Artificial Intelligence Approaches to Spacecraft Scheduling

Mark D. Johnston

Space Telescope Science Institute/Space Telescope-European Coordinating Facility

Summary: The problem of optimal spacecraft scheduling is both important and difficult. Efficient utilization of spacecraft resources is essential, but the accompanying scheduling problems are often computationally intractable and are difficult to approximate because of the presence of numerous interacting constraints. We have applied artificial intelligence techniques to the problem of scheduling astronomical observations and other spacecraft activities for the NASA/ESA Hubble Space Telescope (HST). This presents a particularly challenging problem since a yearlong observing program can contain some tens of thousands of exposures which are subject to a large number of scientific, operational, spacecraft, and environmental constraints. We have developed new techniques for machine reasoning about scheduling constraints and goals, especially in cases where uncertainty is an important scheduling consideration and where resolving conflicts among conflicting preferences is essential. These techniques have been utilized in a set of workstation-based scheduling tools for HST. Graphical displays of activities, constraints, and schedules are an important feature of the system. High-level scheduling strategies using both rule-based and neural network approaches have been developed. While the specific constraints we have implemented are those most relevant to HST, the framework we have developed is far more general and could easily handle other kinds of scheduling problems. This paper describes the concept and implementation of the HST scheduling tools and how they could be adapted to other domains.

1. Introduction

To obtain the maximum benefit from expensive space facilities it is important to schedule spacecraft operations in an “optimal” manner. Since truly optimal scheduling is usually intractable, in practice it is therefore necessary to determine the best schedule possible subject to resource and time constraints on the computational effort that can be invested.

The fundamental requirements of optimal spacecraft scheduling are similar in many ways to those of other scheduling problems, e.g. those encountered in commercial and industrial domains. These problems have been found to be notoriously difficult to solve in practical settings. In this paper we discuss the source of some of these difficulties and how the use of advanced software technology (“artificial intelligence”, or AI) can be applied to help overcome them. We describe the progress made at Space Telescope Science Institute (STScI) in developing AI scheduling tools for Hubble Space Telescope (HST), and conclude with a discussion of how these tools can be adapted for other spacecraft scheduling problems.

2. Approaches to Scheduling

Computer techniques for optimal scheduling have been investigated for many years for a number of applications (see, e.g., [1] for a comprehensive review and bibliography). Much of this classical work has focused on versions of the idealized “job-shop” scheduling problem. This problem and related ones are NP-complete, meaning essentially that there are no efficient algorithms for finding optimal solutions (see, e.g., [2]).

The basic problem with these classical results is that they require key features of the problem to be abstracted away, so that even an “exact” solution to the abstracted problem is often of little relevance to the original “real” problem. Approximate solutions to the abstracted problem suffer from the same limitations. It is clear that classical approaches can be useful for problems which are sufficiently simple: in practice this often means that schedule optimization is driven by a *single* overriding criterion. For the problem of scheduling complex modern space facilities, however, this is not the case: more powerful techniques are required.

In recent years a variety of new software methodologies have been developed under the general term of "artificial intelligence" (AI). This refers to a collection of software development techniques and tools that have evolved in the course of computer science research as effective ways to represent and solve certain kinds of problems. These techniques have moved from the laboratory into widespread use in applications as their effectiveness has been demonstrated.

Several artificial intelligence research efforts have considered scheduling as a domain where AI techniques can be fruitfully applied. Of particular interest is the factory scheduling work of Fox, Smith, and co-workers (e.g.[3,4]) who have developed a rich constraint representation and versatile reasoning process for attacking realistic factory scheduling problems. While factory scheduling shares a number of common features with spacecraft scheduling, there are some obvious and important differences.

At Space Telescope Science Institute we initiated a project (SPIKE) in early 1987 for the purpose of developing AI scheduling tools [5,6,7] for Hubble Space Telescope (HST). HST scheduling is an extremely demanding task, requiring the scheduling of some tens of thousands of observations per year subject to a large number of proposer-specified and operational constraints [8]. Our overall approach to HST scheduling was inspired by the work of Smith and co-workers on the factory scheduling problem but has drawn on a number of other lines of research as well: as part of the SPIKE project we have developed a new framework for representing and reasoning with scheduling constraints [9] (based on discrete uncertainty reasoning for rule-based expert systems) and new techniques for searching the space of possible schedules [10] (based on recent developments in artificial neural networks).

There are four notable features of spacecraft scheduling that make it a difficult problem: interacting constraints, uncertainty, optimization criteria, and search. Realistic scheduling problems will typically involve a large number of different types of constraints, both strict and preference. Trading-off and balancing conflicting constraints adds greatly to the complexity of scheduling. Uncertainty can enter in a variety of ways, ranging from chaotic (completely unpredictable) scheduling factors to a smooth degradation of confidence in an extrapolated model. Optimization criteria are often complex and situation-dependent: there is usually no single criterion that can be used to indicate schedule optimality. All of these considerations must be incorporated into a search process which grows exponentially with the size of the problem.

3. AI Strategies for Optimal Scheduling

In this section we survey some of the AI techniques that can help deal with the problems described above. These techniques have been implemented in the HST SPIKE scheduling tools.

Separate constraint reasoning from strategic search

This is a statement about the overall architecture of the scheduler. The intent is to separate those aspects of the system that reason about *constraints* from those that reason about (partial or complete) *schedules*. The reason for this separation is that these reasoning processes take place on very different levels. Constraint reasoning is low-level and determines feasible and preferred scheduling times among which choices can be made; strategic reasoning evaluates one or more schedules and actually makes the choices. There may be more than one source of strategic knowledge available to work on one scheduling problem: all would, however, make common use of the results of constraint reasoning.

Use uncertainty reasoning methods for reasoning about constraints

There has accumulated a large body of theoretical and practical results on reasoning with uncertainty in the context of discrete rule-based expert systems. Based on this work we have developed a continuum version of uncertainty reasoning that can efficiently represent a wide variety of scheduling constraints [9]. Our framework is well-suited to the weighing of evidence for and against different scheduling hypotheses, thus providing essential inputs to trade-off decisions.

Provide multiple control mechanisms for strategic scheduling and search:

Based on the constraint-reasoning layer it is possible to implement a variety of strategic search mechanisms, any of which may be invoked depending on the nature and state of the problem. To date we have implemented three such mechanisms in the SPIKE scheduling tools:

- procedural search: this includes standard search techniques such as best-first or most-constrained-first algorithms. These tend to be computationally expensive and often encounter deadends which result in grossly sub-optimal schedules.
- rule-based heuristic search: this mechanism includes *search rules* to examine the state of a collection of partial schedules to identify the most “promising”, and *commitment rules* to decide how to extend the schedule by making some scheduling decision [7]. The rules communicate with the constraint-reasoning layer through “frames” or “schema” that hold summary information about the partial schedules. This general approach is well suited to the representation of quite complex scheduling heuristics.
- neural networks: a very different approach makes use of an “artificial neural network” [11] to represent a set of discrete scheduling choices. These networks are conceptually composed of a large number of simple processing elements operating in parallel whose computational power comes from their massive interconnection.

Formulate and attack the problem hierarchically:

A common and important problem-solving strategy is to formulate and solve a simpler higher-level problem, then attack the resulting lower-level subproblems by constraining them with the higher-level solution. In the scheduling domain there are two obvious ways to accomplish this: by scheduling *groups* of related activities at once, and by limiting the *time granularity* of the schedule. Both of these are exploited by SPIKE.

Provide explicit user visibility and control:

The approach we have taken in SPIKE is that automated scheduling is fundamentally a support tool for the people who are responsible for making scheduling decisions. In this approach one of the most important characteristics of the scheduler is how it interacts with the user. The user must have *visibility* into all aspects of the scheduling problem and the evolving schedule. The user must also have *control*, i.e. the ability to override any decisions made by the scheduler, and the ability to create and evaluate alternative schedule fragments.

4. Adapting HST Scheduling Tools to Other Domains

As an experiment to test the adaptability of SPIKE to other scheduling problems we have applied SPIKE to the problem of scheduling European IUE programs. Figure 1 illustrates the workstation screen showing one of the end results of this experiment. The large window shows a one-year time period and a selection of IUE programs (or program fragments in the case of repeated sequential observations). Plotted against time is the “suitability” of scheduling a selection of programs: the height of the suitability function indicates the degree of preference for scheduling a program at a given time, derived from the visibility constraints on all of the targets in the program. The solid bars show the weeks to which programs were assigned by the SPIKE software. The bottom plot in the same window shows the amount of observing time allocated to each week (solid line) compared to the maximum amount available (dashed line). This problem required the assignment of 171 programs or program fragments to 52 weeks. Extending SPIKE to handle this problem required less than one day of effort.

5. Conclusions

It is clear that software technology and approaches to scheduling have reached a sufficient level of development that intelligent spacecraft scheduling is a realistic goal. The use of artificial intelligence techniques makes it possible to develop and adapt software, such as the HST SPIKE scheduling tools, for a variety of telescope scheduling problems (see [12] for a discussion of the experimental use of SPIKE on ground-based telescope scheduling). The advantages of using these techniques are primarily a rapid software development cycle, a concise but expressive representation of scheduling data, flexibility in the definition and modification of scheduling constraints, powerful facilities for expressing search strategies, and the ability to incorporate a graphics-oriented user interface to help the user understand and modify the schedule.

Fig. 1: Workstation screen showing some of the IUE programs as scheduled by the HST SPIKE scheduling tools.

Acknowledgements: The author is grateful to G. Miller, J. Sponsler, S. Vick, K. Lindenmayer, and R. Jackson (STScI), D. Rosenthal (NASA Ames Research Center) and H.-M. Adorf (ST-ECF) for many useful discussions, and for the hospitality and support of the Space Telescope European Coordinating Facility and the European Southern Observatory (Garching) where some of this work was conducted.

Space Telescope Science Institute is operated by the Association of Universities for Research in Astronomy for the National Aeronautics and Space Administration.

References:

[1] King, J.R., and Spachis, A.S. 1980: "Scheduling: Bibliography and Review," *Int. Journal of Physical Distribution and Materials Management* **10**, p. 105.

[2] Garey, M., and Johnson, D. 1979: *Computers and Intractability*, (W.H. Freeman & Co.: San Francisco).

[3] Fox, M, and Smith, S. 1984: "ISIS: A Knowledge-Based System for Factory Scheduling," *Expert Systems* **1**, p. 25.

[4] Smith, S., Fox, M., and Ow, P. 1986: "Constructing and Maintaining Detailed Construction Plans," *AI Magazine*, Fall 1986, p. 45

[5] Miller, G., Rosenthal, D., Cohen, W., and Johnston, M. 1987: "Expert System Tools for Hubble Space Telescope Observation Scheduling," in *Proc. 1987 Goddard Conference on Space Applications of Artificial Intelligence*; reprinted in *Telematics and Infomatics* **4**, p. 301 (1987).

[6] Johnston, M., 1988: "Automated Telescope Scheduling," in *Proc. Conf. on Coordination of Observational Projects*, Strasbourg, Nov. 1987, (Cambridge Univ. Press: Cambridge), p. 219.

[7] Miller, G., Johnston, M., Vick, S., Sponsler, J., and Lindenmayer, K. 1988: "Knowledge Based Tools for Hubble Space Telescope Planning and Scheduling: Constraints and Strategies", in *Proc. 1988 Goddard Conference on Space Applications of Artificial Intelligence*.

[8] "HST Planning Constraints", 1987, Space Telescope Science Institute, SPIKE Report 87-1.

[9] Johnston, M. 1988: "Reasoning with Scheduling Constraints and Preferences," in preparation.

[10] Johnston, M., and Adorf, H.-M. 1988: "Scheduling with Neural Networks", in preparation.

[11] Hopfield, J., and Tank, D. 1985: "Neural Computation of Decisions in Optimization Problems," *Biological Cybernetics* **52**, p. 141.

[12] Johnston, M. 1988: "Automated Observation Scheduling for the VLT", in *Proc. ESO Conference on Very Large Telescopes and their Instrumentation*, Garching, March 1988.