Challenges in Representing and reasoning with spacecraft operations constraints: A case study with Earth Observing One

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, USA

Abstract

We describe the modeling and reasoning about operations constraints in an automated mission planning system for an earth observing satellite – EO-1. We first discuss the large number of elements that can be naturally represented in an expressive planning and scheduling framework. We then describe a number of constraints that challenge the current state of the art in planning systems and discuss how we modeled these constraints as well as discuss tradeoffs in representation versus efficiency. Finally we describe the challenges in efficiently generating operations plans for this mission. These discussions involve lessons learned from an operations model that has been in use since Fall 2004 (called R4) as well as a newer more accurate operations model in final testing before going operational in April 2009 (called R5). We present analysis of the R5 software projecting a significant (>50%) increase in the science return of the EO-1 mission and that R5 produces schedules within 15% of an upper bound on optimal schedules.

Introduction

Spacecraft operations have been a major area of application for automated planning and scheduling with successful applications 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. 2007], and Orbital Express [Chouinard et al. 2008]. Automated planning has even flown as a technology demonstration on the Deep Space One (DS1) Mission in 1999 [Muscettola et al. 1998] and as the primary operations system on 3CS [Chien et al. 2001] in 2004 and EO-1 [Chien et al. 2005b] 2004-present.

The research described in this publication was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

Spacecraft operations are of interest for planning & scheduling applications as follows.

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 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.
4. Limited onboard computing often requires algorithms that are not computationally or memory intensive.
5. Because of the complex nature of science operations priority and optimization are often involved either implicitly or explicitly.

In the remainder of this paper we first describe the EO-1 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 heuristic approach to generating schedules for the EO-1 mission. We then present an analysis of impact on operations. Finally we present related work and conclusions.

Disclosure: Note that many of the specific operational constants (e.g. instrument temperatures, warmup times, etc.) in this paper have been altered for export control purposes. Impact and problem scale numbers (e.g. # of observations per week, numbers of resources scheduled, have not been modified).

Background

The Earth Observing-1 (EO-1) satellite was launched in 2000 into an orbit with a ~16-day repeat track, with at least 5 day and 5 night over-flights per 16-day cycle separated by a <10-degree change in viewing angle.

EO-1 carries three instruments: the Advanced Land Imager (ALI), the hyper-spectral Hyperion Imager, and the Atmospheric Corrector (AC). Together the three instruments collect over 20-Gbits of science data to the onboard solid-state data recorder for each scene (which consists of one science and two calibration images for ALI and Hyperion - the AC is no longer used in nominal observations).

The EO-1 spacecraft has two Mongoose M5 processors. One of these M5's is available for partial use by onboard autonomy software. Each M5 runs at 12 MHz (for ~8 MIPS) and has 256 MB RAM. Both M5's run the VxWorks operating system. The Autonomous Sciencecraft (ASE) [Chien et al. 2005b] AI software operates on the secondary WARP M5 processor.

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 significantly beyond the pre-launch success criteria of 7 scenes per week.

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, automated existing operations policies rather than trying to increase the number of science observations acquired by the mission (to reduce the risk and cost of 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 observations per week. The 2004 automation has operated flawlessly and has acquired over 20,000 scenes in the almost 5 years of operations (a scene includes one science and two calibration images).

More recently (2008-2009), the ground and flight mission planning software for EO-1 is being upgraded again. This upgrade (R5) emphasizes: 1. increasing operational flexibility to change scenes immediately before acquisition and 2. acquiring more science scenes.

EO-1 has a 90-minute orbital period, meaning that in any given week it has ~ 112 orbits. Typically under R4 orbits would be used as follows: ~60 orbits EO-1 takes a single scene, ~30 orbits EO-1 takes two scenes, the remainder EO-1 performs instrument calibration or instrument decontaminations and therefore be unable to acquire science images. In a typical week from 2004-2009 under R4 software control EO-1 might acquire 120 scenes. Restrictions limiting the number of scenes include:

Visibility – even though the spacecraft might be unused it cannot 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 observations 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.

In the following sections we describe our approach to modeling these many constraints. In our implementation we use the ASPEN [Chien et al. 2000] modeling system with an emphasis on the timeline-based modeling capability but our observations are relevant to most fixed time, timeline based scheduling approaches.

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.

Easily 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

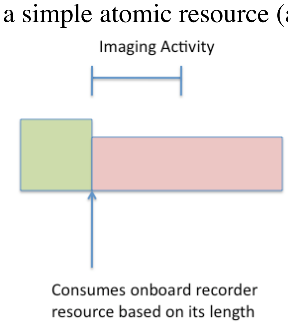

Figure 1: Activity consuming depletable resource

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 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.

Discrete states – there are numerous discrete state constraints including transition constraints and state constraints. For example, the solid state recorder has several states (record, playback, idle, standby,…) with a specific subset of legal transitions and activities to change the state as specified in the planning model. As another example, the ALI instrument has a cover with 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 (HSI) prep activities and post activities. Figure 3 shows the first level of decomposition for a Hyperion Lamp Calibration activity set. While these decompositions can include subgoals solved by means end analysis and alternative achievement methods these are not required by the EO-1 domain (e.g. it is a scheduling problem not a planning problem).

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).

More challenging operations constraints

In this section we describe modeling and non-modeling of several operations constraints – thermal, pointing and wheel biasing, power, prioritization, and others.

Modeling Instrument Thermal Constraints

One challenging constraints in the R5 model upgrade was thermal modeling for the Hyperion instrument. The Hyperion instrument has two imaging subsystems: a visible and near infrared module (VNIR) and a short wave infrared module (SWIR) with somewhat decoupled behavior. VNIR and SWIR gradually increase when imaging and cool when not in use.

VNIR and SWIR have minimum and maximum operating temperature requirement for both precise imaging and instrument protection. 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. Thus the Hyperion 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.

The VNIR module is tightly temperature controlled such

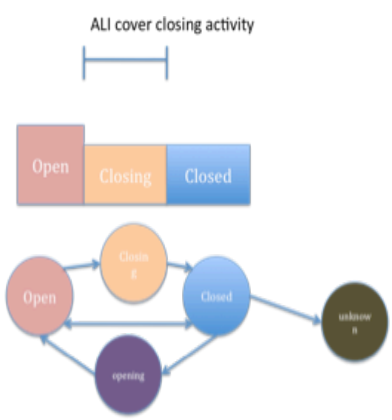

Figure 2: ALI cover state (partial)

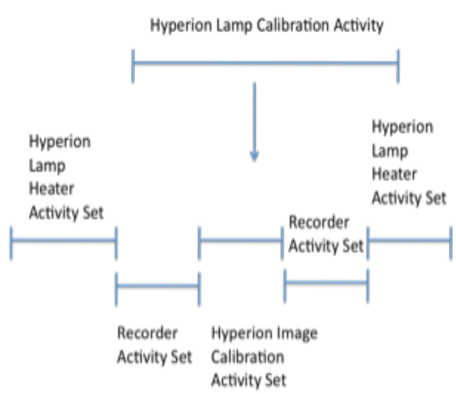

Figure 3: Hyperion Lamp Calibration Activity Decomposition

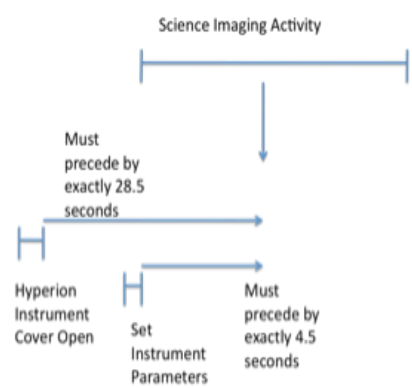

Figure 4: Temporal Constraints in Science Imaging

that it always remains above 311 to prevent the instrument from damage in the cold of space. The VNIR is thusly readily available to image at its minimum operating temperature 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 observations) can cause this temperature to build up without a chance to cool down. Because the EO-1 mission flight rules include an instrument maximum rated operating temperature of 415 as well as a maximum design temperature of 515, the planner must space its observations so the duty cycle does not lead to unacceptable temperature build up.

Similarly, the SWIR module has a maximum operating temperature of 415 and a maximum design temperature of 515, but it does not have any defined optimal operating temperature minimum. The SWIR module is allowed to cool arbitrarily, eventually reaching an oscillating equilibrium with the rest of the spacecraft, the sun (if visible), and space at between -19 and -16. Notably, at these lower temperatures, the SWIR module is much less effective at dissipating heat (and more suceptible 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.

We began by implementing a simple model using a non-depletable resource in which the planner modeled the SWIR and VNIR temperature as instantaneously increasing when turned on and cooling after turned off with the time duration of the increase based on the on-time. This model had the advantage of only requiring local timeline propagation (e.g. the duration of the effect on the temperature timeline is local based on a calculated on-time duration). Each time instrument operations are added or deleted from the timelines, this model requires re-computation of the instrument temperature over the duration of the on-time chance.

We took this indirect approach of modeling a depletable resource as a non-depletable resource due to a number of efficiency concerns. First modeling run time is a concern because we only have a 5 MIPS onboard computing budget for all of the autonomy software. Non-depletable activities only change the resource timeline twice, once at the start of the activity and once at the end of the activity. The natural heating and cooling activity driven model is periodic and requires a number of activities (and potential resource changes) proportional to the length of the modeled schedule – significantly increases the CPU time & RAM for the temperature model.

Unfortunately, the non-depletable model is very inaccurate when consecutive imaging events occur before the instrument is allowed to completely cool to ambient temperature, as often occurs for the SWIR subsystem in EO-1 operations.

Our next iteration model used the starting temperature of the instrument to calculate its duration and therefore again caused a localized effect to the temperature timeline but had a longer duration the higher the input temperature. This model still retains the non-depletable efficiency in runtime and RAM. However, this model also produces inaccurate estimates in cases where a large number of observations occur consecutively.

We finally directly modeled the timeline temperature with a periodic heating and cooling timestep affecting the timeline temperature based on a sampling of if the instrument is on or off. This model directly accounts for the heating and cooling effects most accurately but has the downside of being the most costly computationally to update and propagate during planning. Even this model does not have the most accurate parameters (such as a variable heating and cooling rate based on the current temperature) - the instrument points heat more slowly and cool more quickly when at higher temperatures.)

Figure 5 shows the SWIR temperature as observed in flight, modeled in simulation, and modeled by the planning system. Figure 6, 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 observations exceed the temperature limit" and "how long must the instrument warmup so that this observation 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) this data only included single and dual observations per orbit. In order 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 observations each.

Pointing and wheel Biasing

Another challenging operations constraint for EO-1 is pointing. The EO-1 spacecraft has three reaction wheels for pointing plus a magnetic torquer bar for momentum dumping. Reaction wheels change the orientation of a spacecraft by Newton's Third Law (equal and opposite reaction). Intuitively, spinning a wheel at one end of the spacecraft will cause a rotation in the spacecraft in the opposite direction. Because the spacecraft is in orbit around the Earth, if it continually points directly downward towards the Earth, it will make one 360 degree rotation per orbit. From a spacecraft stability standpoint, for ideal imaging each reaction wheel should be at a target speed of 100 rpm in either direction. Faster or slower speeds are less desirable for imaging quality and reaction wheel wear.

The worst case for image quality occurs when the wheels change direction from spinning in one direction to the opposite (e.g. rate going from positive to negative called a "zero crossing") as the spacecraft will shudder and cannot acquire a high quality image for a period of time. In order to prevent momentum buildup EO-1 has a torquer bar, which applies torque to the spacecraft based on interaction between the magnetic field from running an electrical current through the bar and the Earth's magnetic field.

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. One challenge is that computing these maneuvers is a challenging flight dynamics problem – the maneuver planning software in fact uses a heuristic method to attempt to design such maneuvers that respect rate constraints, timing constraints, and instrument pointing constraints. 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 observation O2 followed by observation O3 is not possible. But the same software might return that starting from nadir, taking O1 followed by O2 followed by O3 is possible. Clearly this means that moving from nadir to O2 to O3 is possible but that the solution through O1 was not found by the maneuver planning software when planning for only O2 and O3. 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 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 planning system cannot directly represent the pointing and momentum state of the spacecraft, we implemented these constraints in the goal generation process (see below). 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.

The EO-1 spacecraft has gimbaled solar arrays that track the sun when visible to generate power. Operationally, the EO-1 spacecraft may generate less power when it is imaging more because its pointing actions make it more difficult for the solar panels to track the sun and generate energy. Also, imaging more implies that more subsystems are powered so they are using more power. Preliminary analyses indicate that even with significantly increased imaging (150 images per week) power will not limit imaging but as the upgrade becomes operational and is checked out the EO-1 power situation will be tracked and analyzed if necessary.

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.

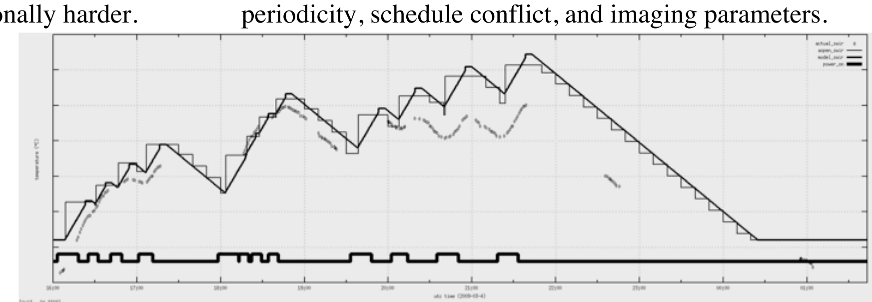

Figure 5: SWIR temperature as: modeled in the planner (square model)
simulated (upper dark line)
observed in flight (stream of circles)
instrument power state (dark line at bottom)

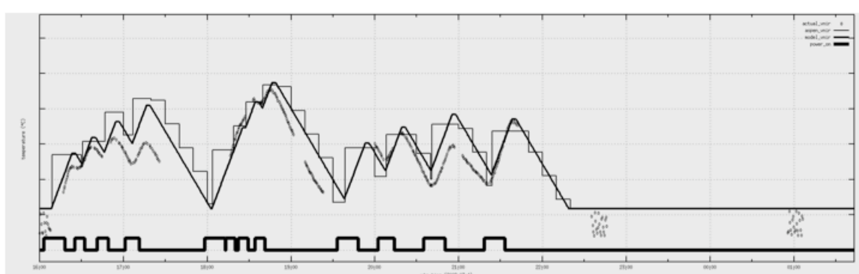

Figure 6: VNIR temperature as: modeled in the planner (square model)
simulated (upper dark line)
observed in flight (stream of circles)
instrument power state (dark line at bottom)

Creating Schedules ground and flight: integrating modeled & implied constraints

One of the challenges of EO-1 operations is to tractably generate schedules given the large numbers of combinations of observations and heavily interacting constraints. On EO-1 there are two very different version of the scheduling problem – weekly schedule generation and onboard replacement scheduling.

On the ground a weekly schedule of EO-1 operations is maintained. This schedule is generated in several versions 3-5 days prior to its start (e.g. the schedule is generated Wednesday for the week starting the following Sunday 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 onboard version of the scheduling problem is more constrained. Due to limited computational resources onboard the spacecraft the onboard scheduler cannot consider the weekly scheduling problem and instead considers a small number of new tuples within a schedule horizon of 8 hours potentially inserting or replacing existing tuples while respecting priorities and operations constraints.

One challenge relating to schedule generation is science priority. 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 (Ground)

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 the greater 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 in three conditions: a) the new tuple can be added without needing to remove anything from the schedule (e.g. does not create any conflicts); b) it can be added and can be conflict free after deleting all subsumed observation tuples currently in the schedule (e.g. if adding a tuple with observations A B and C, if deleting the tuple A B enables A B C to be added without conflict accept ABC); or c) if the new tuple conflicts with an X-band, only add the tuple if after deleting the X-band a) or b) above holds.

The net effect of this scheduling algorithm is that it starts out with tuples (note that single observations are degenerate tuples) with only high priority observations. It then adds lower priority observations where they fit in between high priority observations (not too common) or by growing the tuples with high priority scenes by adding lower priority scenes to the tuples. 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 a larger number of lower priority scenes.

```
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

Onboard Scheduling (Flight)

The onboard scheduling problem is much simpler than the ground (weekly) scheduling problem. In flight the scheduler must accept incremental changes to the baseline (weekly) schedule derived on the ground. These changes are generally due to the addition of new higher priority goals but in theory could also be due to anomalies onboard. The onboard scheduling algorithm is the iterative repair, local search algorithm outlined in Rabideau et al. [Rabideau et al. 1999] but heuristically informed to delete the tuple with the lower priorities. This algorithm is shown below as Algorithm 2.

```
repair
  while conflicts
    heuristically choose a conflict
    heuristically choose a repair method
    apply repair method

choose repair method
  if conflict is between tuples
    lexicographically compare priorities of
scenes in tuples
    return tuple with smallest priority
comparison
```

In a lexicographical comparison, the priority of tuple T1 is less than the priority of tuple T2 iff the highest priority scene of T1 (T1S1) is less than the highest priority scene of T2 (T2S1), or they're equal and T1S2 is less than T2S2, etc. With all priorities being equal, a tuple with fewer scenes is lower priority.

Algorithm 2

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).

When we decided to upgrade the model, we decided to investigate if the total number of scenes could also be significantly increased as part of the upgrade. In order to assess this potential gain we ran a number of simulations with rough constraints – these indicated the potential to increase the number of scenes acquired through better thermal management of the instruments and reducing 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. O3 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.

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. 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.

Discussion, Related Work, and Conclusions

One of the recurring themes in space operations is validation and reliability. Because of the high costs of space missions, reliable operations are very important. 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, which have the advantage of repeatability to facilitate testing. It is an unusual event to find a novel way of doing something in space operations - the more common case is large scale scheduling of combinations of repeated sequences (such as observations).

As listed in the introduction, space mission operations have been a fertile area of applications and research for automated planning and scheduling. This paper has tried to focus on details of constraint representation for a specific mission model, EO-1 as well as our specific heuristic scheduling algorithm.

This paper has described a number of challenges in representing operations constraints and automatically scheduling operations for an earth observing satellite, the EO-1 spacecraft. We described a large number of operations constraints that were naturally modeled in an expressive planning and scheduling system including states, resources, temporal constraints, and decompositions. We then described a number of constraints that were more challenging to model including thermal, location/pointing, and science/image quality. We then described our heuristic approach to EO-1 schedule generation: (1) documenting its significant increase in science observations; and (2) showing its performance approaches that of an upper bound on optimal scheduling

References

J. L. Bresina, A. K. Jonsson, P. H. Morris, K. Rajan, Activity planning for the mars exploration rovers Proc 15th International Conference on Automated Planning and Scheduling, 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 et al. , “ASPEN - Automating Space Mission Operations using Automated Planning and Scheduling” International Conference on Space Operations. Toulouse, France. 2000

S. Chien, et al. , “An Autonomous Earth Observing Sensorweb,” IEEE Int Sys, May-Jun 2005a, pp. 16-24.

S. Chien wt al. , “Using Autonomy Flight Software to Improve Science Return on Earth Observing One, Jnl Aero Computing, Information, & Comm, April 2005b, AIAA.

C. Chouinard, et al. , Automated & Adaptive Mission Planning for Orbital Express, Space Operations Symposium, Heidelberg, Germany, 2008.

M. Deale et al, “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, et al. , "Improving the Observing Efficiency of Hubble Space Telescope," AIAA Aerospace, AIAA Press, 1993.

G. Rabideau, et al. , "Iterative Repair Planning for Spacecraft Operations in the ASPEN System," Intl Symp AI Robot & Aut in Space, Noordwijk, Netherlands, 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.