N95- 23741

A Constraint-Logic Based Implementation of the "Coarse-Grained" Approach to Data Acquisition Scheduling of the International Ultraviolet Explorer Orbiting Observatory

Bruce McCollum
Science Programs, Computer Sciences Corporation
Code 684.9, Goddard Space Flight Center
Greenbelt, MD 20771
tel: (301) 286-7537 fax: (301) 286-7642
e-mail: mccollum@iuesoc.gsfc.nasa.gov

Mark Graves
Department of Cell Biology
Baylor College of Medicine
One Baylor Plaza, Houston, TX 77030
e-mail: mgraves@bcm.tmc.edu

KEY WORDS AND PHRASES

Artificial Intelligence, Constraint Logic Programming, IUE Astronomy Satellite, Satellite Scheduling

BACKGROUND

In 1994 there are six large, long-lived astronomy satellites in operation. Additional missions are planned by the U.S. and other countries by the end of the decade. The general problem of setting up a yearly schedule of science observations for an astronomy satellite is a challenge which will exist in various incarnations for the foreseeable future.

Every year, each orbiting observatory typically carries out observations for several dozen different science programs, collecting data on up to a few hundred different objects in the sky by the end of the year. The number of distinct observations ("exposures") carried out each day with each satellite varies from one to more than ten, depending on the particular satellite. For every satellite, an annual schedule for these hundreds of observations must be set up which obeys the physical and operational constraints of the satellite and the scientific constraints of the many different science programs. In general terms, the problem of science scheduling in a satellite mission is usually cast as attempting to find the best schedule out of an enormous number of possible schedules.

The International Ultraviolet Explorer (IUE) satellite observatory has been in operation continuously since 1978. It typically carries out several thousand observations per year for over a hundred different science projects. These observations, which can occur in one of four different data-taking modes, fall under several satellite-related constraints and many other constraints which derive from the science goals of the projects being undertaken.

One strategy which has made the scheduling problem tractable has been that of "coarse-graining" the time into discrete blocks of equal size (8 hours), each of which is devoted to a single science program, and each of which is sufficiently long for several observations to be carried out. We call it "coarse-graining" because the schedule is done at a "coarse" level which ignores fine structure, i.e., no attempt is made to plan the sequence of observations occurring within each time block. Planning science observations on a "fine" level, within each time block, is done by the guest investigator whose program has been allocated that time block.

Coarse-graining the schedule has several advantages. It reduces the number of time blocks composing a schedule from several thousand to 730. Because most time blocks can be scheduled independently, it permits rapid rearrangements of the schedule with a minimal effect on the overall schedule. It also gives guest investigators the freedom to make last-minute changes in their observations based on new results or new thinking, which can significantly enhance the quality of science; although important in science, such qualitative human judgement cannot readily be represented in any scheduling algorithm.

Another advantage is that coarse-graining increases the observatory's ability to make significant changes in the schedule on short notice with minimal impact to the rest of the schedule, because the time blocks are usually mutually independent. In a fine-grained schedule where a linear sequence of (e.g.) a thousand distinct observations must be planned, moving the time of one observation causes a change in the time of all the rest. This is due to the fact that the time required to obtain one data set of one target (the "exposure time", as in photography nomenclature) varies from seconds to hours. In a fine-grained schedule, if a short exposure time (small time slot) is replaced by a long exposure time (large time slot), then every observation for the rest of the schedule must be moved later in time if the schedule is to avoid "gaps". In a coarse-grained schedule, however, the schedule is divided into (fewer) time blocks having equal length which are usually independent and interchangable. Thus an observer can change the length of exposure or number of exposures within his or her time block without affecting the other time blocks in the schedule. Some ability to make schedule changes on short notice is necessary in most astronomy satellites because a number of important classes of astronomical objects (e.g. novas, supernovas, transient X-ray sources) appear suddenly and unpredictably and may fade rapidly so that data must be obtained quickly before the target is no longer detectable.

SOFTWARE SOLUTION

We have incorporated the IUE's coarse-grained approach in new software which examines the science needs of the observations and produces a limited set of alternative schedules which meet all of the instrument and science-related constraints. With this algorithm, the IUE can still be scheduled by a single person using a standard workstation, as it has been. We believe that this software could be adapted to a more complex mission while retaining the IUE's high flexibility and efficiency. This has the potential of improving the efficiency and scientific return of future satellite missions.

Our first step was to develop a representation for the constraints sufficient for scheduling relatively simple satellites and implement a constraint logic program which accesses the representation and discovers a set of coarse-grained schedules [2]. Our coarse-grained scheduler can find a collection of schedules which satisfy the overarching spacecraft, instrument, target, and scientific program constraints. A human scheduler can then choose from the set an optimal schedule which maximizes the quality of science, after consulting with guest investigators, if necessary, about priorities and trade-offs.

Data Structures for Domain Representation

Developing an appropriate representation is the first step in developing a system which is data driven or knowledge based. We set up a representation for the programs, targets, instruments, and instrument exposures and a representation for general constraints on them, including constraints on the spacecraft operation.

A schedule consists of a collection of investigator programs which are assigned a number of shifts. The investigator's shifts are scheduled into one of 730 shifts during in the year. The shifts are 8 hour blocks of time which define the level of granularity for the coarse-grained approach.

An investigator requests observations of certain targets. Each observation may consist of one exposure or a set of exposures, and each exposure has a specified instrument (data collection mode) and exposure time. Each target may be viewed only on certain days which depend upon its angle to the sun.

Our satellite scheduling language can represent constraints specific to particular science programs, such as various types of temporal constraints, target observation constraints, and instrument exposures. The constraints which must be represented generally fall into two categories: constraints inherent in the spacecraft and instrumentation which are always in effect, and constraints which reflect the scientific needs of the different programs.

In the case of the IUE, spacecraft/instrument constraints include, among other things:

1. There are four data-taking modes, which can be used only one at a time.
2. The spacecraft is restricted in the directions it can point relative to the Sun. This has the effect of restricting the time of year during which a particular object can be observed.
3. Only short (less than 1 hour) exposures can be taken during part of each day due to high background radiation. This period occurs at the same time each day.

Typical constraints which are due to the specific objectives of the various science programs being carried out include:

1. Each science program is annually allotted a fixed, exact amount of time in which it may use the instrument.
2. The choice of detectors, targets, and exposure times is specified by the guest investigator. This information is solicited at the beginning of the scheduling year.
3. For science reasons, some observations must include or avoid certain dates or ranges of dates.

Overview of the Algorithm

The scheduling algorithm is a constraint logic program which finds all valid schedules and presents them one at a time to the human scheduler. The human scheduler determines whether the presented schedule is sufficient or whether the program should attempt to find an alternative one. The input to the algorithm is a series of constraints which must be met to create a valid schedule. The output is a collection of valid schedules.

C-5

Searching the space of all possible schedules is not tractible. Constraint logic programming addresses this issue by restricting the search space. Constraints on the observation date restrict the schedules which are considered, thus reducing the search space. It is useful to reduce the search space even more using techniques such as priority scheduling, which is described in the next section. After the reduced search space is determined, a simple logic program with backtracking is used to create a list of possible schedules and present them to a human scheduler.

The input to the system is a collection of investigator programs. Each investigator's program consist of a collection of target observations. There are two main constraints on the day which an observation may be scheduled. They are the angle the target has in relation to the sun and the request of the investigator. These constraints are combined to create a constraint on the day of observation for the target. These observation constraints are combined into constraints on possible days to which the entire program can be assigned. The combining of constraints occurs by unifying them in a constraint logic program, which is explained in [2].

Sometimes there are problems with the input, such as inconsistencies. Investigator programs may be inconsistent in two ways: they may be inconsistent individually or as part of a collection. When a program is approved by the IUE review process for observation it is assigned a specific number of shifts, which may not be sufficient to observe all proposed targets. In addition, two or more programs may have conflicting constraints. In either case, the investigator(s) must decide which target observations are more important and inform the human scheduler which ones have higher priority. The algorithm can assist in the process by listing alternative schedules.

Even in consistent schedules, there are additional scheduling tasks which must be supported. In a program with many targets, the investigator's program will generally have to be split across multiple days to meet all the constraints, though this does not have to be handled solely by the algorithm. The constraints used in creating a complete schedule of all observations can also be used to generate options for each investigator's program. There are two kinds of listings that can be created. The first is a list of all possible scheduling days for a program. The second option is to use an existing schedule and list all the ways it could be changed to reschedule an investigator's program. This option is especially important for use when unexpected observation opportunities arise.

Implementation

Our algorithm is implemented in the constraint logic programming language LIFE, which is a fusion of object-oriented, functional, and logic programming paradigms developed at Digital Equipment Corporation [1]. We have selected LIFE for this project because it is especially well suited for handling constraints in a declarative manner.

Because there are usually several schedule changes a week, it is important that the scheduling algorithm have an efficient implementation and support incremental updates. One way to make scheduling more efficient is to prioritize the scheduling of observations based on how constrained the observation day is. All programs are still scheduled and the scheduling priority has no relation to any priorities the investigator may set within a program. The search is made faster by considering observations in order of the severity of constraints on possible observation days, with the most time-restricted observations placed in the schedule first. This prioritization is independent of the dates which are allowed. For example, if the constraints limit one observation to one of five days in the year, then that observation is scheduled before observations which can occur during one of 60 days. There can and will be conflicts which will require backtracking, but there will generally be fewer conflicts than with an arbitrary ordering. (This is the same strategy which has already been used successfully for the IUE.)

The technique used to implement the prioritization is priority scheduling. Priority scheduling is a technique from operating systems research where each "job", in this case an observation request, is placed on an ordered set of queues. All the jobs on the first queue are scheduled before any job on the next one, and the process repeats until all requests are scheduled. If a scheduling year contains 365 days, then the algorithm creates a priority queue with 365 levels. Each observation is placed in the level corresponding to the number of days during the year in which it could be scheduled. The observations are scheduled beginning with the first level and proceeding through all 365 levels. If a conflict occurs, then the algorithm backtracks to remove the conflict.

Other satellite scheduling programs exist which use constraints, but none make use of the coarse-grained approach. Some, such as SPIKE [3] use constraint satisfaction to create fine-grained schedules. However, constraint-logic programming has an advantage over constraint-satisfaction programming in that constraint logic programming provides a mechanism for solving constraints within the framework of a high-level programming language. Within constraint satisfaction, there are typically a limited collection of domains and operators which can be used. Within constraint logic programming, constraints can be placed on any variables which occur in a logic formula. We have chosen to use constraint logic programming because it is easier to develop more complex systems of constraints within it.

DISCUSSION

For the past fifteen years, the International Ultraviolet Explorer (IUE) astronomical satellite has been successfully scheduled by "coarse-graining" the time into large (8-hour) discrete blocks, each of which is devoted to a single science program, and each of which is sufficiently long for several sets of data to be acquired. This approach has worked well. The IUE has established a reputation for a high quantity of observations per year as well as high quality of science resulting from them. Coarse-graining greatly simplifies the scheduling problem by not seeking to find the optimum schedule of all possible schedules, but instead to develop a schedule which meets a minimum but adequate set of scientific and instrument constraints.

Our implementation in its present form would have to be modified to work for satellites other than the IUE. However, as discussed above, the coarse-grained approach has some advantages (flexibility, inclusion of scientific judgement) which would be desirable in most space observatories. Furthermore, most of the science and instrument constraints of the IUE are shared to some degree by most space observatories. We believe that adapting our IUE implementation to other satellites would in most cases be possible without too much difficulty. A modified form of our algorithm could be used to schedule a ground-based telescope by a single person using a standard workstation. Whether our approach would be the best implementation to schedule a particular future mission would require further study in the context of planning that mission.

References

[1] Ait-Kaci, Hassan, 1991. "An Overview of LIFE". In *Next generation information system technology: Proceedings of the First International East/West Data Base Workshop*, 42. Vol. 504 of *Lecture Notes in Computer Science*, eds. J.W. Schmidt and A.A. Stogny, Springer-Verlag.

[2] Graves, M. and McCollum. B. 1994. "Coarse-Grained Scheduling for Astronomy Satellites", 1. In *Knowledge-Based Artificial Intelligence Systems in Aerospace and Industry*, SPIE Technical Conf. 2244.

[3] Johnston, Mark D., 1990. "SPIKE: AI Scheduling for NASA's Hubble Space Telescope", 184–190. In *Proc. 6th Conference on AI Applications*, IEEE.