Agile Spacecraft Imaging Algorithm Comparison for Earth Science

Zaki Hasnain, James C Mason, Jason Swope, Joshua Vander Hook, Steve Chien
Jet Propulsion Laboratory
California Institute of Technology
Pasadena, CA, USA

Abstract

Earth surface observation via satellite imaging is limited by data storage, downlink, and imager constraints. Therefore, both asset management and the quality of scientific studies can be optimized by on-board minimization of cloud coverage in data. Algorithms described here plan a sequence of surface targets to image which minimize cloud coverage given satellite constraints. The proposed algorithms build on cloud classification systems and focus on the planning problem of finding an optimal set of surface targets which can be imaged given mission constraints. We provide greedy and graph search based algorithms and test them on a dataset of 50 labeled images which have varying amounts of clear and cloudy pixels. These planning algorithms are compared to a dynamic programming optimal solution, and performance metrics which balance runtimes and the amount of usable data in generated plans are reported. The performance of the greedy and graph search methods depends on both the fraction and the distribution of clear pixels in the imageable domain. A proposed adaptive grid graph search method finds 96-100% of the mean fraction of clear pixels found by the optimal method over the test dataset. The greedy search results vary broadly, and its plans represent 29-100% of the clear pixels found by the optimal method.

List of Symbols

$\mathbf{D}$	binary image matrix of image with clear/cloudy classifications
Π	plan consisting of image center coordinates
d	distance satellite travels in along direction in single time step
d_x	Maximum off-nadir looking distance in across direction
d_y	Maximum off-nadir looking distance in along direction
h	half length of sub-image in along direction
N	number of sub-images in a plan
p	image center point
s_x	Maximum slew distance in across direction between image centers in a plan
s_y	Maximum slew distance in along direction between image centers in a plan
w	half length of sub-image in across direction
x, y	across and along coordinates on swath
x_S	fixed satellite position in across direction
$y_S(t)$	time varying satellite position in along direction

Introduction

Remote sensing of earth surface unlocks a cascade of scientific exploration and engineering solutions (Chien et al. 2005), however, some of these applications are hindered by cloud coverage (Beaumet, Verfaillie, and Charmeau 2011) which may span approximately two-thirds (King et al. 2013) of the Earth's surface at a given time. For instance, cloud coverage compromises the detection and classification of surface elements such as ships (Yang et al. 2013), and environmental phenomenon such as glaciers (Paul et al. 2015) at various stages of experimentation and analysis. Further the increasing volume of onboard imaging requires data prioritization schemes which can mitigate downlink and storage limitations (Sandford et al. 2020). Therefore, autonomous spacecraft-based remote sensing missions can increase yield of scientifically relevant data by limiting the amount of undesired cloudy observations.

Due to rapid and unpredictable cloud motion, it is not feasible to account for cloud coverage during a typical spacecraft remote sensor's observation planning phase. Ideally clouds in a downstream swath would be identified and planned around just minutes before an observation. This requires planning to be performed onboard the spacecraft (Beaumet, Verfaillie, and Charmeau 2011). However, the onboard planning paradigm is significantly different than current practice, which typically plans observations days in advance (Yelamanchili et al. 2019; Moy et al. 2019).

Presently, most approaches to the overall problem of data prioritization focus on on-board cloud screening after collecting observations (Sandford et al. 2020). Specifically, the collected data is analyzed and cloudy regions are removed, reducing the storage and downlink transmission costs. Other

approaches (He et al. 2019; Wang et al. 2020) plan observations hours or days in advance on the ground by using cloud coverage forecasts. While these approaches likely reduce cloudy observations, a more optimal approach would process actual cloud knowledge in real time.

Similar ongoing work at NASA Jet Propulsion Laboratory includes the Smart Ice Cloud Sensing (SMICES) smallsat concept, a radar application that intelligently targets storms and clouds. Despite the opposite goal of the cloud-minimization problem, it operates in a similar manner by picking an area inside of its view to analyze. SMICES targets images at a rapid rate, on the order of seconds, and its planner runs as a continuous problem, and has a power constraint of a 20% duty cycle. Furthermore, SMICES uses multiple cloud labels to identify different targets instead of the binary cloudy/clear labels used for the cloud avoidance problem studied here. This gives SMICES flexibility on which targets it analyzes and allows scientists to tailor its algorithm to target the clouds that best align with their scientific interests. Here, we focus on the converse problem of avoiding clouds, therefore do not distinguish between cloud types.

We describe a selection of onboard planning algorithms for finding a set of surface targets that can be imaged given instrument and vehicle constraints as well as cloud coverage information. Given a swath, for which cloudiness is known, a plan of observations within the swath is generated and evaluated based on the resulting average cloudiness. We develop a greedy algorithm which is limited with no look-ahead cloud data, a grid-based graph search algorithm, and an optimal dynamic programming algorithm.

Data source

The Moderate Resolution Imaging Spectroradiometer (MODIS) instrument on the Terra and Aqua satellites provides wide range multispectral data, covering the Earth's surface every 1 to 2 days (Platnick et al. 2003). We use the 1km MODIS Cloud Mask product (Ackerman and et al. 2015) to determine if an area is clear or cloudy. We classify a pixel as clear if it is flagged as "probably clear" or "confidently clear", and cloudy otherwise. The product is broken into 5 minute swaths which typically cover 2030 or 2040 km, as shown in Figure 1. The ground sample distance in the dataset is 1 km, and typical images are 1354 across and either 2030 or 2040 pixels along. The fraction of clear pixels in the imageable domain for this dataset ranges [0.011, 0.847] with an average of 0.323, and represents a variety of sparse and dense cases.

Methods

Problem

A plan $\Pi = \{p = (x, y) \in \mathbb{R}^2\}$ is defined as a series of target points p on a swath, where each target point is the center of a purported sub-image to be captured (Figure 1). Here, x and y are the distances across and along an image respectively. The swath is given as a binary matrix $\mathbf{D}$ where each element $d_{i,j}$ is 1 if the pixel is cloudy and 0 otherwise for the position i in the along direction and j in the across direction. To maximize the usability of satellite imagery, the optimization methods described here use the fraction of clear pixels given by a plan for a given MODIS 5-minute swath as the objective function. We describe the usability of a plan by the mean fraction of clear pixels (MFC) over all of its sub-images. Plans are generated on the MODIS dataset by i) the nadir ground track, ii) greedy search, iii) graph search, and a iv) dynamic programming (DP) based method which provides optimal solutions. Performance is measured by comparing the MFC of generated plans and the benchmark DP optimal plans. The optimization problem is summarized as maximizing the MFC of a given input swath image, while obeying the constraints shown in Figure 2 and described below.

Constraints

Search for plans is limited by four constraints, as shown in Figure 2. Each plan corresponds to a satellite travelling d = 200 km along the swath in a time step, while maintaining a fixed position $x_S = 677$ km in the across direction, for all time t (Figure 2a). The initial along-distance of the satellite is set to $y_S(t = 0) = 100$ for the MODIS dataset. Each sub-image in a plan is of size $2w + 1$ across and $2h + 1$ along (Figure 2b). Constraint III limits the domain of the sub-image centers p to be $d_x = 350$ km from nadir in the across direction, and $d_y = 100$ km from nadir in the along direction (Figure 2c). Finally, in constraint IV, the maximum slew as measured by ground distance between consecutive sub-image centers is constrained to be at most $s_x = 100$ in the across direction, and at most $s_y = 50$ in the along direction (Figure 2d). However, since the satellite moves along a swath a distance d per time step, consecutive centers may be a maximum distance $d + s_y$ in the along direction. There is sufficient time to slew between observations given the corresponding slew distances do not exceed (s_x, s_y). Therefore, given the choice of constraint parameters described in Figure 2, and a 5-minute swath, a plan consists of $N = 10$ sub-images of size 21 km along and 41 km across ($w = 40$ km, $h = 20$ km).

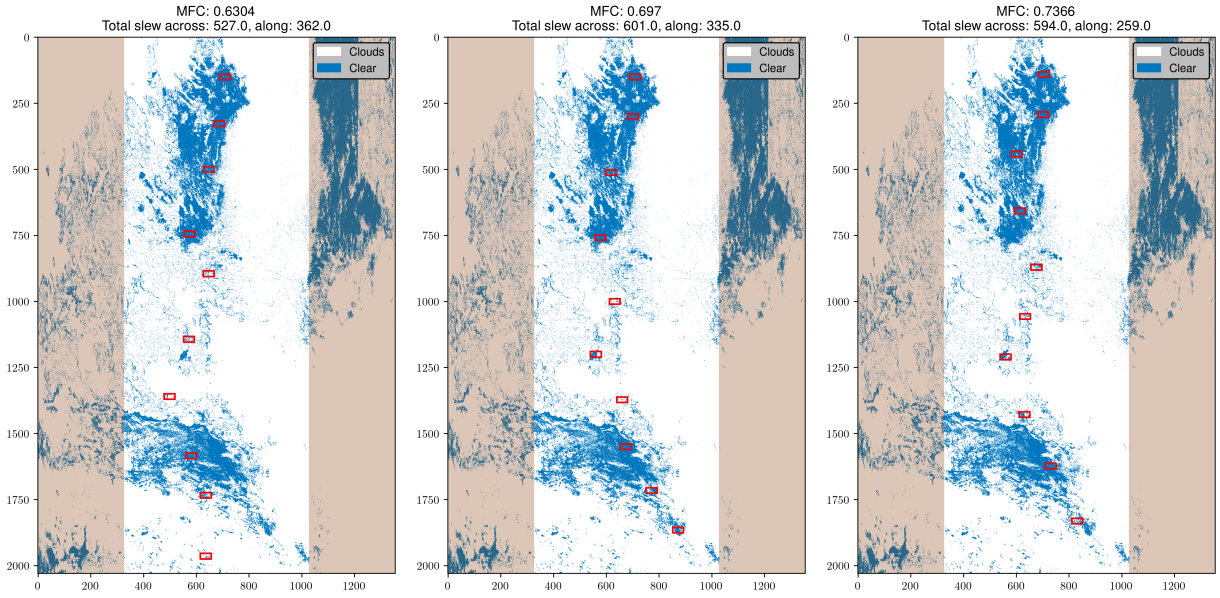
Combined, we have the problem statement:

$$\max_{\Pi=\{(p_x(i),p_y(i))\}} \frac{1}{NA} \sum_{i=0}^{N-1} \sum_{p_x(i)-w}^{p_x(i)+w} \sum_{p_y(i)-h}^{p_y(i)+h} \mathbf{D}(x, y) \quad (1)$$

Subject to,

$$p_x(0) \in x_S \pm s_x \quad (2)$$
$$p_y(0) \in y_S(t = 0) \pm s_y \quad (3)$$
$$\forall i > 0 : p_x(i) \in p_x(i - 1) \pm s_x \quad (4)$$
$$\forall i > 0 : p_y(i) \in p_y(i - 1) + d \pm s_y \quad (5)$$
$$\forall i > 0 : p_x(i) \in x_s \pm d_x \quad (6)$$
$$\forall i > 0 : p_y(i) \in y_s(t = t_i) \pm d_y \quad (7)$$

where $A = (2w + 1)(2h + 1)$ is the area of each sub-image.



(a) Greedy search (b) Graph search (c) Optimal solution

Figure 1: An example of the 1km MODIS Cloud Mask product, with plans generated from the (a) greedy, (b) graph, and (c) optimal dynamic programming (DP) methods for a case that has ∼16% clear pixels in the imageable domain. The graph search method is adaptive grid (AG) with bin size 40×40, $\rho = 50$, $r_b = 1/2$, $r_\rho = 1/2$. The graph search path's mean fraction of clear pixels (MFC) is 94.6% of the DP solution and the two paths are nearly identical. In contrast, the greedy solution fails to image the sparse clear areas, and its MFC is only 85.6% of the DP solution.

Optimal Implementation with Dynamic Programming

Note that Eq (1) trivially satisfies optimal substructure and overlapping subproblems criteria due to the independent sums. To see this, note that we sum over nearby pixels many times as the point p and its neighbors are evaluated, and that any choice of a given $p(i)$ implies that the sub-path $p(0)...p(i-1)$ is independently optimal (otherwise we could find a better path $p(0)...p(i-1)$ to reach $p(i)$). Thus, it is amenable to a dynamic programming solution, which we outline here.

Note, that the dynamic programming solution, when implemented single-threaded (for fair comparison and in a space-like computing environment), even though it exhibits good cache localization, is quite slow compared to the adaptive algorithms discussed. For our purposes, it only matters that it produces the optimal results, for baseline comparison of the algorithms proposed. An optimized and parallelized implementation of this algorithm will be presented and analyzed separately when considering a hypothetical multi-core processor for spaceflight.

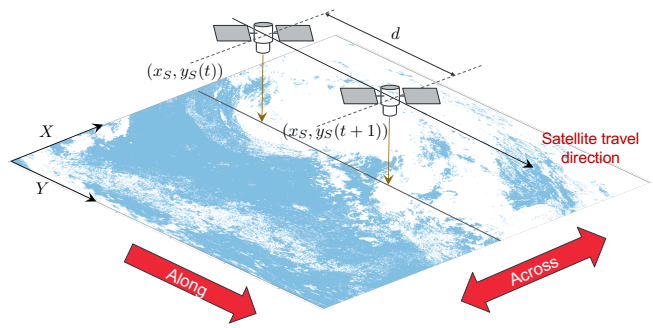
Algorithm 1: Dynamic Programming Solution

input : zero-padded matrix $\mathbf{D}$
output: Target points p of a plan Π with N observations.

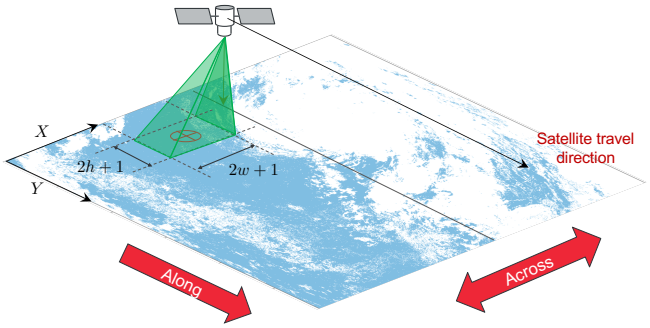
for *each pixel,* x, y *in* $\mathbf{D}$ **do**
  $\mathbf{V}(x,y) \leftarrow \texttt{sumAround}_{w,h}(\mathbf{D}, x, y)$
  $\mathbf{V}(x,y) \leftarrow \mathbf{V}(x,y) + \texttt{maxAround}_{s_x,s_y}(\mathbf{D}, x, y-d)$
end
$p(N) \leftarrow \texttt{argMaxAround}_{d_x,d_y}(\mathbf{V}, x_S, y_S(t = 0) + d \cdot N)$
for $i \leftarrow N-1$ *to* 0 **do**
  $p(i) \leftarrow argMaxAround_{s_x,s_y}(\mathbf{V}, p(i+1) - (0,d))$
  `Print` (p(i))
end

Greedy algorithm

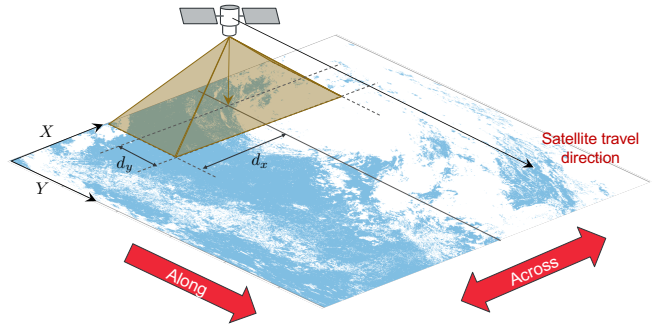
The greedy method steps forward through time and discovers a plan by searching over the area of imageable centers at each time step t. The center p_t associated with the clearest sub-image at time t is added to the plan, and the imageable search area for the next time step $t+1$ is calculated based on the location of p_t. Therefore, the greedy selection of p_t limits the search area for subsequent sub-image centers. This greedy strategy is described in Algorithm 2 where the routine is initialized with the furthest planning distance along a MODIS image $d_{max} = 2000$. In each iteration of the greedy search in Algorithm 2, the window W of imageable centers p, is found based on the constraint parameters and the satellite position. Next, each pixel in W is allowed to be a candidate point p for the plan Π, and the cloudiness about these points is calculated for a sub-image of size $2w+1 \times 2h+1$.



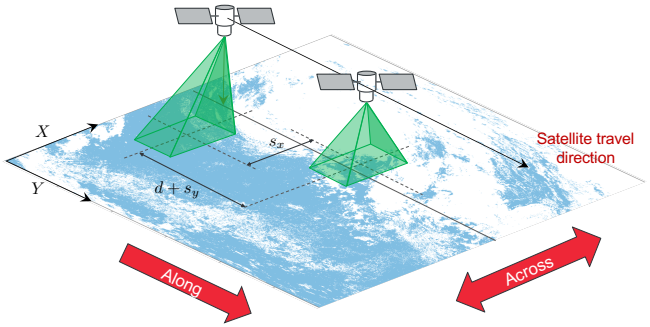
(a) Constraint I: distance d travelled by the satellite along a swath in a single time step is constant



(b) Constraint II: the dimensions of a single sub-image are $2w+1 \times 2h+1$.



(c) Constraint III: the satellite can scan up to a distance away from nadir in both directions.



(d) Constraint IV: satellite slewing between consecutive sub-images is constrained.

Figure 2: The surface target planning problem has four constraints.

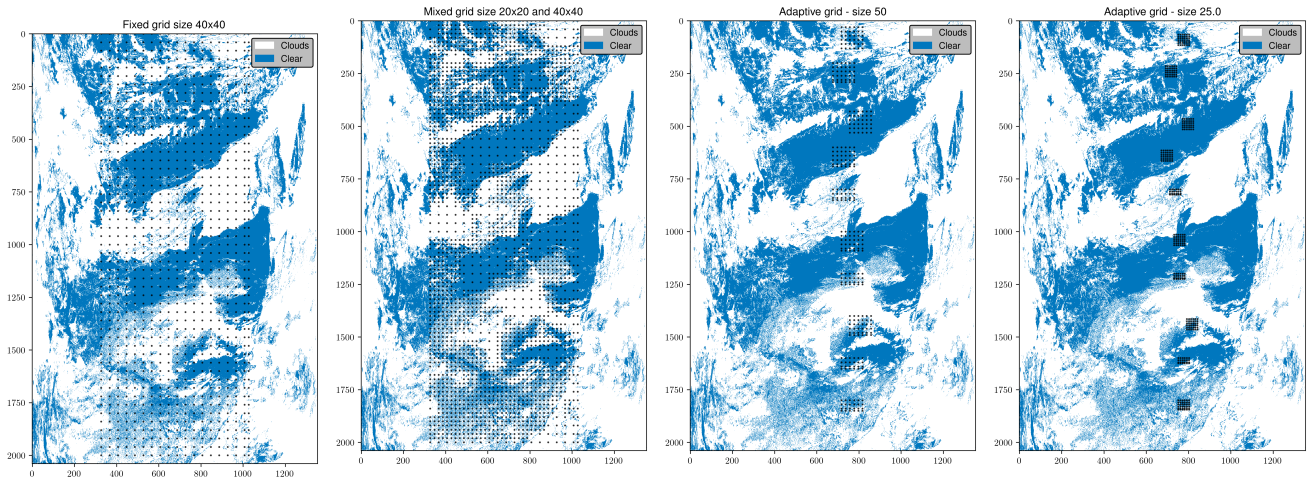
The set of least cloudy candidate points in W is found, and the one with the smallest L_1 (Manhattan) distance to the previous point in the plan is selected.

Graph search algorithm

Graph search methods discretize a swath into a grid of nodes which summarize the cloudiness in their neighborhood, and edges between nodes indicate a valid sequence of sub-images given the set of satellite imager constraints Θ. The nodes represent sub-image centers p of a plan Π. The incoming edges of a node carry a weight equal to the cloudiness in a sub-image centered at the node. Constraints I (Figure 2a) and III (Figure 2c) are used to divide the grid of nodes into sets such that each set has all the nodes which the satellite can scan at a given time step. Specifically, nodes are divided into sets $S_t = \{(x, y) \in \mathbb{R}^2 : x \in [x_S(t) - d_x, x_S(t) + d_x], y \in [y_S(t) - d_y, y_S(t) + d_y]\}$ for a given time step t by the domain constraint distances d_x and d_y, where $x_S(t)$ and $y_S(t)$ give the position of the satellite at time t. For instance, the first set of nodes S_o is comprised of nodes imageable at the satellite's original position, and the final set of nodes S_f is all the nodes imageable at the satellite's final position. Edges only connect nodes from a given set to the subsequent one, and this results in a directed acyclic graph (DAG) which is topologically sorted. Furthermore, constraint IV (Figure 2d) is enforced when discovering edges between nodes. Consequently, a viable plan is given by a path on the graph which connects a node from S_o to a node in the final set S_f at the end of the swath. Therefore, the shortest viable path on the graph represents a sequence of sub-image centers which minimizes cloud coverage. MFC of a plan is computed by averaging the weight of edges in the shortest distance path.

Modulating the density and placement of nodes in a grid allows for a higher or lower fidelity representation of the original swath and also controls the computational performance of the graph search routine. To explore these tradeoffs in performance, we study three types of grids, i) fixed grids (FG) which place nodes on a rectangular lattice of fixed density (e.g. Figure 3a), ii) mixed grids (MG) which increase density of nodes in heterogeneous sections of the swath and reduce node density in homogeneous areas which are characterized by a majority of clear or cloudy pixels (e.g. Figure 3b), and iii) adaptive grids (AG) which iteratively improve the plan by placing more nodes in a narrow window around the previous plan's sub-image centers (e.g. Figure 3c-3d).

Algorithm 3 describes the routine for finding a plan using fixed grids. A FG is defined by bin sizes B_x and B_y which are spacing between nodes in the across and along directions, respectively. Thus, only two hyper-parameters are required beyond the standard constraint parameters for the FG method. On the other hand, the MG method allows for



(a) FG: 969 nodes. Bin sizes $B_x = B_y = 40$.

(b) MG: 2649 nodes. Bin sizes $B_x = B_y = 40$ and $b_x = b_y = 20$.

(c) AG, iteration #2: 312 nodes. Bin sizes $B_x = B_y = 20$, $\rho = 50$.

(d) AG, iteration #3: 324 nodes. Bin sizes $B_x = B_y = 10$, $\rho = 25$.

Figure 3: Examples of the grids generated on the same swath using a (a) fixed grid, (b) mixed grid, and (c)-(d) the second and third iterations of an adaptive grid where the first iteration uses a bin sizes $B_x = B_y = 40$. The fraction of clear pixels in domain for this image is 0.460.

Algorithm 2: Greedy algorithm for minimizing cloud coverage in satellite imagery

output: Plan $\Pi = \{p = (x, y) \in \mathbb{R}^2\}$ of sub-image centers which minimize cloud coverage

input : Satellite image swath data matrix $\mathbf{D}$, imager constraint parameters Θ, d_{max} maximum distance to search along image

1 $\Pi = \emptyset$
2 $t = 0$
3 **while** *True* **do**
4 $W : (x_{min}, x_{max}, y_{min}, y_{max}) \leftarrow$ getWindow(Θ, $y_S(t)$) // candidate points p across and along limits
5 **if** $y_{max} \geq d_{max}$ **then**
6 break;
7 **else**
8 $\mathbf{C} \in \mathbb{R}^2 \leftarrow$ getCloudiness(W, $\mathbf{D}$, Θ) // total cloudy pixels in sub-image centered at each candidate point p in W
9 $p_t \leftarrow$ closestLeastCloudyPoint($\mathbf{C}$, p_{t-1}) // point with least cloudy sub-image that has smallest L_1 distance to previous sub-image center
10 $\Pi \leftarrow \Pi \cup p_t$ // add point to plan
11 $t = t + 1$ // increment time step
12 **end**
13 **end**
14 **return** Π

more control over the placement of nodes but requires more hyper-parameters. A MG uses B_x and B_y to divide the image into bins, and computes the MFC in each of these. If the bin clarity is within an intermediate range $[c_{min}, c_{max}]$, then the bin is categorized as heterogeneous because it contains a mixture of both clear and cloudy pixels whose placement an optimal search may require the knowledge of. These heterogeneous bins are divided into smaller bins of size b_x across and b_y along. Furthermore, the same heterogeneity range of clarity $[c_{min}, c_{max}]$ is not suitable for all input images. Therefore, the actual MFC for the entire domain of an image is computed to separate sparse and dense cases (line 1 in Algorithm 3). In practice, better plans are found by using c_{min} is orders of magnitude smaller for images with a low domain MFC. Therefore, the two sets of bin sizes, two sets of clarity ranges, and a threshold for domain MFC are required as hyper-parameters for the MG method.

Adaptive grids are defined by bin sizes B_x and B_y which are reduced by a ratio of r_b per iteration. During each of these iterations, a new grid is created by placing nodes around each of the current plan's sub-image centers p. The grid nodes around each p must be no further than ρ in either the across or along directions. This grid span distance ρ is also reduced in each iteration by a ratio r_ρ. Figures 3c and 3d show two iterations of an adaptive grid where the bin size and grid span reduction ratios are $r_b = 1/2$ and $r_\rho = 1/2$ respectively. Note that the grid nodes must still obey constraints I and III, therefore a different number of nodes may exist at each time step. Additionally, two stopping criteria parameters are required i) ϵ which is the minimum improvement in MFC required to continue iterating, and ii) ρ_{min} which is the smallest grid span allowed. A new plan is accepted as long as the change in MFC (ΔMFC) does not decrease, which may occur if new grid points omit pre-

Algorithm 3: Fixed or mixed grid graph search algorithm for minimizing cloud coverage in satellite imagery

output: Plan $\Pi = \{p = (x, y) \in \mathbb{R}^2\}$ of sub-image centers which minimize cloud coverage
input : Satellite image swath data matrix $\mathbf{D}$, imager constraint parameters Θ, d_{max} maximum distance to search along image, H grid generation hyper-parameters

$MFC_{domain} \leftarrow$ getDomainMFC // only if MG
$N = d_{max}/(2d_y)$ // number of sub-images in plan
$L \leftarrow$ getNodeSetLimits(Θ, N) // limits of node sets S
$S \leftarrow$ getNodeSets($\mathbf{D}$, L, H, Θ) // grid of nodes separated into sets S
$\mathbf{G} \leftarrow$ discoverEdges(S, Θ) // find allowed edges and make DAG
$\Pi \leftarrow$ shortestDistance($\mathbf{G}$, Θ) // get plan by finding the shortest viable path in $\mathbf{G}$
return Π

viously detected optimal regions. If a new plan's MFC is worse than the best plan's MFC, then the new plan is rejected and another iteration is forced to plan around the same plan again but with smaller bin sizes and grid span. Therefore, the MFC of a plan monotonically increases with iterations of the adaptive grid scheme given in Algorithm 4. The initial plan is found by placing nodes around a nadir ground track plan Π_N with a grid span of $\rho = \infty$. For bin size $B_x = B_y = B$ and N sub-images, the number of nodes processed in an iteration i of the adaptive method is of the order $\mathcal{O}(4N\rho^2(r_\rho/r_b)^{2i}/B^2)$, therefore when $r_\rho = r_b$ the number of nodes processed per iteration remains approximately constant, as is the case in Figure 3c-3d where $r_\rho = r_b = 1/2$.

Results

Examples of plans are shown in Figure 1, where all three methods find plans with MFC far greater than the domain MFC of 0.16. Performance of algorithms is not only dictated by the domain MFC, but also the distribution of clear pixels. Particularly, cases with large and contiguous areas of moderate to high clarity are easiest to plan. Therefore the greedy solution is a natural approach to discovering the least cloudy areas of a swath in scenarios where there is either very low cloud coverage or an even distribution of clear areas. The relationship between MFC found by a plan and the MFC in domain is shown in Figure 4a where an exponential model is fit to the MFC found by optimal, greedy, and AG graph search methods. Figure 4b shows that the graph search performs well for both sparse and dense cases which have very low and high MFC in domain respectively. It is the intermediate range of clarity (MFC in domain = [0.2, 0.5]) which leads to bigger differences between the graph search and the optimal plans. In contrast, the greedy method's differences from the optimal plans are less correlated with MFC in domain (Figure 4b).

Algorithm 4: Adaptive grid graph search algorithm for minimizing cloud coverage in satellite imagery

output: Plan $\Pi = \{p = (x, y) \in \mathbb{R}^2\}$ of sub-image centers which minimize cloud coverage
input : Satellite image swath data matrix $\mathbf{D}$, imager constraint parameters Θ, d_{max} maximum distance to search along image, H grid generation hyper-parameters

$N = d_{max}/(2d_y)$ // number of sub-images in plan
$L \leftarrow$ getNodeSetLimits(Θ, N) // limits of node sets S
$\Pi_{best} \leftarrow$ getInitialPlan($\rho = \infty$, Π_N, L) // initial plan
while $\rho > \rho_{min}$ & $\Delta MFC > \epsilon$ **do**
  $S \leftarrow$ getNodeSetsAroundPlan($\mathbf{D}$, L, ρ, B_x, B_y, Π_{best}) // grid of nodes around current plan centers p, separated into sets S
  $\mathbf{G} \leftarrow$ discoverEdges(S, Θ) // find allowed edges and make DAG
  $\Pi_i \leftarrow$ shortestDistance($\mathbf{G}$, Θ) // get new plan by finding the shortest viable path in $\mathbf{G}$
  if $\Delta MFC < 0$ **then**
    ΔMFC$= \infty$ // reject new plan and force next iteration with same plan
  else
    $\Pi_{best} = \Pi_i$ // accept new plan
  end
  $(B_x, B_y) = r_b \cdot (B_x, B_y)$ // reduce bin size
  $\rho = r_\rho \rho$ // reduce grid span
end
return Π

In addition to optimal, greedy, and nadir results, the FG graph search is run with four different square bin sizes $B_x = B_y = 10, 20, 30, 40$. Two MG graph search cases are run by mixing large square bins of size $B_x = B_y = 40$ with smaller square bins of size $b_x = b_y = 20$, and bins of size $B = 30$ with small bins of size $b = 10$. Finally three AG graph searches are performed using square bins of initial size $B_x = B_y = 20, 30, 40$, and $\rho = 50$, $r_b = 1/2$, $r_\rho = 1/2$. Table 1 and Figure 5 show two main metrics which include i) runtime and ii) percentage of the optimal MFC found over the 50-image dataset.

The graph search results offer greater MFC in exchange for runtime when bin sizes are reduced. For instance the FG 10×10 case has the highest MFC values of all FG cases, but is an order of magnitude slower than even the FG 20×20 simulations. To alleviate these extremes, the results of the MG simulations lie between results of FG cases with bin size B and FG cases with bin size b, but the MG method re-

quires careful tuning of hyper-parameters which define heterogeneous areas of the swath. In comparison to the other techniques, the AG graph search simulations yield higher median MFC over the dataset in less runtime. Specifically, the AG 20×20 case has the highest minimum (96%) and median (99.6%) percent of the optimal MFC (Table 1 in roughly fourth of the time of the FG 10×10 method.

Future work

The utility of algorithms described here can be expanded by by additional constraints which make the observation planning problem more realistic. For instance, spacecraft energy and data storage constraints could alter number of observations. Further, the slew model can be modified to better address slew angle constraints and varying degrees of agility in the across and along tracks. Imager performance limitations can be better incorporated by using a utility map which discounts the value of observations that are further from nadir. Rewards for observations in areas of high interest may be used to increase the overall value of a plan. The current plans require observations at a constant cadence, which could be generalized to non-constant intervals and multiple observations of the same target area. As a part of ongoing work, the algorithms provided in this work are being ported to flight hardware to study mission applicability, and are being generalized to broader application scenarios by the modular addition of aforementioned constraints and utility maps.

Acknowledgements

The research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration (80NM0018D0004).

References

Ackerman, S., and et al. 2015. *MODIS Atmosphere L2 Cloud Mask Product*. NASA MODIS Adaptive Processing System. USA: Goddard Space Flight Center.

Beaumet, G.; Verfaillie, G.; and Charmeau, M.-C. 2011. Feasibility of autonomous decision making on board an agile earth-observing satellite. *Computational Intelligence* 27(1):123–139.

Chien, S.; Cichy, B.; Davies, A.; Tran, D.; Rabideau, G.; Castano, R.; Sherwood, R.; Mandl, D.; Frye, S.; Shulman, S.; et al. 2005. An autonomous earth-observing sensorweb. *IEEE Intelligent Systems* 20(3):16–24.

He, L.; Liu, X.-L.; Chen, Y.-W.; Xing, L.-N.; and Liu, K. 2019. Hierarchical scheduling for real-time agile satellite task scheduling in a dynamic environment. *Advances in Space Research* 63(2):897–912.

King, M. D.; Platnick, S.; Menzel, W. P.; Ackerman, S. A.; and Hubanks, P. A. 2013. Spatial and temporal distribution of clouds observed by modis onboard the terra and aqua satellites. *IEEE transactions on geoscience and remote sensing* 51(7):3826–3852.

Moy, A.; Yelamanchili, A.; Chien, S.; Eldering, A.; and Pavlick, R. 2019. Automated scheduling for the oco-3 mission. In *International Workshop for Planning and Scheduling for Space (IWPSS 2019)*, 195–203.

Paul, F.; Bolch, T.; Kääb, A.; Nagler, T.; Nuth, C.; Scharrer, K.; Shepherd, A.; Strozzi, T.; Ticconi, F.; Bhambri, R.; et al. 2015. The glaciers climate change initiative: Methods for creating glacier area, elevation change and velocity products. *Remote Sensing of Environment* 162:408–426.

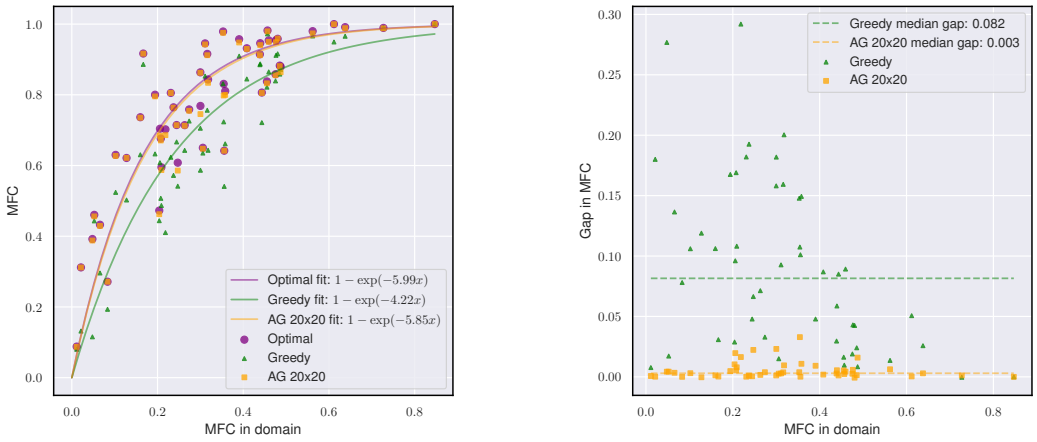
Platnick, S.; King, M. D.; Ackerman, S. A.; Menzel, W. P.; Baum, B. A.; Riédi, J. C.; and Frey, R. A. 2003. The modis cloud products: Algorithms and examples from terra. *IEEE Transactions on geoscience and Remote Sensing* 41(2):459–473.

Sandford, M. W.; Thompson, D. R.; Green, R. O.; Kahn, B. H.; Vitulli, R.; Chien, S.; Yelamanchili, A.; and Olson-Duvall, W. 2020. Global cloud property models for real time triage onboard visible-shortwave infrared spectrometers. *Atmospheric Measurement Techniques Discussions* 1–17.

Wang, J.; Demeulemeester, E.; Hu, X.; and Wu, G. 2020. Expectation and saa models and algorithms for scheduling of multiple earth observation satellites under the impact of clouds. *IEEE Systems Journal* 14(4):5451–5462.

Yang, G.; Li, B.; Ji, S.; Gao, F.; and Xu, Q. 2013. Ship detection from optical satellite images based on sea surface analysis. *IEEE Geoscience and Remote Sensing Letters* 11(3):641–645.

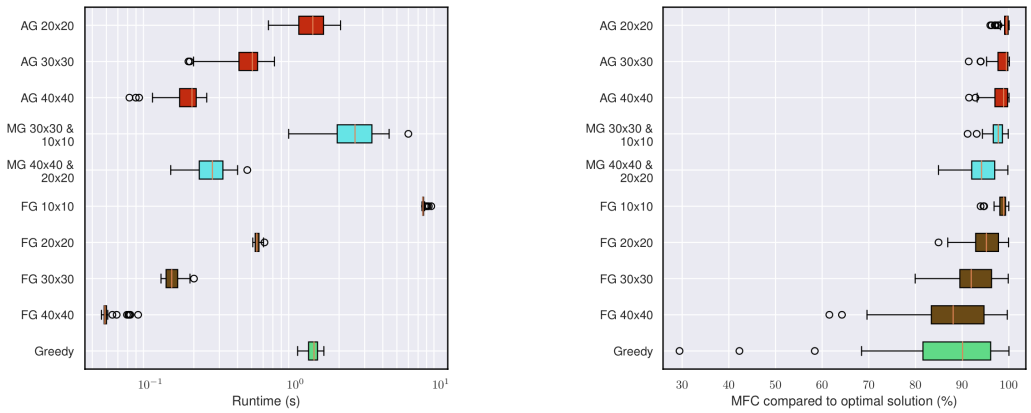
Yelamanchili, A.; Chien, S.; Moy, A.; Shao, E.; Trowbridge, M.; Cawse-Nicholson, K.; Padams, J.; and Freeborn, D. 2019. Automated science scheduling for the ecostress mission. In *International Workshop for Planning and Scheduling for Space (IWPSS 2019)*, 204–211. Also appears at ICAPS SPARK 2019.



(a) Mean fraction of clear pixels (MFC) per MODIS swath compared to MFC in the imageable domain.

(b) Difference in MFC per MODIS swath compared optimal solution.

Figure 4: Performance of surface target planning algorithms over the 50-image MODIS dataset compared to MFC in domain. The AG algorithms use $\rho = 50$, $r_\rho = 1/2$, and $r_b = 1/2$.



(a)

(b)

Figure 5: Performance of surface target planning algorithms over the 50-image MODIS dataset measured by runtimes (5a) and MFC compared to optimal solution (5b). The AG algorithms use $\rho = 50$, $r_\rho = 1/2$, and $r_b = 1/2$.

Table 1: Median performance metrics over the 50-image dataset. % improvement is relative to the greedy method

Method	Median MFC	Median slew across (km)	Median slew along (km)	Max runtime (s)	Min % of optimal MFC	Median % of optimal MFC
AG 20x20	0.803	659	308	2.0E+00	96.0	99.6
AG 30x30	0.800	634	299	7.2E-01	91.4	99.5
AG 40x40	0.793	616	298	2.5E-01	91.5	98.9
MG 10x10 & 30x30	0.799	650	290	6.0E+00	91.2	97.7
MG 20x20 & 40x40	0.753	640	260	4.7E-01	84.9	94.2
FG 10x10	0.805	660	310	8.6E+00	94.0	98.9
FG 20x20	0.769	650	280	6.1E-01	84.9	95.2
FG 30x30	0.740	645	325	2.0E-01	79.9	91.9
FG 40x40	0.702	570	260	8.3E-02	61.5	88.1
Greedy	0.686	505	247	1.6E+00	29.4	90.1
Nadir	0.293	0	0	1.7E-02	0.1	38.2
DP optimal	0.809	657	306.5	-	-	-