Onboard Planning for Geological Investigations using a Rover Team

Tara Estlin, Daniel Gaines, Forest Fisher, and Rebecca Castaño
Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Dr.
Pasadena, CA 91109-8099
{*firstname.lastname*}@jpl.nasa.gov

Abstract

This paper describes an integrated system for coordinating multiple rover behavior with the overall goal of collecting planetary surface data. The MISUS system combines techniques from planning and scheduling with machine learning to perform autonomous scientific exploration with cooperating rovers. A distributed planning and scheduling approach is used to generate efficient, multi-rover coordination plans, monitor plan execution, and perform re-planning when necessary. A machine learning clustering component is used to deduce geological relationships among collected data and select new science activities. A key concept promoted by this system is the use of goal interdependency information to perform plan optimization and increase the value of collected science data. We discuss how we represent and reason about goal dependency and utility information in our planning system and explain how this information can change dynamically during system use. We show through experimental results that our approach significantly increases overall plan quality versus a standard approach that treats goal utilities independently.

1 Introduction

NASA recently demonstrated that mobile robotic craft are a viable and extremely useful option for exploring the surface of other planets. The Mars Exploration Rovers (MER) have already gathered valuable scientific data that will be used to answer many questions about the Martian terrain. Future missions are being planned to send additional robotic explorers to Mars as well as to the moon and outer planets (JPL, 2004). Rover teams will also be an important component to any manned mission to Mars, both in performing science activities and building and maintaining necessary structures. In order to enable certain types of activities and to significantly increase overall science return, many of these future missions will require larger sets of rovers. These rovers will need to behave in a coordinated fashion where each rover accomplishes a subset of the overall mission goals and shares acquired information with other rovers and mission personnel. Furthermore, a key aspect of these missions will be highly autonomous rovers that can efficiently work together and require only limited communication with scientists and engineers on Earth. These rovers will be able to make many decisions on their own as to what new science data should be collected and how to perform the data gathering process.

The Multi-rover Integrated Science Understanding System (MISUS) provides an approach for autonomously achieving planetary science goals using multiple robotic explorers (Estlin, et al., 1999). This system integrates techniques for machine learning and data analysis with those for planning and scheduling to enable autonomous multi-rover coordination. Steps performed by the system include analyzing science data, evaluating what new science observations to perform, and deciding how to successfully perform them. Requested science observations are handled by a distributed planning and scheduling system which is responsible for delegating goals to rovers, achieving as many high priority science observations as possible given resource and operation constraints, and sharing information between rovers on related goals. This system is also integrated with a simulation environment that can model different planetary terrains and the results of science data observations within them.

A key feature of MISUS is its ability to reason about interdependent science observations. Most planning systems allow only simple, static dependencies to be defined among goals where these dependencies remain constant between different problems. However, in many domains, including space and planetary exploration, goals can be related through detailed utility models that significantly change from problem to problem. For instance, in one problem a particular goal's utility may increase if other related goals can be achieved. In another problem, this utility increase may differ or actually decrease if the same combination of goals is achieved. We consider these types of goals to be *interdependent* and have implemented a methodology for representing and reasoning about them and their relevant utilities (Estlin and Gaines, 2002). We have also designed our distributed planning system to specifically handle this type of information when both formulating and executing plans.

The remainder of this paper is organized as follows. We begin by giving an overview of the full MISUS system. Next, we will further describe our distributed planning approach as well as our approach to handling interdependent science goals. We then present the results of a set of experiments designed to test the benefit our approach and how well it handles interdependent goal information. In the final sections, we discuss related work and present our conclusions.

2 MISUS Overview

The MISUS system is comprised of three major components:

- **Data Analysis:** A machine-learning system that creates a distribution model of the different rock types from the observed terrain. A clustering approach is used that employs an objective function for inferring geological relationships among data. This component also contains a prioritization algorithm that suggests new prioritized science observations to best increase the accuracy of its learned model.
- **Distributed Planning and Scheduling:** A distributed planning and scheduling system that produces rover-operation plans to achieve science goals. Planning is divided between a central planner, which creates a global plan for all rovers, and a distributed set of planners, which create detailed operation plans for individual rovers. Planning is continuous where plans are monitored during execution and re-planning is performed when necessary.
- **Environment Simulation:** A multiple rover simulator that models different geological environments and rover-science operations within them. The simulator manages science data, tracks rover operations within the terrain, and reflects readings by rover science instruments. Currently, two types of instrument data are supported: visual texture data, which can be produced from rover camera images, and spectral data, which can be produced using a boresighted spectrometer.

As shown in Figure 1, MISUS operates in a closed-loop fashion where the data analysis system can be seen as driving the exploration process based on its current model of the environment. Data gathered by each rover is used in a clustering algorithm to model the distribution of rocks according to their mineralogical composition and locations. Using this model, a prioritization algorithm generates new science goals based on their scientific value and ability to improve model accuracy. For example, if only limited data has

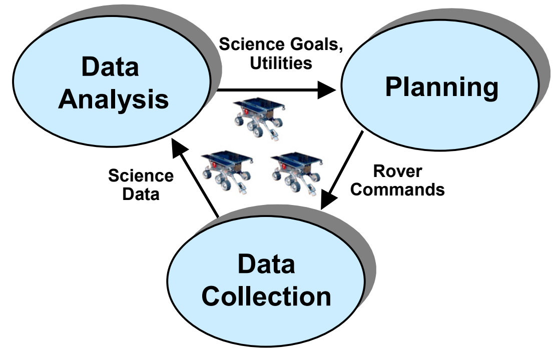

Figure 1: MISUS Closed-Loop Data Flow

been collected on a certain rock class, the algorithm may suggest new observations for that type of rock. New science goals are passed to the distributed planning system, which assigns goals to rovers in a way that minimizes required traverse distance and resources. A set of actions is produced for each rover that achieves the most valuable subset of goals given rover resource and operation constraints. During plan execution, the planning system continually monitors plan status. Re-planning can be used to repair or modify plans if unexpected events occur. Science goals may also be re-assigned to other rovers dynamically if the currently assigned rover can no longer achieve them.

Currently we are applying MISUS to a planetary science application, which was designed through collaboration with JPL geologists and represents an example of how multiple rovers could investigate new areas of Martian (or other planetary) terrain. The primary science objective given to MISUS is to evaluate the distribution of rocks over a particular area of terrain. A team of three rovers is used where each rover has a camera and spectrometer to collect data. Rockfields are generated in our environment simulator, which maintains information on rock types, sizes and locations. Science goals consist of taking panoramic or point (i.e., local) measurements with each instrument. Goals are also given utilities that reflect their overall scientific value. Each rover is assumed to have a standard set of onboard resources and sensors, such as a solar panel and battery for power, memory to hold science data, and antennas that allow communication with Earth and/or other rovers. Note that the overall MISUS architecture could be used for many different science objectives. What drives the science process is the underlying model the data analysis system is tasked to learn. Other models could also be applied such as searching for a particular type of mineral composition or determining what process formed an area of terrain (e.g., volcanic, fluvial).

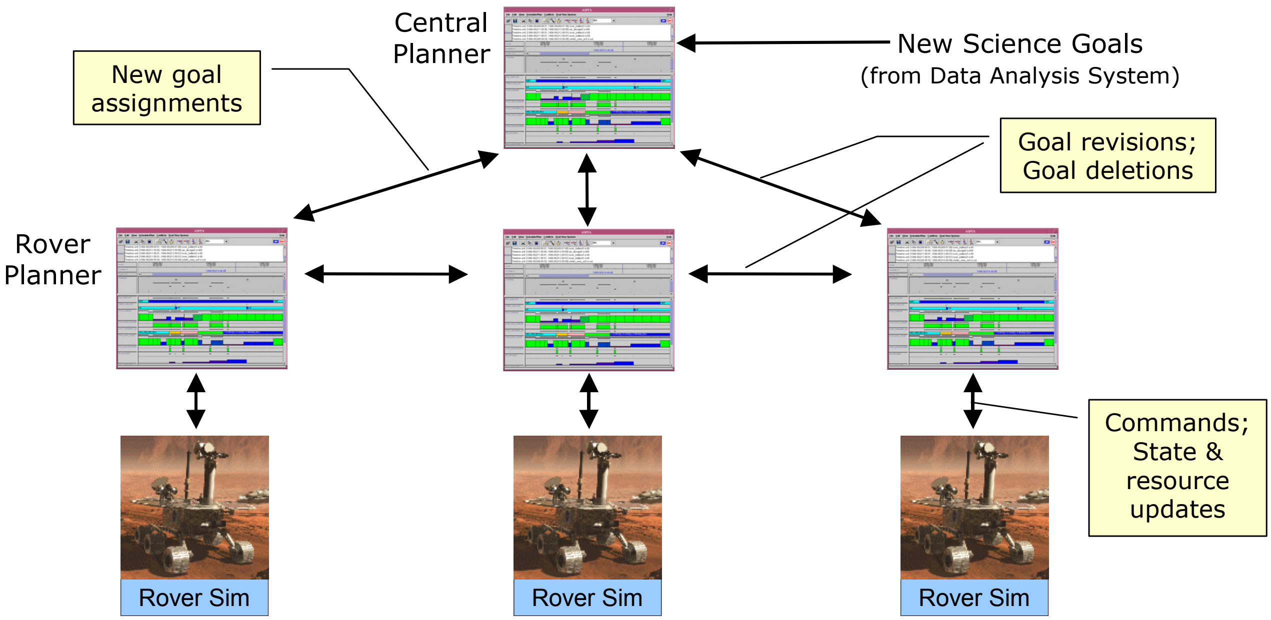

Figure 2: Distributed Planning System Architecture

3 Planning and Execution for Multiple Rovers

To produce and coordinate plans for a team of rovers, we have developed a distributed planning system that enables plans at different abstraction levels to be continually updated with current goal, resource and state information. It also enables science goals to be dynamically redistributed to the most appropriate rover based on current conditions. As shown in Figure 2, our system is comprised of a central planner, which coordinates plans among rovers, and an onboard planner for each rover, which creates and manages detailed operation plans for that rover. New science goals are given to the central planner, which can be located on either a lander or one of the rovers. This planner creates a global plan for the rover team and is responsible for distributing goals among rovers. The central planner has limited knowledge of rover resources and states, which it uses to divide goals in an attempt to minimize overall traverse distance. Each individual rover planner is responsible for creating its own detailed operations plan, which ensures no operation or resource constraints will be violated. This distributed framework was chosen due to its ability to encourage globally optimal plans while still operating under limited communication. MISUS was designed to handle rover teams where the amount of communication between team members can vary. In some applications rovers may all operate in a general area where communication is relatively inexpensive, e.g., several rovers working in close range to build a structure or habitat. In other applications rovers may be out of communication for varying or long periods of time, e.g., surveying a large terrain area that has hills or large rocky areas that can obstruct communication.

3.1 Continuous Planning

The CASPER continuous planner (Chien, et al., 2000) is used as the base system for both the central and individual rover planners. CASPER was developed to address dynamic planning and scheduling where plans can be continually modified based on changing state and goal information. Unlike batch planners, where each plan must be created from scratch, CASPER continually updates its plan based on new information. When an unexpected event occurs, CASPER can quickly modify the plan to handle the new event while still achieving its objectives. CASPER's main components include:

- An expressive modeling language to allow the user to easily represent different domains.
- A constraint management system for representing and maintaining domain operability and resource constraints.
- A set of search strategies and repair heuristics
- A temporal reasoning system for representing and maintaining temporal constraints.
- An optimization system that allows users to define objective functions and preferences.
- A graphical interface for visualizing plans as well as predicted effects on resources and states.
- A real-time system that monitors plan execution and modifies the current plan based on activity, state and resource updates.

CASPER takes as input a set of science and engineering goals and automatically generates an activity sequence that achieves the input goals. One of the primary search algorithms used to produce a valid sequence is iterative repair (Zweben, et al., 1994), which attacks plan conflicts individually. Conflicts occur when a plan constraint has been violated and can be temporal or involve a resource or state. Conflicts are resolved by performing one or more schedule modifications, such as moving, adding or deleting activities. An example of a conflict would be a rover that is in the incorrect location for a scheduled science observation. Resolving this conflict typically involves adding a new drive command to send the rover to the designated target location.

In MISUS, CASPER is used to provide planning and re-planning capabilities for the central and individual rover planners. For the central planner, CASPER creates an abstract plan that divides goals up among rovers and monitors goal execution status. To make goal assignments that best use rover resources, the central planner uses a set of Multiple Traveling Salesman Problem (MTSP) search heuristics, which encourage plans that minimize overall traverse distance. The central planner also monitors goal achievability during plan execution. If a goal cannot be achieved by a particular rover, the central planner may choose to dynamically reassign the goal to another rover or delete the goal if it deems it unachievable.

For the individual rover planners, CASPER creates a detailed execution plan using TSP heuristics and relevant constraints to order science targets, and then monitors that plan and its effect on rover states and resources. For example, it continually monitors information on states such as rover position, resources such as power, and execution status for plan activities. If the plan does not proceed as expected, CASPER can iteratively re-plan to accommodate any unexpected events. These events could simply be activities finishing early or problems that may cause plan conflicts such as an unexpected obstacle blocking the rover's path or a science activity taking more power than expected. Currently we use a rover hardware simulator that models operations of different JPL rovers. This simulator can be used to randomly cause unexpected or faulty behavior during plan execution.

For this application, planning goals correspond to prioritized science observations for taking images or spectrometer measurements, which will be explained in more detail in the next section. The final plan for each rover is a sequence of commands, which typically includes drive operations to different locations, specific instrument operations at those locations, and communication operations. Science and drive activities require a varying amount of power and time depending on parameters such as the distance being driven or the science operation being performed. Science activities also require different amounts of memory for storing gathered data. A number of different resource and state constraints are modeled. Each rover has a limited amount of available power and memory onboard. There is a limited time window (or horizon) each day within which activities can be scheduled. Each rover must also perform a communication activity each day within certain time constraints. Individual rover planners are aware of all of these constraints. The central planner is primarily aware of science operations and their related constraints. Rover planners also receive a number of state and resource updates from the underlying rover control system. These updates include current status on power and memory available, rover estimated position, and the success or failure of executed drive, science, and communication activities.

3.2 Distributed Communication

As shown in Figure 2, several pieces of information are communicated between the different planners. First, the central planner sends new goal assignments to the individual rovers. Second, the individual rover planners broadcast information on their goal execution status to other rover planners as well as the central planner. These status updates relay information such as whether a rover can no longer achieve a particular goal (and thus is releasing it back to the central planner) and what time the goal is scheduled to occur. As mentioned previously, if a rover has shed a goal, the central planner can attempt to reassign it. Each rover planner uses information about the goals that have been assigned to other rovers to evaluate the quality of its own current plan and chosen goal set. Our plan optimization approach uses this information and is explained in more detail in the next section.

To provide a communication mechanism between planners, we have adopted the Shared Activity Coordination (SHAC) framework (Clement and Barrett, 2003), which provides generic capabilities for continually coordinating multiple agents and for rapidly designing and implementing coordination protocols to govern the communication process. Information on goals is communicated between planners using SHAC's shared activity model, which captures the information that multiple agents must share, including control mechanisms for changing that information. For MISUS, SHAC enables goal parameter information, such as duration, start time, target position, and memory required, to be shared among planners. SHAC coordination protocols are also used to signal to the master when goals have been shed by a particular rover and thus can be re-assigned to a new rover.

Other communication constraints can also be represented in SHAC. Currently in MISUS, goal information is communicated between planners as soon as available. However, if communication was more restrictive, the system could easily be modified to only communicate information during certain time windows. The architecture is designed to allow planning and execution to proceed whether or not current data can always be broadcast.

4 Interdependent Planning Goals

A unique feature of our system is its ability to represent and reason about interdependent planning goals. A limitation of most planning techniques is that they define relationships between input goals in a simple, static manner, which cannot be easily adjusted for different problem situations. In many domains, goals can be related in complex and varying ways that are best represented through utility metrics. These metrics, however, cannot always be included as part of a standard domain definition, since they are often dependent on current data or state and can vary widely from problem to problem.

Many planning systems allow you to define utility information that represents an overall plan quality or score. For examples, goals may be assigned priorities that help a planner decide what goals to try to achieve first. Other general metrics may also be defined, such as minimizing makespan, avoiding missed deadline costs, or minimizing the usage of a particular resource (Williamson and Hanks, 1994; Joslin and Clements, 1999; Rabideau, et al., 2000). Most planning systems also allow you to define static dependencies between goals. For instance, two goals could be related in a domain model, through the decomposition of a parent goal, or through pre- or post-conditions. However, in all these approaches, goal relationships and utility metrics are pre-defined in the domain description or an objective function and typically remain constant between problem instances. Furthermore, it is difficult to define utility metrics that involve specific goal instances as opposed to general quality concepts that apply to a certain class of goals (e.g., increasing the number of orders filled). No current planning systems enable *dynamic dependencies* among goals, i.e., dependencies that significantly vary from problem to problem and thus must be defined as part of the problem specification instead of in the original domain description or model. When planning for rover missions, goals are often dictated by science data that has just been collected and/or what new science opportunities are available. Furthermore, there are many situations where the value of a science goal will be changed if other related science goals can be achieved. For instance, collecting images of a particular rock from different angles and distances often increases the value of all images taken of that rock since a better overall analysis of the rock can be performed.

The MISUS distributed planning system provides a method for handling interdependent planning goals while performing plan optimization. In this approach, interdependencies between goals can be formulated dynamically and provided to the planning system as part of the goal input. The central planner and all local rover planners can then reason about these dependencies and incorporate them into the objective function they use to rate plan quality and direct their search process. To implement our approach, we have extended the base optimization framework already available in CASPER (Rabideau, et al., 1999). We have also tested our approach on a series of problems based on the previously described scenario of rovers performing a rock distribution survey of the surrounding terrain area.

4.1 Interdependent Goal Representation and Objective Function

To represent a goal's value, we have extended a typical utility representation where goals can have individual rewards representing their importance, so that complex interdependencies and their relevant utilities can be represented and used by a planning system. Furthermore these interdependencies and utility values can change between problem specifications without requiring any changes to the planning domain model. In our representation a list of goals (g_1, g_2, …, g_n) and goal combinations (c_1, c_2, …, c_m) are provided to the planning system, where each goal combination c_i consists of a tuple of goals $<g_i, g_j, \ldots, g_k>$. For each goal and for each goal combination there is an associated weight indicating the value that will be added to the plan if the plan includes those goals. This representation allows us to express singleton goal values, that is a goal whose contribution to the plan does not change as other goals are added, and any n-ary goal relationship to indicate the value that combination of goals adds.

We currently use a simple objective function to calculate the plan quality with respect to its achieved goals. Let G be the set of goals that occur in the plan. The value of plan P is then give by Equation 1. This

$$O(P) = \sum_{g \in G} o(\langle g \rangle) + \sum_{\langle g1, g2, \ldots gn \rangle \in C} o(\langle g_1, g_2, \ldots g_n \rangle)$$

where $o(\langle g_1, g_2, \ldots g_n \rangle) = \begin{cases} w(\langle g_1, g_2, \ldots, g_n \rangle) & if\ \{g_1, g_2, \ldots, g_n\} \subseteq P \\ 0 & otherwise \end{cases}$

Equation 1: Objective function for calculating plan utility when using interdependent goals

function sums up the values of all goals that occur in the plan along with the weight for each goal combination, where all named goals appear in the plan.

4.2 Optimization Approach

To use the above objective function, we have also provided an improvement heuristic that can suggest what changes CASPER should make to the plan to increase the score. To create and optimize a plan we use a random hill-climbing search with restart. First, a plan is created that achieves any mandatory goals or activities that must be added to the plan. We then perform a series of optimization steps where each step consists of *i* iterations. At each iteration, if there are no conflicts in the plan, we use an improvement heuristic to suggest the next goal to add. If there are conflicts, we perform an iteration of repair. Whenever we have a

Goal	Reward
A: Long-Range Image of a Rock	Rock + 11
B: Close-Up Image of a Rock	Rock + 6
C: Close-Up Spectrometer Read of a Rock	Rock + 1

Table 1: Individual Goals and Rewards

Goal Combination	Reward
<Goal A, Goal B>	(Rew(A) + Rew(B))*1.75
<Goal A, Goal C>	(Rew(A) + Rew(C))*2.25, 90%
	(Rew(A) + Rew(C))*10.0, 10%
<Goal B, Goal C>	(Rew(B) + Rew(C))*1.25

Table 2: Goal Interdependencies and Rewards

conflict-free plan, if its score is the best seen so far, we record its point in the search space and begin the next optimization step. This approach protects against the possibility of adding a goal to the plan that cannot be solved.

To select the next goal to add during this process, we use a simple, greedy improvement heuristic that considers all goals and picks the one that would lead to the highest score if it were added to the plan. We also include an element of randomness to avoid repeatedly adding an unachievable goal. With probability $1 - \varepsilon$ we add the highest scoring goal, otherwise a goal is picked at random.

5 Evaluating Planning Performance with Interdependent Goals

5.1 Testing Methodology

We performed a series of experiments to evaluate whether or not explicitly taking into account goal interdependences during optimization would significantly improve the quality of the overall team plan. We expected to see some improvement over a system that did not use goal interdependences, but were not sure if the improvement in quality would be worth a potential increase in time to produce the plans. For these tests we compared our distributed version of CASPER with support for interdependent goals (which we will refer to as CASPER+IDGS) to two other distributed versions of CASPER: CASPER+Random and CASPER+SimpleReward. All three versions used the randomized hill-climbing algorithm described in the previous section. The only difference is in how each of the three selects the next goal to add to the plan. CASPER+IDGS uses the objective function from Equation 1 to pick the next goal. CASPER+Random simply selects a goal at random without considering rewards. Finally, CASPER+SimpleReward uses an objective function that looks at individual goal rewards without considering goal interdependencies.

We ran each distributed system on a set of generated problems from the previously explained Mars exploration domain. For these particular tests, we did not use the data analysis component to generate goals, but instead used a random problem generator to produce problems of varying degrees of difficulty. In particular, problems varied in the number and location of the science goals, as well as the size of the terrain area to be explored. Table 1 shows the types of goals that are given to the planner along with the possible rewards for each individual goal. The importance of an individual rock is chosen randomly from the range 1-14. Each problem specification contains a set of (optgoals to take images and spectrometer measurements of particular rocks in the selected area. Problems ranged in size from 30 to 90 different goals to examine 10 to 30 rocks in the surrounding terrain. The rovers are given 2 Martian days to complete these goals. Due to domain resource and temporal constraints, most of the generated problems are too large to fully complete. Thus the planning system will have to take into account the different goal utilities to determine which subset of goals to achieve.

Each problem description also included a randomly generated set of goal interdependences, which were based on preferences derived from conversations with planetary geologists and represent the type of utility values considered by human experts. Table 2 shows the goal combinations used for the experiments and the associated rewards. To increase the variance among goal combinations, we used two different factors for computing the value for one of the goal pairs (pair A and C). A certain percentage of the time the rewards for this pair was significantly increased. Finally, for a given rock, each of the three goal combinations is removed with probability 0.5.

We generated a set of 30 problems and ran each version of distributed CASPER on each problem 5 times. The systems were run on a Linux 3.06GHz P4 workstation with 1GB of RAM. To run tests in a reasonable time frame, we ran each planner in a batch mode where the planners were synchronized after each optimization session. This allows planners to still periodically communicate status information and for the central planner to re-assign goals to another rover when shed by one rover.

5.2 Results

At the end of each optimization step we recorded the current plan score based on the objective function from Equation 1. We also recorded several other statistics, including the number of seconds spent during each optimization step, the current number of goals in the plan, and the cumulative traverse distance required by the current plan.

Figures 3-5 present the results from these runs. Figure 3 shows the objective function scores and that

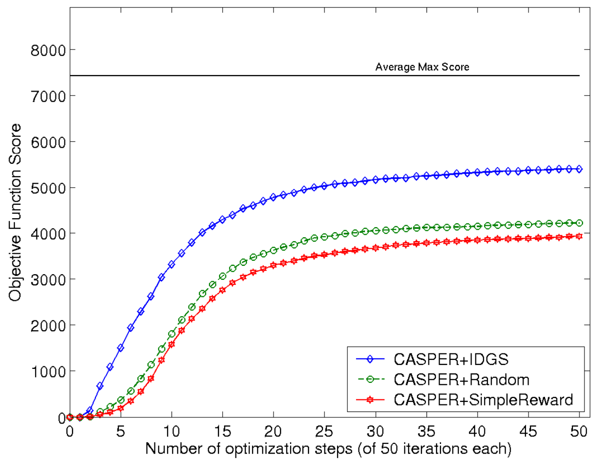

Figure 3: Objective Function Score

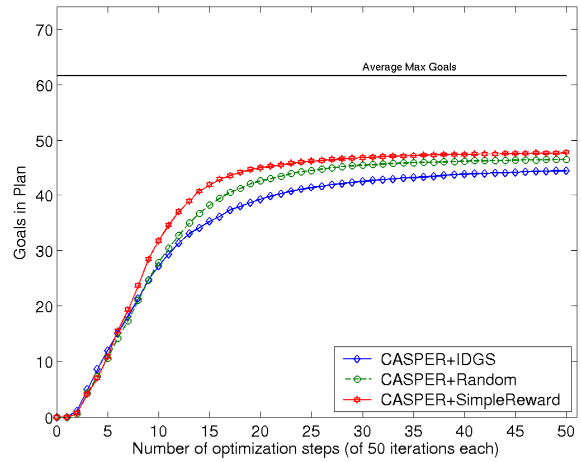

Figure 4: Number of Goals Achieved

CASPER+IDGS outscores both the other approaches. In fact, CASPER+IDGS shows a statistically significant improvement over both approaches at all but the first few optimization steps. Figure 4 shows that for the majority of data points, CASPER+IDGS added fewer goals to the plan. This factor is important because it shows that a higher score can be achieved using fewer goals. Note, that none of the planners were able to achieve all the goals and in such cases it becomes particularly important to achieve the higher quality subset. Another gathered statistic (not shown in a figure) was the average traverse distance required by each plan. These results showed that the plans created by CASPER+IDGS required the rovers to travel up to 15% shorter distances than the other planners, while still achieving a higher quality plan.

It is also important to note that CASPER+IDGS's biggest improvements in performance occur in the early optimization steps. Thus, if the planner is under tight time constraints, using CASPER+IDGS will allow the planning system to find a much higher quality set of goals. This feature is especially important in real-world problems where planning time can be tightly bounded.

Figure 5 shows that reasoning about interdependent goal values does not require additional planning time. This benefit is important when a planner is given more goals than it can achieve as well as when the planner is under time constraints and may not have enough time to plan for all its goals (even if achieving all goals is feasible).

6 Related Work

Many cooperative robotic systems use reactive techniques to coordinate robot behavior (Mataric, 2003; Parker, 1998; Huntsberger, 2003). These systems have been shown to exhibit low-level cooperative behavior in both known and noisy environments. However, they have not been shown useful for mission planning where a set of high-level goals must be achieved in a predictable manner and while obeying a series of resource and state constraints.

Some systems have used planning techniques to determine robot behavior. One example is FIRE (Goldberg, et al., 2002), which coordinates actions of multiple robots at several layers of abstraction. The top planning layer uses a market-based strategy to distribute tasks among robots, where robot travel time is the primary measure of cost. Another example is GRAMMPS (Brummit & Stentz, 1988), which has a central planner and a low-level planner on each robot, however does not consider multiple resources or exogeneous events. Our design has some similarities to teamwork approaches (Tambe, 1997), where the central planner is the leader and rover planners are followers, however, in MISUS each team member can fully re-plan based on current goal and resource knowledge. Furthermore, none of these techniques consider information on interdependent goals or are integrated with a data analysis system to provide new goals.

Work in planning optimization has used utility models to improve on static quality measures, such as missed deadlines or minimizing resource usage (Williamson and Hanks, 1994; Joslin, 1999; Rabideau 2000). Our approach, however, allows for the representation of utility for specific goal combinations that can change from problem to problem. The goal combinations used in this paper could be encoded into a Markov Decision Process (Boutilier, et al., 1999), however MDPs have yet to be demonstrated on problems of significant size in domains with time and resource constraints.

Previous work in decision analysis has looked at decision making with multiple objectives (Keeney and Raiffa, 1993) enabling one to develop preferential structures over decision outcomes. Our representation of goal interdependences is a simple type of preference structure that allows the planner to select among

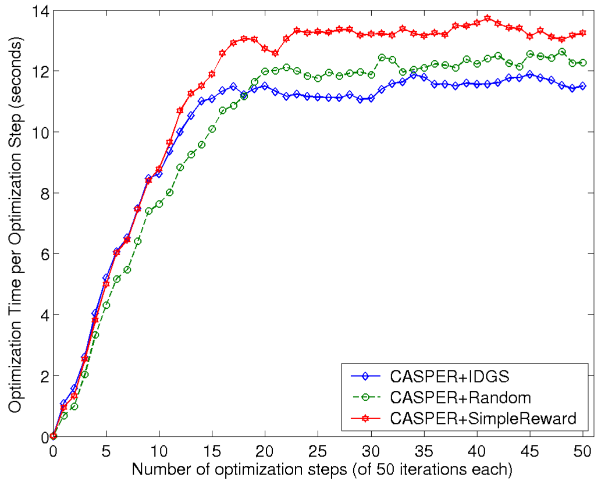

Figure 5: Plan Generation Time

alternate actions. In the future, we plan to incorporate more results from decision analysis to support more complex goal relations and uncertainty about goal pay-off.

7 Conclusions

This paper presents an approach for coordinating multiple rovers in achieving planetary science goals. The system integrates techniques from planning and scheduling with machine learning to autonomously analyze, request and obtain new science data. An important feature of our system is its ability to represent and reason about interdependent science goals. We have shown how this information is used in our distributed planning system and presented a set of experimental results that show how this approach can significantly improve plan quality.

In future work, we plan to apply the full MISUS system to other areas of planetary geology and exploration. In particular, we would like to expand the system to cover the testing of particular hypotheses or the handling of more closely coordinated tasks such as science observations that require more than one rover to execute. We also plan to consider more complex goal interdependencies including relations among more than two goals, relations in which only so many of a certain set of goals should be achieved, and situations in which adding certain combinations of goals can decrease plan quality. Finally, though currently this system is operated only in simulation, we intend to ultimately test its capabilities using real rovers examining actual terrain features.

Acknowledgements

The research described in this paper was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. We would like to acknowledge Bradley Clement for providing the SHAC framework and helping us employ it within MISUS. We thank Wolfgang Fink for his work in developing the data analysis approach. We also want to thank Martha Gilmore and Ashley Davies for their valuable expertise in planetary geology.

References

Boutilier, C., Dean, T., and Hanks, S. "Decision-Theoretic Planning: Structural Assumptions and Computational Leverage," *Journal of Artificial Intelligence Research*, 11:1-94.

Brummit, B., and Stentz, A., "GRAMMPS: A Generalized Mission Planner for Multiple Mobile Robots in Unstructured Environments," *Proceedings of the IEEE Conference on Robots and Automation*, Philadelphia, PA, 1988.

Chien, S., Knight, R., Stechert, S., Sherwood, R., and Rabideau, G., "Using Iterative Repair to Improve the Responsiveness of Planning and Scheduling," *Proceedings of the Fifth International Conference on Artificial Intelligence Planning and Scheduling*, Breckenridge, CO, April 2000.

Clement, B. and Barrett, T., "Continual Coordination through Shared Activities," *Proceedings of the 2nd Int'l Conference on Autonomous Agents and Multi-Agent Systems*, Melbourne, Australia, 2003.

Estlin, T. and Gaines, D., "An Optimization Framework for Interdependent Planning Goals," *Proceedings of the AIPS Workshop on Planning and Scheduling with Multiple Criteria*, Toulouse, France, April 2002.

Estlin, T., Gray, A., Mann, T., Rabideau, G., Castano, R., Chien S., and Mjolsness, E. "An Integrated System for Multi-Rover Scientific Exploration," *Proceedings of the Sixteenth National Conference on Artificial Intelligence*, Orlando, FL, July 1999.

Goldberg, D., Cicirello, V., Dias, M., Simmons, R., Smith, S., Smith, T., and Stentz, A., "A Distributed Layered Architecture for Mobile Robot Coordination: Application to Space Exploration," *Proceedings of the Third International NASA Workshop on Planning and Scheduling for Space,* Houston, TX, October 2002.

Huntsberger, T., Pirjanian, P., Trebi-Ollennu, A., Nayar, H., Aghazarian, H., Ganino, A., Garrett, M., Joshi, S., Schenker, P., "CAMPOUT: A Control Architecture for Tightly Coupled Coordination of Multi-Robot Systems for Planetary Surface Exploration," *IEEE Transactions on Systems, Man & Cybernetics, Part A: Systems and Humans,* 33(5):550-559, 2003.

Joslin, D., and Clements, D., "Squeaky Wheel Optimization," *Journal of Artificial Intelligence Research* 10:353-373, 1999.

JPL, http://www.jpl.nasa.gov/missions/, 2004.

Keeney, R., and Raffa, H., *Decisions with Multiple Objectives*, Cambridge University Press, New York, NY, 1993.

Parker, L., “Alliance: An Architecture for Fault Tolerant Multirobot Cooperation,” *IEEE Transactions on Robotics and Automation*, 14(2):220-240, 1998.

Mataric, M., Sukhatma, G., and Ostergaard, E., “Multi-Robot Task Allocation in Uncertain Environments,” *Autonomous Robots*, 14:255-263, 2003.

Rabideau, G., Engelhardt, B., and Chien, S., “Using Generic Preferences to Incrementally Improve Plan Quality,” *Proceedings of the Fifth International Conference on Artificial Intelligence Planning and Scheduling*, Breckenridge, CO, April 2000.

Tambe, M., “Towards Flexible Teamwork,” *Journal of Artificial Intelligence Research*, 7, 1997.

Williamson, M., and Hanks, S., “Optimal Planning with a Goal-Directed Utility Model,” *Proceedings of the Second Int’l Conference on Artificial Intelligence Planning Systems*, Chicago, IL, June 1994.

Zweben, M., Duan, B., Davis, E., and Deale, M., “Scheduling and Rescheduling with Iterative Repair,” In *Intelligent Scheduling*, Morgan Kaufmann, San Francisco, CA, 1994.