



# Scheduling algorithms for rapid imaging using agile Cubesat constellations

Sreeja Nag<sup>a,\*</sup>, Alan S. Li<sup>a</sup>, James H. Merrick<sup>b</sup>

<sup>a</sup> NASA Ames Research Center, Bay Area Environmental Research Institute, N210/Rm226, Moffett Field, CA 94035, United States

<sup>b</sup> Management Science and Engineering Department, Huang Engineering Center, Stanford University, 475 Via Ortega, CA 94305, United States

Received 25 June 2017; received in revised form 19 October 2017; accepted 9 November 2017

Available online 21 November 2017

## Abstract

Distributed Space Missions such as formation flight and constellations, are being recognized as important Earth Observation solutions to increase measurement samples over space and time. Cubesats are increasing in size (27U, ~40 kg in development) with increasing capabilities to host imager payloads. Given the precise attitude control systems emerging in the commercial market, Cubesats now have the ability to slew and capture images within short notice. We propose a modular framework that combines orbital mechanics, attitude control and scheduling optimization to plan the time-varying, full-body orientation of agile Cubesats in a constellation such that they maximize the number of observed images and observation time, within the constraints of Cubesat hardware specifications. The attitude control strategy combines bang-bang and PD control, with constraints such as power consumption, response time, and stability factored into the optimality computations and a possible extension to PID control to account for disturbances. Schedule optimization is performed using dynamic programming with two levels of heuristics, verified and improved upon using mixed integer linear programming. The automated scheduler is expected to run on ground station resources and the resultant schedules uplinked to the satellites for execution, however it can be adapted for onboard scheduling, contingent on Cubesat hardware and software upgrades. The framework is generalizable over small steerable spacecraft, sensor specifications, imaging objectives and regions of interest, and is demonstrated using multiple 20 kg satellites in Low Earth Orbit for two case studies – rapid imaging of Landsat’s land and coastal images and extended imaging of global, warm water coral reefs. The proposed algorithm captures up to 161% more Landsat images than nadir-pointing sensors with the same field of view, on a 2-satellite constellation over a 12-h simulation. Integer programming was able to verify that optimality of the dynamic programming solution for single satellites was within 10%, and find up to 5% more optimal solutions. The optimality gap for constellations was found to be 22% at worst, but the dynamic programming schedules were found at nearly four orders of magnitude better computational speed than integer programming. The algorithm can include cloud cover predictions, ground downlink windows or any other spatial, temporal or angular constraints into the orbital module and be integrated into planning tools for agile constellations.

© 2017 COSPAR. Published by Elsevier Ltd. All rights reserved.

**Keywords:** Small satellites; Responsive space systems; Rapid replanning; Remote sensing; Attitude control; Dynamic programming optimization

## 1. Introduction

Earth-science processes are intrinsically dynamic, complex, and interactive. To achieve an all-embracing

understanding of the emergence and evolution of these processes requires the collection and assimilation of enormous amounts of data, using complementary measurements in space and time. Spatial measurements from multiple vantage points – space, air, ground and water – help resolve measurement and model uncertainties. Distributed Space Missions (DSMs) such as formation flight and constellations are being recognized as important solutions to

\* Corresponding author.

E-mail addresses: [sreejanag@alum.mit.edu](mailto:sreejanag@alum.mit.edu) (S. Nag), [alanli@alumni.stanford.edu](mailto:alanli@alumni.stanford.edu) (A.S. Li), [jmerrick@stanford.edu](mailto:jmerrick@stanford.edu) (J.H. Merrick).

increase measurement samples over space and time (D'Errico, 2012), especially when augmented with aircraft. The National Research Council (NRC), in its mid-term assessment of NASA's implementation of the 2007 Decadal Survey recommended a “*more agile and cost-effective replacement of individual sensors. . . moving away from a single parameter and sensor-centric approach toward a systems approach that ties observations together to study processes important to understanding Earth-system feedbacks*” (National Research Council, 2012). DSMs also minimize launch and operational risks by adding redundancy in numbers, and allow for deploying evolved technology as they become available. However, they carry a risk unexpected failures in case of poorly understood interdependencies. Continued effort for creating and maintaining an interoperable environment for a diverse set of sensors (land, marine, air, space) using software and internet is underway in NASA Sensor Webs (Mandl et al., 2006). The goal is to allow sensors to operate in a semi-automated, collaborative manner for scientific investigation, disaster management, resource management and environmental intelligence. The development of software tools to design DSMs, in keeping with customizable science objectives for rapid pre-PhaseA trade studies is currently underway at NASA Goddard Space Flight Center (LeMoigne et al., 2017). While this initiative is expected to foster innovation on multi-satellite solutions in the Earth Science community, it does not model operational planning, scheduling or autonomy.

Response and revisit requirements for Earth Observation (EO) vary significantly by application, ranging from less than an hour to monitor disasters, to daily for meteorology, to weekly for land cover monitoring (Sandau et al., 2010). Geostationary satellites provide frequent revisits, but at the cost of coarse spatial resolution, extra launch costs and no polar access. Lower Earth Orbit (LEO) satellites overcome these shortcomings, but need numbers and coordination to match GEO's responsiveness. While adding satellites to a constellation or optimizing their orbits can significantly improve revisit/response, adding agility to the satellites and autonomy to the constellation can improve the revisit/response for the same number of satellites in given orbits. Moreover, human operators are expected to scale linearly with constellation nodes (Eickhoff, 2011) and as satellites and ground stations scale to hundreds or more, operations staffing may become cost prohibitive. Deployment, maintenance, imaging, downlink, maneuver, de-orbit and other satellite operations within scarce resources are complex scheduling problems (Lin et al., 2005), and NP-hard unless formulated and bounded very carefully to make the design space tractable (Arkali et al., 2008). Scaling the problem to multiple satellites and including uncertainty of control subsystems adds complexity even further. Early investment in autonomy will increase management efficiency of the inevitably numerous space nodes, including better fault detection and isolation.

Large, single satellites with agile attitude control capabilities have demonstrated rapid image collection from space, mostly individually but also sometimes as an ad-hoc constellation. Fig. 1 (not to scale) shows six images of interest that are imaged by a single satellite by pointing appropriately over three orbits, which would not have been possible with a fixed viewing sensor. Autonomous, agile steering of the spacecraft body allows image acquisition over a much larger field of regard, thereby improving coverage and revisit. Planning and scheduling algorithms have been successfully developed for single large satellite missions, examples being Automated Scheduling and Planning Environment (ASPEN) for EO-1 (Sherwood et al., 1998), scheduling for the Advanced Spaceborne Thermal Emission and Reflectance Radiometer (ASTER) (Muraoka et al., 1998) on the Terra satellite, high resolution imagery from the IKONOS commercial satellite (Martin, 2002) and scheduling observations for the geostationary GEO-CAPE satellite (Frank et al., 2016). Scheduling image strips over Taiwan by ROCSAT-II was formulated as an integer programming problem and Lagrangian relaxation is used to decompose it into separable sub-problems (Lin et al., 2005). The hyperspectral instrument CHRIS, carried on the Proba spacecraft, demonstrated dynamic pointing for multi-angle imaging of specific ground spots that it is commanded to observe (Barnsley et al., 2004).

The problem of tasking multiple, diverse sensors was preliminarily addressed for aerial flight paths using the orienting algorithm and demonstrated on NASA's INTEX-B flight data (Oza et al., 2008). Scheduling resource-constrained observations for large satellite constellations has been formulated in detail for the French PLEIADES project (Damiani et al., 2005; Lemaitre et al., 2002) with recommendations dependent on weak to strong coordination between the space agents. Scheduling for the COSMO-SkyMed constellation of synthetic aperture radars was proposed using a deterministic constructive algorithm with look-ahead and back-tracking capabilities to allow for updates on resources and changes to requests (Bianchessi and Righini, 2008). Evolutionary algorithms have been proposed and computationally simulated for single spacecraft (Xhafa et al., 2012), multiple payloads (Jian and Cheng, 2008) and comparative merits documented for satellite fleets (Globus et al., 2002), but they are very limited in mission applications. Algorithms for agent-based autonomous scheduling have been implemented on NASA's Deep Space 1, and simulated for a cluster on a real-time testbed (Schetter et al., 2003). (Abramson et al., 2013; Robinson et al., 2017) have developed a coordinated planner that can handle a continuous stream of image requests from users, by finding opportunities of collection and scheduling air or space assets to maximize collected utility.

In the last decade, Cubesats have increased in size (27U or ~40 kg standard in formulation) and emerged as increasingly capable platforms for Earth observation

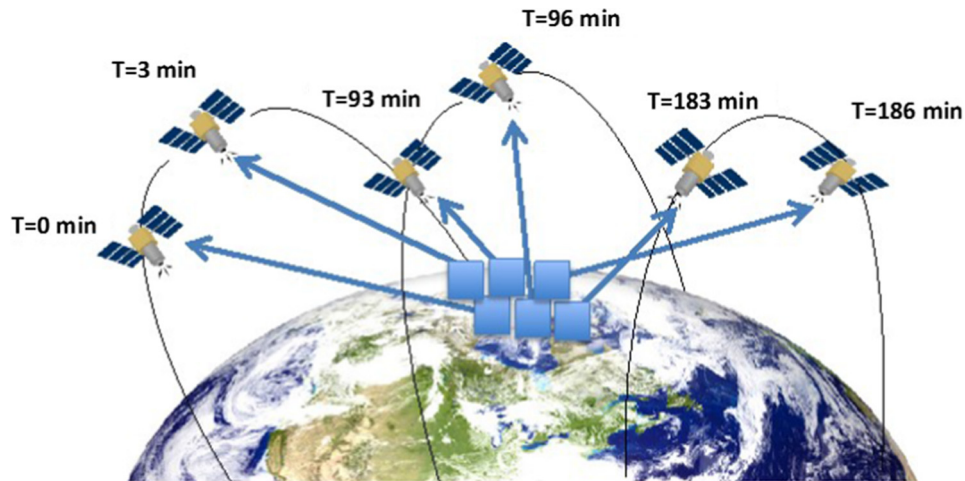


Fig. 1. Cartoon showing a single satellite dynamically pointing to observe six images over three orbits. Our proposed algorithm defines the attitude control, steering schedule for a constellation of such satellites and global image requests.

(Space Studies Board, 2016). Small satellites with one or few instruments provide a cost effective way to deploying a large constellation, thereby leveraging economies of scale and redundancy in numbers. In comparison, large bus-sized satellites with many instruments need redundancy in their components and higher reliability because the cost of their failure is higher. Simulation studies have optimized the scheduling for single Cubesat downlink to a network of ground stations (Spangelo and Cutler, 2012) or multiple payloads' downlink to existing stations (Jian and Cheng, 2008), within available storage, energy and access time constraints. Studies have also combined single satellite scheduling (using integer programming applied to greedy search) with information sharing across satellites for a weak consensus on feasible charging, downlink and observation schedules (Kennedy and Cahoy, 2015) using fixed view imagers. Theoretical studies have shown that multiple satellites when downlinking to ground stations with overlapping visibility can be scheduled in polynomial time only for special cases when station reconfiguration time is near zero (Arkali et al., 2008), and greedy and linear programming algorithms suggested.

Given the increasingly precise attitude control systems emerging in the commercial market for Cubesats (ARC Mission Design Center, 2016), small spacecraft now have the ability to slew and point as per command, within few minutes of notice. While academic literature addresses satellite scheduling/coordination for large, steerable satellites and small, fixed view satellites, we found very little work reported on algorithms for controlled pointing and distributed target observation for imaging constellations, given current Cubesat maneuverability, accurate pointing and image angular constraints. Assuming a known Cubesat, small satellite constellation or Sensor Web structure (in terms of orbits, sensors, ground stations, images of interest, etc.), this paper demonstrates a scheduling algorithm that steers each spacecraft attitude in a manner that

maximizes collected images and/or imaging time. The goal is to inform design studies of agile constellation operations because agility comes at the cost of ground segment complexity and associated schedule optimization. (Morris and Dungan, 2007; Morris et al., 2009) has simulated a workflow model and a model-based observation based planner, which adapts to changes in its own configuration, recognizes opportunities for modifying data acquisition plans to improve overall performance and coordinates resources and tasks accordingly. (Witt et al., 2008) demonstrated the planner on NASA's ST-5 mission for lights-out operations. Our proposed, ground-based planner will also be model-based and adaptive, however, for the purpose of scheduling steering and imaging operations for multiple spacecraft.

## 2. Proposed methodology

This section proposes a scheduling framework for the attitude control of multiple Cubesats, so that they can (together) image as many given targets as possible, given constraints from orbital mechanics (OM) and attitude control systems (ACS), and summarizes a review of relevant scheduling literature used to inform this framework. The innovative aspects are the modularity of the OM, ACS and optimization modules, which allows independent numerical solution, uncertainty modeling and innovation within each, as will be described in the upcoming subsections. The algorithm can be applied to any satellite constellation structure, ACS, target point list, space and time varying cloud cover or new requests for data, and can be modified easily for multi-angular imaging applications such as single CHRIS or satellite formations (Nag et al., 2016). It is expected to run, automated, on the ground station computers and the resultant schedules uplinked to the satellite at frequent overpasses. The length of the planning horizon is determined by (and greater than) the latency between expected overpasses, based on available ground

station networks. The ground stations are expected to estimate future satellite states using downlinked GPS data and orbit propagation, and able to communicate across the global ground network near immediately. Future work is envisioned to allow implementation of the presented algorithm onboard spacecraft, for more autonomous operations similar to (Damiani et al., 2005). This will entail onboard processing trades, onboard orbit determination, inter-satellite communication to exchange states and plans, and other hardware and software upgrades that are beyond the scope of this article.

The agile Cubesat constellation scheduling problem for imaging operations can be formulated using the variables in Table 1. The orbital parameters or states of the satellites ( $s \in S$ ), geometric constraints of the payload (FOR, FOV,  $\theta$ ), location and expected distortion of images ( $i \in I$ ,  $\Lambda(\theta)$ ) and specifications of the ACS system ( $C$ ,  $\tau$ ,  $M_c$ ,  $T_{max}$ , IN) are assumed known. Image importance can also be modeled by replacing distortion with functional quality  $\Lambda(I)$ , in case the scheduling problem has to be formulated as a set of images to be observed with varying urgency in civil vs. military applications. Any uncertainty in values can be added as stochastic variances (e.g. in ACS jitter) or as random space–time functions (e.g. cloud cover obfuscating some images). The solution will be in the form of  $\Omega(s)$ , which describes the series of attitudes that the satellite will command in order to maximize the images observed, after optimizing over mission and geometric requirements and hardware constraints. Our proposed framework and algorithm is generic over most geometric requirements and satellite/imager constraints and presented case studies are realistic examples of such. However, the formulation can only support a minimum FOV for the decoupled ACS module (Section 2.5). Also, beyond a certain maximum satellite size, it is far more efficient to steer the imager alone, instead of the entire satellite, as demonstrated by Ball Aerospace’s autonomous lower level software for real-time data collection at the sensor level, applied to adaptive lidars to control tens to a hundred individually steerable laser beamlets (Lieber et al., 2017).

The most common algorithms in Earth Observation Scheduling (EOS) applied to missions are those based on greedy approaches (e.g. ASTER on Terra (Muraoka et al., 1998)) and local search methods (e.g. knapsack algorithm and Tabu search on SPOT-5 (Vasquez and Hao, 2001), ASPEN on EO-1 (Sherwood et al., 1998)). Global optimization methods are expected to improve coverage compared to local search, over all satellites, images and times horizon. However, evolutionary or heuristic search methods, by themselves, are very dependent on initial conditions (genetic algorithms), need exponential time to converge (simulated annealing) or large training sets (neural nets). Analytical approaches are very quick in comparison, however lack the ability to numerically model uncertainties in satellite subsystems with customizable fidelity. For example, (Bunkheila et al., 2016) demonstrates scheduling the scan of single, agile satellites by dividing the areas of

interest into scan-able and chooses the sequence of strips, based on various distance functions for an ideal orbit. It uses fixed scanning times based on roll and pitch angles, to compute the temporal feasibility of pointing, without modeling the ADC system or its uncertainties.

EOS is well suited for the mixed integer linear programming (MILP) approach because to perform any activity at any time instant (or not) can be modeled as a binary integer (e.g. GEO-CAPE (Frank et al., 2016) or ROCSAT-II (Lin et al., 2005) scheduler) and CPLEX can solve such formulations efficiently. However, MILP allows for only linear constraints and a single objective function, and there is no guarantee of reaching an optimum in linear time. The first drawback can be addressed using linear bounds to otherwise non-linear variables (under-constrained formulation (Spangelo and Cutler, 2012)). The lack of a single, linear objective can be addressed using a Lagrangian sum of multiple objectives or a convex function representation of the objective. However, such approximations take away from accuracy and cutting plane bounds need not always be reliable (Lemaître et al., 2002). Nonetheless, MILP has been successfully formulated for EOS in flight missions - PLEIADES and the SPOT series (Lemaître et al., 2002), GEO-CAPE (Frank et al., 2016) - and adapted as Constraint-based Interval Planning in the EUROPA planner for Deep Space 1 (Frank et al., 2001). Constraint programming is not restricted to integer variables and linear equations/inequalities. Variables can be intervals or anything in the finite domain and constraints can be arithmetic or symbolic (Verfaillie et al., 2010).

EOS can also be framed as an orienteering problem (Oza et al., 2008; Vansteenwegen et al., 2011) because it is a selective traveling salesman problem (TSP) where agents are required to visit as many checkpoints as possible within a time frame, each associated with a weight. The target images can be assigned weights, and orbital or subsystem constraints set up to represent travel times between images and the objective is to maximize the weighted sum. The prize collecting TSP not only minimizes travel time but also penalizes for unvisited cities. It has been modified for scheduling stereo images using scanning satellites, a convex value function and the Russian doll approach to solve nested sub-problems (Benoist and Rottembourg, 2004). However, the interplay between orbital access to required images by fast-moving LEO satellites and ACS-dependent slewing times causes the time for the traveling salesman to go from one image to another to be highly dependent on the absolute time either image is accessed. Variable travel times further add to TSP solution complexity.

While dynamic scheduling across the full state space is known to generate astronomical number of paths, unsolvable in polynomial time, branch and bound like approximations to the dynamic programming (DP) approach have demonstrated a practically sub-optimal resolution of the NP-hard non-restricted problem (Lemaître et al., 2002). DP has also been applied to the dynamic scheduling

Table 1  
Problem definition.

I	Full set of images (latitude, longitude, altitude) to be observed
i	Image number $\in I$ , the set of all images left to be observed
T	Time horizon within which objective to be maximized
t	Time step $\epsilon T$
S	Full set of N satellites in the constellation
s	Single satellite $\in S$ , represented by Keplerian elements [Semi major axis (SMA), eccentricity, inclination, argument of perigee, right ascension of ascending node (RAAN), mean anomaly]
FOV	Field of view of the imaging sensor on s
FOR	Field of regard of the imaging sensor on s, constrained by $\theta_{\max}$
$\theta$	Off-nadir pointing angle of s, variable with t due to 3-axis ACS
$\theta_{\max}$	Maximum off-nadir pointing angle of s allowed for a feasible image
$\Lambda(\theta)$	Spatial distortion of image i when taken at off-nadir angle $\theta$
Az	Azimuth of s at any time t
El	Elevation of s at any time t (isoformic with altitude and $\theta$ )
$\Omega(s)$	Schedule of [Az,El] per s, to be executed by s over T
C	Rotation matrix between desired and estimated orientations
$\tau$	Control moment on rotor wheels
$M_c$	Control moment in satellite body directions
$T_{\max}$	Maximum torque per rotor
IN	Moment of Inertia of the satellite

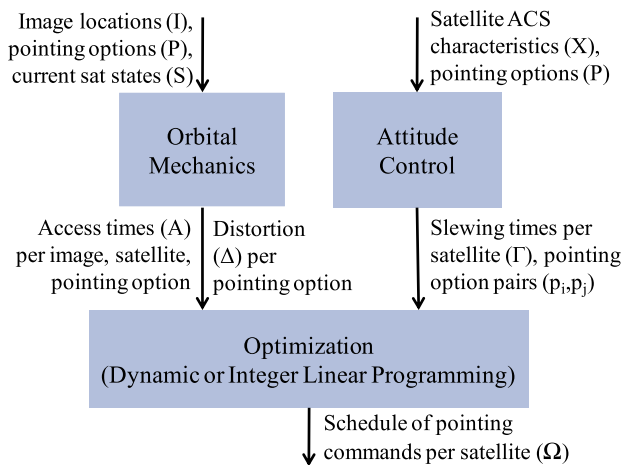


Fig. 2. Optimization of spacecraft attitude control schedules for maximizing required images covered, modularized into two subsystem components and one optimizer. The variables are defined in Table 1 or Sections 2.1 and 2.2.

problem after it is decomposed into several simple, static problems defined by a rolling horizon, which are automatically appended with waiting tasks, as current tasks are completed (Dishan et al., 2013). However, the ACS system was modeled at linear swing speeds and no validation of optimality is presented for the proposed heuristic algorithm. DP has been proposed for onboard scheduling of images to observe, within a reasoning horizon, power and memory (Damiani et al., 2005). The satellites are provided their image superset by ground control, also determined by DP applied to their orbits, priority image positions and downlink opportunities. However, the study does not account for ACS capability, time, energy or the coordination across satellites over the reasoning horizon.

Our approach to finding the most optimal schedule for controlling satellite attitudes within a constellation, for maximizing images observed, as a function of satellite

orbits, satellite characteristics, ACS subsystem specifications, angular and field of view constraints, is summarized in Fig. 2. Two satellite subsystems (OM and ACS) are modularly separated from the optimization algorithm, such that they simulate independently but provide parallel inputs into the optimizer. Dynamic programming is chosen as the algorithm for optimization, with custom heuristics to make the search space tractable, followed by MILP to assess solution optimality. To allow decoupling between the subsystems and for the purpose of representative simulation, the sensor attitude is forced to take one of a finite number of pointing options ( $p \in P$ ), which will be shown to span the allowable pointing space. The subsystems and their inputs/outputs will be described in the sections below. The presented framework allows for the addition of more subsystems and constraints, while using the same algorithm for optimization. For example, ground stations may be added as additional ground targets to the orbits module and the optimization would add downlink access to the schedule, as focused upon in (Arkali et al., 2008; Jian and Cheng, 2008; Kennedy and Cahoy, 2015; Spangelo and Cutler, 2012). Power and Data Handling can be added as separate subsystems, which would add more constraints, as inputs into the optimization module. The developed algorithm is expected to run on the ground using estimates of  $s \in S$ , as informed by orbit determination, and images required to be seen ( $i \in I$ ) within specified time bounds, as informed by ground experts of automated ground-based predictive algorithms.

### 2.1. Orbital mechanics (OM)

The OM module is responsible for computing coverage and access. It assumes the knowledge of all satellites' Keplerian elements and current states, defined in Table 1, as downlinked from the satellites' GPS systems or tracked

by ground-based systems, followed by orbit determination. The constellation is assumed to be optimized via regular tools (D’Errico, 2012; LeMoigne et al., 2017) or flying Ad-Hoc, and outside the scope of this paper. The orbits for the known satellites are simulated using MATLAB-controlled AGI’s Systems Tool Kit (STK) via the Microsoft Connect language. A satellite’s field of regard (FOR) is defined as the area that its sensor *can* see at any instant in time. Simplistically put, it is dependent on the sensor’s maximum off-nadir angle ( $\theta_{\max}$ ), as defined by science and technical requirements. The FOR of any satellite is assumed to be a cone and discretized into multiple, densely packed conical fields of view (FOV) corresponding to the FOVs of the satellite sensors and representing the full set of pointing options  $P$ . The algorithm used for discretization is 2D circle packing in a circle (Stephenson, 2005), for which analytical solutions<sup>1</sup> as well as online solvers<sup>2</sup> are readily available. The circular FOR spot is discretized into several concentric layers of FOV spots as a function of desired density, as seen in Fig. 3 (right). Each pointing option  $p \in P$  is characterized by  $[Az, El]$ , the azimuth and elevation of the satellite required to achieve that pointing direction.  $[Az, El]$  can be easily computed from orbital geometry and the location of the FOV spot from the FOR’s center. A FOR discretized into 19 or 66 pointing options is shown in Fig. 3 (left or right), with an FOV of  $15^\circ$  and  $3.6^\circ$  respectively. A sensor is allowed to point in only one of these 19 or 66 directions at any instant in our proposed simulation. The swath at any given time is a function of the sensor FOV, orbital altitude and off-nadir pointing  $\theta$ .

The lower limit of pointing options is the ratio of the solid angles of FOR and FOV. In theory, there is no upper limit because infinite options represent seamless pointing. However, since the ACS module will slew and stabilize at any given option, a large number of pointing options with no new images to capture will expend computational resources for no gain. The upper limit is thus set at the ratio of the solid angle FOR and solid angle subtended, at the sensor, by the Nyquist sampling distance between the required images, on the ground.

The orbits module provides constellation access  $A(t, i, p, s)$  as output.  $A(t, i, p, s)$  is a binary variable that = 1 if satellite ‘s’, when pointing in direction ‘p’, can access image ‘i’ at time ‘t’. Time is discretized into one-second steps for the case studies presented in this paper, since the resolution was sufficient for 20 kg satellite dynamics and required integration time for the instruments, but can easily be changed for other mission studies. The module can provide this data for any constellation structure and number of satellites. The module also provides the image spatial distortion  $\Delta(p)$ , i.e. the percentage increase of the ground footprint due to off-nadir pointing. The footprint is a func-

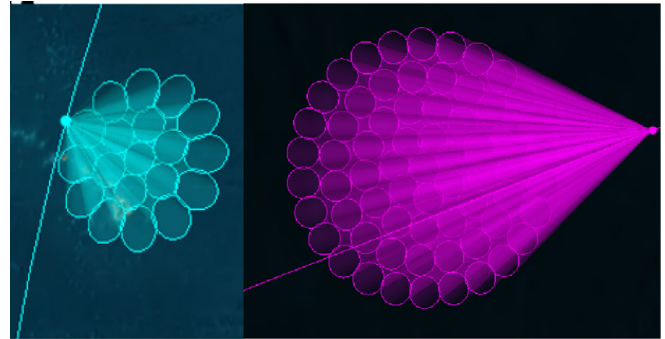


Fig. 3. A snapshot of the 19 discrete pointing options in three concentric layers for a satellite conical sensor in orbit (left). The simulation can accommodate different fields of view and higher densities or layers. For example, 66 discrete options packed into five layers (right).

tion of the orbit altitude and sensor FOV, and its distortion a further function of the off-nadir angle  $\theta$ . Nadir pointing causes 0% distortion.

## 2.2. Attitude control systems (ACS)

The ACS module is responsible for computing steering or slewing time between different attitude modes, and associated uncertainties. The decoupling of the ACS from the optimization routine allows for the independent formulation and numerical solution of the attitude problem. In general, the described ACS can ingest any given satellite mass, moments of inertia, sensors and actuator specifications to achieve a calculation of the time required between successive reorientations of the satellite pointing vector. The result is two  $n(P) \times n(P)$  matrices giving the times and energy required to orient between any two sets of attitude profiles or pointing options (with 0 on the diagonal), where  $n(X)$  is the cardinality of set  $X$ .

For representation in this paper, we simplify the satellite to an evenly mass-distributed cube composed of three sensors (sun sensor, magnetometer, and rate gyro) with four actuator wheels, of which three are aligned along the orthogonal axes and one aligned diagonally. The actuator wheels are commercially available from Blue Canyon Technologies (product RWP500), chosen as a specialized mid-range attitude control solution for Cubesats. The volume is listed at  $10.8 \times 10.8 \times 10.8$  cc with a mass of 0.75 kg, and momentum and torque specifications viable for our requirements. The wheels exhibit low jitter, operate at 200 Hz, exhibit extremely accurate torque control (Daniel, 2016) and have been proposed for interplanetary small sat missions. The satellite and ACS specifications used as a representative example in this paper are listed in Table 2, but the control strategies described are generic for any satellite actively controlled with wheels with some modifications required if other actuators are used. For example, for thrusters, the ACS analysis would need to reflect torque generated through thruster impulses, as well as additional considerations for fuel mass and rise and fall

<sup>1</sup> [https://en.wikipedia.org/wiki/Circle\\_packing\\_in\\_a\\_circle](https://en.wikipedia.org/wiki/Circle_packing_in_a_circle).

<sup>2</sup> <http://www.engineeringtoolbox.com/smaller-circles-in-larger-circle-d-1849.html>.

Table 2  
Satellite, sensor, and actuator properties. Rotor properties taken from the RWP500 from Blue Canyon Technologies.

Component	Property	Value
Satellite	Mass	20 kg
	Inertial matrix	$\begin{bmatrix} 0.245 & 0 & 0 \\ 0 & 0.245 & 0 \\ 0 & 0 & 0.245 \end{bmatrix} \text{ kg} \cdot \text{m}^2$
Sensors	Sun sensor	$\sigma_{\text{pointing\_error}} = 0.1^\circ$
	Magnetometer	$\sigma_{\text{pointing\_error}} = 0.1^\circ$
	Rate Gyro	$\sigma_{\text{rate\_error}} = 0.1^\circ/\text{sec}$ bias = [0.01 -0.01 0.02] rad/s
Actuators	Number of rotors	4
	Rotor orientation (relative to spacecraft)	$\begin{bmatrix} 1 & 0 & 0 & \frac{1}{\sqrt{3}} \\ 0 & 1 & 0 & \frac{1}{\sqrt{3}} \\ 0 & 0 & 1 & \frac{1}{\sqrt{3}} \end{bmatrix}$
	Moment of Inertia per rotor	0.0005 kg · m <sup>2</sup>
	Maximum torque	0.025 Nm
	Maximum angular momentum	0.5 Nms

times of thruster operation. Control moment gyros, while able to provide more torque, will require more complex control due to the changing orientation of the actuator spin axes relative to the satellite. Table 2 takes into account the resource constraints of a 20 kg satellite. For example, RWP500's specifications confirm the availability of 23 W for peak power, 6 W for power at full momentum and 3 W at half momentum. An approximate inertia of  $\sim 0.245 \text{ kg} \cdot \text{m}^2$  is computed for a 20 kg satellite and  $0.0005 \text{ kg} \cdot \text{m}^2$  for the wheels. The maximum allowed torque of 0.025 Nm and 23 W peak power restricts the rotor spin rate to 245 rad/s, which allows 0.5 rad/s of maximum angular velocity for the satellite (last row of Table 2). Given the specifications of the satellite and its ACS, our automated framework computes the available thresholds and restricts the control phases (see Fig. 5) accordingly.

The primary requirement for attitude control is to reorient the satellite to its necessary pointing direction as quickly as possible. We found the bang-bang control method to be an efficient solution, with a PD control (or PID in the case of disturbances) when the satellite neared its desired attitude for smoother operation. Two general strategies were implemented:

- (1) *Full alignment*, where all three axes are aligned according to a specified orientation. This mode is suitable in conditions where parallel requirements must be met, such as directing the solar panels towards the sun, while achieving pointing requirements.
- (2) *Fast alignment*, where only the camera sensor needs to be oriented towards a certain direction. This mode is suitable for normal operation since it consumes less power and is in general faster than full alignment.

The ACS maneuver is assumed completed when the satellite is able to keep to within  $0.2^\circ$  of its desired pointing direction. As will be seen in Section 2.5, decoupling between ACS and optimization for the described control algorithm, is possible for a minimum FOV of  $3.6^\circ$  and a  $\sim 5\%$  error is assumed reasonable for Cubesat mission requirements. An error of  $0.2^\circ$  at a 710 km orbit corresponds to 2.5 km on the ground, which is similar in magnitude to orbit determination and control errors for Cubesats.

The general attitude control algorithm can be visualized as per the block diagram shown in Fig. 4. The dynamics are propagated using the Euler equations in quaternion space, in which the sensors measure as vectors and angular rates. A Kalman filter is then used to estimate the attitude and rotation rate of the satellite assuming a certain noise level and bias of the sensors (Lefferts et al., 1982; Markley, 2003; Wen and Kreutz-Delgado, 1991). This feeds into the control law, which consists of a bang-bang control method (Wertz, 2012) for large angular displacements and two levels of PD control for small angles (Trawny and Roumeliotis, 2005; Wie and Lu, 1995). Commands are then relayed to the actuators taking into account saturation and maximum torque levels. The feedback loop is completed by incorporating the moments generated by the rotors back into the dynamic simulation. We assume no external moments for the current investigation, as the purpose of our study is to find an approximate minimum time to complete successive reorientation maneuvers, but the modularity of the ACS formulation allows for its incorporation, if required.

The estimation portion of the system consists of a Kalman filter which takes vector measurements provided by a sun sensor and a magnetometer, as well as angular rates through a rate gyro. Employing a first order quaternion

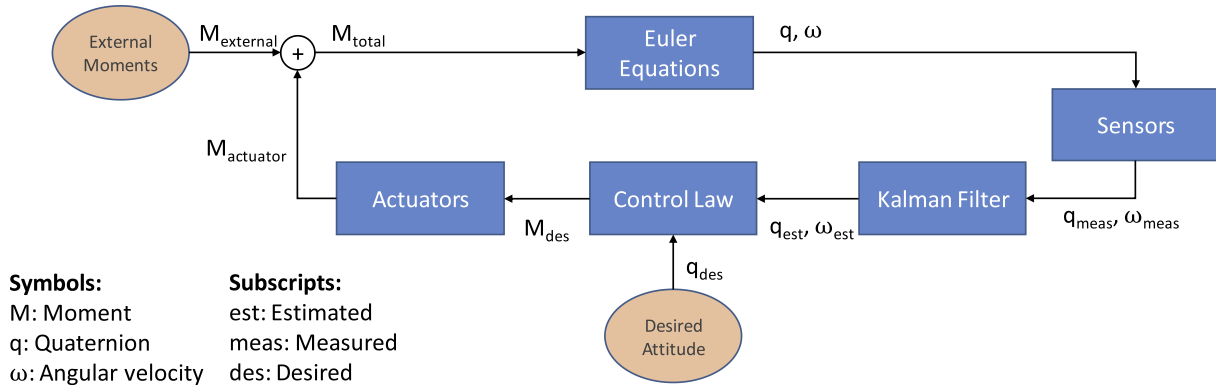


Fig. 4. Simplified block diagram of attitude control algorithm.

integrator and a zeroth order angular velocity propagator to estimate the relevant attitude and rates (Lefferts et al., 1982; Trawny and Roumeliotis, 2005) according to the specifications listed within Table 2, results in a pointing angle uncertainty of roughly  $\sigma_{\text{error}} = 0.03^\circ$ . The system is also able to estimate the bias of the sensors within a few seconds of initial operation. Initialization assumes that the spacecraft is stable (not tumbling) and the estimate and actual pointing attitudes are roughly coincident.

The control law is segmented into two portions: a bang-bang and a PD control component. The former orients the satellite quickly into the desired orientation at the maximum available torque provided by the wheels, while the latter provides optimal smooth control near the desired attitude. The pseudocode for the control law is shown in Fig. 6, where  $K_p$  is the Proportional control constant,  $K_d$  the differential control constant,  $v_{\text{eig}}$  is the vector of rotation,  $A_{\text{rot}}$  is the rotor orientation matrix,  $\theta$  is the angle of rotation,  $u$  the control signal in satellite body directions,  $\omega_{\text{est}}$  are the estimated angular rates, trace is the Trace func-

tion of a matrix, max is the Maximum function, ‘+’ is Pseudoinverse and sgn is the Sign function. The algorithm first calculates the rotation matrix, Euler axis, and Euler angle, before assessing if the satellite is currently within the bounds of bang-bang or PD control (Wertz, 2012). The control moment is calculated accordingly and limited by the maximum rotor torque available. For angles below  $10^\circ$ , the small angle approximation can be used to linearize the problem. Control constants are chosen such that the system exhibits a natural frequency of 0.5 and a damping ratio 1.5. As shown by the attitude phase in Fig. 5 for a specific scenario, the Euler angle is initially approximately 1.8 radians from desired with no initial spacecraft angular velocity. After engaging the rotors to maximum torque, the spacecraft starts slewing towards its targeted attitude, with the rotors engaging in the opposite direction approximately halfway through the entire maneuver, along the optimal second order bang-bang switching line. When the measured attitude is within  $10^\circ$  of its desired attitude, the algorithm switches to a PD control methodology. Finally, when

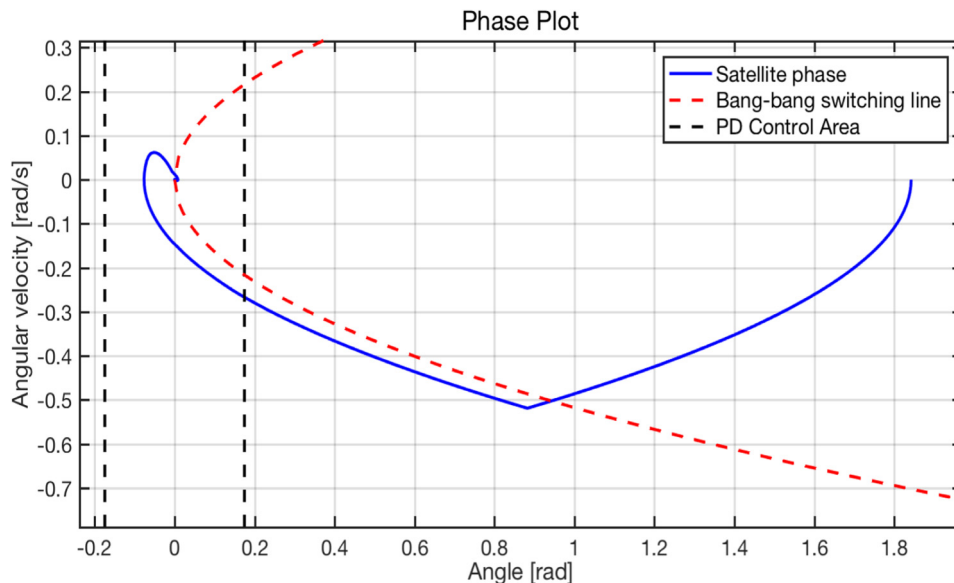


Fig. 5. Phase plot of the bang-bang and PD control modes. Note that the satellite moves along the optimal bang-bang control line until it reaches the PD control area, in which it then smoothly approaches the origin.



```

1  C = Aest* (Ades)T //Find error rotation matrix
2  veig = real (eig (C)) //calculate axis of rotation
3  θ = cos-1 (1/2*trace (C)-1) //calculate rotation angle

4  if abs (θ) < 10°
5      u = -Kp* [θsat] -Kd [ωest] //PD control law
6      τ = (Arot)+u //Desired moment on wheels
7      if τ > Tmax
8          Mc = TmaxArot*τ/max (τ) //Maximum torque check
9      else
10         Mc = u
11 else
12     τ = (Arot)+veig //Desired torque per wheel
13     u = Tmax*sgn (θ+1/2*ωest | ωest | ) Arot*τ/max (τ)
// limit to maximum of wheel
14     Mc = Tmax*sgn (θ+1/ (2* | u | ) *I*ωest | ωest | ) Arot*τ/max (τ)
// Control moment

```

Fig. 6. Control law pseudocode, where the variables are defined in the text or Table 1.

the satellite achieves a bounded  $0.2^\circ$  within its desired attitude, the algorithm will assume that the maneuver has successfully terminated.

The current implementation gives an  $n(P) \times n(P)$  array of the required time to switch from one attitude to another. Furthermore, it is desirable for the triangle inequality between attitude maneuvers to be held, such that:

$$P_{i,j} + P_{j,k} \geq P_{i,k} \quad (1)$$

where  $P_{i,j}$  is the time required to change the attitude from position  $i$  to  $j$ . Failure to uphold this condition leads to complications within the optimization algorithm (Section 2.3) such that there might be shorter paths towards a particular attitude profile through an intermediate pro-

file, rather than transitioning directly. In the current formulation, the triangle inequality holds for a pure bang-bang/PD control architecture, but it can be broken by simply switching between control laws during arbitrary phases of the attitude maneuver. This is due in part to what we credit as a “successful” attitude transition (currently set to a bound of  $0.2^\circ$ ), and changes to the control law during a maneuver may cause overshoot or instability in the final result. This suggests that a LQR or model predictive control methodology might fare better than PD in these cases. The outputs for the fast re-orientation for all combinations of pointing options in Fig. 3, are shown as a color chart in Fig. 7. There are three concentric circles of equal off-nadir pointing for the case with 19 pointing options (1, 6, 12

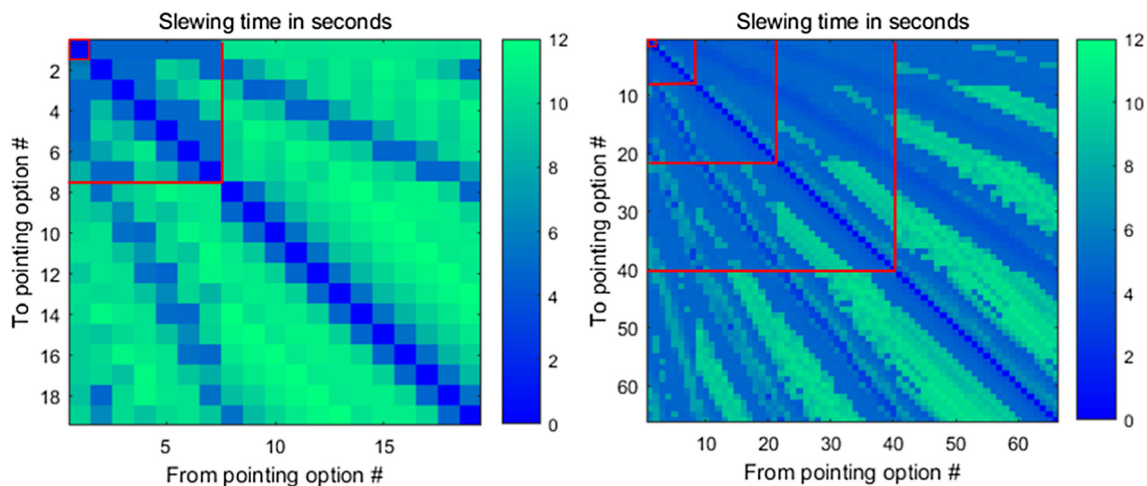


Fig. 7. Time required to slew among the 19 (left) and 66 (right) pointing options shown in Fig. 3, using the fast slewing algorithm. The maximum off-nadir angles are  $28^\circ$  (left) and  $16.2^\circ$  (right). The red lines represent the transition between concentric layers. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

options each as indicated by the red lines) and five concentric circles in the case with 66 options (1, 7, 13, 19, 26 each). Fig. 8 shows a visualization of re-orienting from option #3 ( $Az = 60^\circ$ ,  $EI = 75^\circ$ ) to option #18 ( $Az = 315^\circ$ ,  $EI = 62^\circ$ ), as shown in Fig. 3-left. The corresponding control moments, calculated as per Fig. 6, to effect this re-orientation are shown in Fig. 9.

Although the above formulation gives a good estimate of a Cubesat ACS to rapidly slew and capture images, it does not take into account disturbance moments (e.g. gravity gradient, solar radiation, aerodynamic torques), internal forces (e.g. wheel friction, damping) and maneuvers that might cause the system to saturate. In the event of saturation, torque rods or gravity gradient maneuvers to dump the angular momentum buildup are necessary to ensure continued operation and are assumed available. Additional operational considerations include solar panel

alignment with the sun vector for power, diurnal dependence of sensors (e.g. sun sensor), and downlink requirements with ground stations. For increased estimation and control capability, a star tracker and magnetic torquers or control moment gyros may be deployed and the proposed framework allows for such changes quite modularly. While the presented ACS is appropriate for the framework in Fig. 2, ACS optimality can be improved as well as narrower FOVs simulated if the attitude control process is incorporated within the optimization routine, at the cost of computational efficiency.

### 2.3. Optimization by dynamic programming

Our proposed algorithm optimizes the constellation imaging schedule by using two levels of heuristics, applied to select the ‘best’ path in the time-pointing option space

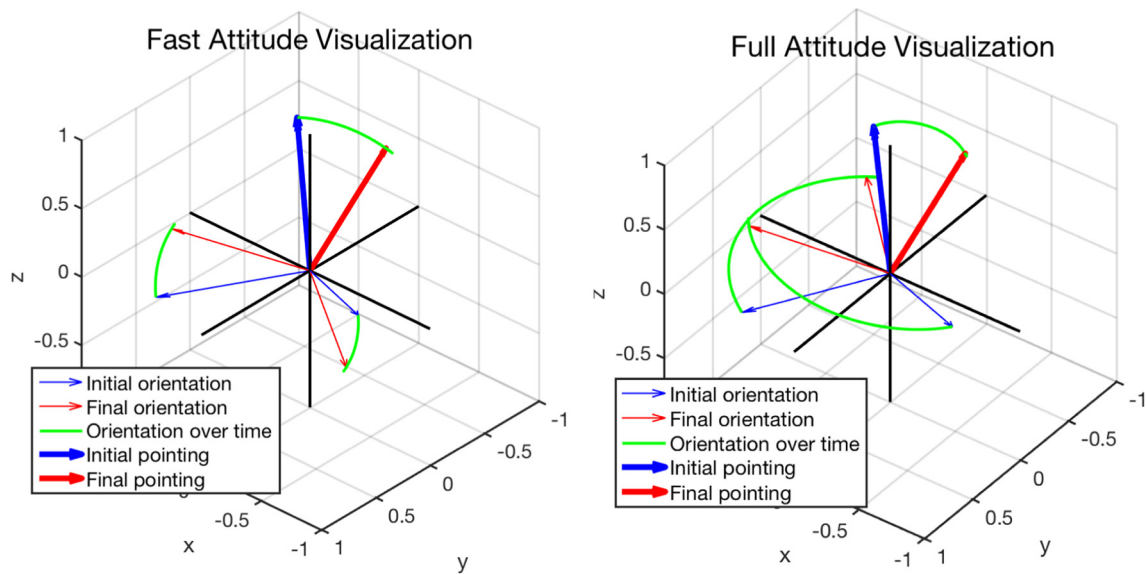


Fig. 8. Attitude visualization between the fast method and the full method. Note that the final pointing directions are similar, but the general orientations are different, with the full attitude method rotating significantly more to align all 3 axes.

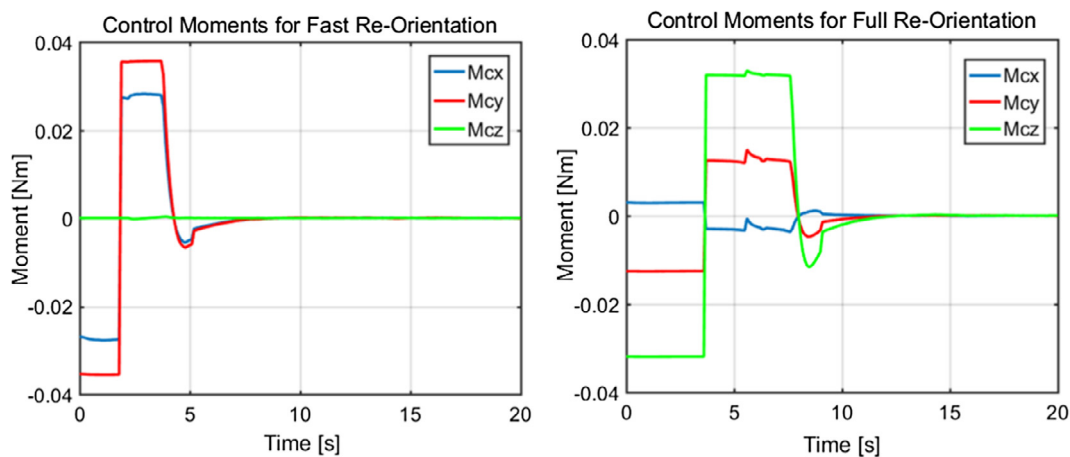


Fig. 9. Control moments per axes for fast vs. full attitude modes, with the full attitude control mode requiring more torque and time to reorient. Note the limiting available torque per axes due to wheel constraints.

per satellite. The innovative aspects are the speed of execution due to usage of dynamic programming, linear time and space complexity, and the ability to schedule for any flexible time horizon. Fig. 10 describes the algorithm in brief and Fig. 11 shows a cartoon of the option space and an example ‘best’ path. For every satellite, the state of pointing  $p$  at time  $t$  can be represented as a node – circle in Fig. 11. Depending on the locations of the images to be observed, node  $[p,t]$  is capable of observing an image (filled blue circle) or not (hollow white circle). A feasible path in this state space is one that connects the nodes, such that the time to slew from one node (say,  $j$ ) to another (say,  $i$ ) satisfies the slewing and stabilization time matrix  $\Gamma(p_i, p_j)$  as  $t_j = t_i - \Gamma(p_i, p_j)$ . Any node will have up to  $n(P)$  feasible paths leading to it. Fig. 11 shows three such paths leading to node  $[p = 5, t + 4]$ . The proposed DP algorithm starts at  $t = 0$ , and goes over every pointing option and its feasible previous options. It selects and stores the ‘best’ path/s at any node, as it progresses through the simulation by incrementing time. Since any node considers feasible previous nodes, one can simply add the path (and heuristic values) since the previous node, to the stored path (and heuristic values), without the need of any re-calculation.

To ensure that memoization per node does not become astronomical, the dominated paths are eliminated in two consecutive steps of heuristics.

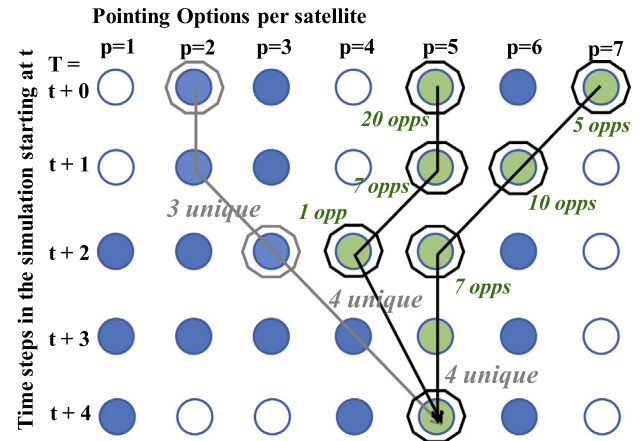


Fig. 11. Cartoon of a very simple example of the proposed algorithm.

First, all  $n(P)$  paths are compared in terms of the weighted number of unique images seen. Unique images seen by the three paths in Fig. 11 are enclosed in a hollow decagon, other images in the path will not be counted. The weights obviously depend on the mission or application. An example of weights is the image spatial resolution distortion, as computed in the orbits module – Fig. 2. Unique images seen at maximum off-nadir pointing are less valuable than those seen at nadir pointing, so they can be

```

1 for every t in T:
2   for every pi in P:
3     //pick the best path/s among the n(P) possible leading to pi
4     for every pj in P:
5       check if path pj->pi is possible in Γ
6       if path possible,
7         update path at node [pi,pj] by adding:
8           new images seen at node [pj,pi] to images stored
9             at node [pj, t-Γ(pj,pi)]
10          [pj->pi] to path stored at node [pj, t-Γ(pj,pi)]
11
12          compute heuristic #1 and heuristic #2 for this new
13            path at node [pi,pj]
14          record running max(heuristic #1) across all paths
15            leading to pi
16
17          delete paths among n(P),
18            where heuristic #1 < max(heuristic #1)
19          if paths have equal heuristic #1:
20            record running min(heuristic #2)
21            delete paths where heuristic #2 >
22              min(heuristic #2)
23          store leftover paths (max heuristic#1,
24            min heuristic #2) at node [pi,pj]
25 Optimum path till t, is one with max heuristic#1 across all pi
26
27 heuristic #1: weighted sum of unique images in path ending at node
28               [p,t]
29 heuristic #2: weighted opportunities after t, to see images in path
30               ending at node [p,t]

```

Fig. 10. High level pseudocode of the proposed dynamic programming algorithm, per satellite, where  $n(P)$  is the cardinality of  $P$ . The modified algorithms for multiple satellites are discussed in Section 3.2.

weighted at  $(1-X\%)$  where  $X\%$  is the fraction of GSD increase. Another example of weights is the continuous time available to observe any unique image. Unique images that can be observed for a greater time window are more valuable, due to increased exposure and signal to noise ratio, so they can be weighted at  $Y$ , where  $Y$  is the number of continuous observation seconds. We found that a large fraction of the feasible paths are equally good if only one heuristic was used, therefore another was added to eliminate further among them.

Second, the leftover paths were compared in terms of the minimum weighted number of opportunities left to see the image at  $[p,t]$ . For example, two of three shown paths in Fig. 11 see the maximum number of unique images (four) and are shaded black. The path eliminated in step 1 is shown in grey. Among the leftover paths, the one on the left has 20, 7 and 1 opportunity remaining to see its unique images later, and the one on the right has 5, 10 and 7 opportunities remaining to see its unique images later. The DP algorithm eliminates the path on the right because the path on the left has the smallest opportunity remaining to see one its images. The second heuristic favors the less frequently seen images. Since the algorithm simulates in consecutive time steps and greedily captures the best possible path until then, the optimum schedule over any time period is available by stopping the simulation at any time step.

As the number of satellites increase, they serve as a multiplicative factor to the pointing options i.e. increase the horizontal axis in Fig. 11. The algorithm in Fig. 10 is expected to run as a separate thread for every satellite, however heuristic information exchange is necessary across the satellite loops to allow for coordinated observations. The reasoning behind and algorithms for this exchange are discussed in depth in Section 3.2. The proposed algorithm's time complexity is  $O(n(S) \times n(T) \times n(P)^2)$ , where  $n(X)$  is the cardinality of set  $X$ . Thus, the complexity of the algorithm is linear with the size of the constellation and planning time horizon, quadratic with the discrete pointing space and independent of the number of images requested. Since the pointing options are strictly bounded and not expected to increase much compared to infrastructure, images or mission duration, the proposed solution can be assumed to be near-linear in time. The algorithm's space complexity is  $O(n(I) \times n(S) \times n(P))$ , for the worst case scenario when all unique images have been seen by all feasible paths. Even so, the memoization is linear and easily handled by modern-day computers.

#### 2.4. Optimization by mixed integer linear programming

Our problem can be formulated exactly as a Mixed Integer Linear Program (MILP). Such a formulation can be computationally challenging to solve, but has the advantage of, in contrast to heuristic methods, the ability to provide a guarantee of optimality with a solution, or bounds on how far the solution is from optimum. As discussed in

the results below, while MILP as formulated here will be too slow for rapid response applications, it can provide a useful offline check on the quality of solution provided by the DP algorithm, especially upon availability of the DP solution as an initial condition. In addition, the capability to solve mixed integer optimization approaches has advanced rapidly in recent years, making more problems of this type tractable. For example, (Bertsimas et al., 2016) outlines how software and hardware improvements combined have enabled a factor of 450 billion speedup in the period 1991 to 2015 for solving MILP problems.

To formulate the problem as an MILP, let  $o_{t,p,s}$  be a binary variable which is 1 if satellite  $s$ , when pointing in direction  $p$ , captures a required image at time  $t$ , for any given schedule. Image distortion  $\Delta[p]$  and access  $A[t,i,p,s]$  are obtained as outputs from the orbits module (Section 2.1). If  $S_{max}$  is the maximum number of satellites available,  $P_{max}$  the total number of pointing options available to any satellite,  $T_{max}$  the time window of the simulation in seconds and  $I_{max}$  is the total number of images that can be seen, let value be defined as the weighted sum of unique images seen at any given time instant.

$$v_t = \sum_{s=1}^{S_{max}} \sum_{p=1}^{P_{max}} o_{t,p,s} * \Delta[p] * a_{t,p,s} \quad (2)$$

$$a_{t,p,s} = \sum_{i=1}^{I_{max}} A[t,i,p,s] \quad \forall s \in S, p \in P, t \in H \quad (3)$$

$a_{t,p,s}$  is the number of images that can be seen, from an orbital feasibility perspective, given  $s$ ,  $t$  and  $p$ . If the pointing options are discretized within the limits described,  $a_{t,p,s}$  should be either 0 or 1. The sets  $S$ ,  $P$  and  $H$  are inclusive ranges where  $S = [1, S_{max}]$ ,  $P = [1, P_{max}]$  and  $H = [1, T_{max}]$ .

$$\text{maximize} \sum_{t=1}^{T_{max}} v_t \quad (4)$$

$$\sum_{p=1}^{P_{max}} o_{t,p,s} \leq 1 \quad \forall s \in S, t \in H \quad (5)$$

$$\sum_{s=1}^{S_{max}} \sum_{p=1}^{P_{max}} \sum_{t=1}^{T_{max}} o_{t,p,s} * A[t,i,p,s] \leq 1 \quad \forall i \in I \quad (6)$$

$$o_{t+k,p_i,s} + o_{t,p_j,s} \leq 1 \quad (7)$$

$$\forall k \in [1, \Gamma(p_i, p_j)], s \in S, p \in P, t \in H$$

The MILP objective is to maximize the scalar in Eq. (4), i.e. the total number of unique images collected over the simulation. The constraints are enumerated in the following equations. Eq. (5) ensures that any satellite can point only in one pointing direction at a time. Eq. (6) forces any image to be captured at most once, so that only unique images are counted in the value function  $v_t$ . Eq. (7) ensures that there is sufficient time for the satellite to slew between two pointing options, between capturing images. If any  $o_{t,p_i,s} = 1$  (for a particular pointing option  $p_i$ ), then this con-

straint confirms that no  $o_{t,p_j,s}$  (for another pointing option  $p_j$ ) can be = 1 within at least  $k$  seconds, where  $k$  is the time required to slew and stabilize from  $p_i$  to  $p_j$ . The time constraint ( $\Gamma$ ) is an asymmetric 2D matrix, obtained as outputs from the ACS module. While Eq. (7) is a group of at least  $P_{max}^2$  equations and at most  $\max(\Gamma) * P_{max}^2$  equations per satellite and time step, it can be simplified by summing across  $p$  for any 't + k' because there can be only one on zero  $o(s, p, t + k)$  by Eq. (5). This reduces Eq. (7) to Eq. (8), which is a set of  $\max(\Gamma)$  equations per P, S and H.

$$\sum_{p_i=1}^{P_{max}} o_{t+k,p_i,s} + o_{t,p_j,s} \leq 1 \quad (8)$$

$$\forall k \in [1, \Gamma(p_i, p_j)], s \in S, p \in P, t \in H$$

## 2.5. Case studies

We present two case studies, based on narrow FOV, space sensors, to demonstrate the utility of the above algorithms to agile, small satellite constellations. The first is derived from the Operational Land Imager (OLI) flying onboard heritage Landsat satellite, the longest-running enterprise for acquisition of satellite imagery of Earth. The OLI is a pushbroom sensor with a 185 km swath and 30 m ground sample distance (GSD). Since Landsat-8 is in a sun-synchronous 710 km altitude, 98.2° inclination orbit, it takes approximately 24 s to traverse 185 km in the along track direction. The OLI (Knight and Kvaran, 2014) takes pushbroom slices at 236 Hz and integrates these into a single, square image tile over 30 s. The extra time over 24 s is to account for data buffering and scene-to-scene overlap. For the purpose of simulation, the OLI can be assumed to have a ~15° conical FOV since the ground spot will sit cleanly within the square image tile. The simulation also assumes that a single image capture takes less than a second, in keeping with the fast slice rate.

OLI has a 16-day repeat and a 14-day revisit frequency. Orbital mechanics analysis has shown that at least 14 simultaneous Landsat satellites with OLIs will be needed to cover the global images, everyday. Increasing the FOV three times reduces the need to four satellites, however increasing the FOV will add more spatial distortion to the edge pixels and complex corrections for bi-directional reflectance distribution functions (BRDF) to the image. Instead, if OLI-like narrow FOV sensors on a small satellite constellation were commanded using our proposed algorithms to slew their sensor over time, they would be able to coordinately cover the required Landsat images much faster without compromising on spatial resolution or BRDF effects.

The locations of the Landsat images are publicly available through the United States Geological Survey's<sup>3</sup> Worldwide Reference System (WRS), which is a global

notation used in cataloging Landsat data. We imported the locations of 16,896 land and coastal images on the WRS-2 grid for this case study. The minimum elevation angle at the furthest pixel measured off-nadir was limited to 55° (view zenith 35°) to prevent BRDF complexities, therefore the maximum off-nadir pointing angle limited to 28°. For a 15° FOV sensor, a minimum of three concentric circles of 19 pointing options were needed to cover the FOR. Fig. 3-left's example shows the arrangement, and Fig. 7-left shows the slew-and-stabilize times computed for the option transitions. We assumed a 20 kg small spacecraft with specifications, assumptions and generalizability described in Section 2.2 for all simulations presented. The heuristic used was to maximize the number of unique images seen over any given time horizon. The captured images in any schedule were weighted by [1-X] where X = [0 0.08 0.36] for the inner, middle and outer concentric circles of pointing options, respectively. As explained in Section 2.3, X represents the spatial distortion caused by off-nadir pointing. For example, the GSD of the middle layer is 8% larger than the inner layer.

The second case study is derived from new instruments being developed for high resolution imaging of coastal aquatic systems and their response to climate and anthropogenically-driven forcings. The Canadian Space Agency is building a Coastal Ocean Colour Imager (COCI) (Bergeron and Craig, 2017) based on the heritage Hyperspectral Imager for the Coastal Ocean (HICO) flown onboard the HypSPiRI satellite. COCI is expected to fly at an altitude of 675 km with a swath of 240 km, ~20° FOV. The agile ability of small satellites to point at coasts, while imaging, increases their coverage and also provides more multi-angular image data needed for computing ocean color, atmospheric properties, phytoplankton concentration and many other products. Imaging of coastal targets over extended time periods also enables space-based Fluid Lensing. Fluid Lensing (Chirayath and Earle, 2016) is a novel mechanism where in high-resolution, video data of shallow coral reefs is processed and underwater reefs imaged by exploiting time-varying optical lensing events caused by refractive distortions from travelling surface waves. From the ACS perspective, video data can be collected from space by tracking one image at a time for a long duration, with a stable, high-frame rate imager.

The United Nations Environment Program (UNEP)'s World Conservation Monitoring Centre provided NASA with the most comprehensive dataset of warm-water coral reefs<sup>4</sup> up to 2010, for the Fluid Lensing project. The dataset contains the global distribution of shallow, underwater coral reefs in tropical and subtropical regions, at ~1 km resolution. We uniformly sampled the dataset such that 22,718 reef locations were used as targets of interest in the coastal case study, as shown in Fig. 12 (red). 13,172

<sup>3</sup> WRS-2 grid points – <https://landsat.usgs.gov/what-worldwide-reference-system-wrs>.

<sup>4</sup> UNEP Coral Reef database – <http://data.unep-wcmc.org/datasets/1>.

images of this dataset can be accessed (blue) within the field of regard (FOR) of the reference sensor at the reference orbit, described in the next paragraph. The heuristic used was to maximize the number of unique images seen over any given time horizon, where each image was weighted by the continuous time available to capture it. Varying results for varying heuristics relevant to extended time observation will be shown to demonstrate the flexibility of the DP for many imaging mission objectives. Spatial distortion was not used in the objective function, instead we imposed a 10% GSD distortion limit on the set up. Both dwell time and distortion can be factored into the heuristic as weights, if required.

We assume a 675 km altitude (COCI reference mission) and 35° inclination for the reference orbit (corresponding to 35.8% coverage of the UNEP images). Since all the warm-water corals are found within 35° parallels of latitude, their coverage over a 12-h simulation with the reference sensor reduces when observed by increasing inclinations. Only 17.6% are observed at perfectly polar inclination and 18.61% by sun-synchronous orbits. The instrument for Fluid Lensing, tested so far on drones only, has a very narrow FOV of less than 1°. Decoupling the ACS module from the optimization module (Fig. 2) enforces a minimum FOV of  $\sim 3.6^\circ$  for the sensor because the time required to traverse the FOV on the ground (at 6.7 km/s of ground velocity) must be more than the time required to slew from one pointing option to its adjacent. If not, the satellite will not have enough time to slew and image an off-nadir image before it loses orbital access to that image.

A  $3.6^\circ$  FOV for the coastal case study is representative of a narrow FOV and can be easily increased, for example to  $20^\circ$  for COCI, without any loss in generality of the algorithmic implementation. The maximum off-nadir pointing angle is limited to  $\sim 16.5^\circ$  to limit the GSD spatial distortion at the edge pixels to within 10% of nadir-view pixels. Sixty six  $3.6^\circ$  FOV conical sensors can be packed into five concentric shells within the given FOR, in the arrangement shown in Fig. 3-right. Since  $\max(\Gamma)$  for the  $66 \times 66$   $\Gamma$  matrix is 6 s, a 40 km ground spot allows for enough slew time. Moreover, the FOV traversal time at maximum off-nadir is more than four seconds, therefore a one-second time resolution in the simulation is sufficient to register even distant accesses.

### 3. Results using dynamic programming

We applied the DP algorithm described in Section 2.3 to both case studies in Section 2.5, and found a large improvement in the number of unique images seen, compared to a static sensor. The case studies are different in that the Landsat case has globally distributed coarse images that need to be observed with a wider sensor, while the coral case has sparsely distributed but fine images that need to be observed with a narrow sensor. The Landsat case is a *rapid imaging mode* while the coastal case is an *extended*

*time imaging mode*, even if both are applied to global images. They are similar in that the goal is to maximize the required images seen in slightly different ways. We will present the proposed dynamic programming algorithm as, first, applied to a single satellite case and then, to a two satellite constellation in a manner generalizable for any satellite constellation. We highlight the sensitivity of results and methods constellation structure, therefore the possible limitations of DP applied to this problem. Finally, we apply the DP algorithm to the coral case study to show its versatility of application across targets of interest, sensor types and heuristics.

Since the algorithm runs in linear time and care was taken to minimize loops, all presented simulations ran within a couple of hours on a MATLAB environment installed in a Mac OS X version 10.10.5 with a 2.5 GHz processor and 16 GB of 1600 MHz memory. Code conversion from MATLAB to a low-level language like C++ ensures 10 to 100x speed up in runtime, because the rate determining steps are the loops in Fig. 10. In other words, scheduling two satellites for 12 h will take  $\sim 20$  min on a simple laptop, ensuring efficiency within available memory of ground processors. Section 4 will validate the results in this section and show better performance since MILP provides a better guarantee of optimality than a heuristic-based approach, at the cost of speed.

We acknowledge that our algorithm does not optimize for signal to noise ratio (SNR) and that image quality taken on rapidly steering Cubesats (weighing  $\sim 40$  kg) will be subpar compared to flagship instruments like the OLI, carried on larger platforms (weighing  $\sim$  metric ton). Constellations of small satellites are intended to serve as complements to flagship missions, not replacements (National Research Council, 2000). Our results below show the improvement of coverage from fixed sensors to body-steerable sensors, assuming the same instrument quality on both. Moreover, our algorithm allows the time step size or image capture time to be altered, based on the mission scenario, to allow for sufficient integration time to meet a threshold SNR. For example, in the rapid imaging case based on OLI, the time step is a second (Section 2.1) and the functional OLI takes a pushbroom image every 0.004 s (Section 2.5). The ACS algorithm ensures that the imaging vector is stable within  $0.2^\circ$  of its desired pointing direction (Section 2.2) for one full time step (1s) so that the image can be captured in a stable way, thereby allowing far more buffer over the OLI single-slice integration time (0.004 s). The capture geometry and time constraints can also be tailored to meet image overlap needs, for further improvement in achievable SNR.

#### 3.1. Single satellite case for rapid imaging

Over a full day's worth of simulation or 86,400 s in the Landsat case study, the FOR shown in Fig. 3-left is capable of seeing 14,723 of the 16,896 land and coastal images on the LandSat WRS-2 grid. Of these, 11,900 were seen using

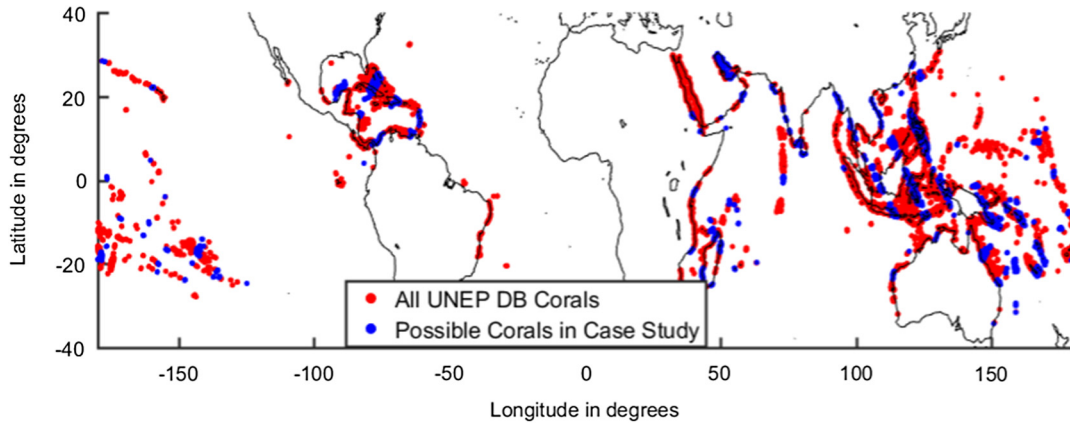


Fig. 12. Global distribution (red) of shallow, warm water coral locations and the fraction of them (blue) that can be seen within the field of regard of a 12-h simulation with two satellites in one orbital plane. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

the DP algorithm, using only unique images without weights as the heuristic in both levels of elimination. In comparison, a maximum of 4894 images were seen using the static, nadir viewing concept of operations and 3079 images using a whiskbroom or scanning approach, where the sensor swung from side to side at a predetermined rate. The proposed DP algorithm was able to cover 77.5% of the possible images, which comprised 87% of the total database. The second number (87%) can be improved by changing constellation orbits (beyond the scope of this paper) or by increasing the FOR (constrained by maximum slew angle). The first number (77.5%) indicates the effectiveness of the DP algorithm, which causes the steerable satellite to observe 2.5 times the number of images than the static satellite case and 3.86 times than the scanning satellite case.

Fig. 13 shows the first two minutes of the simulation and the selected schedule ( $\Omega$  as a black solid line) that samples across 18 of the 21 unique images possible to be observed.

Since pointing option #1 has no distortion, #2 to #7 have 8% distortion and #8 to #19 have 36% distortion, it is obvious from Fig. 13 that to meet the goal of maximizing non-weighted images observed, the schedule also prefers greater spatial distortion of those images. An image quality analysis over the full 24-h schedule showed that less than 6% of the seen images are nadir-viewing and more than 65% have maximum distortion (36% of nadir GSD). To correct for this, image distortion was added to the heuristics by weighting the seen images by a factor of  $(1-X\%)$  and opportunities left to see any image with  $(1+X\%)$ . If an image is seen more than once in a schedule, the capture with the least spatial distortion is recorded. The re-run of the same simulation with the new heuristics found an ‘optimum’ schedule where more images were observed with 0% and 8% GSD distortion instead of 36%. Fig. 14 shows the improvement in the distribution of image distortion. However, adding distortion minimization to the heuristic caused 11,418 unique images to be observed, instead of the previous 11,900. As before, the nadir-pointing approach was able to observe only 32% of the possible

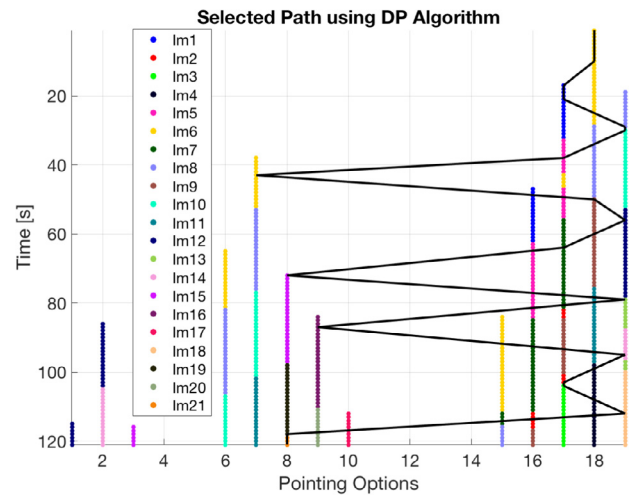


Fig. 13. Example of a path (black line) between different pointing options (X-axis) over time (Y-axis) using the dynamic programming algorithm with no weights for distortion, for a selected 2 min of simulation time where in 21 unique images (colors) can be seen. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

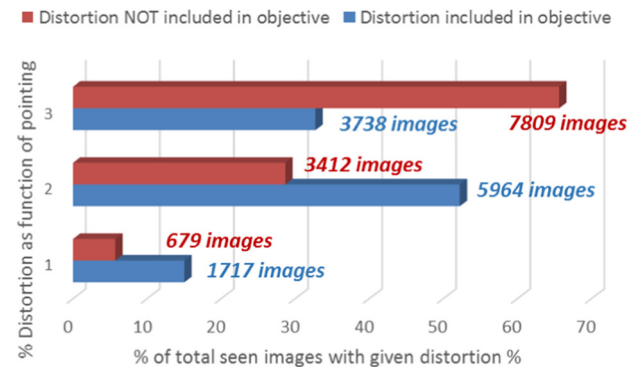


Fig. 14. Histogram distribution of images when distortion weights are added (or not) to the objective function arranged by increasing distortion (1.0%, 2.8%, 3.36%).

images, therefore the steering option supported by our DP algorithm does  $\sim 2.4$  times better than the static option, even with distortion factored in. In fact, a total of 9548.5 effective images (weighted by ‘1-X%’) were seen instead of 8785.4 effective images, without the distortion factored in. Thus, the changed heuristic improved overall image quality.

At any time step in the simulation, there could be up to 19 feasible schedules ending at the 19 pointing options. We found that the number of unique images across the 19 possible schedules at any time in the simulation were different from each other by up to 35 unique images. This difference oscillates randomly between zero and 35 in periods of 15–45 min. Distortion in the heuristic adds to the variability, but even without it, no monotonic trends could be seen. For example, after 23.85 h in the simulation there was no addition of a new unique image and all schedules showed 11,900 seen. We thus reduced the simulation time to 12 h, to reduce runtime in future demonstrations while keeping the representativeness of the results. DP applied to optimization over 12 h provided an ‘optimum’ schedule that observed 6506 unique images and 5394.6 effective images. Distortion weights reduced the number of effective images, and 836, 3418 and 2253 images were observed at nadir to off-nadir pointing, respectively. Compared to a static pointing approach, 2.19 times more images were seen effectively and 2.65 times more in total number.

### 3.2. Multi satellite case for rapid imaging

A two satellite constellation in the same orbital plane,  $180^\circ$  apart in mean anomaly was simulated for 12 h. There were 14,164 observable images within the FOR of the two satellites out of the total 43,200. The ‘optimum’ schedule, as per the DP algorithm in Fig. 10, commanded independently on each satellite as a separate thread, observed

10,847 unique images. In comparison, 4366 images were observed if the sensors were always pointed nadir. The proposed algorithm covered 76.6% of the observable images and showed a 148.4% improvement over the no-agility case. The difference in coverage is apparent in Fig. 15. Agility allows the sensor to effectively expand its cross track swath, without compromising on pixel coarseness. Instead, if a larger FOV sensor had been used for the same focal plane array, the effective pixels would have been larger (aside of uniform edge distortion).

There were 2229 unique images that were common between the two satellites’ observation schedules. Since the DP algorithm uses a satellite’s own history to compute uniqueness of a new image, we need to add other satellites’ histories to the heuristic to prevent redundant observations. With knowledge about the others’ schedules to determine how many unique images have been successfully captured so far (line 11 and 19 in Fig. 10), the DP algorithm can push any satellite to observe images that others have not. The most precise way to address this to make each path in DP, i.e. line 2 and 4 in Fig. 10 (now  $n(P)$  nodes per satellite), a full factorial of  $n(P)$  nodes across  $n(S)$  satellites. This would generate  $n(P)^{n(S)}$  possible paths to eliminate from at any time step, instead of the current  $n(P) \times n(S)$ . The time complexity of this full factorial algorithm would be  $O(n(T) \times n(P)^{2 \times n(S)})$ , instead of  $O(n(S) \times n(T) \times n(P)^2)$ , therefore making it unsolvable in polynomial time with increasing satellite numbers.

Knowledge transfer between the parallel DP optimizations across satellites could be enabled with some assumptions, to keep complexity at  $O(n(S) \times n(T) \times n(P)^2)$ . One assumptive option is that, at any intermediate time, each DP optimization thread could assume that the unique images seen at any node is equal to those seen by that thread’s satellite plus those seen by all possible paths by other satellites, until that node. All possible paths for the

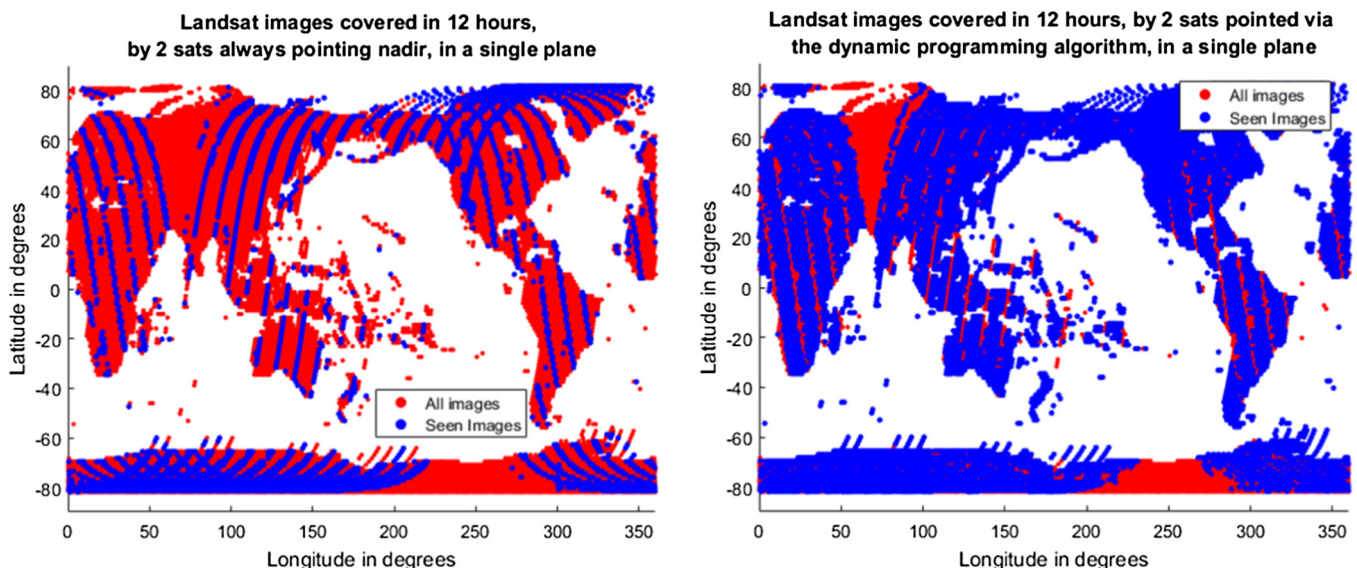


Fig. 15. Comparison of number of images seen using simple Nadir Pointing (left) and using Agile Pointing, guided by our proposed algorithm (right).



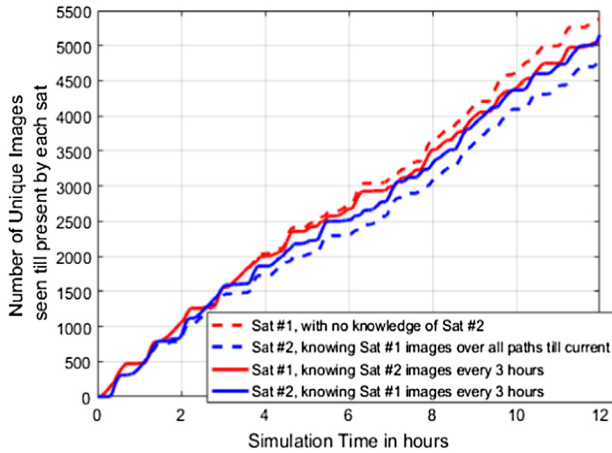


Fig. 16. Number of unique images seen in a two-satellite constellation where the second satellite is aware of all path possibilities of the first (dashed line) vs. where both are aware of each other's optimum paths intermittently (solid line). The associated text describes these are the second (ranked-satellite-based information sharing) and third (time-horizon-based information sharing) options for knowledge transfer between parallel DP threads.

others should be considered because the optimization window is 12 h and an optimum path per satellite has not been determined yet. We found that this approach eliminates overlapping image captures significantly, but comes at the opportunity cost of missing images that are assumed to be seen, but are not. A second assumptive option is to rank the satellites in order of the maximum to minimum images that they are capable of seeing. Each DP thread is made aware of all possible paths, until the present, from all DP threads associated with higher ranking satellites. We found that this approach asymmetrically increased the number of images seen by the higher ranking satellites because they performed the most unconstrained or greediest optimization – dashed lines in Fig. 16. A third (fairer) assumptive option would be to make all DP threads aware of the optimum schedule ( $\Omega$  in Fig. 2) of all others after fixed time horizons. Fig. 16's solid lines show the unique images observed by each satellite over time using this approach. The second option observed 11,076 unique images in total excluding 1363 repeats, i.e. a 39% drop in repeated images compared to the no-knowledge-transfer case. The third option observed 11,027 unique images excluding 1410 repeats, for a time horizon of three hours. While it would appear that the second is a better option than the third, the latter is positively sensitive to constellation structure

and time horizon, as well as ensures a fairer distribution of collected images. After 12 h, the satellites observed 5379.5 and 4834.6 effective images respectively, with ranked-satellite-based sharing of information (Fig. 16-dashed), and 5046.6 and 5153.3 effective images respectively, with time-horizon-based sharing of information (Fig. 16-solid). Moreover, the difference in number of images captured by the satellites in the constellation increases linearly over time of simulation or planning horizon, for ranked-satellite-based sharing of information.

Finding the right time horizon is a conflicting trade. While the different DP threads corresponding to different satellites should know the other's optimum images as frequently as possible to avoid repeats, each DP thread should also have as long a time horizon as possible for global optimization. In a satellite mission scenario where our proposed algorithm will take into account expected satellite states and image requests, shorter time horizons will improve the accuracy of both due to more frequent injection of information. State information can be updated using state of health telemetry or ground surveillance of satellites, which is then used to propagate orbits for future expected states. The satellites can also let known which requests have been actually fulfilled for more accurate estimates of the two heuristics (line 19 and 20 in Fig. 10) within the algorithm.

To demonstrate sensitivity to the time horizon of optimization, Table 3 shows the increase in unique images observed and the decrease of repeated images within decreasing horizon length (X). In comparison, of 14,163 unique images possible to be seen with the FOR of 2 satellites, 10,847 were seen without any information sharing (i.e.  $X = 12$  h) and 4366 were seen without any agile pointing. It is clear that more frequent exchange of optimum paths allows for better optimization and a limit is reached at 45 min of horizon. Shortening windows beyond this limit did not impact results because, as seen in Fig. 15-right, consecutive orbits do not have overlapping FOR swaths. An image within a satellite's FOR can be seen again by the same satellite only after the Earth rotates entirely under the orbit. For the Landsat orbit at given FOR, this takes approximately a week. Therefore, unless the simulation window is more than a week long (unlikely for low Earth orbit planning), the single-satellite DP algorithm optimizes for the best path only within twice its FOR. However, an image within a satellite's FOR can be seen by another satellite in the constellation. Since the described constellation

Table 3

Results of the proposed algorithm on the 2-sat constellation simulation where a satellite's path is optimized every X hours, and each satellite's optimizer is made aware of images seen by the optimum schedule/path of all others every X hours.

X	Unique Images seen	Repeat images (not counted)	Improvement from no agility	Improvement from X = 12 hrs
6 h	10,948	1751	150.7%	0.9%
3 h	11,027	1410	152.5%	1.6%
1.5 h	11,137	1018	155%	2.7%
45 min	11,430	0	161.8%	5.4%

has two satellites phased out by  $180^\circ$  in the same orbit, this opportunity comes at least 45 min later. The multi-satellite DP can behave like a single-satellite DP until this point because there are no potential common images to optimize around. The best schedule generated by DP using the above inter-thread knowledge sharing approach, was able to observe 8999.48 effective images, which is more than twice the effective images (4366) by fixed view, nadir pointing.

### 3.3. Sensitivity to constellation size and clustering

The time horizon for knowledge sharing is a balance between optimization across possible pointing paths for any satellite thread and knowledge sharing across all these threads. As mentioned in Section 3.1, the difference in the number of unique images across possible paths decreases with time, therefore there is little gain for computationally choosing among those paths over a long horizon. For the 2-sat constellation, the common images across all possible paths till any node increases quickly with time, i.e. the paths become less unique, and starts flattening after an hour. The images not common across the paths oscillate at less than 40, so the difference in images seen for a particular schedule vs. another, after 45 min is not very different (<5%) than after 12 h. Since there is no scope of potential path overlap between the two satellites in those 45 min due to the orbital constraints described earlier, that time horizon is a good balance between the conflicting objectives for the 2 satellites phased at  $180^\circ$  on one orbital plane.

For a 4-sat Walker constellation where all satellites are in the same plane (cheapest launch manifest), the uniform inter-satellite phasing is  $90^\circ$ . Polar images will be visited four times an orbit i.e. every  $\sim 20$  min, therefore knowledge sharing should ideally occur every 20 min. Fig. 17 shows the number of unique images that are common across all possible paths (19) per satellite as a fraction of the total

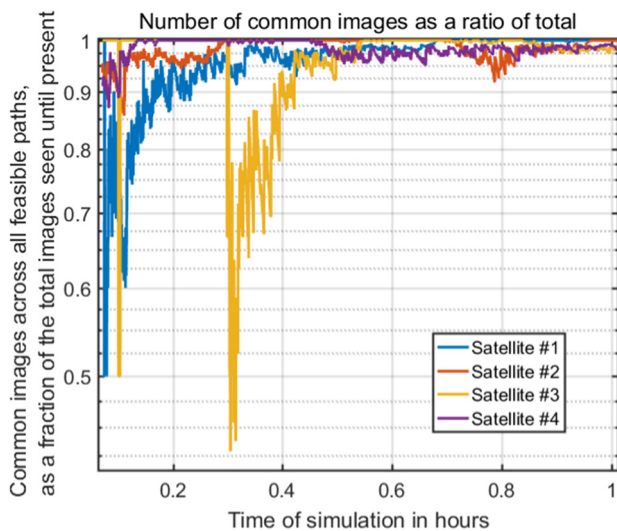


Fig. 17. Fraction of common images across all possible paths to any node (i.e. pointing option and time step) at different simulation times.

unique images seen, as the simulation proceeds. The Fig. shows only the first hour because, beyond that, more than 95% of the unique images seen are the same across all possible paths, irrespective of cross-thread knowledge sharing. However, the paths are not very unique 20 min into the hour because of one satellite (#3 as a yellow line), whose number of common images across its 19 pointing options is less than 75% of the total. Significant difference between the paths implies significant room for optimization by DP on #3's thread. While 45 min for two satellites was a good cutoff, a 20 min horizon may not be a long enough time for finding an optimum solution for four satellites. Table 4 shows the improvement in unique images observed and reduction in repeated images with more frequent inter-thread exchange. As expected, exchanges every 22 min ensures no repeats. Comparing to Table 3, the improvement over the control case with no agile pointing is less for the 4-sat (49.63%) than the 2-sat constellation (161.8%). Adding more satellites to a single orbital plane of a constellation adds diminishing marginal utility, irrespective of agility. However, the DP algorithm at best observed 80.7% of the images within their field of regard (11,430 of 14,164) when applied to two satellites and 95.2% (14,779 of 15,509) when applied to four satellites. Adding satellites not only increased the accessible images, but also the fraction of them observed.

As the number of satellites in a constellation or clustering between satellites increase, the potential revisit time for any image by a satellite becomes shorter and ideal horizons may be too short for the DP optimization. Ranked-satellite knowledge sharing is a possible approach, however it will not perform well for any region that all satellites can access near simultaneously, but not observe completely on their own. For four or more satellites, we propose a strategy that is a combination of full factorial path optimization at  $O(n(t) \times n(P)^{2 \times n(S)})$  and single path optimization at  $O(n(t) \times n(P) \times n(S))$ . We divided the simulation period  $T$  into unequal time horizons ( $t$ , where  $\sum t = T$ ) corresponding to when the images of interest can be observed by  $n_s$  satellites ( $n_s \in [1, n(S)]$ ). The horizons when the images are accessible by any one satellite were optimized using the DP algorithm with no knowledge sharing across DP threads because no overlaps are expected. Horizons ( $t$ ) when images are accessible by two or more satellites ( $n_s$ ) were optimized by eliminating from a full factorial  $n(P)^{n_s}$  possible paths, instead of the  $n(P)$  loops in line 2 and 4 in

Table 4

Results of the proposed algorithm on a 4-sat constellation simulation where in a satellite's path optimization is for every X hours and each satellite is made aware of the images seen by the optimum schedule/path of all others every X hours.

X	Unique Images seen	Repeat images (not counted)	Improvement from no agility
3 h	14,038	10,400	42.13%
45 min	14,594	3929	47.8%
22 min	14,779	0	49.63%

Fig. 10, per satellite and per time step. The length  $t$  should be long enough for the heuristic difference between the possible paths to drop below a threshold. For example, it took 30 min of a time horizon, for all possible paths to be at most 10% different from each other, for the four satellite case. Nonetheless, these horizons  $t$  are expected to be short with low  $ns$ , thus divergence from linear time due to exponential complexity  $O(t \times n(P)^{2 \times ns})$  can be handled without a large computational load.

### 3.4. Multi satellite case for extended time imaging

The coastal case study has a denser concentration of image locations than the Landsat case, however very sparsely distributed globally. The distance between image locations is  $\sim 11$  km, therefore there can be more than a dozen images within a circular swath at  $3.6^\circ$  FOV. The UNEP database was downsampled 10x to avoid hundreds of images within the FOV, which would have added computational needs for no added benefits to demonstrate the proposed algorithm. Downsampling any further would risk underestimating coastal access because their distribution is not as uniform as WRS-2.

This case study will be used to demonstrate the applicability of the DP algorithm to different mission objectives (therefore heuristics) and requested images. We show only results from a two satellite constellation at 675 km altitude  $35^\circ$  inclination,  $180^\circ$  apart in mean anomaly. All results described in the Landsat case are applicable here, unless mentioned otherwise. For example, frequent information sharing across the satellite threads is not necessary because the orbital mechanics module found no overlapping images between the two orbits in 12 h. The unique image sets can be attributed to lower FOR than before and the lack of polar overlaps due to a low latitude area of interest. Nonetheless, leaving the 45-min sharing window as is, is computationally efficient because there are a large number of common images across the competing paths per satellite (akin to  $>1$  h statistics in Fig. 17). Owing to the redundancy in optimizing for long periods and that COCI/Fluid Lensing needs daytime images only, the simulation time was reduced from 12 to 6 h. Reducing further would not affect results or efficiency of the DP algorithm, but 6 h serves as a decent time period for comparing results across different heuristics, allowing 9923 of the 22,718 coral images in the downsampled UNEP database to be accessed. If more satellites were to be added or the constellation structure were to change, the appropriate time window can be computed or a hybrid method used, as described in Section 3.3.

Extended imaging of coastal regions entails imaging as many unique images as possible but also, more importantly, image them for as long as possible to allow image acquisition at multiple angles for COCI and longer video frames for Fluid Lensing. Larger off-nadir angles allow for more imaging time but also cause more distortion at the edges, as seen in Fig. 18. For this case study, we limited

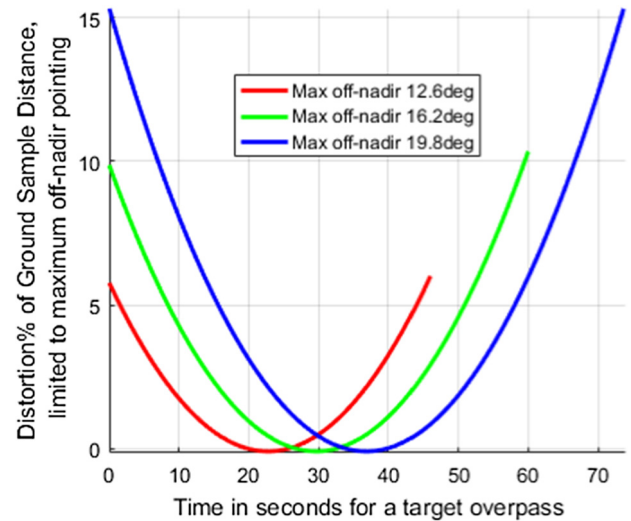


Fig. 18. Percentage distortion of GSD over time, as the reference sensor at 675 km approaches and recedes from any target, limited by the maximum allowed off-nadir angle.

the maximum look to  $16.2^\circ$ , corresponding to at most 10% distortion and a minute of total imaging time. BRDF-related complexities can be assumed negligible because these are more conservative constraints than the Landsat case. When the weights used in Fig. 10's heuristics are 'sum of maximum continuous time over all seen, unique images', 5706 unique images are observed over 21,600 s of simulation. These images are observed for between one to ten seconds, continuously at best. In comparison, 1147 images are observed for one to seven seconds, continuously at best, when no agile pointing is used. As seen in Fig. 19 (brown histogram), the best continuous time available for observing all images in the no-agility case is almost uniform because the images that fall within a satellite's static FOV, as the Earth rotates under it, can be assumed random. In contrast, the blue histogram has nearly five times more unique images but they are seen for a lower, average continuous time (2.3 s vs. 4.3 s) across all images. The DP approach still sees more images per imaging duration than the control case.

The obvious drawback of the above DP heuristic which maximizes the product of image number and time, is that there is no preference for extended imaging or selecting images for which extended imaging is possible. If the objective is to maximize the data per image, the first level heuristic can be modified to the 'maximum continuous time spent in observing any image', as an average across all seen, unique images. Unlike the previous heuristic, this one does not try to maximize the observed unique images. However the second level heuristic does give preference to paths with minimum opportunities left to see any image, just as before. When the modified first level heuristic is used over 6 h of simulation, 28 images are observed for at best 8 to 41 continuous seconds each. Fig. 19-right shows the modified (blue) histogram for the simulation results with the changed heuristic. Since there is no preference for observing

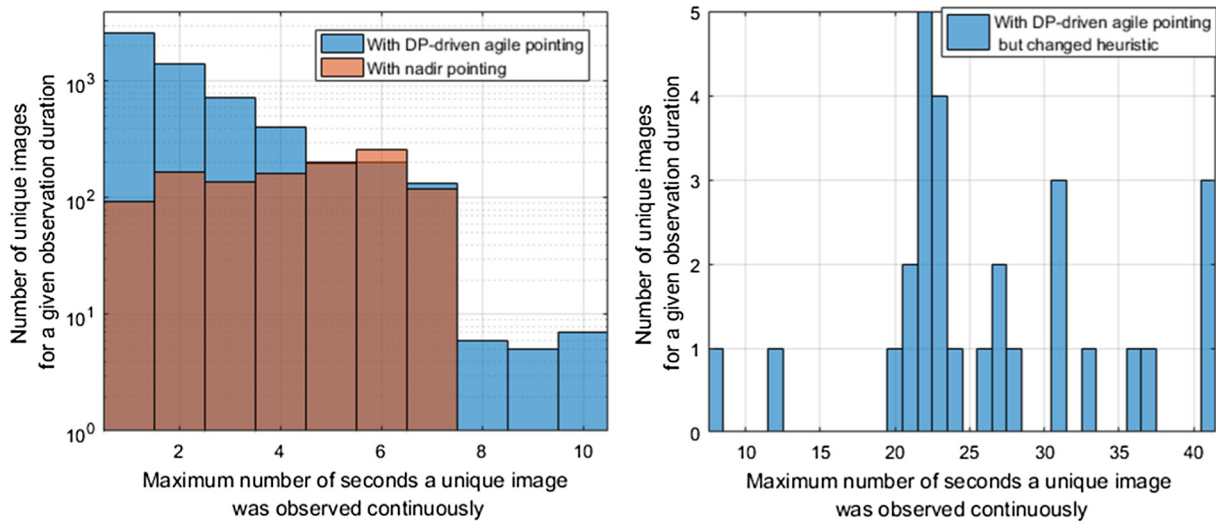


Fig. 19. Histograms of continuous imaging times for all unique images seen, when unique images weighted by ‘maximum continuous time any image is seen’ is used as heuristic [left, blue] vs. when ‘maximum continuous time per unique image seen’ is used as heuristic [right, blue]. The brown bars [left] correspond to the control case with no agility. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

more images, the algorithm avoids observing any images which cannot be seen long enough to better the average observation time, resulting in forty times less images in the schedule. Instead, the algorithm maximizes the access interval for getting as much angular and/or video data as possible (average 26.4 s) and all the images are captured for longer than any in the no-agility case.

There are several ways to address both (or more) objectives used as first-level heuristics in Fig. 19 – number of unique images, distortion, continuous observation time. One, the Lagrange multiplier method can be used where the heuristic in the DP algorithm is a weighted sum of both, in several levels if one level does not cause sufficient elimination. The weights would be a function of the importance of either objective and the need to scale their values to the same order. For example, the first level heuristic values for the optimum schedule, at the end of the simulation for Fig. 19 – left and right were 1292 and 27 respectively. Unless scaled, the two orders of difference in magnitude will hamper the relative utility of increasing one, as perceived by the DP algorithm. Second, a hybrid, time-dependent heuristic can be used in a scheduling re-run, after computing the optimum schedule with changed heuristics, where 28 images are observed for extended durations. In the re-run, the DP algorithm with original heuristics can be used to compute the schedule for periods of time between those extended 28 observations. The hybrid approach is feasible because the first schedule spends less than 15 min in 6 h slewing and observing the said 28 images, allowing for plenty of downtime for capturing more images, even if for non-maximum durations. Third, a convex function can be used as a heuristic, which forces the number of unique images seen to be higher than a required threshold (for example, 1147 for the nadir pointing case) before maximizing observation duration per

image. Results from the above case studies demonstrated that the DP algorithm allows for multiple, non-linear heuristics over multiple levels, enabling optimization over a vast number of mission requirements.

#### 4. Verification using integer programming

We implemented the MILP formulation of Section 2.4 using the General Algebraic Modeling System (GAMS), with subsequent solution by the CPLEX 12.7 solver. The model was solved on the Stanford Sherlock Computing Cluster using the 16 threads of a dual socket Intel(R) Xeon(R) CPU E5-2650 v2 @ 2.60 GHz (8 core/socket) with 64 GB RAM. The full problem comprises 9 million constraints and over 800,000 binary variables, with Eq. (7) being the major source of rows, necessitating high RAM availability. The solution required tuning of CPLEX, including the custom setting of tolerances, and the use of an interior point algorithm for both the initial linear relaxation solve and the subsequent MILP subproblems also. For this problem, the interior point algorithm was found to have the best performance relative to alternatives such as the simplex algorithm. The efficient use of threading by the interior point method can provide it with an advantage on large problems, that offsets the disadvantage of not being able to conduct warm starts on the subproblems. Further tuning and harnessing of the problem structure could yield more efficient formulation or a set of CPLEX tuning parameters that enable faster solution of the MILP.

Importing an initial solution from the DP implementation allowed the MILP search space to be greatly decreased, reducing solve time and allowing us to focus on the purpose of MILP here – an assessment of the quality of the DP solution. Table 5 illustrates results from the first

Table 5  
MILP results for first 45 min of simulation (after taking 15.5 h of solution time on the Stanford Computing Cluster).

	MILP	DP
Objective Value	818.43	779.23
Best possible bound	864.7	
Number images captured	930	917
Images at increasing distortion	207, 546, 177	135, 528, 254

45 min of simulation in the multi-satellite rapid imaging case study (Landsat). The objective value represents the sum of the images captured, weighted by distortion (Eq. (2)), as termed ‘effective images’ in Section 3. The best possible bound represents, *at the time of solution*, the theoretic bound on what the maximum achievable value could be. This bound allows an assessment of the quality of a MILP solution. Further running of the MILP problem allows the optimality gap to be closed by either increasing the objective value i.e. finding better solutions, or reducing of the best possible bound i.e. revising the bound estimate.

We see that MILP quickly finds an improved objective value of 818.43 compared to DP’s 779.23, and shows that no integer solution can exist with an objective value greater than 864.7 i.e. 5.65% better than the found solution. MILP was able to improve upon the number of images observed by the DP schedule, and schedule more images to be observed at 0% and 8% distortion (207 vs. 135 and 546 vs. 528 respectively), therefore better image quality. However, it took the DP algorithm a minute to find its presented schedule on a Mac 2.5 GHz processor @ 1.6 GHz, 16 GB RAM and is expected to take negligible time on the on the Stanford Computing Cluster, and the MILP algorithm took 15.5 h to find its presented schedule. While the DP solution is 4.79% less in quality to the found MILP solution and 9.88% to the best possible bound, it can be reached more than three orders of magnitude faster and without the need of an initial estimate. The MILP simulation also confirmed that image quality notwithstanding, the DP schedule can observe within 1.5% of the optimum number of images for any given satellite.

Comparison of solutions for the full 12-h simulation provides a sense of the quality for the constellation schedule. Due to the large scale nature of the 12-h problem with current formulation and implementation, we generated only a relatively crude upper bound by solving the linear relaxation of the MILP problem within 16 h of runtime on the Stanford Computing Cluster. DP’s best schedule showed an objective value of 8999.48, which is 22.56% lower than the best possible *bound* generated by MILP (11,622). Due to the large memory requirements of the full problem, MILP *found* a solution 3% better than DP’s solution – objective value of 9294 and total images at 11,396 (compared to DP’s 8999.48 and 11,400 respectively). 1182, 5836, 4378 images were observed by MILP at increasing distortion, compared to 750, 5905, 4776 images by DP over 12 h. The results show that MILP improves the quality of the solution far more due to improved image

quality (fewer images with more distortion), than due to the improved number of images seen. While the DP solution is 22% less optimal in the 12-h case than the 45-min case, 22.56% represents the worst-case optimality gap because the bound is expected to improve with computation time and refinement of the implementation.

## 5. Summary and future work

This article proposed a simple, modular framework for scheduling the attitude control operations for a constellation of small satellites to maximize the observation of requested images around the globe. The orbital mechanics and ACS modules can be simulated independently, as a function of satellite and image request specifications, and their outputs applied to a DP algorithm and the solution schedule improved upon by the MILP algorithm. Independence of the different modules allows for numerical simulation and uncertainty modeling within each, therefore ensuring higher fidelity than an idealized, analytical approach.

The DP-based algorithm for agile constellations applied to rapid imaging observed more than 2.5 times the images compared to constellations with no agility. Images observed can be increased by intermittent exchange of optimum schedules between the optimization threads per satellite. The algorithm applied to extended imaging observed more than nearly five times the number of unique images or six times the observation time per image, depending on the heuristic used, compared to the static sensor case. As shown through two case studies of ground point targets and sensor characteristics, the framework is applicable to any Earth imagery and a few degrees or higher FOV sensors. Our proposed framework of decoupling satellite dynamics from optimization and the DP algorithm applied to this framework is generic enough to be applied to changing mission objectives, orbits, constellation structure, satellite number, images locations or sensor characteristics. The MILP implementation showed that the DP solution is within 10% of the true optimum, and reached it in a computationally efficient manner (3–4 orders of magnitude faster than MILP) for each satellite’s optimization thread. Also, the number of images seen by the DP schedule was 1.5% less than the MILP schedule. Since the DP algorithm is structured in a way such that the solution at any given simulation time step is the optimum till that time step, a schedule over any time period can be compared to MILP for verification. The MILP algorithm provides bounds on the optimum solution for a time period, even if it does not find the optimum schedules in finite time. For the 12-h simulation, the worst case optimality gap for the DP schedule was found to be 22.56%.

The presented results are an initial step in exploring the capabilities that state-of-the-art commercial ACS systems are enabling in Earth observation by small satellites. The modularity of our proposed framework allows for the inclusion of many complex space–time constraints to image selection and schedule, such as avoiding images covered by

time-dependent cloud cover (Orbits module), avoiding image collection during ground station downlink windows (Orbits module), ACS constraints for full-body alignment of solar panels for charging at the same time as imaging (ACS module), adding duty cycle upper limits if power is tight (Optimization module), scheduling for inter-satellite calibration or any other nominal mode operations (Optimization module). A more complex orbits module, such as one that outputs illumination conditions to be used as imagery constraints, or a more complex ACS module, such as one that includes uncertainty characterization as a function of momentum wheel limitations or position/attitude knowledge error, may also be introduced. One can also explore better solutions to the presented problem using improved MILP, other optimization formulations and a more integrated ACS-scheduling optimization, perhaps at the cost of computational efficiency. For example, in orienteering, each image at a given time will have neighborhood of  $n(P) \times n(P)$  nodes representing the transition pairs and the reward per node would be the same as the heuristics proposed in this paper.

The proposed framework and algorithms can be integrated with constellation design software such as NASA GSFC's TAT-C to assess the full potential of any design, given currently capable agile satellites. The scheduling framework proposed can be integrated with other schedulers and applied to a diversity of existing assets such as unmanned aerial vehicles (UAVs) or research balloons. In keeping with NASA's Sensor Web approach, such scheduling will help optimize operations of many flying platforms by leveraging their positives – spatial resolution and local targeting by UAVs, long distance access by balloons, global access by satellites – and enable more responsive remote sensing. The presented framework can be adapted for running onboard the satellite for autonomous scheduling, contingent on onboard processing capability, and if the satellites can propagate their states accurately for a short time horizon and communicate their states with all other satellites, within the same horizon, with or without relay (both technologies are currently under development for small spacecraft). This will allow satellite clusters to make schedules in flight without ground-in-the-loop, based on image observation requests and their states, and enable more autonomous remote sensing.

## Acknowledgements

The Quick Response System Grant, awarded by NASA's Earth Science Technology Office, funded the presented work. The authors are grateful to Dr. Jeremy Frank at the NASA Ames Research Center, and Prof. Marco Pavone and Stefan Jorgensen at Stanford University for very useful discussions that have improved the quality of this paper. The integer programming portion of the computing for this research was performed on the Sherlock cluster at Stanford University, where we would like to

thank the Stanford Research Computing Center for providing these computational resources.

## References

- Abramson, M.R., Kowitz, S., Robinson, E., Poppe, D., 2013. Earth Phenomena Observation System (EPOS) for Coordination of Asynchronous Sensor Webs. In: *AIAA Infotech@ Aerospace (I@ A) Conference*, p. 4813.
- ARC Mission Design Center, 2016. Small Spacecraft Technology State of the Art (No. NASA/TP–2015–216648/REV1). NASA Ames Research Center, Moffet Field, CA.
- Arkali, G., Dawande, M., Sriskandarajah, C., 2008. Scheduling support times for satellites with overlapping visibilities. *Prod. Oper. Manage.* 17, 224–234.
- Barnsley, M.J., Settle, J.J., Cutter, M.A., Lobb, D.R., Teston, F., 2004. The PROBA/CHRIS mission: A low-cost smallsat for hyperspectral multiangle observations of the earth surface and atmosphere. *IEEE Trans. Geosci. Remote Sens.* 42, 1512–1520.
- Benoist, T., Rottenbourg, B., 2004. Upper bounds for revenue maximization in a satellite scheduling problem. *Q. J. Belg. Fr. Ital. Oper. Res. Soc.* 2, 235–249.
- Bergeron, M., Craig, S.E., 2017. Report on ocean colour activities. In: *International Ocean Colour Science Meeting*. Lisbon, Portugal.
- Bertsimas, D., King, A., Mazumder, R., 2016. Best subset selection via a modern optimization lens. *Ann. Stat.* 44, 813–852.
- Bianchessi, N., Righini, G., 2008. Planning and scheduling algorithms for the COSMO-SkyMed constellation. *Aerosp. Sci. Technol.* 12, 535–544.
- Bunkheila, F., Ortore, E., Circi, C., 2016. A new algorithm for agile satellite-based acquisition operations. *Acta Astronaut.* 123, 121–128.
- Chirayath, V., Earle, S.A., 2016. Drones that see through waves—preliminary results from airborne fluid lensing for centimetre-scale aquatic conservation. *Aquat. Conserv. Mar. Freshw. Ecosyst.* 26, 237–250.
- Damiani, S., Verfaillie, G., Charneau, M.-C., 2005. An earth watching satellite constellation: How to manage a team of watching agents with limited communications. In: *Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems*. ACM, pp. 455–462.
- Daniel, H., 2016. FlexCore: Low-Cost Attitude Determination and Control Enabling High-Performance Small Spacecraft. In: *Small Satellite Conference*. Logan, Utah.
- D'Errico, M., 2012. *Distributed Space Missions for Earth System Monitoring*. Springer Science & Business Media.
- Dishan, Q., Chuan, H., Jin, L., Manhao, M., 2013. A dynamic scheduling method of earth-observing satellites by employing rolling horizon strategy. *Sci. World J.* 2013.
- Eickhoff, J., 2011. *Onboard Computers, Onboard Software and Satellite Operations: An Introduction*. Springer Science & Business Media.
- Frank, J., Do, M., Tran, T.T., 2016. Scheduling ocean color observations for a GEO-stationary satellite. In: *Proceedings of the Twenty-Sixth International Conference on Automated Planning and Scheduling*. AAAI Press, pp. 376–384.
- Frank, J., Jonsson, A., Morris, R., Smith, D.E., Norvig, P., 2001. Planning and scheduling for fleets of earth observing satellites.
- Globus, A., Crawford, J., Lohn, J., Morris, R., 2002. Scheduling earth observing fleets using evolutionary algorithms: Problem description and approach.
- Jian, L., Cheng, W., 2008. Resource planning and scheduling of payload for satellite with genetic particles swarm optimization. In: *Evolutionary Computation, 2008. CEC 2008*. (IEEE World Congress on Computational Intelligence). IEEE Congress on. IEEE, pp. 199–203.
- Kennedy, A.K., Cahoy, K.L., 2015. Onboard operations scheduling for a cooperative earth remote sensing small satellite constellation. In: *9th International Workshop on Satellite Constellations and Formation Flying*. Delft, The Netherlands.

- Knight, E.J., Kvaran, G., 2014. Landsat-8 operational land imager design, characterization and performance. *Remote Sens.* 6, 10286–10305.
- Lefferts, E.J., Markley, F.L., Shuster, M.D., 1982. Kalman filtering for spacecraft attitude estimation. *J. Guid. Control Dyn.* 5, 417–429.
- Lemaître, M., Verfaillie, G., Jouhaud, F., Lachiver, J.-M., Bataille, N., 2002. Selecting and scheduling observations of agile satellites. *Aerosp. Sci. Technol.* 6, 367–381.
- LeMoigne, J., Dabney, P., De Weck, O.L., Foreman, V., Grogan, P.T., Holland, M., Hughes, S., Nag, S., 2017. Tradespace Analysis Tool for Designing Constellations (TAT-C). In: IEEE International Geoscience and Remote Sensing Symposium. Presented at the IEEE International Geoscience and Remote Sensing Symposium, Fort Worth, TX.
- Lieber, M., Weimer, C., Rohrschneider, R., Ruppert, L., 2017. Model Predictive Control Architecture for Optimizing Earth Science Data Collection (PCAES) – Computational Challenges and Initial Test Results.
- Lin, W.-C., Liao, D.-Y., Liu, C.-Y., Lee, Y.-Y., 2005. Daily imaging scheduling of an earth observation satellite. *IEEE Trans. Syst. Man Cybern.-Part Syst. Hum.* 35, 213–223.
- Mandl, D., Frye, S.W., Goldberg, M.D., Habib, S., Talabac, S., 2006. Sensor webs: Where they are today and what are the future needs? In: *Dependability and Security in Sensor Networks and Systems, 2006. DSSNS 2006. Second IEEE Workshop on.* IEEE, pp. 65–70.
- Markley, F.L., 2003. Attitude error representations for Kalman filtering. *J. Guid. Control Dyn.* 26, 311–317.
- Martin, W., 2002. Satellite image collection optimization. *Opt. Eng.* 41, 2083–2087.
- Morris, R., Dungan, J., 2007. Harnessing the sensor web through model-based observation. In: *Aerospace Conference, 2007 IEEE.* IEEE, pp. 1–8.
- Morris, R., Dungan, J., Votava, P., 2009. A workflow model for earth observation sensor webs, in: *Space Mission Challenges for Information Technology, 2009. SMC-IT 2009. Third IEEE International Conference on.* IEEE, pp. 69–76.
- Muraoka, H., Cohen, R.H., Ohno, T., Doi, N., 1998. Aster observation scheduling algorithm. In: *Proceedings of the International Symposium Space Mission Operations and Ground Data Systems.* Citeseer.
- Nag, S., Gatebe, C., Miller, D.W., De Weck, O.L., 2016. Effect of satellite formation architectures and imaging modes on global albedo estimation. *Acta Astronaut.* 126, 77–97. <https://doi.org/10.1016/j.actaastro.2016.04.004>.
- National Research Council, 2012. *Earth Science and Applications From Space: A Midterm Assessment of NASA's Implementation of the Decadal Survey.* National Academies Press, Washington, DC.
- National Research Council, 2000. *The Role of Small Satellites in NASA and NOAA Earth Observation Programs.* National Academies Press.
- Oza, N., Morris, R.A., Strawa, A., Kurklu, E., Keely, L., 2008. *Automated Data Assimilation and Flight Planning for Multi-Platform Observation Missions.*
- Robinson, E., Balakrishnan, H., Abramson, M., Kowitz, S., 2017. *Optimized Stochastic Coordinated Planning of Asynchronous Air and Space Assets.* *J. Aerosp. Inf. Syst.* 14 (1), 10–25.
- Sandau, R., Roeser, H.P., Valenzuela, A., 2010. *Small Satellite Missions for Earth Observation: New Developments and Trends.* Springer Verlag.
- Sherwood, R., Govindjee, A., Yan, D., Rabideau, G., Chien, S., Fukunaga, A., 1998. Using aspen to automate eo-1 activity planning. In: *Aerospace Conference, 1998 IEEE.* IEEE, pp. 145–152.
- Space Studies Board, 2016. *National Academies of Sciences and Medicine. Achieving Science with CubeSats: Thinking Inside the Box.* National Academies Press.
- Spangelo, S., Cutler, J., 2012. Optimization of single-satellite operational schedules towards enhanced communication capacity. In: *AIAA Guidance, Navigation, and Control Conference.* p. 4610.
- Stephenson, K., 2005. *Introduction to Circle Packing: The Theory of Discrete Analytic Functions.* Cambridge University Press.
- Schetter, T., Campbell, M., Surka, D., 2003. Multiple agent-based autonomy for satellite constellations. *Artif. Intell.* 145, 147–180.
- Trawny, N., Roumeliotis, S.I., 2005. Indirect Kalman Filter for 3D Attitude Estimation.
- Vansteenkoven, P., Souffriau, W., Van Oudheusden, D., 2011. The orienteering problem: A survey. *Eur. J. Oper. Res.* 209, 1–10.
- Vasquez, M., Hao, J.-K., 2001. A “logic-constrained” knapsack formulation and a tabu algorithm for the daily photograph scheduling of an earth observation satellite. *Comput. Optim. Appl.* 20, 137–157.
- Verfaillie, G., Pralet, C., Lemaître, M., 2010. How to model planning and scheduling problems using timelines. *Knowl. Eng. Rev.* 25, 319–336.
- Wen, J.-Y., Kreutz-Delgado, K., 1991. The attitude control problem. *IEEE Trans. Autom. Control* 36, 1148–1162.
- Wertz, J.R., 2012. *Spacecraft Attitude Determination and Control.* Springer Science & Business Media.
- Wie, B., Lu, J., 1995. Feedback control logic for spacecraft eigenaxis rotations under slew rate and control constraints. *J. Guid. Control Dyn.* 18, 1372–1379.
- Witt, K.J., Stanley, J., Smithbauer, D., Mandl, D., Ly, V., Underbrink, A., Metheny, M., 2008. Enabling Sensor Webs by utilizing SWAMO for autonomous operations. In: *8th NASA Earth Science Technology Conference,* pp. 263–270.
- Khafa, F., Sun, J., Barolli, A., Biberaj, A., Barolli, L., 2012. Genetic algorithms for satellite scheduling problems. *Mob. Inf. Syst.* 8, 351–377.