Exp. B, 3-hr	20,266	0.11	T/O	2,045	22,144	24,153	23,275	20,366	0.951	0.9951
5 : Exp. B,15%,12hr	19,858	0.01	T/O	2,044	17,008	23,924	17,113	19,882	0.994	0.9988
6*: Exp B, 20%	19,568*	0.00	1 min	2,054	15,328	23,770	15,363	19,577	0.998	0.9995

Figure 6: MILP results for 6 test cases (* = optimal)

uses an abstract model which is inexact because it does not include many model details.

The MILP tests have two experiment parameters: *Value Threshold* and *Time Limit*. Value threshold is a minimum science reward threshold which is used to filter out low-reward targets. All targets are considered if no threshold is provided. We ran Experiment B with a value threshold of 0.15 (removes 7,145 targets), and with a threshold of 0.2 (removes 7,912 targets). Time limit tells the solver the maximum amount of time it can run before returning the best answer it found so far. The solver will terminate before the time limit when it finds an optimal solution. The MCTS baseline tests have one experiment parameter: The number of rollouts. MCTS is a form of reinforcement learning, and each Monte Carlo rollout represents a unique training case, so 10,000 rollouts means 10,000 training cases. The objective generally improves with more training (more rollouts).

The largest MILP model in Figure 6 is Experiment B with all targets, which has 47,812 variables and 23,827 constraints. The smallest model is experiment B with value threshold of 20%, which has 39,517 variables, and 15,915 constraints. This case solves optimally in 1 minute.

Baseline Comparison: Figure 5 shows the objective score for MCTS baselines vs. MILP for our two data sets. Experiment A results show the MCTS objective with a single rollout (no reinforcement learning) vs. 10K rolls vs. 20K rolls vs. MILP. MCTS takes about 1 hour per 1000 rolls, and scales linearly, so it takes ∼10 hours for 10K rolls and ∼20 hours for 20K rolls. The plan includes 99.1% of all visible targets, and collects 99.97% of the total value available if all visible targets were observed.

These results show that our best MCTS score on Experiment A was 51% of the optimal objective, and was 42% of optimal on Experiment B. The poor MCTS score is likely due to the fact that the MCTS tree is so deep that reinforcement learning only helps with a tiny prefix of the whole plan. The tree is 92,000 levels deep, because the model is at 1 second resolution, and includes both observation and downlink command choices. Figure 5 is the only chart referencing MCTS baselines. All other charts show MILP results only.

Analysis of MILP results: Figure 6 summarizes our MILP results, sorted from highest objective to lowest. The first and last cases are optimal solutions. The first row is experiment A, which achieved the optimal score of 31,364 in 72 minutes, and found a nearly equal objective of 31,352 within 8 minutes. This case solved optimally so quickly, there is little need to try different planner configurations the way we did for experiment B. We ran experiment B in five configurations: with solver time limits of 24-hr, 6-hr, and 3-hr, and also with two different *target value thresholds*, which filter

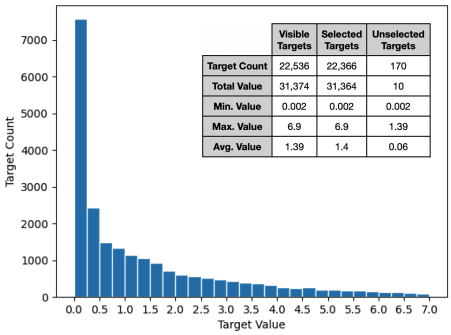

Figure 7: Test 1 Histograms: Experiment A (*Optimal*)

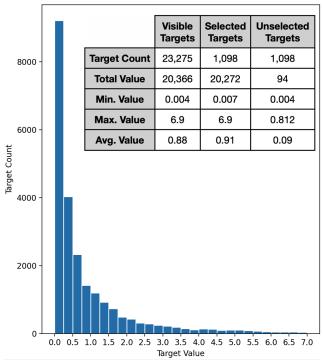

Figure 8: Test 2: Experiment B, 24hr T/O (*Best objective*)

out targets with rewards below the threshold. We ran experiment B with thresholds of 0.15 (15%) and 0.2 (20%). The 15% case also had a 12-hr time limit. The other columns are as follows: *Obj* is the objective score, *Gap %* is the MIP Optimality Gap, which is an upper bound on suboptimality. *Time* is the solver time to reach optimality, or timeout (T/O) when the time limit is reached.

The *Plan* section shows the number of images and targets in the plan. The *total reward* in the plan is the same as the objective column. The *Available* region shows the number of images, targets, and total reward available. The *Plan Efficacy* region on the right shows we scoop up almost all of the targets and rewards. Target% = (selected target count)/(available target count), and Rwd% = objective/Available Total Reward.

In both experiments, the objective increases as the number of selected targets increase. In the experiment B timeout cases, as planning time increases, so does the number of observed targets because the number of targets per image increases. In Experiment B, the optimal objective (19,567) is *lower* than the suboptimal tests. This demonstrates an interesting trade between the number of available targets, reward, and optimality. Filtering out low-reward targets allows the solver to find an optimal solution faster, but some reward is left on the table (regret). One lesson is perhaps optimality is the not the most important metric if it increases regret.

Figure 7 shows the target value histogram for Test 1 (experiment A). The embedded table shows more target value statistics. The first row shows the the number of available (visible) targets, the number of targets selected in the plan,

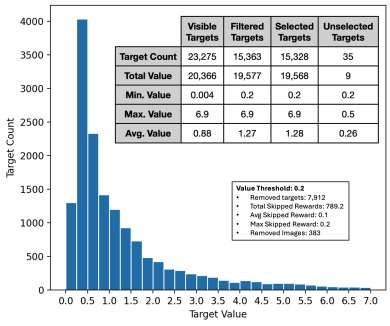

Figure 9: Test 6: Expt. B, 20% value threshold (*Optimal*)

and the number of remaining (unselected targets). In this case, only 170 targets out of 22,536 remained unobserved, with a total reward value of 10 out of 31,374 remaining uncollected. This shows that the planner is scooping up all of the high-reward targets. The *Total Value* row shows the total value available (visible) vs. the value in the plan (the objective), vs. the residual value which is not collected.

Figure 8 shows a similar histogram and table for Test 2, the experiment B case with the best objective (20,272). This is the case shown at the far right of Figure 5. In this case, the solver timed out after 24 hours with a MIP gap of 0.08, which is nearly optimal. A total of 1098 targets out of 23,275 remained unobserved for a total uncollected reward of 94.33 out of 20,366. The plan covered 95.3% of all visible targets and collects 99.5% of total available reward.

Figure 9 shows the histogram and table for Test 6, the *optimal* experiment B case, which ironically has lower objective (19,568) than the suboptimal objective (20,272) shown in figure 8. This case has a target value threshold of 0.2, which means it filters out any targets with value $\leq$ 0.2. The *Filtered Targets* column describes the targets which remained after filtering out targets below the threshold. This optimal solution is found is *less than 1 minute*. This removes 7,912 low value targets from the full set of 23,275 targets in experiment B, which means skipping 789 potential reward points. Each skipped target eliminates a y_j decision variable. If all of the targets in an image are filtered out, then we remove the whole image from the model, removing a x_i decision variable. This plan selects 99.8% of the remaining targets, and collects 99.5% of the remaining available reward. Only 35 targets out of 15,363 were skipped.

Plan Quality Trades: Comparing Figures 8 and 9 demonstrates the *trades between solver time, optimality and objective*. Increasing the value threshold allows us to reach an optimal solution quicker, but also reduces the objective's upper bound. Here we achieve an optimal objective in less than one minute, but we have regret about filtering out some additional reward. That skipped reward, however, would come from many low reward targets, which may or may not be of interest to wildfire scientists. This raises science questions about if/when to ignore low-value targets.

For comparison, with a slightly lower threshold of 0.15 (more targets), the solver reaches an objective of 19,856 in 1.27 hours, then times out after 12 hours with almost the same objective of 19,858. It found an optimal solution in 1

minute with threshold of 0.2 (15,328 targets), but timed out after 12 hours with threshold of 0.15 (17,008 targets). This shows how sensitive solve times are to the number of targets.

Related research

There is a large body of research addressing different aspects of satellite remote sensing problems, in various combinations. We focus our discussion below on systems which combine planning for data collection and downlinking with limited data capacity. Within that work, interval-based models rather than time-indexed, and MILP models in particular are most closely related.

(Xiao et al., 2019) present a variation of the "flow shop" problem where there are only 2 machines: observe and downlink. They present a MILP model for a "multi-satellite observation and data downlink scheduling problem". They represent time intervals rather than individual timepoints. They integrate observations and data downlinks but have *no storage limit*, and all targets must be observed, in contrast with our application. Their problem size is relatively small: 10 "tasks" in a four day plan horizon, or 20 tasks in 8 days, while ours is $\sim$23,000 tasks in 24 hours. Each *task* is an observation and downlink pair. They present experiments with up to two satellites. Their objective function is different. Since all tasks must be completed, they are minimizing completion time (makespan). In contrast we are given the completion time (our 24-hour plan horizon) and then maximize the science rewards collected for the subset of targets which get selected for the plan.

(Cho et al., 2018) present a MILP solution for generating observation and downlink plans for up to three satellites, with data storage capacity limits. Their model includes *time-indexed* decision variables for the start time of each observation and downlink command. They propose a 2-stage MILP process: First, construct and solve a MILP to resolve ground station (GS) conflicts, when for multiple satellites are competing to downlink to the same GS. Second, after assigning satellites to GS in the first phase, then construct and solve a second MILP to assign observations given the downlink assignments. They construct and solve different planning problems for observation and downlink planning. The downlink windows generated by the first step are used to constrain the observation tasks, while maximizing rewards for all observation tasks, and enforcing a constraint that data storage never exceeds capacity. They study non-greedy downlink planning, where GS have different energy/transfer rates. Their evaluation uses orbits from historical missions, with up to 700 *randomly generated tasks (targets)* for 1 satellite, or 500 for 3 satellites.

(Herrmann and Schaub, 2023) use MCTS for satellite observation scheduling and downlinking with capacity constraints for a *single satellite*, but they focus on a very different application than ours. They use MCTS to "play" against a high-fidelity Markov Decision Process (MDP) in order to generate training examples for a neural net. The neural net produces a reactive state-action *policy* for onboard (online) execution, to optimize observation collection for a single satellite. MCTS is used offline to generate state/value estimates to train a neural net and produce a policy for online execution (on-board the satellite). No "optimal" plan is produced as the MCTS plans are only training cases for the reactive online policy. They evaluate how well their policy handles situations which are outside the training set. In contrast, we focus on optimizing complete plans for specific situations known in advance.

(Spangelo et al. 2015), present the Single-Satellite Multiple Ground Station Scheduling Problem (SMSP). They present a MILP model similar to ours, but it's for *downlinks only*, and also for only a *single satellite*. Data collection times are an input parameter, not a variable. Their objective is to maximize the total amount of data downlinked. They consider cases where downlinking is a choice, not mandatory (greedy) as it is for us. In their case, the planner may choose to remain idle even when it has data while passing over a ground station. Different ground stations may have different data, energy and efficiency (reliability) rates, so it may save energy or be more reliable to skip over one downlink opportunity and wait for the next one. They split the plan horizon for each satellite into intervals delimited by passes over ground stations, similar to our model.

(Frank et al., 2016) present MILP and Constraint Processing (CP) formulations and comparative results for scheduling ocean color observations for a *single* satellite, where cloudiness plays a factor in plan success. They schedule observations only, without downlinking. They evaluate how planning is impacting by two potential sensor designs.

Conclusion

We presented a mathematical model for a novel earth observing satellite problem variant with the following elements: Multiple satellites, integrated data collection and downlinks, limited storage, real world data and large problem size (number of satellites and targets, plan horizon). We have 8 satellites, $\sim$23,000 targets, a 60 image limit, and 24 hour plan horizon. We demonstrated the optimal solution results in 99% of available targets and science rewards being collected. Our model may be viewed as a novel knapsack variant, where each satellite has a knapsack which may be repeatedly filled and unfilled throughout the plan horizon.

We identified trades between science value threshold, optimality gap, and objective score (see Figures 8 and 9). Optimality may be achieved by reducing the problem size. These trades (optimality vs. maximum reward) raise science questions about the importance of observing low-value targets.

Contributions include finding the first optimal solutions for this real-world wildfire danger monitoring problem. We presented an interval-based abstraction for integrated observation and downlink planning for a large-scale real-world earth science application. We demonstrated this new model can be solved optimally in less than 2 hours, within the project requirement of 3-hour solve times. Understanding what an optimal solution looks like for this real-world scenario, in terms of cost, reward, and trades between them, may help inform future mission design. Another contribution is making our large, real-world data sets available for other researchers.

Acknowledgments

References

Cho, D. H., Kim, J. H., Choi, H. L., Ahn, J. (2018). Optimization-based scheduling method for agile earth-observing satellite constellation. Journal of Aerospace Information Systems, 15(11).

Congressional Research Service. (2023). Wildfire Statistics. Retrieved from https://sgp.fas.org/crs/misc/IF10244.pdf Congressional Budget Office. (2022, June). Wildfires. https://www.cbo.gov/publication/57970

Frank, J., Do, M., Tran, T. (2016). Scheduling Ocean Color Observations for a GEO-Stationary Satellite. Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS 2016). https://doi.org/10.1609/icaps.v26i1.13780

Gurobi. 2024. https://www.gurobi.com Accessed: March 3, 2024.

Herrmann, A., Schaub, H. (2023). Reinforcement Learning for the Agile Earth-Observing Satellite Scheduling Problem. IEEE Transactions on Aerospace and Electronic Systems.

Iwama, K., Taketomi S. Removable online knapsack problems (2002). In Proc. of the 29th Int. Colloq. on Automata, Languages and Programming (ICALP). 2002.

Levinson, R., Niemoeller, S., Nag, S., Ravindra, V. (2022). Planning Satellite Swarm Measurements for Earth Science Models: Comparing Constraint Processing and MILP Methods. Proc. of the Int'l Conference on Automated Planning and Scheduling (ICAPS 2022)

Levinson, R., Nag, S., and Ravindra, V. 2021. Agile Satellite Planning for Multi-payload Observations for Earth Science, Proceedings of the International Workshop on Planning and Scheduling for Space (IWPSS). 2021.

Nag S., et al., 2024. "Distributed Spacecraft with Heuristic Intelligence to Monitor Wildfire Spread for Responsive Control," IGARSS 2024 - 2024 IEEE International Geoscience and Remote Sensing Symposium, Athens, Greece, 2024, pp. 699-703, doi: 10.1109/IGARSS53475.2024.10640833.

Nag S., et al., 2019. "Autonomous Scheduling of Agile Spacecraft Constellations with Delay Tolerant Networking for Reactive Imaging," presented at the International Conference on Planning and Scheduling (ICAPS) SPARK Applications Workshop, Berkeley, California, U.S.A., 2019

Nag, S., Moghaddam, M., Selva, D., Frank, J, Ravindra, V., Levinson, R., Azemati, A., Aguilar, A., Li, A., Akbar, R., 2020. D-Shield: Distributed Spacecraft with Heuristic Intelligence to Enable Logistical Decisions, Proc. of the 2020 IEEE International Geoscience and Remote Sensing Symposium.

Nag, S., Moghaddam, M., Selva, D., Frank, J., Ravindra, V., Levinson, R., Azemati, A., Gorr, B., Li, A., Akbar, R. 2021. "Soil Moisture Monitoring using Autonomous and Distributed Spacecraft (D-SHIELD)", IEEE International Geoscience and Remote Sensing Symposium, Virtual, July 2021

NASA, FIRMS: Fire Information for Resource Management System. Retrieved 2023 from https://firms.modaps.eosdis.nasa.gov/

Spangelo, S., Cutler, J., Gilson, K, Cohn, A. 2015. Optimization-based scheduling for the single-satellite, multi-ground station communication problem. Computers Operations Research 57 (2015). https://www.sciencedirect.com/science/article/abs/pii/S0305054814002809

Ravindra, V., Ketzner, R., Nag, S. (2021, July). Earth Obser-vation Simulator (EO-Sim): An Open-Source Software for Observation Systems Design. In 2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS (pp. 7682-7685). IEEE.

Ravindra, V., Roy Singh, S., Moghaddam, M., Nelson, K., Levinson, R., Melebari, A., Kannan, A. (2023). Distributed Spacecraft with Heuristic Intelligence to Enable Logistical Decisions for Global Navigation Satellite System Reflectometry (GNSS-R) of Wildfires. Presented at Earth Science Technology Forum. https://esto.nasa.gov/forums/estf2023/Presentations/S5P3RavindraSinghESTF2023.pdf

Ruf, C. S., Chew, C., Lang, T., Morris, M. G., Nave, K., Ridley, A., Balasubramaniam, R. (2018). A new paradigm in earth environmental monitoring with the Cygnss small satellite constellation. Scientific reports, 8(1), 8782.

United States Geological Survey. (n.d.). BAER FAQs. Retrieved from https://burnseverity.cr.usgs.gov/baer/

U.S. Geological Survey. (2023). WFPI-based Large Fire Probability (WLFP). Retrieved from https://www.usgs.gov/fire-danger-forecast/wfpi-based-large-fire-probability-wlfp

Xiao, Y., Zhang, S., Yang, P., You, M., Huang, J. (2019). A two-stage flow-shop scheme for the multi-satellite observation and data-downlink scheduling problem considering weather uncertainties. Reliability Engineering System Safety, Vol. 188.