Where is my coverage? Using explainable automated scheduling to inform mission design of an Earth-observing constellation

Adrien Maillard, Christopher Wells, Shadi Oveisgharan, Paul Rosen, Steve A. Chien
Jet Propulsion Laboratory, California Institute of Technology
4800 Oak Grove Drive
Pasadena, California 91109
adrien.maillard@jpl.nasa.gov, christopher.r.wells@jpl.nasa.gov

Abstract

The Surface Deformation and Change Mission Architecture Study (SDC) aims at designing the next synthetic aperture radar mission to measure geophysical properties of the Earth's land surfaces, focusing on surface deformation and disruptive change. Multiple mission architectures are evaluated in a rigorous selection process. One step of this process is the generation of realistic mission schedules that are used to compute the science performance of each architecture. For that purpose, we adapt and use CLASP, an automated scheduling software used in the design and operations of many missions. We describe the specific constraints linked with the SDC study. We highlight the challenges with using automated scheduling software to create explainable plans that the science team can understand and trust. We also detail current and planned changes to the scheduling software for that purpose.

Introduction

NASA has recently launched the Surface Deformation and Change Mission Architecture Study (SDC) to address the research and applications community's needs in the United States over the next few decades. The SDC mission is being developed to gather data on global surface displacement and disruptions with a precision of up to one millimeter. These measurements will be useful to both scientific communities interested in estimating global biomass and soil moisture and emergency responders dealing with geohazards.

In the past, two surveys had recommended missions using synthetic aperture radar (SAR) to measure geophysical properties of land surfaces. The 2007 survey recommendation led to the development of the NASA-ISRO SAR (NISAR) mission, which is close to launching. The 2017 Decadal Survey recognized the importance of continuing these measurements beyond NISAR, with an emphasis on surface deformation and disruptive surface change. Surface Deformation and Change observables were defined as a top priority for the next decade (National Academies of Sciences, Engineering, and Medicine 2018). However, unlike in 2007, the 2017 survey only prescribed the observations without providing any guidance on the mission's underlying architecture or instrument suite and concept of operations.

The goal of the SDC mission study is the evaluation of mission architectures that would support SDC observables and provide the most value to the diverse science and applications communities it serves.

There is a systematic process that the architecture team has derived for defining and evaluating mission architectures (Horst et al. 2021). After an initial brainstorming period that involved identifying desired capabilities and mission architecture classes that might deliver those capabilities the team went through an iterative process that considers a broad number of architectures at a very cursory level, to be followed by a narrowing of the candidate architectures and more detailed analysis on the remaining architectures during the second phase of the study. In each case, the criteria for selection or rejection is a cost-benefit analysis of the cost of the mission compared to the science value it delivers.

One component of the process is the evaluation of the Science performance of the architecture. The SDC Science Performance Model consists of a set of tools that calculate spatially-varying measurement uncertainties for a given set of point target locations on the surface of the Earth. The model takes into account instrument parameters and orbital calculations combined with a mission plan, information about global conditions of the Earth's surface (e.g., terrain type, snow and vegetation cover, topography), and time-dependent models that represent propagation delays through the troposphere and ionosphere. The tool combines all the aforementioned information to calculate seasonal error statistics for a set of targets, which can then be combined into long-term performance estimates for a given architecture.

Our goal has been to produce the mission plans for each architecture so they can be used to compute the Science performance of each architecture. For this work, we build on top of CLASP, an existing mission scheduling software that has been used multiple times for mission design and mission operations.

In the remainder of this paper, we describe the base software and its algorithms as well as the specific constraints of the scenarios for the SDC study. We describe the lifecycle of a scenario and how we translate the intent of the Science team into inputs for our software. Another section addresses the interpretation of the produced mission plans, which are the result of thousands of automated decisions.

We focus on the inherent difficulty of this process and the approaches we have put into practice to help, highlighting the need for decision-making software to provide some level of explainability.

CLASP

CLASP (Knight and Chien 2006) is a long-range scheduler for space-based or aerial instruments that can be modelled as pushbrooms, that is 1-dimensional line sensors dragged across the surface of the body being observed. It addresses the problem of choosing the orientation (if steering is possible) and on/off times of a pushbroom instrument or collection of pushbroom instruments such that the schedule covers as many target points as possible, without oversubscribing memory and energy. Orientation and timing of observations are derived from geometric computations that CLASP performs using the SPICE ephemeris toolkit (Acton Jr 1996). Inputs of CLASP notably include (1) campaign files describing desired geographical areas to be observed along with required instrument mode and geometric and temporal constraints. Each campaign is assigned a priority which determines the order in which resources (steering capability, energy, memory) will be allocated to it; (2) constellation definition which can be made of one or more spacecraft, each of which has one or more body-mounted instruments with specific swaths modeled as pushbroom sensors. Geometry of the sensors are parameterized by minimum and maximum look angles as angles rotated about the velocity vector of the spacecraft from the nadir look vector, looking 90 degrees off of velocity. Figure 2 shows how we go from a visibility, computed during the discretization phase, to an observation by choosing a roll angle and a sensor (there might be several sensors/modes with different swaths). The discretization approach is shown schematically in Figure 1. After discretization, an algorithm goes through all target points in priority order and greedily schedules observations in a non-chronological manner, propagating constraints at each insertion in the schedule. A more thorough description of the combinatorial coverage problem as well as algorithms to solve it are described in (Maillard, Chien, and Wells 2021).

Using automated planning and scheduling tools for mission design is not novel, such as in the study for the Pluto Fast Flyby (Sherwood et al. 1997) or more recently D-SHIELD (Nag et al. 2020) for constellations. (Sipps and Magruder 2023) developed a constellation analysis tool that uses a scheduling algorithm called envelope grid-point approach (EGPA). Their constellation design approach has a greater focus on orbit design and optimization and does not appear to currently incorporate resource constraints (data volume, energy, etc.). In (Schaffer et al. 2016) or (Do et al. 2013), the authors use a systematic approach and automated scheduling tools to explore the space of possible configurations and outline the design which would yield the best science return. CLASP has been used for scheduling science observations in recent missions involving instruments mounted on the ISS such as the Orbiting Carbon Observatory-3 (Yelamanchili et al. 2021a), EMIT (Yelamanchili et al. 2021b), and ECOSTRESS (Yelamanchili et al. 2019) missions. It was previously used in the operations of the Intelligent Payload Experiment Cubesat Mission (Doubleday et al. 2015), in the mission design and operations of the NASA-ISRO Synthetic Aperture Radar Mission (Doubleday 2016), in coverage analysis for the Europa Clipper and Jupiter Icy Moons Explorer (Troesch, Chien, and Ferguson 2017), and was proposed for operations of the Thermal Emission Imaging System on the Mars Odyssey spacecraft (Rabideau et al. 2010).

Each mission has some desired behaviors, inputs, and outputs that are not directly available in the base CLASP software. Fortunately, CLASP has been designed to be adapted to these mission-specific needs. For SDC, we added several algorithmic layers to (1) reduce slewing (2) allow interferometric observations and other combinations of observations (3) allow energy modeling (4) perform latency analyses. We detail those in the following sections.

Lifecycle of a scenario

As mentioned before, the SDC project as a whole has a process for selecting and evaluating architectures. We will now see how we process an architecture at the mission planning level. First, we receive an architecture definition for the case, such as seen in Table 1. As described in another section, we automated the generation of inputs for CLASP from these definitions. What is not shown in Table 1 is the science intent or campaign definitions, that is the sets of polygons and associated illumination and temporal constraints that we want to observe. We built a base set of campaigns from the set of science campaigns currently baselined for NISAR and for the sake of being able to compare the outputs of all the cases, we usually only modify this base set of campaigns in two ways: (1) scale up the desired number of observations to fit the increased capacity of a constellation (compared to one NISAR spacecraft) (2) add specific campaigns fitting a scenario (e.g. disaster areas, formation-flying requirements for a given campaign). These requirements tied to a specific scenario sometimes require new software development, either new scheduling algorithms or new output plots. Then the case can be run on a planning horizon corresponding to the repeat cycle of the base spacecraft (10 or 12 days) and outputs are examined. When new algorithms are developed, particular attention is given to the correctness of the behavior. Observation plans containing hundreds of observations are examined in Google Earth with the science team. We also look at various specialized plots for coverage, duty cycle, slewing, and campaign satisfaction. In a coming section, we describe how this step can be difficult and how we manage it. Then the science performance tool is run on the case and the results get an external evaluation from the science team. For cases that are downselected, we want to get more details and thus the case is run again with slight variations and adjustments.

Scheduling algorithms for SDC

We developed several additions to the base scheduling algorithm of CLASP.

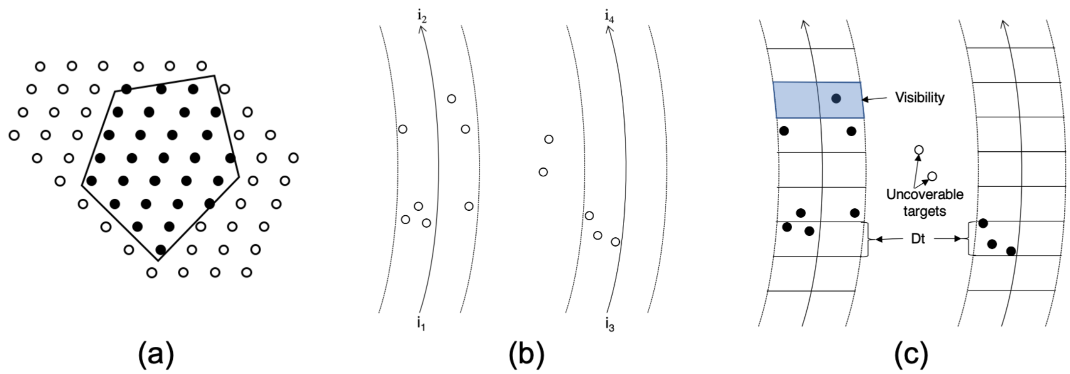

Figure 1: Schematic view of the discretization process in CLASP. (1) A grid of point is projected onto the body and intersected with target polygons which result in sets of target points. (2) The footprint of the instrument onboard the spacecraft is projected onto the body and intersected with the target points. (3) This footprint is time-discretized in fixed increments. The resulting polygon is intersected with the target points and called a *visibility*. Visibilities that do not contain any targets are discarded.

Name	# of s/c	orbit configuration	revisit time (days)	elevation beamwidth of each s/c (deg)	swath start look angle of each s/c (deg)	steerable	duty cycle constraint
L1A	1	NISAR	12	12	30	No	No
L2A	2	NISAR, 6-days apart	12	12	30	No	No
L4A	4	Sentinel-1, 3 days apart	12	12	30	No	No
L6A	6	NISAR, 2-days apart	12	2	30, 32, 34, 36, 38, 40	Yes, 0.04 deg/sec	No
L5A	5	L5A, 2-days apart	10	3	30, 32.4, 34.8, 37.2,39.6	Yes, 0.04 deg/sec	No
L4C	4	Inclined, 3-days apart	12	3	30,33,36,39	Yes, 0.04 deg/sec	No
L12C	12	NISAR, 1-day apart	12	3	30, 30, 30, 33, 33, 33, 36, 36, 36, 39, 39, 39	Yes, 0.04 deg/sec	15%
L12D	12	NISAR, 1-day apart	12	2	30, 30, 32, 32, 34, 34, 36, 36, 38, 38, 40, 40	Yes, 0.04 deg/sec	15%
L4B	4	NISAR, 3 days apart	12	3	30, 33, 36, 39	Yes, 0.04 deg/sec	No
L9A	3	NISAR, 4-days apart	12	4	30, 34, 38	Yes, 0.04 deg/sec	No
L6F	2	Inclined, 6-day apart	12	6	30, 36	No	No
L6C	2	Sentinel-1, 6-day apart	12	12	30	No	No
L18A	6	NISAR, 2-days apart	12	3	30, 30, 33, 36, 36, 39	Yes, 0.04 deg/sec	15%
L8A	8	NISAR, 6-hours apart	12	1.5	30, 31.5, 33, 34.5, 36, 37.5, 39, 40.5	Yes, 0.04 deg/sec	No
L8D	8	NISAR, 6-hours apart	12	2	30, 32, 32, 34, 36, 38, 38, 40	Yes, 0.04 deg/sec	No
L8E2	8	NISAR, 6-hours apart	12	2	30, 32, 32, 34, 36, 38, 38, 40	Yes, 0.04 deg/sec	18%
L6C2	6	2 formation-flying constellations of 3 s/c each (1 mothership + 2 co-flyers)	12	12 for mothership and 4 for co-flyers	30 for motherships 30 and 34 for co-flyers	Yes for co-flyers	20% for mothership

Table 1: Evaluated architectures.

Reduce slewing via a multipass algorithm

In several architectures, the spacecraft are steerable, see Table 1. We have seen that the base algorithm of CLASP goes through targets by priority order and schedules observation in a non-chronological way. It does not try to actively minimize slewing but considers it is a constrained resource. This combination can lead to the spacecraft slewing a lot between targets (see top plot of Figure 3) which, as we cannot take observations while slewing, results in a suboptimal use of possible observation time. To remedy this, a multipass algorithm was developed and the the space of slewing angle has been discretized. Each spacecraft is assigned a set of look angles it can slew to (instead of a continuous range) and a preferred angle (a lane in other words). During both passes, the algorithm goes through all targets, as the original algorithm, but during the first pass, it can only schedule observations in the preferred lane of each spacecraft. The spacecraft are only allowed to slew during the second pass. Thus if a spacecraft slews, it means there were no more targets in its lane at that time. That may be the case if an area that has a lot of overflights has already been observed enough or along coastlines. It can be seen in Figure 3 that this mechanism significantly reduces the number of slews and increases the amount of time spent observing target areas.

Interferometric and combined observations

The CLASP algorithm schedules observations one at a time with the implicit assumption that the reward is linear with the number of observation realized up to the repetitivity specified by the user. In the context of interferometric SAR, the reward might be obtained only when we can assemble two observations taken at different times. There is a general assumption that observations produced during one planning horizon can be paired with observations produced during another planning horizon to ultimately produce interferometric pairs. But in some cases, we want to produce complete interferometric pairs from observations within the same planning horizon. We augmented the CLASP algorithm with a mechanism that schedules observations in pairs for certain

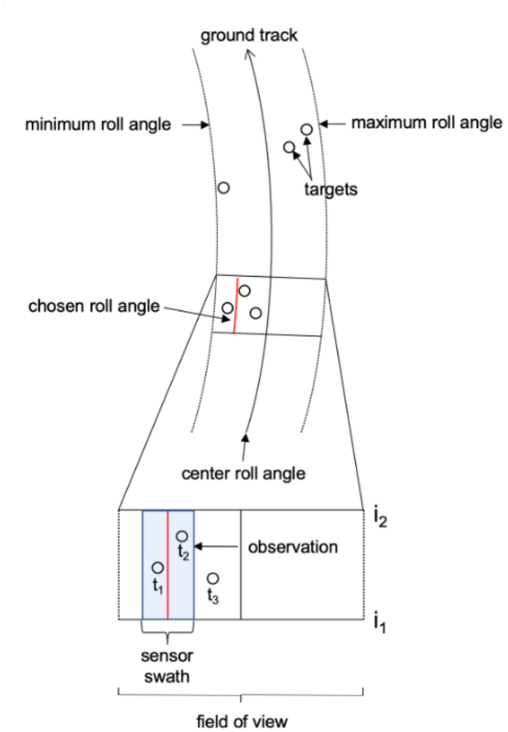

Figure 2: How CLASP produces an *observation* from a *visibility* by choosing a roll angle at a particular timestep. Roll angle is a decision variable at each timestep. CLASP uses a least-commitment strategy to assign this variable by using a timeline with propagation which ensures the constraint is never violated.

campaigns. This requires more search and backtracking than previously because for a given target, it is necessary to find two observations taken in the same orbital conditions that would both satisfy constraints.

Later, we added a capability to schedule simultaneous observations for the case of formation-flying spacecraft. In this configuration, three spacecraft: a mothership, a leader, and a follower are offset by tens of seconds and are able to observe the same point on the ground (by offsetting the pitch of the leader and follower spacecraft). In this context, the reward for a certain campaign is only obtained when simultaneous observations are taken of the same target. In the *focused* mode, the three spacecraft must observe the same point on the ground at the same time. In *extended* mode, the instrument on the mothership is off and the leader and follower are observing the ground targets side-by-side. Figure 6 shows a pass over South America where a formation-flying constellation is observing campaigns with different mode requirements.

Latency analysis

Urgent response in case of disaster is another use case for the architecture evaluation in the SDC study. After a disaster has happened, we want to be able to use the spacecraft to image the affected area as soon as possible and as much as possible.

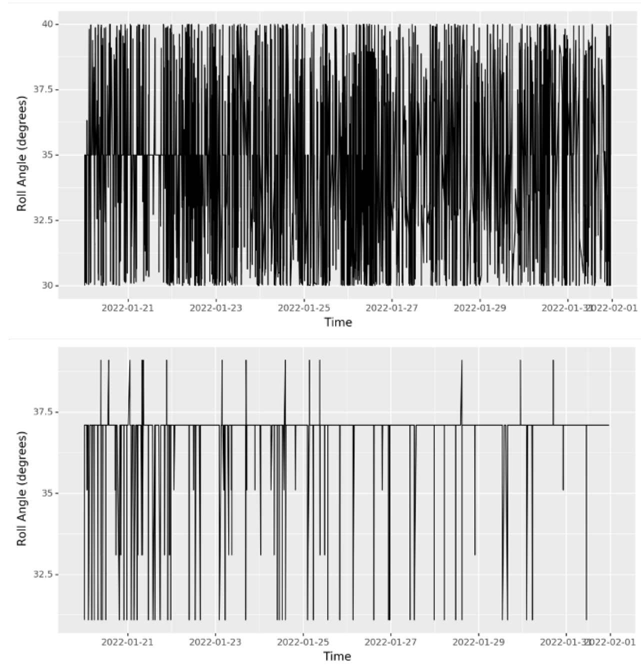

Figure 3: Roll angle as a function of time for one spacecraft member of a 6-spacecraft constellation on a 12-day planning horizon. Top: without 2-pass algorithm. Bottom: with 2-pass algorithm. 10-fold decrease in number of slews

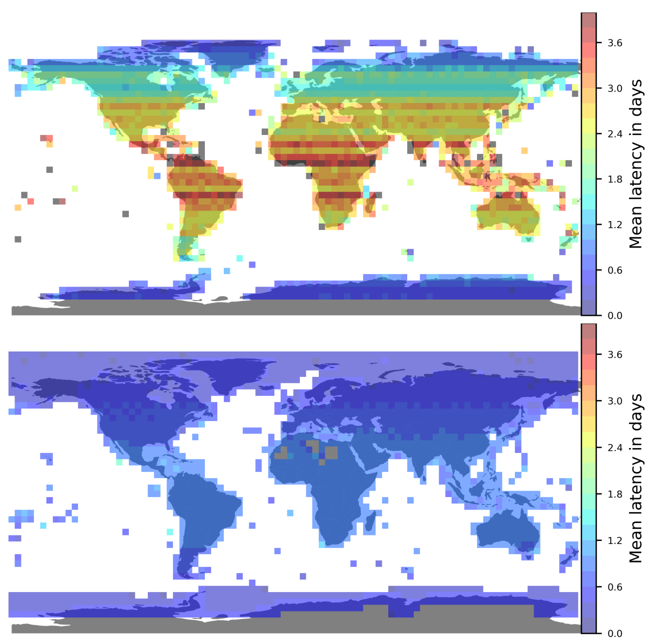

Figure 4: Latency analysis for two architectures. Lower values are better. Top: 1 full-swath spacecraft. Bottom: 6 1/6-swath steerable spacecraft.

To compare the capacities of architectures in terms of urgent response, we designed a *latency analysis*. We generate a pseudo-random latitude/longitude point p in the landmass where the disaster is supposed to happen. Then we generate a pseudo-random time t in the planning horizon to represent the time at which the disaster occurs. Then we compute the latency, which is the duration between t and the start time of the next opportunity to image the disaster target p for any spacecraft of the constellation. We do this thousands of times and we average the results geographically. Figure 4 shows the comparison between two architectures of equal imaging capability (in terms of swath) but with different numbers of spacecraft. Unsurprisingly, by having six steerable spacecraft spread over the planning horizon instead of one, we are able to significantly reduce the latency for imaging disaster sites.

Automated generation of scheduling inputs and figures

Due to SDC intending to analyze and compare different scenarios, we decided early on to develop some infrastructure around the existing CLASP software to make it easier to construct, run, and analyze the results of new scenarios. We created some pre-processing code that takes in a declarative definition of a scenario (number of spacecraft, the relationships between their orbits, campaigns, etc.) and generates inputs to CLASP that would normally be written or generated manually (orbit files, spacecraft and sensor definitions, etc.). In order to specify the consteallations to CLASP, we need to have ephemeris for each of the individual spacecraft. Using SPICE (Acton Jr 1996) we generate offset ephemeris from base orbits (e.g. NISAR or L5A or Sentinel-1 in Table 1) that are generated at undisclosed location (for review). We also created some post-processing code that takes data outputted by CLASP and automatically produces data visualizations that we and the science team can examine. (see Figure 5 for more information)

Having this automation around CLASP has allowed us to more rapidly run and analyze the results of different scenarios and iterate as we found changes that we wanted to make to the scenarios and as new architectures were proposed.

Explainability and dialogue between mission planning and science teams

Why it is difficult

When generating and presenting mission plans for various scenarios we found it important to produce plans that are correct and that the science team could understand and trust. Since the mission plans would be used to make decisions about which architectures to study and eventually select, ensuring that the mission plans are correct is important in order to make sure that those decisions are properly informed. Since the science team will be relying on the mission plans being correct, it is important that we ensure that the results are explainable and trustworthy.

In generating mission plans, we would occasionally notice unexpected and problematic effects in the results (gaps in coverage, spacecraft slewing too often, etc.). In such cases we attempt to identify the cause of the effect with it typically falling into one of three types: (1) bugs in the scheduling software, (2) problems with the scheduling inputs, and (3) differences between the science team's intent and what was implemented in the scheduling software and inputs. For software bugs it proved to be important to both find and fix the bug and to also do verification to prove to the science team that the bug was fixed and thus that the new results were trustworthy. For problems with the scheduling inputs we find and correct the defect, and if it was not an effect we had seen previously we would show and explain to the science team how the defect caused the unexpected effect. For differences between science team intent and scheduling inputs, we would discuss the results with the science team in order to identify the difference in intent and correct the scheduling inputs or software to align with those intents.

We also found that communicating to the science team how the scheduling software made the scheduling decisions that it did to be difficult. The underlying greedy scheduling algorithm in CLASP has impacts due to it's greedy decisions that can be clear to people with a background in AI planning and scheduling algorithms or sequential decision algorithms, but can be difficult for others to understand and intuit. In particular, the strict campaign priority scheme that the greedy algorithm follows can yield unintuitive coverage effects.

When attempting to explain coverage effects in a particular mission plan, there are two types of constraints that differ in how difficult they are to explain: (1) local constraints and (2) global constraints. Some scheduling constraints like slewing have effects that are mostly local in time and thus some of their effects can often be explained by examining a particular segment of the mission plan that is clearly effected by the constraint. Such local effects can result in some global patterns (ex. characteristic gaps in coverage) that can be explained by showing such an example case and building up to a more global explanation (see Figure 6 for an example). However, the effects of more global constraints such as energy and duty cycle can be more difficult to explain, often requiring us to detail how CLASP's greedy algorithm would impact which geographic areas are scheduled first (and thus have more energy and duty cycle available). Additionally, this can be further complicated when there are multiple constraints that are interacting with each other.

Another complicating factor is that the relationship between the orbital parameters and the ground track can be unintuitive. There are some heuristics that can be used to reason about what the ground track will be like based on the orbital parameters, but once you want to start reasoning about coverage achieved by the sensors on a global scale it can be more difficult. For example, when we are examining the coverage for a scenario we often try to reason about how much coverage we should get over different ranges of latitude. For most architectures, at the Equator there are some small overlaps between the swaths of the spacecraft (when pointed at their preferred lanes). At more extreme latitudes the swaths of the spacecraft in the constellation start to overlap with each other more, and in cases like L6A (see Table 1) allow us to achieve 1 observation at all longitudes without needing to observe with all 6 spacecraft. However, reasoning about how much overlap to expect at intermediate latitudes is non-trivial.

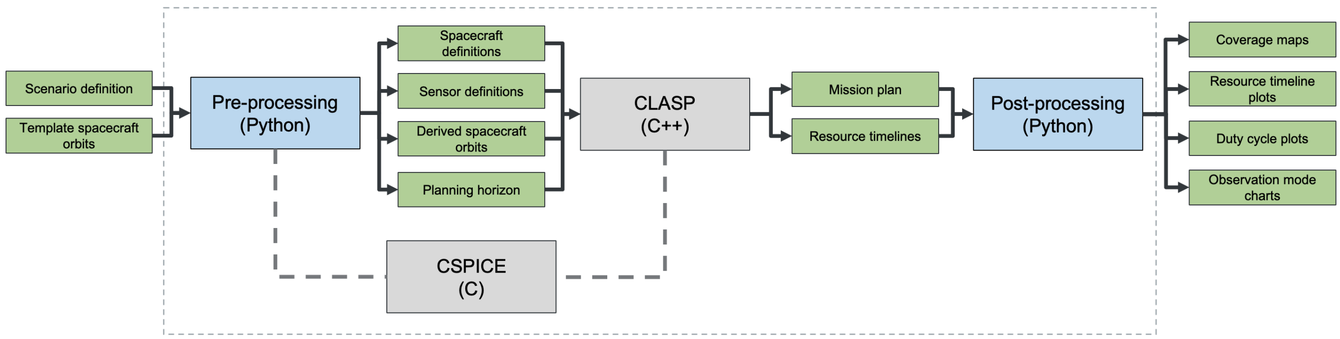

Figure 5: Diagram of the architecture and data flow of the SDC CLASP adaptation. Notice that the pre-processing stage generates the many different inputs that CLASP needs, including using CSPICE to generate new orbits based on template orbits. Both CLASP and CSPICE are existing tools that we leveraged, though with CLASP we made changes to it to support new scheduling algorithms that we needed.

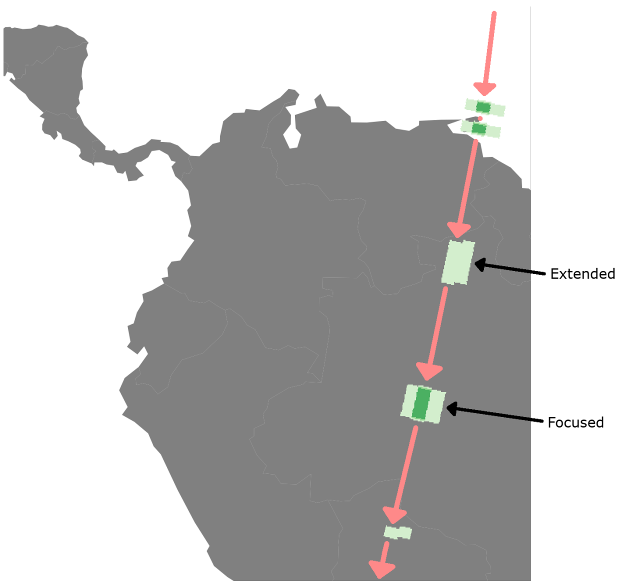

Figure 6: An example of a descending pass over South America for the L6C2 architecture where the constellation switches repeatedly between focused and extended mode (shown by the swaths overlapping and the two neighboring swaths respectively), which results in observation time lost due to slewing. Repeated occurrences of overflights like this lead to a noticeable pattern of coverage gaps over a 12 day period.

When defining which geographic areas that we want the spacecraft to observe, we have to assign each such campaign a priority level, specify how many observations we want of the area over our planning horizon, and specify a radar mode to observe the area with. Many of the scenarios we are running have a considerable number of different campaigns, some of which overlap geographically while requiring different radar modes. In such geographic areas, the priorities of the campaigns will determine which radar mode will be prioritized, with the other mode(s) allowed to be used

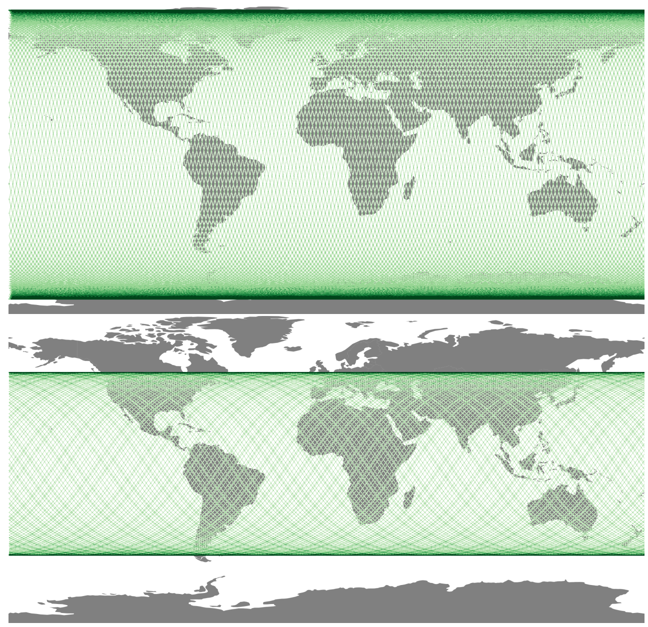

Figure 7: Examples of 12 days of the ground track for two different orbits. Top: Ground track of NISAR's reference orbit. Bottom: Ground track of the ISS' orbit.

if there are remaining visibilities that can be allocated to cover the area in that mode. Additionally we often have campaigns that ask for more than one observation or even "as many observations as possible" (the latter encoded as a very large number). In those cases, the exact priority ordering of campaigns is important, as if campaigns that request a lot of observations are made higher priority than other campaigns they can result in those lower priority campaigns getting too few observations.

Use cases of explainability and solutions

Slewing In a previous section, we have seen how a two-pass algorithm has reduced the number of slews that the

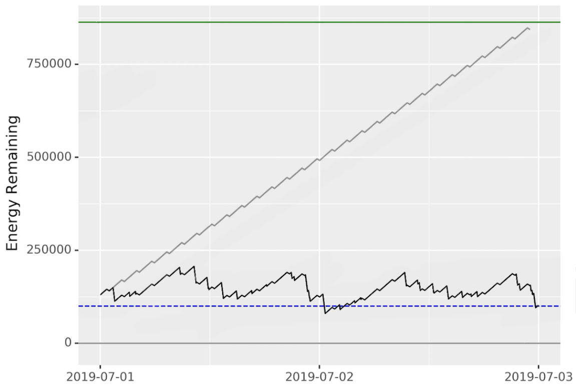

Figure 8: Energy diagrams. Green line is the maximum capacity of the batteries. Blue line is the minimum handover level of energy. Grey line: energy timeline simulated before scheduling observations. The energy is increasing thanks to solar panels. The dips are due to eclipses. Black line: energy timeline simulated at the end of the scheduling process. It can be seen that the minimum handover level of energy has been reached and it is probable that no other observation can be inserted in the plan without violating this constraint.

spacecraft perform in a mission plan. CLASP models the slewing of spacecraft as a resource timeline. The timeline maintains the minimum and maximum attainable slewing at each timestep. At the beginning of the planning horizon, the schedule is empty and thus there is no constraint on the timeline. Adding an observation to the schedule corresponds to constraining the slewing angle for the duration of the observation. Then, attainable angles are propagated outwards from the observation into the future and the past, taking into account the slew rate. This least-commitment approach is similar to approaches of iteratively reducing the domain of decision variables in constraint satisfaction. This algorithmic approach has obvious advantages during scheduling but is hard to explain when trying to relate it with what is seen in the observation plan. This is why we developed diagrams showing the current pointing (see Figure 3) as well as showing the angle bounds for each timestep. While the observation plans do not explicitly show slewing periods, we can follow them on these diagrams if necessary.

Aggregation of campaign definitions A campaign consists of a set of polygons (or points) associated with a instrument mode, a priority, and illumination and temporal constraints. Each campaign is defined in a Keyhole Markup Language (KML) file. As there might be tens of these campaigns covering the globe, it was difficult to get a clear global picture as one would have to look at each individual KML file.

In a previous section we discussed the explainability challenges with large complex campaign sets. In order to make understanding such campaign sets easier we developed software that can aggregate all target polygons and campaigns into one KML file which you can examine to see the geographic relationships between the campaigns.

In scenarios with one or more resource constraints, there is usually a point during the scheduling process when the scheduler cannot insert more observations without violating a resource constraint. It can then be difficult to understand why some areas have not been observed without having a good understanding of the campaigns and their relative priorities. In the case that an area is starved for coverage, the campaign(s) that are using up the resource are certainly higher priority, but are not necessarily geographically close if the resource is global in effect (such as energy). Scheduling an observation can impact the ability to schedule observations both before and after it. To help in reasoning about such cases we developed an output spreadsheet that lists all of the campaign definitions which allows us to quickly see how the campaigns' priorities relate to one another.

Future work

Targetpoint scheduling failure reason heatmaps

Currently CLASP tracks some information internally on why particular targetpoints have failed to be scheduled. When attempting to schedule observations for a targetpoint CLASP will check if a particular observation violates any scheduling constraints (ex. data volume, illumination constraints, duty cycle, etc.) and if that observation violates a constraint then the corresponding lookup table "constraint bucket" for that targetpoint will be incremented. Currently CLASP does not produce any output product to allow for analyzing this "failure reasons" information for explainability purposes. We would like to add the capability for CLASP to export this information so that we can generate geographic heatmaps showing where targetpoints are failing to be scheduled for each different type of failed constraint. This would help us to examine cases where we see unexpected coverage gaps in order to find out why the coverage gaps exist. For an example, see Figure 9. Similar work aimed at providing explainations for scheduling failures to the user has been done for Crosscheck (Agrawal, Yelamanchili, and Chien 2020), the explainability module for the automated scheduler of the Mars 2020 rover mission.

Slewing heatmaps

In order to help with explainability related to slewing, we would like to generate heatmaps of geographically where the spacecraft are slewing. This would be able to help us identify and demonstrate to the science team which neighboring sets of campaigns are causing potential coverage to be lost because we are requiring the spacecraft to slew between them when overflying one after the other (for example when transitioning between focused and extended configurations for architecture L6C2).

Observation plan by scheduling order

Currently CLASP does not provide any information on the order in which it scheduled observations. We can reason based on campaign priorities to figure out when an observation was scheduled (relative to observations of other campaigns), but this can be unituitive to the science team when there are a lot of different campaigns (some with overlapping

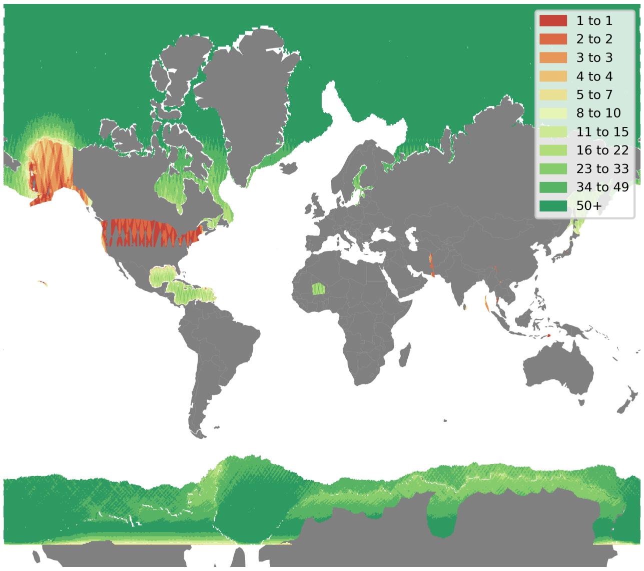

Figure 9: Heatmap showing targetpoints that were not fully satisfied due to a duty cycle constraint. Larger values indicate more times that a targetpoint had an overflight that was not scheduled because it would violate a spacecraft's duty cycle constraint.

Figure 10: An opportunity analysis for a off-nadir pointing spacecraft in a sun-synchronous orbit over a 12 day horizon. Top left: Coverage achieved with all visibilities without considering constraints. Top right: Coverage achieved when accounting for a solar zenith angle constraint. Bottom: Coverage achieved when accounting for both the solar zenith angle constraint and a duty cycle constraint. Going from the top left to top right maps you can see that the missing coverage in the upper latitudes is due entirely to the solar zenith angle constraint (with the banding in the middle latitudes being due to the sensor pointing off-nadir). Going from the top right to the bottom map you can see that the spotty coverage is due to the duty cycle constraint.

areas) and this does not help us to reason about scheduling order within a particular priority level. It would be helpful to have CLASP produce an output product that detailed the order in which the observations in the schedule were scheduled, as it would allow us to see how exactly the schedule was constructed. Additionally it would be able to serve as a visual aid in explaining the scheduling algorithm.

Opportunity analyses

In order to help make reasoning about the scheduling algorithm easier, we could produce a "scheduling story" by creating coverage heatmaps starting with all observation opportunities and progressively filtering down with our constraints until we arrive at the final schedule. This would allow us to more clearly demonstrate the effects of particular constraints and help to give a clearer upper bound on how much coverage we should expect to get for a given scenario. For an example of an opportunity analysis, see Figure 10.

Conclusion

In this paper, we have presented the use of automated scheduling software in a mission design framework aimed at designing a next-generation Synthetic Aperture Radar mission. From the specifications we received from the science team, we semi-automated the generation of inputs that are fed into an adaptation of the CLASP scheduling software. Observation plans are produced for evaluation by the mission planning and science teams. Special cases required the development of ad-hoc scheduling algorithms, for reducing slewing or scheduling interferometric observations. We have emphasized the challenges of explaining schedules produced by the aggregation of thousands of sequential non-chronological decisions. We have shown the current and future outputs and visualizations we used to inform the decision process and ensure that the implemented behaviors were the intended ones.

We have seen that while automated scheduling software can produce plans that fulfil the science needs of the mission, producing them in a way that the science team can understand and trust is vital to the software's use in mission design. We have also seen that there are approaches to allow for explainability that are achievable and result in mission plans that are a reliable and trustworthy contribution to the mission design process.

Acknowledgments

The authors gratefully acknowledge the broad set of contributors to the SDC study including, but not limited to: Shaun A. Deacon, Jonathan D. Chrone, Vianni Ricano Cadenas, Evan Clark, and Joshua Doubleday. 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

Acton Jr, C. H. 1996. Ancillary data services of NASA's navigation and ancillary information facility. *Planetary and Space Science*, 44(1): 65–70.

Agrawal, J.; Yelamanchili, A.; and Chien, S. 2020. Using Explainable Scheduling for the Mars 2020 Rover Mission. In *International Workshop of Explainable AI Planning (XAIP) at the International Conference on Automated Planning and Scheduling (ICAPS)*.

Do, M.; Feather, M.; Garcia, D.; Hicks, K.; Huang, E.; Kluck, D.; Mackey, R.; Nguyen, T.; Shah, J.; Stylianos, Y.; et al. 2013. Synthesizing Fractionated Spacecraft Designs as a Planning Problem. In *Proc. of SPARK 2013*.

Doubleday, J.; Chien, S.; Norton, C.; Wagstaff, K.; Thompson, D. R.; Bellardo, J.; Francis, C.; and Baumgarten, E. 2015. Autonomy for remote sensing—Experiences from the IPEX CubeSat. In *2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 5308–5311. IEEE.

Doubleday, J. R. 2016. Three petabytes or bust: planning science observations for NISAR. In *Earth Observing Missions and Sensors: Development, Implementation, and Characterization IV*, volume 9881, 20–26. SPIE.

Horst, S.; Chrone, J.; Deacon, S.; Le, C.; Maillard, A.; Molthan, A.; Nguyen, A.; Osmanoglu, B.; Oveisgharan, S.; Perrine, M.; Shah, R.; Tymofyeyeva, E.; Wells, C.; Zufall, A.; and Rosen, P. A. 2021. NASA's Surface Deformation and Change Mission Study. In *2021 IEEE Aerospace Conference (50100)*, 1–19.

Knight, R.; and Chien, S. 2006. Producing large observation campaigns using compressed problem representations. In *International Workshop on Planning and Scheduling for Space, Space Telescope Science Institute, Maryland*.

Maillard, A.; Chien, S.; and Wells, C. 2021. Planning the Coverage of Solar System Bodies Under Geometric Constraints. *Journal of Aerospace Information Systems*, 18(5): 289–306.

Nag, S.; Moghaddam, M.; Selva, D.; Frank, J.; Ravindra, V.; Levinson, R.; Azemati, A.; Aguilar, A.; Li, A.; and Akbar, R. 2020. D-SHIELD: DISTRIBUTED SPACECRAFT WITH HEURISTIC INTELLIGENCE TO ENABLE LOGISTICAL DECISIONS. In *IGARSS 2020 - 2020 IEEE International Geoscience and Remote Sensing Symposium*, 3841–3844.

National Academies of Sciences, Engineering, and Medicine. 2018. *Thriving on Our Changing Planet: A Decadal Strategy for Earth Observation from Space*. Washington, DC: The National Academies Press. ISBN 978-0-309-46757-5.

Rabideau, G.; Chien, S.; Mclaren, D.; Knight, R.; Anwar, S.; Mehall, G.; and Christensen, P. 2010. A tool for scheduling themis observations. In *International Symposium on Space Artificial Intelligence, Robotics, and Automation for Space (ISAIRAS 2010)*.

Schaffer, S.; Branch, A.; Chien, S.; Broschart, S.; Hernandez, S.; Belov, K.; Lazio, J.; Clare, L.; Tsao, P.; Castillo-Rogez, J.; et al. 2016. Using operations scheduling to optimize constellation design. In *Proceedings of the 26th International Conference on Automated Planning and Scheduling*.

Sherwood, R.; Chien, S.; Rabideau, G.; and Mann, T. 1997. Design for X (DFX): Operations Characteristics Spacecraft Design Analysis Tool. In *International Workshop on Planning and Scheduling for Space Exploration and Science*, 28–30.

Sipps, J.; and Magruder, L. 2023. Envelope-Based Grid-Point Approach: Efficient Runtime Complexity for Remote Sensing Coverage Analysis. *Journal of Spacecraft and Rockets*, 0(0): 1–13.

Troesch, M.; Chien, S.; and Ferguson, E. 2017. Using automated scheduling to assess coverage for europa clipper and jupiter icy moons explorer. In *International Workshop on Planning and Scheduling for Space (IWPSS 2017)*.

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. *SPARK 2019*, 17.

Yelamanchili, A.; Wells, C.; Chien, S.; Eldering, A.; Pavlickl, R.; Cheng, C.; Schneider, R.; and Moy, A. 2021a. Scheduling and Operations of the Orbiting Carbon Observatory-3 Mission. In *Proceedings Space Operations 2021*.

Yelamanchili, A.; Wells, C.; Chien, S.; Russino, J.; Oaida, B.; and Thompson, D. R. 2021b. Using Automated Scheduling for Mission Design: A Case Study for EMIT. In *Proc. of SpaceOps 2021*.