An Integrated Planning and Scheduling Prototype for Automated Mars Rover Command Generation

Rob Sherwood, Andrew Mishkin, Steve Chien, Tara Estlin,
Paul Backes, Brian Cooper, Gregg Rabideau, Barbara Engelhardt

Jet Propulsion Laboratory, California Institute of Technology
4800 Oak Grove Dr., Pasadena, CA 91109
`firstname.lastname@jpl.nasa.gov`

Abstract. With the arrival of the Pathfinder spacecraft in 1997, NASA began a series of missions to explore the surface of Mars with robotic vehicles. The mission was a success in terms of delivering a rover to the surface, but illustrated the need for greater autonomy on future surface missions. The planning process for this mission was manual, and very time constrained since it depended upon data from the current day to plan the next day. This labor-intensive process was not sustainable on a daily basis for even the simple Sojourner rover for the two-month mission. Future rovers will travel longer distances, visit multiple sites each day, contain several instruments, and have mission duration of a year or more. Manual planning with so many operational constraints and goals will be unmanageable. This paper discusses a proof-of-concept prototype for ground-based automatic generation of rover command sequences from high-level goals using AI-based planning software.

1 Demonstration

We will demonstrate a ground based automated planning prototype for a multi-instrument Mars rover using the ASPEN planner (Chien, et al., 2000). With this software, new goals can be added to the existing plan, resulting in conflicts that will be solved using an iterative repair algorithm. The end result will be a valid sequence of commands for execution on a rover.

2 Introduction

Over the next 10 years, NASA will be sending a series of rovers to explore the surface of Mars. The rover planning process uses specialized tools for path planning and instrument planning, but the actual activity planning and scheduling is a manual process (Mishkin, et al., 1998). We are using AI planning/scheduling technology to automatically generate valid rover command sequences from goals specified by the specialized tools. This system encodes rover design knowledge and uses search and reasoning techniques to automatically generate low-level command sequences while respecting rover operability constraints, science and engineering preferences, environmental predictions, and also adhering to hard temporal constraints.

3 ASPEN Planning System

In ASPEN, the main algorithm for automated planning and scheduling is based on a technique called *iterative repair* (Rabideau, et al., 1999, Zweben et al., 1994). During iterative repair, the conflicts in the schedule are addressed one at a time until conflicts no longer exist, or a user-defined time limit has been exceeded. A conflict occurs when a resource requirement, parameter dependency or temporal constraint is not satisfied. Conflicts can be repaired by means of several predefined methods. The repair methods are: moving an activity, adding a new instance of an activity, deleting an activity, decomposing an activity into subgoals, abstracting an activity, making a resource reservation on an activity, canceling a reservation, connecting a temporal constraint, disconnecting a constraint, and changing a parameter value. The repair algorithm may use any of these methods in an attempt to resolve a conflict. How the algorithm performs is largely dependent on the type of conflict being resolved and the activities, states, and resources involved in the conflict.

4 Rover Motion Planning

ASPEN is able to reason about simple resource and state constraints. ASPEN also has the ability to use simple external functions to calculate parameters for resource usage. Many rover constraints are too complex to reason about in a generalized planning system, or use simple parameter functions to solve. For these, an external program must be used to reason about these constraints. ASPEN can interface with other domain-specific programs (or special purpose algorithms) using input files, library calls, a socket interface, or software interfaces. Motion planning is a good example of a complex rover constraint requiring a specialized tool.

JPL uses a tool called Rover Control Workstation (RCW) for the motion-planning problem (Cooper, 1998). RCW provides a unique interface consisting of a mosaic of stereo windows displaying the panorama of Mars using camera images from both a lander and a rover. The operations team uses the RCW to make decisions about where to safely send the rover and what to do when reaching the goal. RCW calculates the maximum safe tilt angles for the rover traverse goals input by the user. RCW also calculates the parameters for the rover motion commands. The RCW software outputs a set of goals that cannot be changed in ASPEN.

5 Mixed-Initiative Rover Planning

While the goal of this work is an integrated fully automated planning system for generating a rover sequence of commands, the human operator is required to be part of the planning process. There is not enough CPU capability onboard current flight rovers to run autonomy software such as path planning or generalized planning. The JPL developed Web Interface for Telescience (WITS) science-planning tool (Backes, et al., 1998) and the RCW motion-planning tool each require human interaction. These tools allow the user to select rover destinations and science targets in three dimensions using surface imagery. The WITS tool does not actually enforce an order of the goals, but instead relies on ASPEN to build the plan, schedule, and check the resource usage.

Combining these tools with ASPEN creates a “mixed-initiative” end-to-end planning system. The ASPEN operator starts with a set of goals from WITS and RCW, but can then modify the schedule within ASPEN by inserting new goals, changing existing activities, or deleting activities. The schedule is then generated using a forward dispatch algorithm followed by an iterative repair algorithm to fix any conflicts. The repair actions available for each activity are defined within the model for that activity. If the rover resources are over-constrained or under utilized, the user may decide to modify the schedule to optimize the rover resource usage, then re-run the iterative repair algorithm. Several iterations can be performed using ASPEN, WITS, or RCW to modify the goals. This capability allows the rover operations team to try several different scenarios before deciding on the best course of action. The result of this mixed-initiative optimization strategy is a plan with increased science opportunities. Because ASPEN is autonomously checking flight rules and resource constraints, the plan should also be safer than a manually generated plan.

We are also investigating how the user should be interacting with each of the tools involved in building a schedule. The science and engineering users are used to interacting with WITS and RCW, but not with ASPEN. Yet WITS and RCW do not show resource information and activity ordering. Currently the system requires the user to utilize the ASPEN GUI for resource and activity information. In the future, this information could be added directly to the WITS GUI.

6 Difficulties in Modeling Rover Constraints

There are several aspects of modeling the Mars rover domain has proven to be very difficult. The power system is a good example. The rovers planned for 2003 contain solar arrays and rechargeable batteries. During the daytime, the power for rover operations is produced using the solar arrays. If the total power drain from operating the rover exceeds the available power from the solar arrays, the batteries must be drawn upon. Because the battery drain is context dependent, the planner needs to understand all the influences and be able to repair conflicts using this knowledge. Additionally, computing the energy taken from a battery is a function of the battery parameters such as temperature, current, voltage, etc. Representing this in a planning model is very difficult.

To solve the power-modeling problem, we initially used a parameter dependency function to calculate the amount of solar power and battery power as a function of the activity duration, available solar array power, available battery power, and power required by the activity. This technique will only work if there are no overlapping power activities because the calculated solar array and battery usage are based on the amount available at the beginning of the activity. In the ASPEN representation, resource use is assumed to be constant over the duration of the activity. In the same manner, we can only request the existing value of a resource at the start of the activity and we must assume that the existing resource profile remains constant until the end of the activity. In the case of overlapping activities that consume power, the first of the two activities would calculate the required power based on the available power at the start time of the first activity. The power available would change during the activity due to the overlap of the second activity.

Another difficulty with modeling the depletable resources in planning systems is the usage profile. Some examples in the spacecraft and rover domains include the memory buffer resource, battery, and fuel. If an activity that uses the memory buffer resource has duration of several minutes, ASPEN will change the value of the resource timeline at the beginning of the activity. In this case, the entire amount of memory buffer resource used by the activity is unavailable for the entire activity. In the example, the memory resource is set to them maximum value at the start of the timeline. This is the equivalent of consuming an entire tank of gas in a car at the beginning of a trip rather than using the gas gradually over the course of the trip. Likely the actual resource usage is linear over the duration of the activity. For long activities, the depletable resource value near the beginning of the activity can be very inaccurate. One workaround for this problem is to split the activity up into several subactivities, each using an equal fraction of the resource. This solution has several problems. First, it increases scheduling complexity by adding multiple activities into the activity database. Second, it creates the problem of trying to determine how many subactivities is enough to accurately model the resource usage. Third, it’s non-intuitive for the user to see multiple subactivities that don’t represent actual events. The ideal method for modeling resource usage is to use a generalized timeline. Generalized timelines allow modelers to provide a set of functions to describe the depletable resource timeline and its constraints. The generic scheduler can then accurately reason about the described timelines. The example given contains a linear depletable timeline, but any other function could have been modeled as well.

7 Conclusions

Planning and reasoning about complex rover resources is a difficult task to automate. The rover planning process involves interfacing with other specialized planning tools to create a mixed-initiative end-to-end planning system.

Current approaches to rover-sequence generation and validation are largely manual, resulting in a labor and knowledge intensive process. This is an inefficient use of scarce science-investigator and key engineering-staff resources. Automation as targeted by this tool will automatically generate a constraint and flight rule checked, time ordered list of commands and provide resource analysis options to enable users to perform more informative and fast trade-off analyses.

8 Acknowledgement

This work was performed at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

References

P. Backes, K. Tso, and G. Tharp. "Mars Pathfinder mission Internet-based operations using WITS. Proceedings IEEE International Conference on Robotics and Automation," pages 284-291, Leuven, Belgium, May 1998.

B. Cooper, “Driving On The Surface Of Mars Using The Rover Control Workstation,” SpaceOps 98, Japan, 1998.

S. Chien, G. Rabideau, R. Knight, R. Sherwood, B. Engelhardt, D. Mutz, T. Estlin, B. Smith, F. Fisher, T. Barrett, G. Stebbins, D. Tran , "ASPEN - Automating Space Mission Operations using Automated Planning and Scheduling," SpaceOps, Toulouse, France, June 2000.

A. Mishkin, "Field Testing on Mars: Experience Operating the Pathfinder Microrover at Ares Vallis," presentation at Field Robotics: Theory and Practice workshop, May 16 1998, at the 1998 IEEE International Conference on Robotics and Automation, Leuven, Belgium.

G. Rabideau, R. Knight, S. Chien, A. Fukunaga, A. Govindjee, "Iterative Repair Planning for Spacecraft Operations in the ASPEN System," International Symposium on Artificial Intelligence Robotics and Automation in Space (ISAIRAS), Noordwijk, The Netherlands, June 1999.

Zweben, M., Daun, B., Davis, E., and Deale, M., “Scheduling and Rescheduling with Iterative Repair,” Intelligent Scheduling, Morgan Kaufmann, San Francisco, 1994, pp. 241-256.