Planning and Scheduling for Fleets of Earth Observing Satellites

Jeremy Frank[2] Ari Jónsson[1,2] Robert Morris[2], David E. Smith[2]

1. Research Institute for Advanced Computer Science (RIACS)
2. Computational Sciences Division
NASA Ames Research Center, MS 269-2
Moffett Field, CA 94035

May 3, 2001

Abstract

We address the problem of scheduling observations for a collection of earth observing satellites. This scheduling task is a difficult optimization problem, potentially involving many satellites, hundreds of requests, constraints on when and how to service each request, and resources such as instruments, recording devices, transmitters, and ground stations. High-fidelity models are required to ensure the validity of schedules; at the same time, the size and complexity of the problem makes it unlikely that systematic optimization search methods will be able to solve them in a reasonable time. This paper presents a constraint-based approach to solving the EOS scheduling problem, and proposes a stochastic heuristic search method for solving it.

1 Introduction

NASA's growing fleet of Earth-observing satellites employ advanced sensing technology to assist scientists in the fields of meteorology, oceanography, biology, and atmospheric science to better understand the complex interactions among Earth's lands, oceans, and atmosphere. Demand on these satellites is already high, and is expected to increase significantly in the near future. Currently, science activities on different satellites (e.g. the AM Constellation) or even different instruments on the same satellite (e.g. the ASTER instrument on the Terra satellite [11]), are scheduled independently of one another, requiring the manual coordination of observations by communicating teams of mission planners.

It is unlikely that this approach to daily mission planning and scheduling will be viable in the future. As constellation sizes and the number of observation requests grow large, manual coordination will no longer be possible. A more effective way to manage observation scheduling is by allowing customers of the data (viz. the scientists themselves) to request data products, and centrally schedule all requests using information about all possible data gathering resources. Customer preferences will constrain which satellite or satellites will be used to collect the data. Automated techniques can reason about all of the resources that are involved in collecting data, storing the data temporarily on board satellites, and transmitting the data back to Earth. This will enable more efficient management of the fleet of satellites as well as the communication resources that support them.

In this paper we discuss the problem of scheduling observations for a collection of earth observing satellites. We first formulate the problem in Section 2 as a constrained optimization problem, involving a set of observation requests, each with associated constraints that must be satisfied by any solution to the problem, and a set of resources, including imaging instruments, solid state recorders (SSRs), antennae and transmitters, and ground stations. Typically, there will be too many observations to schedule with available satellite resources. Therefore, we assume requests are prioritized, and search for the *best* subset of requests to service, subject to operational constraints. In Section 3, we survey approaches to solving the EOS scheduling problem. In Section 4, we introduce our approach to solving the problem, based on the Constraint-Based Interval Planning (CBIP) paradigm [16]. In CBIP, actions and fluents (or states) are uniformly described as intervals during which a state variable maintains a particular value. CBIP uses a model to specify how states are related to each other in a plan. Candidate

plans are represented by variables and constraints which reflect the temporal relationships between actions, ordering decisions between actions, and the parameters of states or actions. In Section 4 we also formulate our approach to conducting and controlling an algorithm based on Heuristic Biased Stochastic Search (HBSS) [3] using a heuristic related to priority and resource contention.

2 Problem Description

We assume that constellations of the future will contain many satellites with heterogeneous capabilities. The satellites may be in any orbit. Each satellite is equipped with a suite of instruments; some satellites have pointable instruments, giving increased flexibility in what they can observe at any point in an orbit. Further, some imaging instruments are meant to be on almost continuously, in order to ensure global coverage (e.g. the ETM+ on Landsat 7). Others are designed to be operating on a limited basis to obtain high resolution, detailed maps of selected parts of earth's land surface.

Image data acquired by an earth observing satellite are either downlinked in real-time, or recorded on board for playback at a later time. TDRSS satellites and ground stations are available to receive downlinked images. Different satellites may be able to communicate with only a subset of these resources, and transmission rates will differ from satellite to satellite and from station to station. Further, there may be different costs associated with playing back data through different ground stations.

An observation request is typically specified in terms of the type of data and instrument desired, a series of locations and times for the sensing event, and a priority for satisfying the request. A proposed observation sequence must satisfy a number of constraints. These constraints include the requirement that the observation requests be matched with the satellites capable of collecting the requested data, and that observation times must obey duration and ordering constraints associated with the imaging, recording, and downlinking tasks. In addition, SSR capacity, and constraints on communications equipment such as satellite antennae and ground stations must be satisfied. There may also be set-up times associated with satellite systems, which generate further ordering constraints. Servicing requests may involve

coordinating activities among different satellites. For example, a stereo image will involve multiple sensing events of the same location at different viewing angles. In other cases, adequate spectral coverage may require the use of two or more instruments to sense the same land area, or to sense both land use and atmospheric conditions. Finally, scientists may want to image the same area at different times of day.

There will be too many observations to schedule with available satellite resources. Solutions are preferred based on objectives such as maximizing the number of high priority requests serviced and the expected quality of the observations, and minimizing the cost of downlink operations.

EOS science management requires continuous scheduling and rescheduling of activities. Requests can be submitted at any time, and high priority targets of opportunity (e.g., fires, earthquakes, volcanos) may result in the need for updating a partially executed schedule. In addition, there are numerous sources of uncertainty in the satellite observation scheduling domain. One of the most important, and difficult, aspects of the EOS scheduling problem arises from the uncertainty of the weather, specifically, with respect to cloud cover. On the one hand, image quality typically is heavily determined by the amount of cloud cover; on the other hand, many parts of the world have long seasons where clouds are omnipresent, and if a simple "no cloud" scheduling policy were followed, these parts of the world would virtually never be observed. Thus, it is important to enforce a sophisticated scheduling policy which mollifies a "no cloud" cover restriction with the need for coverage.

3 Previous Work

Previously reported work on EOS scheduling problems includes both theoretical investigations using abstract models, as well as operational schedulers for ongoing EOS missions. We divide our survey of previous approaches into two parts: modeling and algorithms.

3.1 Models of EOS Scheduling

Very few theoretical approaches consider multiple satellites or the coordination of observations. Burrowbridge [4] discusses the important problem of managing telemetry and data acquisition (TDA) resources needed by multiple satellites, but does

not treat problems involving observations, data gathering, or downlinking data. Theoretical approaches usually involve simplified models of the satellites and communication resources. For example, Lemâitre et al. [10], Pemberton [12] and Wolfe and Sorensen [18] do not discuss on-board data storage or communications system management. Bensana et. al. [2] describe problems with on-board storage constraints, but without communications system management. Pemberton [12] and Wolfe and Sorensen [18] assume that there are no precedence constraints or any other logical constraints between the requests, while Lemâitre et al. [10] and Bensana et al. [2] compile the complex constraints down to simple binary and trinary exclusion constraints.

There are several operational systems for ongoing EOS missions. The ASTER scheduler described in [11] and the Landsat 7 scheduler [13] are two examples. These schedulers rely on quite detailed models of the satellites and the communications environment when scheduling operations. However, they do suffer from some limitaitons. For example, ASTER scheduling is performed independently of other instruments onboard the Terra satellite. A fixed amount of memory is allocated for this instrument; if it is unused, it can't be used by any other instrument, resulting in suboptimal schedules. Additionally, these models do not account for all of the steps that occur on board the satellites during operations. For instance, the ASTER instrument is aimable, yet there is no accounting for the time required to aim the instrument between observations. Similarly, Landsat requires time to shut down and power up its instrument; this is assumed to take place between scenes. While this may be sufficient for Landsat, it may not be good enough for future satellites with more advanced capabilities. A notable exception is AS-PEN, which was used to model and solve the EO-1 data acquisition scheduling problem [14, 15]. AS-PEN is an integrated planning and scheduling system that can represent complex resources, activities that take time, as well as subgoals of activities. However, the scheduling problem described in [15] does not appear very difficult; EO-1 can only schedule 4 observations a day. It is not clear how their approach scales to many satellites with many instruments of varying capabilities.

As mentioned previously, most of the problems described in these papers are optimization problems. The usual goal is to maximize the weighted sum of the scheduled observations. Wolfe and Sorensen [18] describe a slightly different problem, in which observations are valued based on when they are performed and how much data is collected. This makes the optimization problem more difficult to solve.

3.2 Scheduling Algorithms

Many of the search algorithms described in these papers are incomplete algorithms. The primary reason for focusing on such algorithms is that, even for small numbers of satellites, the problems are large enough that solving them optimally is impractical. The usual approach is to greedily select the next highest priority request to try and schedule, and reject it if there is nowhere for it to go. The ASTER scheduler [11] works exactly this way, as does of the approaches described by Wolfe and Sorensen [18]. Pemberton [12] describes a family of algorithms ranging from strictly greedy to complete search; after sorting the requests, blocks of n requests are scheduled optimally, with all previous allocations acting as constraints on the next set of observations to schedule. Wolfe and Sorensen [18] describe a greedy search which sorts requests by priority, breaking ties using the amount of slack (extra space for the request), then greedily scheduling the request in the best place for it. A modification performs lookahead to decide where the request leads to the best schedule, thereby accounting for its future impact. Another modification generates input lists using a genetic algorithm, with the option of rejecting a request preemptorily. Burrowbridge's scheduler [4] greedily schedules requests based on the earliest finishing time of the request. The Landsat 7 scheduler [13] greedily schedules requests based on the earliest finishing time until resources run out, then preempts previously scheduled observations based on priority. ASPEN [14] uses a local search algorithm that generates an initial schedule, then identifies and repairs conflicts in the schedule by changing variable assignments. This algorithm is quite complex, with 10 distinguished types of conflicts and heuristics required to identify both the conflict to work on and the method of addressing it.

As a final note, the priority of observations is normally derived from a number of factors, some of which are dynamically determined. For example, estimated cloud cover and nearness to the end of the feasibility window are normal inputs. The Landsat 7 scheduler [13] also attempts to find

schedules with long sequences of adjacent scenes to reduce the overhead on data acquisitions.

4 Technical Approach

We believe that effective coordination of EOSs requires high-fidelity modeling of the entire EOS environment. Not only do we need to model onboard satellite resources, communication resources and requests, but we must also model the detailed activity sequences on the spacecraft and on the ground. However, we would like to make use of search techniques developed for solving combinatorial problems. To balance these needs, we use the Constraint-Based Interval Planning (CBIP) framework.

4.1 Constraint-Based Interval Planning

The CBIP framework [16] is based on an interval representation of time. A *predicate* is a uniform representation of actions and states, and an *interval* is the period during which a predicate holds. A *token* is used to represent a predicate which holds during an interval. Each token is defined by the start, end and duration of the interval it occurs, as well as other parameters which further elaborate on the predicate. For instance, a `Take-Image` predicate may have a parameter describing the resolution, which can be either `low` or `high`. The planning domain is described by *planning schemata* which specify, for each token, other tokens that must exist (e.g. pre and post conditions), and how the tokens are related to each other. Figure 1 shows an example of a planning schema. Schemata can specify conditional effects and disjunctions of required tokens. For instance in Figure 1, a `Take-Image` interval can be met by a `Calibration` period if a high resolution image is to be taken. The value of the `?mode` parameter indicates whether or not a `Calibration` period is required. Planning schemata can also include constraints on the parameters of the token. As shown in Figure 1, the `Take-Image` interval has a constraint relating the mode and the amount of data stored by the operation.

EUROPA [6] is a CBIP planning paradigm which continuously reformulates the planning problem as a Dynamic Constraint Satisfaction Problem (DCSP). This is done by mapping each partial plan to a CSP. The temporal constraints form a Simple Temporal Network, which can be ef-

ficiently solved [5], while the rest of the constraints form a general, non-binary CSP represented by procedural constraints [8]. An additional feature includes the ability to produce plans with flexible time; that is, activities may start and end at any time in an interval [9]. This gives the plan some flexibility, should activities take longer or shorter than expected. Figure 2 shows a plan fragment and its induced CSP. Assignments of variables in the CSP correspond either to the adding of new plan steps, or the assignment of parameters of plan steps. As steps are added to or removed from the plan, the CSP is updated to reflect the current partial plan. For example, in Figure 2, adding the `Take-Image` step to the plan requires adding several new variables and constraints to the CSP. At any time, if the CSP is inconsistent, then the partial plan it represents is invalid; if a solution is found to the CSP, then that solution can be mapped back to a plan which solves the problem. The advantage of such a representation is that any algorithm which solves DCSPs can be used to solve the planning problem.
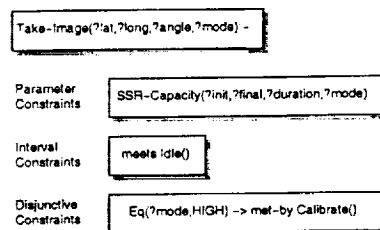

Figure 1: The planning schema for a `Take-Image` interval. This schema consists of four components: the master token of the schema, constraints on the parameters of the schema, a description of other tokens which must exist when the master token is in the plan, and a disjunction of tokens which may exist when the master token is in the plan.

EUROPA has the ability to model various types of resources. A domain model consists of a number of *attributes*, each of which represents an aspect of the objects that interact in the world. Each of these attributes may be in only one state at a time; hence, if a camera is taking an image, it can't also be turning. This permits simple modeling of resources. Complex resources such as fuel and power can be modeled using numerical constraints. In Figure 1, the filling of the SSR is modeled by a constraint that relates the initial amount of storage, the final amount of storage, and the rate at which the data acquisition task fills the buffer.
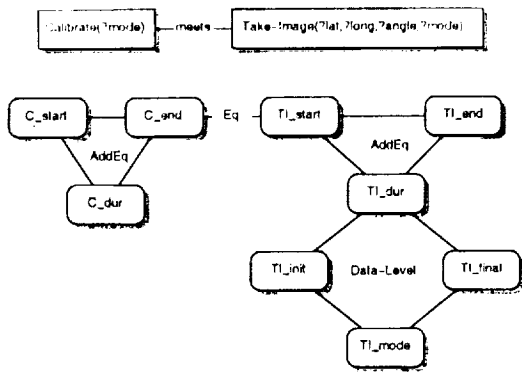
4

Figure 2: A partial plan and its DCSP representation. The partial plan consists of 2 tokens, shown at the top of the figure. The DCSP variables are in rounded boxes. Edges between DCSP variables are labeled with the constraints on those variables.

4.2 A CBIP Model of the EOS Domain

A CBIP model for the EOS domain will describe the attributes of a set of satellites with different types of sensing instruments and resources, as well as different orbital tracks. Resources to be modeled for each satellite include the instruments, the SSR, and a set of antennae and transmitters for downlinking data. We not explicitly model power consumption or satellite maneuver operations, although maneuver periods and power-related duty cycles may constrain the schedule. Other model elements are data receiving stations, either ground stations or TDRSS satellites.

A sensing instrument is defined primarily in terms of the type of data it acquires, its spatial and spectral resolution (for spectrometers), its swath width, and pointing limitations (field of view, slew rate, and so on). A solid state recording device (SSR) is defined by the storage capacity and the rate at which it stores data. Antennae and transmitting devices are defined by whether they are slewable, and also by their data transmission rate. Data receiving stations are associated with a frequency band, and also by the number of downlink channels they support. Each of these entities will correspond to one or more attributes of a model.

Requests are identified by their location, either specified in World Reference System (WRS) units, or latitude and longitude. We may also model a "Quality of Service" (QoS) type for each request. For example, in Landsat 7, requests for images made by non-U.S. international ground stations are usually serviced through direct downlink to the the requesting ground station. By contrast,

so-called "special" requests on Landsat 7 corresponding to exceptional events are typically simultaneously recorded and directly downlinked to a ground station, and later also played back for redundancy. As noted earlier, requests are associated with a user-defined priority, but other, derived priorities emerge during the scheduling process. For example, a request may undergo a boost in priority as a result of the delay since the previous time an image of the area was taken, or because of limited opportunities for capturing the image. Conversely, a request priority may be demoted due to expected excessive cloud cover over the area. A given request may also correspond to a coordinated activity involving multiple instruments. Coordinated observation activities arise for many reasons, for example, to take a stereo image of an area, to sample a region over different spectral regions, or to calibrate instruments.

Each attribute of a CBIP model supports a limited set of activities. Thus, an SSR can be recording, playing back data, or idle, an antenna can be slewing, or pointing to a receiving station, and an imaging instrument can be off, idle, or taking an image. The model will also represent set up events such as warming up an instrument, or slewing for antennae or pointable sensing instruments. Temporal constraints impose restrictions on the duration and ordering of tokens in a plan. Temporal constraints may be associated with a single activity, such as the constraint that an antenna be slewed to a certain location before it can begin pointing at that location; or a temporal constraint can involve pairs of activities, such as the constraint that a ground station must be in contact with a satellite while data is being downlinked. Resource constraints include SSR capacity, communication bandwidth, and duty cycle restrictions on imaging instruments. Figure 3 shows how all of these aspects are combined in a simple model. This model shows the interaction of an instrument attribute and an SSR attribute. The instrument transitions between Pointing, Idle, Calibrating and Take-Image. The SSR transitions between Recording, Playback and Idle. The time required for Pointing, Calibrating, Recording and Playback activities are constrained. In addition, Take-Image and Recording activities must be simultaneous, and whenever a Playback occurs on the SSR the instrument must be Idle.
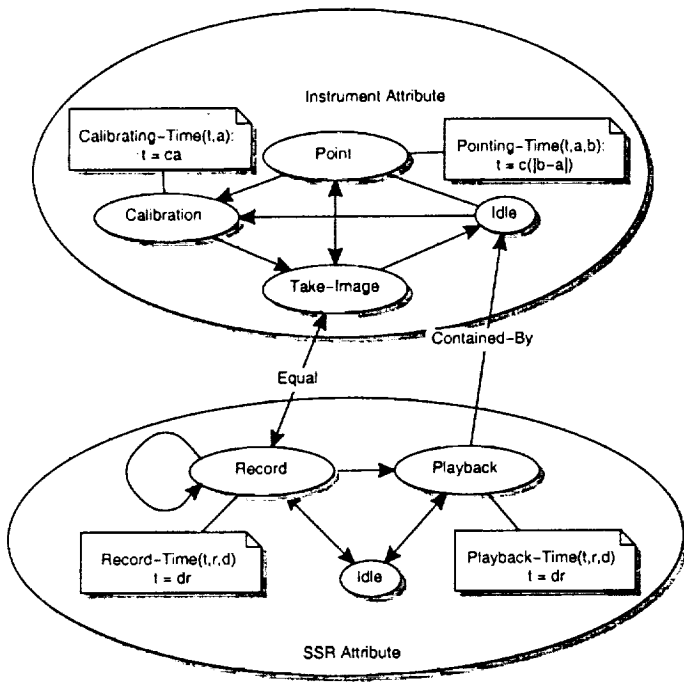
The EUROPA planner supports object-oriented

Figure 3: Simplified model showing the interaction of instrument and SSR attributes.

```
procedure HBSS(Obs)
    P = ∅
    while Obs ≠ ∅
        o = SelectObs(Obs)
        t = SelectTime(o)
        P = P∪ o starting at time t
        Obs = Obs − o
        P = Propagate(P)
        if no plan found return ∅
    end while
    FindPlan(P)
    return P
end
```

Figure 4: HBSS Modified for the EOS Scheduling problem. The algorithm repeatedly selects an observation, then selects a time to schedule the observation or rejects the observation. This assignment is added to the plan, and **Propagate** then performs any inferences that result from the scheduling of the observation. If all observations are scheduled or rejected, **FindPlan** attempts to schedule and subgoals that need to be scheduled, and the resulting plan (there may not be one) is returned. Heuristics strongly drive the **SelectObs** and **SelectTime** steps.

descriptions of models. Most subsystems of satellites are quite similar, so we expect that we can define a relatively large number of different satellites quite easily. We can then vary the parameters of these different satellite models to create more or less challenging EOS domains. For instance, we can vary the transmission rates and SSR capacities of the satellites, the number of ground stations or TDRSS contacts, as well as change the instrument makeup of satellites, to assess the impact of different scenarios for particular sets of requests.

5   The HBSS Algorithm

In theory, the optimal solution to an observation scheduling problem can be found using the well known systematic branch and bound algorithm. Unfortunately, complete search algorithms are simply not practical for most large scheduling problems. Bensana et al. [2] indicate that they were unable to optimally solve problems with more than about 200 observations using Russian Doll Search (a clever but specialized variation on Branch and Bound). Pemberton [12] makes similar observations. The only alternatives are to use some form of greedy search or hill-climbing search, possibly augmented with stochastic variation to escape local optima. Fortunately, for observation

scheduling these approaches tend to work well, because there are usually many local optima that are nearly as good as the global optimum. Thus, by injecting stochastic variation into a greedy search procedure one of these resonably good solutions can usually be found very quickly.

For our purposes, we have chosen to overlay a stochastic gready search algorithm on the constraint-based planning techniques discussed earlier. In particular, the greedy search will choose and schedule observations, and the constraint based planning foundation will propagate constraints to rule out possibilities inconsistent with each observation assignment, and expand individual observations by including any necessary setup and postprocessing steps required by the scheduled observations. The stochastic greedy search algorithm is based on the HBSS algorithm developed by Bresina [3]. The basic algorithm for HBSS looks like a simple greedy search with restarts. A modified version of the algorithm appears in Figure 4.

What distinguishes the HBSS algorithm from ordinary greedy search is the way in which alternatives are chosen in the **SelectObs** and **SelectTime** steps. In a pure greedy search, these choices are made absolutely by a heuristic. In the HBSS algorithm, the heuristic must rank or score the possible alternatives. HBSS then chooses probabilistically from among the alternatives, weighted according to their ranking or score. Thus, possibilities ranked highly by the heuristic have higher probability of being selected, but other lower ranked possibilities are sometimes selected. This means that several alternatives with roughly the same score will have roughly equal probability of being chosen. Because of this stochastic character, alternative schedules are likely to be explored with each successive restart of the algorithm.

The **Propagate** step performs simple inferences after scheduling an observation. These inferences include eliminating choices for observations and otherwise eliminating the values of variables in the DCSP representation of the plan, but may include inserting subgoals into the plan. HBSS only selects observations to be in the plan; these may lead to subgoals, and these also need to be inserted in the plan. Before the HBSS procedure completes, any subgoals that have not been inserted into the plan must be handled; this is done by the **FindPlan** step.

Like most search procedures, the effectiveness of HBSS depends critically on the quality of the heuristic advice. Bresina [3] has shown that HBSS is particularly effective when the ranking heuristics typically give good advice. As the quality of the heuristic advice declines, HBSS must search progressively longer (more restarts) to find near optimal schedules. In the next section we develop contention heuristics for ranking observation choices.

5.1 Contention Heuristic

The success of Greedy search methods depends largely on the heuristic used to decide which variable to assign next, and which value to assign to that variable. These steps correspond to the **SelectObs** and **SelectTime** steps.

For observation scheduling, an obvious heuristic for choosing an observation is to select the one with the highest priority. In general, this will ensure that the schedule is loaded with as many high priority observations as possible before any lower priority observations are considered. However, there may be many observations with the same priority,
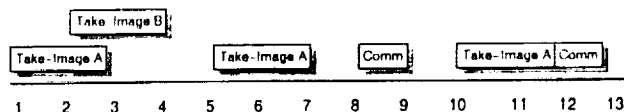

Figure 5: The impact of variable and value ordering. `Take-Image A` has three possible timeslots, while `Take-Image B` has only 1. The temporal constraints imply that scheduling `Take-Image A` at time 1 makes it impossible to schedule `Take-Image B` at all, since it can only start at time 2.

and the order in which we consider these observations can have a dramatic impact on the resulting schedule. For example, consider the simple example shown in Figure 5. Here there are two observations, A and B, of equal priority. As shown, there are several opportunities for scheduling A, but only one opportunity for scheduling B, which overlaps with the first opportunity for A. If we choose observation A first, and foolishly schedule it in the first available time slot, then observation B will not appear in the schedule. In contrast, if we were to schedule B first, other opportunities would still remain for observation A.

These examples suggest a simple rule of thumb for choosing which observation to schedule next: prefer observations having the fewest remaining opportunities. This heuristic resembles the Minimum Remaining Values (MRV) heuristic commonly used in the CSP community [7]. Calculating the number of remaining opportunities for an observation is appealing because it is simple to compute, and provides at least some estimate of how easy it is to schedule that particular observation. However, it does not give any estimate of how much "contention" there is for those opportunities. For example, if there are two remaining opportunities for a high priority observation, but absolutely no contention for one of the time slots, then the observation will always be easy to schedule. In contrast, if there are numerous other observations that could use those time slots, then there is good reason to schedule the observation early, to make sure it gets one of those time slots.

This leads us to a more sophisticated measure of contention. To start with, we will only consider contention for time slots. We first define some terms: Observations(t) is the set of observations that could occur at time t, and Opportunities(o) is the set of discrete opportunities for observation o (noting that each discrete opportunity is exactly

long enough to accommodate the window.) For a given time slot, we could measure contention by counting the number of observations that want that time slot, weighted by the priority of the observation:

$$\text{Contention}(t) = \sum_{o \in \text{Observations}(t)} \text{Priority}(o)$$

However, this measure doesn't incorporate how badly each observation needs the time slot; i.e. if an observation can be scheduled in only that time slot, it needs the time slot badly, but if it can be scheduled in lots of different time slots, it doesn't need the time slot very badly at all.

We can define the *need* of an observation as:

$$\text{Need}(o) = \frac{\text{Priority}(o)}{|\text{Opportunities}(o)|}$$

The *contention* for a particular time slot can then be defined as:

$$\text{Contention}(t) = \sum_{o \in \text{Observations}(t)} \text{Need}(o)$$

The *contention* for a particular observation can then be defined as:

$$\text{Contention}(o) = \min_{t \in \text{Opportunities}(o)} \text{Contention}(t)$$

We take the minimum because there may be an easy place to put an observation, and the contention of an observation should not be lowered by slots that are in higher demand. In other words, adding another opportunity for an observation should never increase the contention measure for that observation. Note, however, that contention should be recomputed as observations are scheduled to account for slots that are no longer available, leading to higher contention for the remaining observations.

In developing the equations above, we regarded observations as if they only required a single scene or time slot, and could only be scheduled for that slot (i.e. no window of opportunity). For observations that involve a sequence or group of scenes we would have to sum up (or maximize over) the contention measures for each of the individual scenes (time slots). With pointable instruments, there is an interval during which a given scene could be

taken. This can also be incorporated (with some further complication of the equations); this would resemble heuristics that attempt to maximize the slack in a schedule [17, 1].

Measuring contention for a global resource like SSR capacity involves generalizing the above contention measure to consider the amount of the resource needed by an observation, the resource capacity, and the interval of time under consideration.

Let $\text{Requires}(o, r)$ be 1 if observation o requires resource r and 0 otherwise, and let $\text{Capacity}(r, i)$ be the capacity of a resource over a time interval i. Thus, an SSR with a capacity of 50 has a $\text{Capacity}(r, i) = 50$; if a playback of 20 units occurs within the interval i, then $\text{Capacity}(r, i) = 70$. We then generalize the above definitions to be:

$$\text{Need}(o, r) = \text{Requires}(o, r) \frac{\text{Priority}(o)}{|\text{Opportunities}(o)|}$$

$$\text{Contention}(r, i) = \frac{\sum_{o \in \text{Observations}(i)} \text{Need}(o)}{\text{Capacity}(r, i)}$$

$$\text{Contention}(r, o) = \min_{i \in \text{Opportunities}(o)} \text{Contention}(r, i)$$

Again, note that these measures change as activities are scheduled. In particular, as activities that empty the SSR are scheduled $\text{Capacity}(r, i)$ may increase, and as observations are scheduled $\text{Capacity}(r, i)$ may decrease. Intuitively, these contention measures provide a more accurate assessment of how hard it is to actually schedule an observation. Using these measures, our variable ordering heuristic is:

> Schedule the observation of highest priority and highest overall contention

where contention will be a weighted sum of contention measures for the different resources (time slots, SSR capacity, ...). This approach assumes that resources are *independent*; while not true, it does provide an efficiently computable approximation. This heuristic provides a ranking of observations suitable for use with the HBSS search procedure.

Given an observation to schedule, we would prefer to put it in the place where it will compete with the fewest other observations. We can use the above contention measures to define a value ordering heuristic:

Schedule an observation in the opportunity with the least contention

Again, this heuristic provides a ranking suitable for use with the HBSS search procedure.

6 Conclusions and Future Work

We have presented the problem of scheduling observations on a collection of Earth Observing Satellites and discussed a candidate representation and solution methodology. In order to produce good plans, we advocated a high-fidelity model incorporating both satellite resources and communications resources. In order to gain maximum flexibility in solving problems, we used the CBIP paradigm, which gives us access to algorithms from the DCSP community. We believe that this problem is large enough and complex enough that a biased greedy stochastic search method with a well-motivated heuristic is the best approach. We have motivated and described such a heuristic, and shown how it can be integrated with a modified form of the HBSS algorithm.

Our next tasks are to choose the final form of the heuristic, select the bias function to be used with HBSS, and select the exact method by which subgoals of scheduled observations will be inserted into the plan. Once this is done, we can then begin experiments to test the effectiveness of this procedure on large, heterogeneous EOS scheduling problems.

References

[1] J. C. Beck and M. Fox. A generic framework for constraint-directed search and scheduling. *AI Magazine*, 19(4):101–130, 1998.

[2] E. Bensana, M. Lemâitre, and G. Verfaillie. Earth observation satellite management. *Constraints*, 3(4), 1999.

[3] John Bresina. Heuristic-biased stochastic search. In *Proceedings of the 13th National Conference on Artificial Intelligence*, 1996.

[4] S. Burrowbridge. *Optimal Allocation of Satellite Network Resources*. PhD thesis, Virginia Tech, Virginia, 1999.

[5] R. Dechter, I. Meiri, and J. Pearl. Temporal constraint networks. *Artificial Intelligence*, 49:61–94, 1991.

[6] J. Frank, A. Jónsson, and P. Morris. On the representation of mutual exclusion constraints for planning. In *Proceedings of the Symposium on Abstraction, Reformulation and Approximation*, 2000.

[7] R. Haralick and G. Elliot. Increasing tree search efficiency for constraint satisfaction. *Journal of Artificial Intelligence*, 14:263–313, 1980.

[8] A. Jónsson and J. Frank. A framework for dynamic constraint reasoning using procedural constraints. *Proceedings of the Euopean Conference on Artificial Intelligence*, 2000.

[9] Ari K. Jónsson, Paul H. Morris, Nicola Muscettola, Kanna Rajan, and Ben Smith. Planning in interplanetary space: Theory and practice. In *Proceedings of the Fifth International Conference on Artificial Intelligence Planning and Scheduling*, 2000.

[10] M Lemâitre, G. Verfaillie, F. Jouhaud, J. Lachiver, and N. Bataille. How to manage the new generation of agile earth observation satellites? In *Proceedings of the International Symposium on Artificial Intelligence, Robotics and Automation in Space*, 2000.

[11] H. Muraoka, R. Cohen, T. Ohno, and N. Doi. Aster observation scheduling algorithm. In *Proceedings of the International Symposium Space Mission Operations and Ground Data Systems*, 1998.

[12] J. Pemberton. Towards scheduling over-constrained remote sensing satellites. In *Proceedings of the 2d International Workshop on Planning and Scheduling for Space*, 2000.

[13] W. Potter and J. Gasch. A photo album of earth: Scheduling landsat 7 mission daily activities. In *Proceedings of the International Symposium Space Mission Operations and Ground Data Systems*, 1998.

[14] G. Rabideau, R. Knight, S. Chien, A. Fukanaga, and A. Govindjee. Iterative planning for spacecraft operations using the ASPEN system. In *Proceedings of the International Symposium on Artificial Intelligence, Robotics and Automation in Space*, 1999.

[15] R. Sherwood, A. Govindjee, D. Yan, G. Rabideau, S. Chien, and A. Fukanaga and. Using ASPEN to automate EO-1 activity planning. In *Proceedings of the IEEE Aerospace Conference*, 1998.

[16] D. Smith, J. Frank, and A. Jónsson. Bridging the gap between planning and scheduling. *Knowledge Engineering Review*, 15(1), 2000.

[17] S. Smith and C. Cheng. Slack-based heuristics for constraint satisfaction scheduling. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, pages 139–44, 1993.

[18] W. Wolfe and S. Sorensen. Three scheduling algorithms applied to the earth observing domain. *Management Science*, 46(1), 2000.