5/0 N93-18669
137236
p. 5

Iterative Refinement Scheduling*

Eric Biefeld
Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive
Pasadena, CA 91109-8099
U. S. A.

Abstract

We present a heuristics-based approach to deep space mission scheduling which is modeled on the approach used by expert human schedulers in producing schedules for planetary encounters. New chronological evaluation techniques are used to focus the search by using information gained during the scheduling process to locate, classify, and resolve regions of conflict. Our approach is based on the assumption that during the construction of a schedule there exist several disjunct temporal regions where the demand for one resource type or a single temporal constraint dominates (bottleneck regions). If the scheduler can identify these regions and classify them based on their dominant constraint, then the scheduler can select the scheduling heuristic.

1 Introduction

Scheduling science experiments for such projects as Viking, Voyager, and Spacelab consumes a large amount of time and manpower. Whenever the Voyager spacecraft encounters a planet, the science experiments must be preplanned and ready to execute. This is a difficult scheduling problem due to the number and complexity of the experiments and the extremely limited resources of a spacecraft.

Since very few opportunities for space science exist, the major goal of mission scheduling is to maximize the number of science experiments that can be performed using the limited resources of the spacecraft. The total amount of requested experiments can be several times the amount that the project can accomplish.

Not only are schedules oversubscribed, they are also dynamic. Although the Voyager spacecraft was built and launched years ago, the flight rules governing the use of the spacecraft have changed. As the scientists learn more about their objectives, the experiment requests are updated. Thus, the mission schedule is a dynamic entity.

The Jet Propulsion Laboratory has performed mission scheduling for many years with a variety of deep space flight projects. The effort in scheduling an entire project such as Voyager can be measured in mancenturies. Because of this huge cost, JPL has been researching advanced software scheduling systems for several years (e.g. Deviser, Plan-It, Switch, Ralph, OMP).

Our current research, the Operations Mission Planner (OMP), is centered on minimally disruptive (non-nervous) replanning and the use of heuristics to limit the scheduler's search space. This paper addresses some of the problems pertinent to mission scheduling. It then defines iterative refinement, one of the basic design goals of our current research. This work has been greatly influenced by discussions with and the observations of the expert mission schedulers for the Viking, Voyager, and Spacelab projects.

2 Definitions

2.1 Resource/State

A resource/state (here after shorten to resource) tracks how a variable describing a state of the system changes through time and the steps which presently reserve this resource. An example is a pooled resource which tracks how many pieces of equipment out of a limited pool is being used at any moment in time. Another example is the direction of an antenna which is a continuous-state resource.

There are five fundamental types of resources: capacity, consumable renewable, continuous-state, and discrete-state [Starbird, 1987]. A capacity resource is basically a pooled resource but can have non-integer value and may have a time varying initial capacity.. Steps allocate a amount of the resource for their duration and then free up the resource for other activities. A consumable resource is one for which there is a limited supply, and once it is used by a step, it is no longer available (e.g. spacecraft fuel). A renewable is a generalization of a consumable, where the resource can be replenished (e.g. storage tape; it is used up during recording, and "replenished" during playback). A state resource represents a resource whose state (configuration, position, etc.) must be a certain value in order to support an activity. A continuous- state resource is one in which the state of the resource can best be described by a continuous vari-

*This research was done at the Jet Propulsion Laboratory, California Institute of Technology, and was sponsored through an agreement with the National Aeronautics and Space Administration.

able (e.g. the direction that an antenna is pointing). A discrete-state resources, on the other hand, are represented by discrete values (e.g. on/off, low-gain/medium-gain/high-gain).

Most domain resources can either be directly mapped into these resource types or be modeled by combining these type of fundamental resources to form a special meta-resource. A ground based Deep Space Network Antenna could be modeled as a meta-resource which combines two continuous- states and a renewable resource. The two continous-states would model the azimuth and declination while the renewable would model the number of times the antenna cables are wrapped around the antenna pedestal.

While the four fundamental types of resources can be used to model most of the resources we have encountered there exist a domain specific resources which could not be easily modeled. An example is the Voyager tape recorder which is a four track tack wire tape recorder. To schedule the tape recorder the schedulers build tape maps of what data is at what physical location on which track. This information is used to determine the order in which data can be removed and how long it takes to position the tape head to the beginning of a particular data track.

Associated with each type of resource is its definition of conflict. A conflict for a capacity resource occurs if the system reserves more then the limit of the pool at any moment in time. The resource is in conflict at the temporal interval for which a oversubscription occurs. A discrete-state is in conflict if either a step "reserves" a state that is not compatible with the state of the resource during the duration of the step or if the resource changes states without having an appropriate state changing step occurring.

2.2 Step/Activity

A step is a temporal interval which "reserve" resources where the meaning of reserve depends on the type of resource. While the resources model the state of the system over time steps model changes or constraints on the system. Along with resource reservations a step can contain constraints that either directly limit the range of choices possible in scheduling a step or links a step to other steps. The most common type of linking constraints are temporal predecessor and successor relations.

An activity is a set of steps and a set of constraints that link the steps together. The temporal constraints are the "glue" that bind the steps into a logical unit. The most common type of temporal linkage is the predecessor and successor relations. Along with the steps and constraints between the steps, an activity includes constraints that act on upon all the steps within an activity. This includes any global temporal windows and other global scheduling preferences like a priority for the request.

The users views an activity as the "primitive" action that must be scheduled to satisfy a user scheduling request. When a user issues a "request" the system finds the one or more activities that satisfy the request. The scheduler heuristically selects one of the activities and schedules the entire activity as a logical unit. Unlike resources the scheduler does not violate the constraints within an activity.

Since the activities interact only through the resource timelines, in some sense the activities are independent. It is possible to modify a previously schedule activity without backtracking or updating any other scheduled activity. Modifying a previously scheduled activity may cause some resource conflicts, but at certain stages of the scheduling process that is acceptable. The scheduler has the ability to note the conflicts for resolution at latter stages in the processing.

3 Focused Iterative Refinement

3.1 Expert Iterative Refinement

Iterative refinement is a technique used by expert spacecraft schedulers. The expert user first lays out the highly constrained activities over which he has little or no control. This forms a background against which the rest of the scheduling is done. The expert user then places the activities which impact large portions of the schedule. These may, for example, be a series of activities that have to be performed at exactly one-hour intervals over a large portion of the schedule. Any changes to this type of activity would cause changes to most of the schedule. If the scheduler gets stuck trying to place such an activity, he may elect to move it, but only as a last resort. Next, the expert user positions the high-priority activities, minimizing the number of conflicts. Finally, to complete the initial loading process, the expert user places the remaining activities on the schedule. If, at this point, some of the lower-priority activities do not fit easily, the expert user may simply ignore them.

After the loading process is done, the schedule is 80sense that most activities are in their final position on the schedule), although some resource contentions may still exist. The expert user has only spent about 20user will spend the remaining time trying to fit a few more activities into the schedule and trying to resolve resource contentions.

Up to this point in the scheduling process the scheduler has been task oriented [Smith and Ow, 1985]. Now the scheduler becomes resource oriented. The expert user focuses on the activities which are causing resource contentions on a particular resource and in a particular time region. After this area is fixed the expert user moves to another. Using this type of planning, the expert user iterates over and over again on the schedule, each time refining it a little more. After each pass through the schedule, the scheduler is willing to do a deeper search on any single activity because the total number of activities needing to be searched will decrease.

By focusing on just one area at a time the expert user may fix a portion of the schedule just to cause conflicts when the next portion of the schedule is processed. After several iterations, a small set of activities will circulate through the problem areas of the schedule. In this stage of scheduling, the expert user once again becomes task oriented. The expert user focuses on this small set of hard-to-place activities and performs the deepest search. The expert user addresses any chain reactions resulting from moving a specific activity. In Voyager scheduling this reasoning recurses about three levels down. In SpaceLab science scheduling the depth cut off is about four levels down. It is important to realize, however, that at this point the expert user has a small list of activities to try. The scheduler also restricts the impacted activities to those that seem flexible.

In the final stage of processing, the expert user looks for under-utilized areas of the schedule. The expert user checks the list of unscheduled activities looking for an activity that could use these resources. This unscheduled activity will, most likely, not fit directly into the schedule without causing some conflicts. Otherwise, the activity would have been scheduled earlier in the process. The scheduler tries to adjust some of the activities in the under-utilized areas in order to make room for the unscheduled activity. This may involve a series of shifts, but since both the activity and the under-utilized areas have been identified, it is a tightly focused search.

The schedule is then evaluated by the mission scientists for its total science return. The scientists negotiate with one another and with the scheduling team about which activities to include in the final sequence. The results of the negotiations must be reflected in the schedule. Therefore, the evaluation process following the generation of the initial schedule often results in requests to change the schedule, and hence the requirement for the replanning capability discussed earlier.

3.1.1 Phases of Iterative Refinement

Iterative planning consists a series of techniques. Each technique is responsible for a different aspect of the overall planning process. The first of these techniques roughs out the plan and identifies areas of high resource-conflict. The later techniques use the knowledge of the resource conflicts to refine the plan and solve many of the schedule problems. The final techniques try to solve the last of the conflicts and "optimize" the plan.

The OMP Load Phase is responsible for drafting an initial schedule. During this phase, the scheduler focuses on the requested activities, fitting them into the schedule with minimal concern for conflicts and levels of oversubscription.

During the Resource Centered Phase, OMP becomes resource oriented [Smith and Ow, 1985]. The scheduler focuses upon a resource region which contains conflicts and uses quick and simple techniques to fix these regions before processing another resource. It is during this phase that the bulk of the schedule is roughed out.

By focusing on just one resource region at a time the scheduler may fix one portion of the schedule but create additional conflicts in other regions. The scheduler discovers the bottlenecks by tracking these interactions between the separate regions. Once a bottleneck has been identified, it is classified and OMP attempts to resolve that bottleneck using techniques specialized for the type of bottleneck.

Once the conflict regions of the schedule have been resolved (which, since this is an oversubscribed domain will involve deleting some activities from the schedule), OMP takes another look at the high priority activities which have been deleted from the schedule and tries to fit them in. At this point, OMP will perform its deepest search in an effort to schedule just one more activity (extremely important in a domain such as deep space mission scheduling where opportunities to perform interplanetary experiments are rare). This phase is called the Optimization , although it doesn't produce a truly optimal schedule as would be defined in an operations research sense. Rather, it refers to fitting in additional activities after a conflict-free schedule has been produced. According to Spacelab scheduling experts, an optimal schedule is one where no one can suggest an improvement [Japp, 1986].

By specializing the planning techniques, each technique can be made more efficient. For example, the first techniques will use shallow searches over a broad spectrum of activities. Later techniques will use deeper searches but the search will only be applied to a limited number of activities. They will use knowledge about the particular schedule (i.e. the current resource conflicts, which activities have changed most often in the scheduling process) to constrain the search space. The techniques will employ either a shallow and broad search or a deep and narrow search. If a planner must perform a broad and deep search, it will not be able compute the schedule in any reasonable time.

3.1.2 Self-Reflective Iterative Refinement

The basic concept of self-reflective search is focusing the search by using knowledge gained from monitoring the search process. The OMP architecture, operating as outlined in the previous section, provides the mechanisms for supporting self-reflective search: the chronologies gather the raw information, the assessment heuristics analyze the information and feed the results to the control heuristics which focus the dispatch heuristics.

During the scheduling process, OMP keeps a chronology [Biefeld and Cooper, 1989] of the effort expended to resolve resource conflicts. In OMP, the chronologies are composed of a set of course grain resource timelines which record the scheduling effort level associated with a given region of the schedule, one measure of which is the number of times the scheduler attempts to resolve conflicts in that region.

During the resource centered phases, OMP focuses on a temporal interval within a given resource that is in conflict. Simple heuristics (which either change the resource used by an activity or temporally shift an activity out of the focus region [Biefeld and Cooper, 1991]) are used to reduce the level conflict in the focus region. The chronologies keep track of the effect of these actions within the region and on other regions which are changed as a result of the scheduling actions.

The system first attempts to find a set of resource assignments which reduces the total amount of conflict in the entire schedule. If the system can not lower the total conflict then it will increase the effort level for the focus region. The system retries the search, again attempting to reduce the conflict level in the region of focus, however this time it can increase the conflict level in other temporal regions for which the effort level is less than the focus region's effort level.

The above process will eventually cause OMP to cycle through the same regions. When the effort level for these regions exceeds the preset threshold, OMP exits the resource centered phase and begins the bottleneck centered phase. The assessment heuristics search through the chronologies and find the regions that have recently been raised to a high effort level. These regions are then collected into a bottleneck. The assessment heuristics then classify the bottleneck depending on its temporal size and its degree of oversubscription.

The current assessments heuristics in OMP distinguish bottlenecks by: 1) the amount of subscription compared to the bottleneck capacity; 2) the temporal extent of the bottleneck regions; and 3) the number of resources the bottleneck spans. Using these ratings the assessment heuristics classify the bottlenecks as either: 1) largely oversubscribed; 2) close to capacity but large in extent; or 3) close to theoretical capacity and small in extent.

If a bottleneck is largely oversubscribed then OMP's control heuristics will delete the low priority activities from the bottleneck region until the demand is only slightly larger then the capacity of the bottleneck. If a bottleneck is close to capacity but large in extent the control heuristics will split the bottleneck into several smaller regions. The first step is to distribute the tasking uniformly across the bottleneck and to reduce the demand slightly by shrinking the duration of the activities. The control heuristics will then focus on the smaller regions and use dispatch heuristics that emphasize local modifications over the global modifications used in the Resource Centered Phase. During this processing the assessment heuristics closely monitor the chronologies to identify small bottleneck regions. OMP processes each of the small bottlenecks as it locates them.

When processing a small bottleneck OMP uses it's most complicated heuristics. They use localized modifications to position one more activity onto the schedule. If the region in conflict is temporally small, the heuristics will either try to clip some activity whose start or end time is near the conflict, or the heuristics will split some activity into two separate activities with a gap equal to the conflict duration. If the conflict region is slightly larger, the heuristics clip and form gaps in a series of activities and align these gaps in such a manner as to reduce the conflict over the focus region.

Some heuristics, such as those for antenna handoff, are domain specific. A antenna handoff is when an activity splits its requirement for an antenna between two or more antenna resources. In the OMP demonstration domain, an activity may use one antenna for the first part and a second antenna for the second part but there must be a period of overlap during which it is using both antennas. In the OMP demonstration domain, if a bottleneck either spans two antennas or the temporal regions on the two antennas are near but do not completely overlay, then a antenna handoff may be practical. The dispatch heuristics attempt to split the activity into two activities and assign the antennas and temporal overlap to reduce resource contention.

This is an example of not only domain specific ways of expanding an activity but also where domain specific heuristics are needed to suggest when and how to try a particular activity expansion. Since the durations of the handoff overlap and the duration that an activity must spend on any single antenna is relatively small compared to the entire duration of an activity, the total number of ways an activity can be sliced up using antenna handoffs is quite large and in most cases not very useful. By identifying the bottleneck regions and then using domain specific heuristics to find particular patterns in the bottleneck regions the search process can be restricted, while still finding most cases were there special configuration tricks are useful.

4 Summary

This iterative planning approach to scheduling arose from attempts to heuristically control the search space of mission scheduling. The source of the heuristics were the human schedulers of Voyager, Viking, and SpaceLab who provided information on the stages of the scheduling process. Earlier stages are concerned with "roughing out" the schedule, placing most of the tasks, and identifying the trouble areas. Later stages then use scheduling heuristics to refine the existing schedule.

Most of these heuristics assume that the scheduler knows which resources are the bottlenecks and which tasks are causing the most difficulty for the scheduler. The best way to identify these critical resources and tasks is from the schedule produced by the earlier stages. In order to know what to try next one must already know what the schedule will be like.

Iterative planning assumes that the information gained by earlier techniques can be used by the later techniques to constrain the search space. Iterative planning also assumes that the schedule will not be changed dramatically by the later techniques. These assumptions seem to hold for the mission scheduling domain, which is extremely under- constrained. There exist many possible schedules for a single set of requested tasks. Two different human schedulers will produce two very different but equally acceptable schedules, given the same set of requested tasks. If, however, one human scheduler must modify another person's schedule, the basic structure of the schedule will not be modified. Therefore, expert schedulers normally perform non-nervous replanning.

References

[Biefeld and Cooper, 1989] Eric Biefeld and Lynne Cooper. Comparison of Mission and Job Shop Scheduling. *Proceedings of the Third International Conference on Expert Systems and the Leading Edge in Production Planning and Control*, pages 483–494, Hilton Head Island, South Carolina, May 1989.

[Biefeld and Cooper, 1989] Eric Biefeld and Lynne Cooper. Scheduling with Chronology-Directed Search. *Proceedings of the AIAA Computers in Aerospace VII*

Conference, pages 1078–1087, Monterey, California, October 1989.

[Biefeld and Cooper, 1991] Eric Biefeld and Lynne Cooper. Bottleneck Identification Using Process Chronologies. *Proceedings of the Twelfth International Joint Conference On Artificial Intelligence*, pages 218–224, Sydney, Australia, August 1991.

[Dean, 1986] Thomas Dean,. Intractability and Time Dependent Planning. *Proceedings of Workshop on Planning and Reasoning About Action*, pages 143–164, June 1986.

[Dean and McDermontt, 1987] Thomas Dean and Drew McDermott. Temporal Data Base Management. *Artificial Intelligence Journal*, 32(1):1–55, 1987.

[Dean *et al.*, 1987] Thomas Dean, R. James Firby, and David Miller. Hierarchical Planning involving Deadlines, Travel Time, and Resources. *Computational Intelligence*, 4(4), November 1988.

[Eskey and Zweben, 1990] Megen Eskey and Monte Zweben. Learning Search Control for Constraint-Based Scheduling. *Proceedings of the Eighth National Conference on Artificial intelligence*, page 908–915, Boston, Massachusette, July 1990.

[Fox and Smith, 1984] Mark Fox and Stephen Smith. ISIS: A Knowledge-Based System for Factory Scheduling. *Expert Systems*, 1(1):25–49, July 1984.

[Japp, 1986] John Japp. Mission Timeline Analysis Demonstration. NASA/Marshall Space Flight Center, June 3, 1986.

[Johnson and Roadifer, 1986] Craig Johnson and James Roadifer. A Look-Ahead Strategy for Heuristic Activity Scheduling. *Joint Conference of the Operations Research Society of America and the Institute of Management Sciences*, October 1986.

[Le Pape and Smith, 1987] Claude Le Pape and Stephen Smith. Management of Temporal Constraints for Factory Scheduling. *Proceedings of the Working Conference on Temporal Aspects in Information Systems*, May 1987.

[McLaughlin and Wolff, 1989] W. I. McLaughlin and D. M. Wolff. Automating the Uplink Process for Planetary Missions. *AIAA 27th Aerospace Science Meeting*, Reno, Nevada, January 1989.

[Miller, 1988] David Miller. A Task and Resource Scheduling System for Automated Planning. *Annals of Operations Research*, 12(1–4):69–198, February 1988.

[Minifie and Davis, 1986] J. Roberta Minifie and Robert Davis. Survey of MRP Nervousness Issues. *Production and Inventory Management*, 27(3):111–121,1986.

[Muscettola and Smith, 1987] Nicola Muscettola and Stephen Smith. A Probabilistic Framework for Resource-Constrained Multi- Agent Planning. *Proceedings of the Tenth International Joint Conference On Artificial Intelligence*, page 1063–1066, Milan, August 1987.

[Ow and Smith, 1988] Peng Si Ow and Stephen Smith. Viewing Scheduling as an Opportunistic Problem-Solving Process. *Annals of Operations Research*,12(1-4):85–108, February 1988.

[Smith *et al.*, 1986] Stephen Smith, Mark Fox, and Peng Si Ow. Constructing and Maintaining Detailed Production Plans: Investigations into the Development of Knowledge-Based Factory Scheduling Systems. *AI Magazine*, 7(4):45–61, 1986.

[Smith and Ow, 1985] Stephen Smith and Peng Si Ow. The Use of Multiple Problem Decompositions in Time Constrained Planning Tasks. *Proceedings of the Ninth International Joint Conference On Artificial Intelligence*, pages 10130–1015, Los Angeles, California, August 1985.

[Starbird, 1987] Tom Starbird. Space Flight Operations Center Sequence Subsystem (SEQ) Functional Requirements Document for Planning. JPL Internal Document D–4697; NASA, Jet Propulsion Laboratory, California Institute of Technology, Pasadena California, August 1987.

[Vere, 1983] Steven Vere. Planning in Time: Windows and Durations for Activities and Goals. *IEEE Transactions on Machine Intelligence PAMI-5*, No. 3, pages 246–267, May 1983.

[Zweben *et al.*, 1990] Monte Zweben, Micheal Deal, and Robert Gargan. Anytime Rescheduling. *Proceedings of the DARPA Workshop on Innovative Approaches to Planning, Scheduling and Control*, pages 251–259, San Diego, California, November 1990.