Timeline-based space operations scheduling with external constraints

Steve Chien[1], Daniel Tran[1], Gregg Rabideau[1], Steve Schaffer[1], Daniel Mandl[2], Stuart Frye[2]

[1] - Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, 91109-8099, USA
[2] - Goddard Space Flight Center, Greenbelt, MD, 20771, USA

Abstract

We describe a timeline-based scheduling algorithm developed for mission operations of the EO-1 earth observing satellite. We first describe the range of operational constraints for operations focusing on maneuver and thermal constraints that cannot be modeled in typical planner/schedulers. We then describe a greedy heuristic scheduling algorithm and compare its performance to both the prior scheduling algorithm - documenting an over 50% increase in scenes scheduled with estimated value of millions of dollars US. We also compare to a relaxed optimal scheduler showing that the greedy scheduler produces schedules with scene count within 15% of an upper bound on optimal schedules.

Introduction

Spacecraft operations have been a major area of application for automated planning and scheduling. Numerous space missions have used automated planning & scheduling on the ground to enable significant operational efficiencies including the Hubble Space Telescope [Johnston et al. 1993], space shuttle refurbishment [Deale et al. 1994], shuttle payload operations [Chien et al. 1999], The Modified Antarctic Mapping Mission [Smith et al. 2002], Mars Exploration Rovers [Bresina et al. 2005], Earth Observing One (EO-1) [Chien et al. 2005a] Mars Express [Cesta et al. 2005], and Orbital Express [Chouinard et al. 2008]. Automated planning has even flown as a technology demonstration on the Deep Space One (DS1) Mission [Muscettola et al. 1998] and as the primary operations system on 3CS [Chien et al. 2001] and EO-1 [Chien et al. 2005b].

Spacecraft operations have a number of interesting attributes from a planning & scheduling applications perspective.

1. Spacecraft operations require modeling of a number of challenging operations constraints including: instrument and subsystem timing and synchronization, thermal, power, data volume, visibility, and spacecraft pointing.
2. Because spacecraft are so expensive ($100M+ US is not unusual), a planning model must be highly reliable to not produce operations plans that might endanger a valuable asset.
3. Because of the complex nature of science operations priority and optimization are often involved either implicitly or explicitly.
4. For onboard schedulers, two additional factors are relevant.
   a. Because communications to spacecraft are limited in frequency and duration, from an AI planning perspective a spacecraft has a flight and ground version of the planning problem. The flight version typically involves embedded replanning in modest context whereas ground planning may tackle large problems involving hundreds or thousands of activities.
   b. Limited onboard computing often requires algorithms that are not computationally or memory intensive.

In the remainder of this paper we first describe the EO-1 weekly operations scheduling problem. We then describe the wide range of operations constraints that are naturally modeled in typical planning & scheduling modeling languages. We then describe a number of more difficult to model constraints including thermal, pointing, and prioritization. We then describe a timeline based scheduling algorithm used for mid range (weekly) ground-based scheduling of the EO-1 mission. We then present an analysis of impact on operations. Finally we present related work and conclusions.

Copyright © 2010, California Institute of Technology. All rights reserved.

Portions of this work were performed by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

Note that some of the specific operational constants (e.g. instrument temperatures, warmup times) in this paper have been altered for export control purposes. Numbers relating to impact (e.g. # of scenes scheduled) have not been altered and any operational constants changed do not affect the technical points of the paper.

Background

The Earth Observing-1 (EO-1) satellite is the first mission in NASA's New Millennium Program Earth Observing series [EO-1]. EO-1 was launched November 21, 2000 and has been in operation since.

Following a one-year primary mission, EO-1 entered extended mission in January of 2002 having surpassed all original technology validation goals. By 2004 continuous improvements in EO-1 conventional operations enabled acquisition of approximately 100 scenes per week, a remarkable improvement over the pre-launch success criteria of 7 scenes per week (a scene is a typical science imaging activity consisting of a 50km long image using the two principal instruments – ALI and Hyperion).

In 2004, onboard and ground-based automated mission planning software was deployed operationally to automate mission planning and sequencing elements of the EO-1 mission [Chien et al. 2005a, 2005b]. This software, called R4, was directed at automating existing operations policies rather than improving the number of science scenes acquired by the mission. This approach was taken because it offered the lowest risk, least costly path to automation. This automation was tremendously successful - enabling an over $1M per year operations costs reduction and allowing more rapid response to science events and anomalies such as ground station failures. This automation was able to continue this pace of ~ 100 scenes per week. The 2004 automation has operated flawlessly and has acquired over 25,000 scenes in over 5 years of operations.

More recently (2008-2009), the ground and flight mission planning software for EO-1 was upgraded (to R5) again with a focus on: 1. increasing operational flexibility to change scenes immediately before acquisition and 2. acquiring more science scenes.

Major constraints limiting the number of scenes include:

Visibility – even though the spacecraft might be unused it might not be able to see a desired science target.

Pointing/maneuver – the spacecraft takes time to move from pointing at one target to the next and must allow time for the spacecraft to stabilize after pointing to enable precise imaging.

Thermal – the instruments have minimum and maximum temperatures at various locations that must be met to acquire valid science imagery. The minimums mean that warmup activities are required. The maximums mean that too many consecutive images will overheat the instrument.

Data volume – the spacecraft can only store a limited number of scenes onboard.

Downlink – the spacecraft can only downlink at pre-scheduled times and overflights of fixed ground stations

Mode – various spacecraft subsystems have operational modes that must all be carefully selected and achieved for valid operations.

The goal of the scheduler is:

1. maximize weighted scene count where higher priority scenes are given more weight; and
2. maximize total scene count;

while respecting all of the above constraints. We present a generate and test approach to weekly scheduling for EO-1 operations that:

1. incorporates pointing, maneuver, and visibility constraints into generation of scene combinations caused "tuples"
2. searches greedily in the space of combinations of tuples
3. using timelines to check the remaining constraints (e.g. thermal, data volume, …)

We analyze the computational complexity of this scheduling algorithm and compare its performance to optimal upper bound schedules in both weighted priority and scene count.

Spacecraft Operations Modeling

In this section we describe the range of spacecraft operations constraints present in the Earth Observing One Model. We begin by describing constraints that are easily modeled in automated planning/scheduling systems and then discuss problematic constraints.

In our discussion, we assume the context of a timeline based modeling framework, in which state and resource values are represented by a fully ordered sequence of values. Furthermore, in our framework, value transitions are grounded in time (e.g. not flexible). For our specific class of scheduling problems this is a reasonable restriction because the overflight times or scene targets and data up/downlink times are fixed (therefore fixing most events of interest).

Naturally modeled spacecraft operations constraints

The updated EO-1 operations domain has a wide range of constraints that can be naturally represented in common planning & scheduling system modeling constructs.

Activity overlap – instances of activities cannot overlap such as those that require an atomic resource. For example, two image sequence parent activities cannot overlap. This is represented by a simple atomic resource (a unit capacity resource) that an image sequence parent activity claims. If a second image activity overlaps it also claims this resource, exceeding the capacity.

Integer capacity – depletable – this is an integer capacity resource reserved by one activity making a portion of the resource unavailable until it is freed by another activity. For example, EO-1 has a mass storage device primarily for science data. The storage device, called the WARP, has two capacity constraints. First, there is a limitation on the total number of files on the WARP

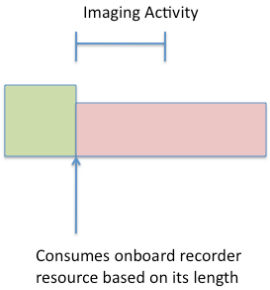

Figure 1: Activity consuming depletable resource

at any given time. The file count is represented as a depletable resource the maximum capacity. When files are created they are counted against the file count resource. When files are deleted after downlink activities the resources are freed. Second, the total size of all of the files (summed) cannot exceed a different bound. This resource is consumed as data is written to a file on the recorder and released when files are deleted (after being downlinked). Usage of these resources can depend on activity parameters – for example the amount of data generated by an imaging sequence is dependent on how long the instrument is imaging as dictated by a function (a base amount plus a fixed rate times the image activity duration). This resource usage is shown in Figure 1. In contrast, non-depletable resources are used only for the duration of the activitiy (e.g. power).

Discrete states – there are numerous discrete state constraints. These both represent transition constraints and state constraints. For example, the solid state recorder has several states (record, playback, idle, standby,…). Furthermore, there may only be a specific subset of legal transitions with activities to change the state. The only means for the WARP state to change from one state to another is via execution of an action with a state changer, As another example, the ALI instrument (one of the two principal science instruments onboard EO-1) has a cover which has specific activities to change its state, and imaging activities require specific states (dark calibrations require closed state, science images require open state). Figure 2 shows some aspects of constraints on the ALI cover state.

Decomposition – often a high level activity consists of several lower level activities. These are represented as Hierarchical Task Network planning decompositions. For example, and imaging sequence high level activity consists of a large number of lower level activities including ALI and Hyperion (the other principal science instrument) prep activities and post activities. Figure 3 shows the first level of decomposition for a Hyperion Lamp Calibration activity set.

Temporal constraint – constraints on the relative timing or ordering of two related activities. For example, in an image sequence, the instrument parameters must be set 4.5 seconds before the image start time and the Hyperion instrument covers must be opened 28.5 seconds before the image start time. Most of these temporal constraints are enforced in the decompositions outlined above.

Some of these temporal relationships utilize dependencies upon timeline values or activity parameters. For example, the Hyperion and ALI warm-up times are dependent on the expected temperatures entering into the imaging activity. If the instruments are already warm from prior image sequences the warmup time can be shortened allowing images to be acquired closer together and preventing the instrument from overheating (this is discussed in the section on thermal modeling below).

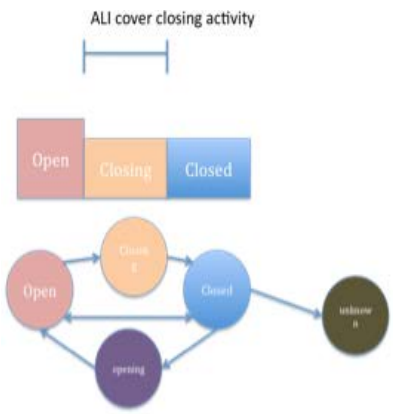

Figure 2: ALI cover state (partial)

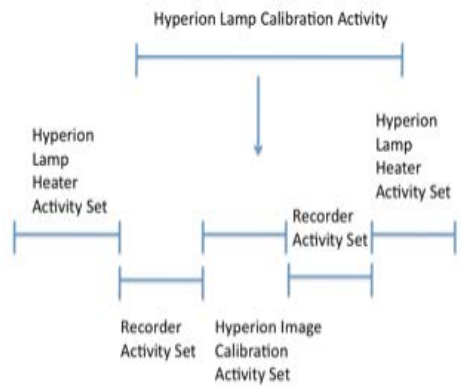

Figure 3: Hyperion Lamp Calibration Activity Decomposition

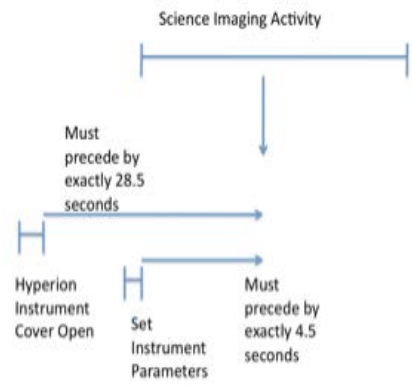

Figure 4: Temporal Constraints in Science Imaging

More challenging operations constraints

In this section we describe modeling and non-modeling of several operations constraints – thermal and maneuver.

Modeling Instrument Thermal Constraints

The Hyperion instrument has two imaging subsystems: a visible and near infrared module (VNIR) and a short wave infrared module (SWIR). These distinct subsystems are physically separated and therefore have somewhat decoupled behavior. Both of these modules are used

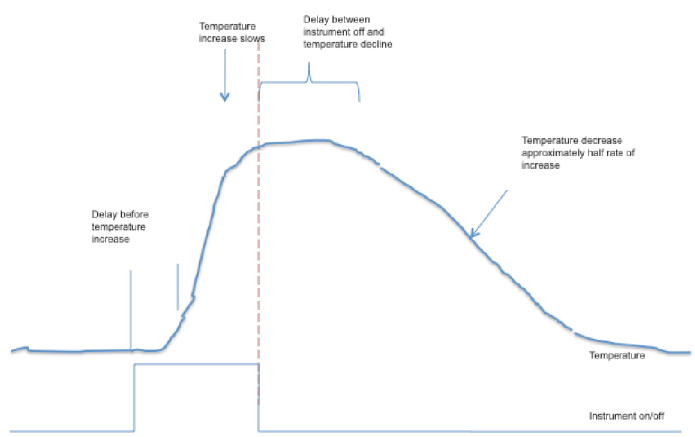

Figure 5: Typical instrument temperature profile

during imaging, and both gradually increase in temperature while they are powered. When unpowered, the instrument (and subsystems) gradually shed heat to the rest of the spacecraft and to space, thusly cooling (see Figure 5). Each of these subsystems has a minimum and maximum operating temperature requirement for (1) instrument health and (2) precise imaging. The Hyperion instrument also has a setup time so that the instrument must be powered on by this amount prior to imaging to allow the instrument to enter the correct mode to accept imaging control parameters prior to imaging. The Hyperion instrument operations challenge is to control the power state of the instrument such that both the SWIR and VNIR are operating within acceptable temperature ranges and the instrument is able to accept imaging parameters for all desired images. This is complicated because

1. the SWIR and VNIR have different minimum and maximum temperatures,
2. SWIR and VNIR heat and cool at different rates, have different ambient temperatures (e.g. if left off it will return to this temperature) and
3. there is only one power control (e.g. we cannot warm up VNIR without warming up SWIR).

The VNIR module has a heating element that maintains its temperature at or above 311 to prevent the instrument from damage in the cold of space. The VNIR has an imaging temperature requirement of 313. A brief warm-up period in advance of an image is modeled to allow the instrument to reach this temperature, if needed. During sparse operations, the instrument then cools back to its set point over a period of approximately an hour. However, extended instrument duty cycles (e.g. during a rapid sequence of adjacent scenes) can cause this temperature to build up without a chance to cool down. EO-1 flight rules include a maximum VNIR operating temperature of 415 as well as a maximum health temperature of 515.

Similarly, the SWIR module has a maximum operating temperature of 415 and a maximum health temperature of 515, but no operating temperature minimum. If the SWIR module is allowed to cool lengthily it will eventually an equilibrium temperature of -19. Notably, at these lower temperatures, the SWIR module is much less effective at dissipating heat (and more susceptible to absorbing heat), as expected from classical Newtonian cooling models. This means that temperature builds more rapidly in the SWIR once the instrument is cycled on, and that it takes much longer for it to return to ambient -- on the order of 12 hours.

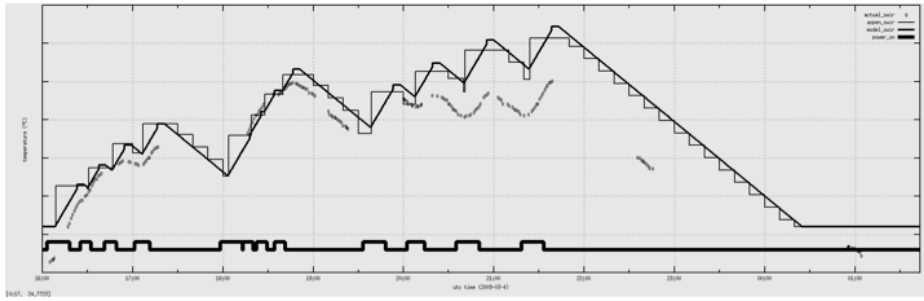

Figure 6: SWIR temperature as: modeled in the planner (square model)
simulated (upper dark line)
observed in flight (stream of circles)
instrument power state (dark line at bottom)

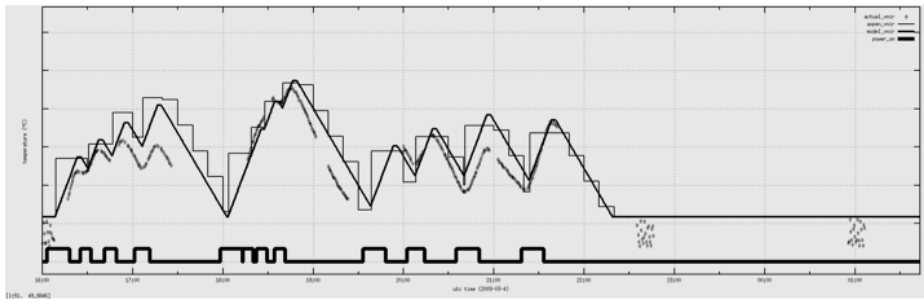

Figure 7: VNIR temperature as: modeled in the planner (square model)
simulated (upper dark line)
observed in flight (stream of circles)
instrument power state (dark line at bottom)

We constructed three different Hyperion instrument temperature models:

1. A scene uses a non-depletable temperature resource for the duration of the scene plus a cool down period. This model requires computation and RAM O(# of overlapping scenes). Unfortunately this model becomes very inaccurate with large numbers of tightly consecutive scenes.
2. Use a non-depletable model but make the duration of the resource usage dependent on the time since the previous scene. This model requires CPU and RAM O(# overlapping scenes). This model was more accurate, it was still too inaccurate because of the long history of SWIR (literally days).
3. Use a stepwise simulation model where the temperature is increased or decreased at each step based on the instrument on/off state. This model is most accurate but requires CPU and RAM O(length of timeline).

Efficiency of the temperature model is important because we only have a 5 MIPS onboard computing budget for all of the autonomy software. While the principal use of the model is for ground scheduling, the same model runs onboard to validate and detect problems in execution as well as to enable onboard replanning of scenes.

Figure 6 shows the SWIR temperature as observed in flight, modeled in simulation, and modeled by the planning

system. Figure 7, shows the corresponding information for the VNIR subsystem. The graphs show that the SWIR and VNIR temperatures appear to increase and decrease in approximately linear segments, with continuous curves between the areas of linear heating and cooling. The planning model only roughly approximates the actual and simulated temperatures but for planning purposes it only need answer the question "will this set of scenes exceed the temperature limit" and "how long must the instrument warmup so that this scene will be at least at the minimum of the operating range."

The above model development was performed using historical operations data. While we had virtually unlimited examples of imaging (thousands of scenes) from prior operations, this data only included single and dual scenes per orbit (because R4 and prior operations only allowed two scenes to be imaged before returning to nadir point and instruments off). In order to get data to further refine the model we performed a flight experiment in which we controlled the power state of the instrument simulating three sequences of four scenes each (as allowed by R5).

Pointing and wheel Biasing

Another challenging operations constraint for EO-1 is pointing (see Figure 8). The EO-1 spacecraft can change its pointing using three reaction wheels for turning quickly (seconds) and a magnetic torquer bar for turning slowly (tens of minutes) [Wikipedia]. Reaction wheels spin in one direction, causing the spacecraft to spin in the opposite direction due to Newton's third law. However, the reaction wheels have a maximum speed (engineering) and an ideal speed (scene quality). The torquer bar can be used to reduce the wheel speeds while maintaining a pointing. Because the spacecraft is in orbit around the Earth, if it continually points directly downward towards the Earth, it will need to make one 360 degree rotation per orbit. Additionally, the spacecraft must roll (turn across track) in order to point at imaging scenes that are not directly underneath the spacecraft). These combined maneuvers, including the return to nadir point after a sequence of scenes presents a considerable maneuver planning problem (Figure 8). In summary, the pointing problem is:

1. to point the spacecraft so it can acquire a sequence of scenes
2. while not exceeding the reaction wheel speed limits at any time and
3. ensuring the wheels are at desired speeds during scene

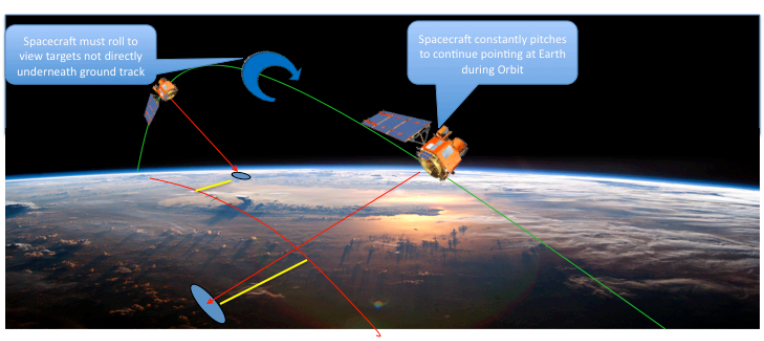

Figure 8: EO-1 Pointing

Operationally, if the mission planner wishes to acquire scene A then scene B, maneuver planning software takes the requests and computes parameters that the spacecraft attitude control system ingests at execution time to achieve the desired pointings (from nadir to A to B to nadir). Computing these maneuvers is a challenging flight dynamics problem – the maneuver planning software in fact uses a heuristic method to attempt to design maneuvers that respect rate constraints, timing constraints, and instrument pointing constraints. This flight dynamics planner is implemented in Matlab and takes approximately 20 seconds to solve a typical request of several consecutive pointings on a high performance linux workstation. From a mission planning perspective these constraints are treated as black box solutions that possess challenging non-monotonic properties. For example, the maneuver planning software may return that starting from nadir pointing, taking scene B followed by scene C is not possible. But the same software might return that starting from nadir, taking A followed by B followed by C is possible. Clearly this means that moving from nadir to B to C is possible but that the solution through A was not found by the maneuver planning software when planning for only B and C. The lack of structure of these returned constraints make the EO-1 mission planning problem computationally harder.

Originally in operations, the spacecraft was "nadir pointed" (i.e. pointed directly at the ground) and "zero biased" (i.e. reaction wheels not spinning) in between every scene. While this is the most straightforward operationally (each scene is independent from a momentum history standpoint) it is not very efficient because considerable spacecraft time is wasted slewing the spaceraft to nadir and slowing the spin of the reaction wheels. One of our upgrades enables EO-1 to go directly from one image to the next without zero biasing or nadir pointing for up to four consecutive images.

Because the timeline planning system cannot directly represent the pointing and momentum state of the spacecraft, we implemented these constraints in the goal generation process (see below) as "external constraints". Basically, when all of the individual scene requests are received, we construct sets of combinations of the scenes (called "tuples") that represent scenes without intervening nadir pointing and zero biasing. The mission planner then operates on these tuples, considering combinations of tuples for a weekly schedule.

The mission planner only indirectly models spacecraft location and therefore image overflights. The mission planner accepts as inputs goals to image targets but it does not directly consider alternate opportunities to image the same target. Because the EO-1 general planning horizon is only

at the one week granularity, it does not offer a direct method of considering among alternate overflights for specific targets. The mission team does often consider alternate overflights but does so outside of the automated mission planning process.

EO-1 operations also have a number of trending and tracking calibration and instrument maintenance activities. These include ALI calibrations (data collected with covers closed and internal lamp on or off) as well as outgassing of the instruments. Other engineering activities include orbit determination calculations, burns to maintain orbit, and fuel calculation. An ideal planning system would track these events and schedule them when needed based on periodicity, schedule conflict, and imaging parameters.

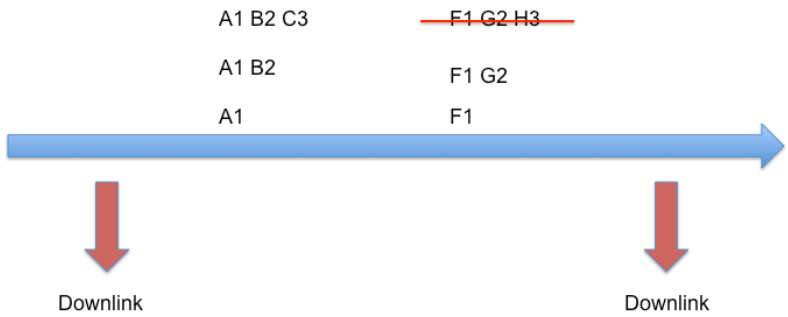

Figure 9: EO-1 Scheduling Example

Scheduling Algorithm

On the ground a weekly schedule of EO-1 operations is maintained. This schedule is generated in several versions 5 days prior to its start (e.g. the schedule is first generated Wednesday for the week starting the following Monday and then refined several times). This schedule must consider hundreds of individual scenes that can comprise thousands of potential tuples that must be heuristically pruned in order to produce a manageable problem.

The EO-1 mission has a simple model of priority that does not fully capture the science and operations constraints of the mission. Within this model priority ranges from 0-999 with 999 being the highest priority. Users have the authorization to submit scenes at a range of priorities. The semantics of the priorities are that a higher priority scene will be selected over any number of lower priority scenes that may conflict. The priorities are incorporated in the core of the scene selection and scheduling algorithm as indicated below.

A better system for representing priority would allow for the scheduling system to be aware of contention (which other scenes are also competing for the overflight), periodicity of the contention (i.e. is this going to happen for every overflight or is it only for some known subset of overflights), urgency (is there a temporal urgency to acquire this scene now – e.g. is it a short lived event such as a ground-truthing, flood, or volcanic event), and age (many targets are designed to be periodically observed and this target may have just been observed).

Weekly scheduling consists of: submission of requests from a set of customer groups, scheduling engineering activities, and scheduling science activities. The weekly scheduling algorithm is shown below as Algorithm 1.

The weekly scheduling algorithm can be understood as follows. First the tuples (combinations of adjacent scenes) are generated from the individual requests. Next the downlink contacts are processed. All of the approved downlink contacts will be S-band engineering downlinks since S-band activities do not interfere with the other spacecraft operations. X-band high rate science downlink however does preclude simultaneous science image acquisition. By default we take all downlink opportunities and schedule them as X-band activities but later in the scheduling algorithm we consider removing them for high priority scenes.

Next we sort the generated tuples by decreasing minimum priority scene in the tuple (so that we consider all tuples that have only high priority scenes first, then all that have only high and slightly lower, and so on…).

As we consider a new candidate tuple, we try to insert it into the schedule (e.g. "ScheduleTuple" below). ScheduleTuple considers whether the new tuple should be added by:

1. adding the new tuple to the schedule
2. deleting any subsumed tuples (e.g. if adding a tuple with scenes A B and C, deleting the tuple A B)
3. delete any X-band in conflict (overlapping) with the new tuple
4. if any timeline constraints are violated FAIL, else succeed

The net effect of this scheduling algorithm is that it starts out with tuples (note that single scenes are degenerate tuples) with only high priority scenes. It then adds lower priority scenes where they fit in between high priority scenes (not too common) or by growing the tuples with high priority scenes by adding lower priority scenes to the tuples. Figure 9 shows this process in which A1, and F1 are inserted and grown until A1B2C3 and F1G2 are present. When F1G2 tries to grow it cannot fit due to downlink constraints and the scheduling process stops. In each case a single higher priority scene is preferred over lower priority scenes. X-bands can be bumped but only if they are not needed for storage of higher priority scene (which would have been already scheduled). Because the scenes are secondarily sorted by number of lowest priority scene the algorithm generally prefers more scenes of a given priority. However it cannot guarantee optimality at this level due to the possibility of a tie-break at a higher level priority precluding choice of a high priority scene that would be compatible with a larger number of lower priority scenes.

This algorithm requires timeline placements O(# tuples). Timeline placement cost is dominated by the thermal model O(time duration of schedule. The number of tuples is exponential in the individual scene density, but only

linear in the schedule length (with scene count density held constant – i.e. scene count increasing linearly with schedule length). In our operational experience several hundred individual scenes are requested and several thousand tuples are considered.

```
scheduleWeekly
  generate tuples from individual requests
  schedule the given S-bands
  schedule one X-band for every S-band
  sort unsatisfied tuples by greater min priority
  for each unsatisfied tuple
    scheduleTuple(tuple)
  repair resulting schedule

scheduleTuple(tuple)
  find satisfied subsets of the given tuple
  if tuple has unsatisfied scenes
        (not part of a subset)
    remove subsets from satisfied tuples
    unschedule subsets
    unschedule X-bands that overlap with a new
scene
    schedule tuple
    if no conflicts
      add tuple to satisfied tuples
      return true
    else // undo
      unschedule tuple
      schedule subsets
      add subsets to satisfied tuples
      schedule overlapping xbands
  return false


schedule(goal)
  expand goal activity and model states/resources

unschedule(goal)
  unexpand goal activity & unmodel
states/resources
```

Algorithm 1

Evaluating EO-1 scheduling effectiveness

Originally the motivation for the R5 software upgrade was to increase flexibility to change the schedule. In R4 once X-bands were selected they could not be later pre-empted by high priority scenes. Additionally, scenes priorities resulted in several unnatural constraints in their implementations: (1) dual collects had to consist of two scenes of the same priority (so that the priority of the dual scene was semantically unambiguous) and (2) replacements of a single or dual scene had to be with the same number of scenes (e.g. a single replacing a single or a dual replacing a dual). However R5 can schedule more scenes due to better thermal management and reduced nadir pointing and zero biasing.

To assess scheduling improvement we ran the R4 and R5 on four weekly schedules from Spring 2009. To simplify analysis we scheduled these weeks without any engineering activities (which require human input). Ideally we would compare R5 schedules against optimal weekly schedules. Unfortunately non-monotonic constraints (slewing and maneuver in creating tuples) and computationally expensive modeling (thermal) and weekly problem size prohibit generating optimal solutions. The problem cannot even be localized to small NP-hard problems between X-bands because X-band selection is part of the search space and tuples can span X-bands. Therefore we developed a series of optimal schedulers that ignore certain hard (e.g. maneuver, slew, temperature) constraints and produce optimal schedules for these relaxed problems – thereby providing upper bounds on optimal schedules for the real problem. The results of these schedule runs are shown below in Table 1. O1A & O1B below used the fixed X-band selections from the R5 algorithm. O2 uses an alternative approach for X-bands. Table 1 shows the number of X-bands and scenes scheduled as well as a priority score of the schedule indicating a weighted score where a scene of each priority level is worth 10x the value of a scene of the next lower priority.

Algorithm	X-bands	Scenes scheduled	Priority Score
R4	32	130	1233
R5	51	217	1243
O1: Optimal no thermal, no maneuver, R5 X-bands	51	243	1286
O1A: O1 removing onboard storage	51	419	1286
O1B: O1 ignoring scene overlap	51	252	1422
O2: O1 but choose all X-bands not in conflict with high priority	48	229	1246

The data shows several interesting points.

1. The most significant constraint limiting scenes is onboard storage (seen by the jump in scenes removing this constraint from O1 to O1A). However, the gained scenes are not important ones as the priority score is unchanged (e.g. there are no gained scenes in the top several priority levels). It is also worth noting that O1, O1A, and O1B all ignore instrument thermal constraints, which would certainly prevent taking of 400 scenes in a week.
2. The biggest constraint preventing acquiring higher priority scenes is scene overlap as indicated by the jump in schedule score from O1 to O1B. Note that maneuver (also unmodeled by O1, O1A, and O1B) would certainly preclude taking many of these high priority combinations even if scene overlap could be relaxed.
3. The R5 scheduler significantly outperforms R4 in scene count increasing average scene count from 130 to 217(+67%) – primarily by enabling two or three scenes to be taken many orbits. Weekly averages for R4 are 70 singles and 30 duals whereas R5 averages 18 singles 45 duals and 37 triples. Note that the R4 algorithm was also constrained to take duals of only the same priority and also have a designated primary scene as the first scene. However most of the additional scenes that R5 acquires are

not high priority – this is because the operations team self selects by not choosing multiple high priority scenes that it believes will conflict – further study will enable us to acquire less biased input requests.

4. R5 also performs well compared to the tightest upper bound on optimal schedules (O1). R5 is within 11% of the optimal upper bound by scene count and within 3.4% by priority score. Given that O1 is an upper bound and maneuver and thermal are significant additional constraints it is likely that R5 is closer to a true optimal schedule.

Empirically the R5 scheduler has performed well. Taking two months of "typical" operations, for September 2009, R5 scheduled from 146 – 187 scenes/week, whereas January 2009 R4 operations ranged from 91-115 scenes per week. The operational numbers tend to be lower than the test runs above due to inclusion of engineering activities.

What is the value of the additional scenes? A conservative estimate based on the 2004 mission cost valued the EO-1 mission at ~$3.6M/year so one measure (scene count) would estimate the value of the additional 50% scenes at $1.8M/year[1] for a total value over the remainder of the mission of millions of dollars US.

Discussion, Related Work, and Conclusions

AI planning specializes in generating novel sequences to achieve combinations of goals. Because of the importance of safety in space operations, novelty in sequencing is discouraged. Consequently, most planning in space operations is performed by hierarchical task network methods (as opposed to more search intensive first principles methods), which have the advantage of repeatability to facilitate testing.

The maneuver constraint also is one or a larger set of spatial reasoning constraints that recur in scheduling problems. An area for future work would be for a scheduler to reason about the distances between the consecutive pointings in the momentum management space and explore these more efficiently (instead of the current method of exhaustively searching all combinations).

Thermal constraints were also hard to model. Representing these sorts of complex quantitative constraints along with scheduling/repair methods are another area for future research.

This paper has described several modeling which arose in developing an automated planning system for EO-1 spacecraft operations – with a focus on thermal and pointing constraints. We then described our heuristic approach to EO-1 schedule generation that isolates the pointing constraint in candidate generation and thermal constraints in a test portion of schedule. We document that our scheduling algorithm significantly increases the mission science scenes; and show its performance approaches that of an upper bound on optimal scheduling

[1] One might argue that the worth of additional scenes is lower per scene because the highest priority scenes would be taken first. However one might also argue that more scenes enables studies at a finer temporal resolution thereby enabling studies not allowed with fewer scenes.

References

J. L. Bresina, A. K. Jonsson, P. H. Morris, K. Rajan, Activity planning for the mars exploration rovers, Proc 15th Intl Conf Auto Plan & Sched, 2005, Monterey, CA.

Cesta, A., Cortellessa, G., Fratini, S., Oddi, A. and Policella. N., An innovative product for space mission planning: an a posteriori evaluation. Proc Intl Conf Automated Planning & Scheduling, 2007, Providence, RI.

S. Chien, G. Rabideau, J. Willis, T. Mann, "Automating Planning and Scheduling of Shuttle Payload Operations," Artificial Intelligence Journal 114 (1999) 239-255.

S. Chien, B. Cichy, A. Davies, D. Tran, G. Rabideau, R. Castano, R. Sherwood, D. Mandl, S. Frye, S. Shulman, J. Jones, S. Grosvenor , "An Autonomous Earth Observing Sensorweb," IEEE Intelligent Systems, May-Jun 2005, pp. 16-24.

S. Chien R. Sherwood, D. Tran, B. Cichy, G. Rabideau, R. Castano, A. Davies, D. Mandl, S. Frye, B. Trout, S. Shulman, D. Boyer . , "Using Autonomy Flight Software to Improve Science Return on Earth Observing One, Journal Aerospace Computing, Information, & Communication, April 2005, AIAA.

C. Chouinard, R. Knight, G. Jones, D. Tran, D. Koblick, Automated & Adaptive Mission Planning for Orbital Express, Space Operations, Heidelberg, Germany, 2008.

M. Deale, M. Yvanovich, D. Schnitzius, D. Kautz, M. Car- penter, M. Zweben, G. Davis, and B. Daun, "The Space Shuttle Ground Processing Scheduling System," in Intelligent Scheduling, Zweben & Fox (eds.), Morgan Kaufman, San Francisco, 1994.

N. Muscettola, P. Nayak, B. Pell, B. C. Williams, "Remote Agent: to boldly go where no AI system has gone before," Artificial Intelligence v103, # 1-2, pp. 5–47, 1998.

M. D. Johnston, R. Henry, A. Gerb, M. Giuliano, B. Ross, N. Sanidas, S. Wissler and J. Mainar, "Improving the Observing Efficiency of Hubble Space Telescope," AIAA Aerospace, AIAA Press, 1993.

G. Rabideau, R. Knight, S. Chien, A. Fukunaga, A. Govindjee, "Iterative Repair Planning for Spacecraft Operations in the ASPEN System," Intl Symp Artificial Intelligence Robotics & Automation in Space, Noordwijk, The Netherlands, June 1999.

B.D. Smith, B.E. Engelhardt, and D.H. Mutz,"The RADARSAT-MAMM Automated Mission Planner," AI Magazine, v. 23, # 2, 2002. pp. 25–36.

Wikipedia, Reaction Wheel, http://en.wikipedia.org/wiki/Reaction_wheel