An Optimization Framework for Interdependent Planning Goals

Tara A. Estlin, Daniel M. Gaines

Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive
Pasadena, California 91109-8099
{firstname.lastname}@jpl.nasa.gov

Abstract

This paper describes an approach for optimizing over interdependent planning goals. 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, goals are related through detailed utility models that may 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. To address these types of problem situations, we have implemented a methodology for representing and utilizing information about interdependent goals and their related utilities using the ASPEN planning and scheduling system. We show through experimental results that this approach significantly increases overall plan quality versus a standard approach that treats goal utilities independently.

Introduction

As the sophistication of planning techniques grows, these systems are being applied to an increasing number of real-world problems. Planning and scheduling techniques are currently being applied with great success to handle problems in manufacturing, logistics, and space exploration. In a typical application, a planner is given a set of goals, and it then constructs a detailed plan to achieve the goals where the plan must respect a specific set of domain rules and constraints. A limitation of most planning systems, however, 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 are hard to include as part of a standard domain definition, since they are often dependent on current data and can vary widely from problem to problem.

When planning for NASA spacecraft or rover missions, planning goals are often dictated by science data that has just been collected. Goal utilities and dependencies for new science measurements are often dependent on a current data model and on what new science opportunities are available. Goal interdependencies can be seen in other domains as well. For instance, consider a travel-planning domain where we are planning a business trip for several people to the same location. Thus, all travelers need to arrive at the same destination and in the same general timeframe. In most cases, they would all prefer to arrive on the same day and time, however, plans that have some travelers arriving one day earlier are still valid and would still be considered. Furthermore, preferences for when people arrive could change from trip to trip. On one trip it may be important that a certain set of people arrive on the same day to attend a particular meeting. On other trips this criteria may be less important or apply to a different set of people. Representing such information in current planning systems would be difficult since most goal dependencies cannot easily change between problem instances based on new preference information.

Approaches to goal handling and representation vary widely among planning and scheduling systems. In some approaches, all goals must be achieved for the planner to even reach a solution. In other approaches, goals can be given different priorities or utilities, and the planner will try to create a plan that achieves the highest utility score where some goals may not be added to the plan. Other approaches enable a planner to accept both goals and other quality objectives, 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). However, even in approaches that allow the usage of more flexible optimization metrics, goal relationships are pre-defined in a domain model and typically remain relatively constant between problem instances. Furthermore, it is difficult to define utility metrics that involve specific goal instances as opposed to a general quality concept that applies to a certain class of goals (e.g., increasing the number of orders filled).

Most planning systems do allow you to define some types of *static dependencies* between goals. For instance, two goal or action types could be defined as related in a domain model, perhaps through a decomposition of a

Goal Num	Target Description	Location (x,y,z)	Reward
1	Spectrometer read for rock type x	(3.4, -34.6, 2.0)	10
2	Spectrometer read for rock type x	(162.3, 43.9, 1.1)	10
3	Spectrometer read for rock type x	(-4.1, 145.8, 0.4)	10
4	Spectrometer read for rock type y in area A	(104.3, -12.1, 1.5)	12
5	Soil sample from area A	(103.5, -13.4, 0.2)	15
6	Rock image for rock type y in area A	(104.3, -12.1, 1.5)	10
7	Dust collection experiment from area A	(105.1, -13.7, 1.5)	12

Table 1: Example sets of science goals given to planning system

parent activity. In a travel domain, you might want to tie a "board-plane" action with a "deboard-plane" action, since both will commonly occur in the same plan. Some static dependencies may also be defined automatically through other parts of the model definition. For instance, pre- and post-conditions links can relate certain goals. A domain model does typically allow goals to be linked in optional ways (e.g., a goal that could decompose to several different sets of actions or goals), however, these options are usually limited to several commonly-seen combinations. Encoding a large number of dependency options in a domain model would be intractable both for modeling ease and search complexity. No current planning systems enable *dynamic dependencies* among goals, i.e. dependencies that significantly vary from problem to problem, that can be easily utilized and defined as part of the problem specification instead of the domain model.

This paper presents a method for handling interdependent planning goals while performing plan construction and optimization. In this approach, interdependencies between goals can be formulated dynamically and provided to the planning system as part of the goal input. The planning system can then reason about these dependencies and incorporate them into the overall objective function it uses to rate plan quality and direct its search process.

This is particularly important when attempting to optimize a plan relative to multiple criteria. One approach to planning with multiple criteria is to combine the different objective functions into a single metric representing overall plan quality. However, for many domains, these objectives will interact in complex (e.g. nonlinear) ways making it difficult to improve plan quality. Our approach represents a step toward addressing this problem by providing the planner with an explicit representation of the interdependent relationships among the individual criteria that contribute to overall plan quality. Our planner uses this information to guide its search toward higher quality plans.

This optimization approach has been implemented on top of the Automated Scheduling and Planning Environment (ASPEN) (Chien, et al., 2000). ASPEN already has a base optimization framework that we have extended to handle this class of problems (Rabideau, et al., 2000). This new approach has been tested on a series of problems based on a team of rovers performing geological experiments in a new terrain. Even with our current implementation's relatively simple objective function and search technique, experimental results show that by using information about related goals, our approach is able to significantly improve plan quality.

Planning for a Multi-Rover Domain

In recent years, NASA has begun to focus on missions that utilize rovers to perform exploration and understanding of planetary terrains. Future missions will likely send teams of rovers to autonomously explore planetary surfaces.

To produce plans for a team of rovers, we have adapted a version of the ASPEN planning system (Estlin, et al., 1999). ASPEN automatically generates the necessary activity sequence to achieve a set of input goals. One of the main algorithms used to produce this sequence is a local, early-commitment version of iterative repair (Minton and Johnston, 1988; Zweben et al., 1994), which classifies plan conflicts and attacks them individually. For the experiments presented in the paper, planning is performed in a centralized fashion, where one planner controls multiple rovers. In future work, these techniques will be migrated to operate in a distributed planning system, where each rover has a separate onboard planner controlling its operations (Estlin, et al., 2000).

Plan Optimization

ASPEN provides an optimization framework that allows the representation of continuous *soft constraints* (i.e., preferences) (Rabideau, et al., 2000). In contrast to traditional *hard constraints*, soft constraints do not have to be satisfied for the plan to be valid. However, satisfying them will improve the quality score for the plan.

In ASPEN, a preference is defined as a mapping from a plan variable (e.g. resource level, goal count, etc.) to a quality metric. Specifically, a preference indicates whether the score is monotonically increasing or decreasing with respect to the plan variable. The overall plan score is the weighted sum of individual preference scores.

An iterative optimization algorithm, similar to iterative repair, is used to improve plan quality. For each defined preference, an improvement expert automatically generates modifications that could potentially improve the preference score. In the following sections we illustrate how we extended ASPEN's optimization framework to deal with interdependent goal combinations.

Interdependent Goals and Utilities

Historically in planning and scheduling systems, goal selection has been a linear process in which goals are independently selected and prioritized based on their expected reward. However, in some applications, this model is insufficient to correctly characterize the utility of a plan. For instance, in the case of performing science experiments in a new planetary terrain, goal priorities should be determined by the expected scientific gain, which is dependent on data already collected and available science targets. There are many situations in this type of domain where the value of a science goal will be increased if other related science goals can also 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 done. Conversely, there are situations in which it is very important to achieve one of a set of goals, but having accomplished one in the set, the others become less important. For instance, we may want a rover to collect one or two more samples of a particular rock type but there are a large number of possible targets from where to collect such a sample. In this situation, we would like to direct the planner to collect a couple samples and then move on to other science experiments. If samples were collected at all target sites, this data would be overly redundant and somewhat lower the utility of the overall set since time had been wasted collecting unneeded data.

To represent a goal's value, we have extended a typical goal-utility representation (where goals can have individual rewards representing their importance) so that complex interdependencies and their relevant utilities can be represented and utilized by a planning system. Furthermore these interdependencies and utilities can change between problem specifications without requiring any changes to the planner domain model. In our representation, a list of goals and goal combinations are provided to the planner. A utility value is also assigned to each goal and to each specified goal combination. As an example, consider the spectral measurement and image goals shown in Table 1, which are from the previously introduced rover domain. Let's assume these goals are interdependent in several ways. First, Goals 1-3 are for spectrometer readings for the same type of rock and it has been deemed necessary to obtain only one such reading and any more would add little value to the current set of collected data. Second, Goals 4-7 are for the same rock or rock area and it has been determined desirable to obtain all of those observations. However, if only a few can be obtained that data would still be beneficial but not provide as much scientific value as the entire set.

These types of goal combinations are difficult to represent in standard planning-optimization approaches. As mentioned previously, a number of systems represent goal rewards in the form of utility functions or preferences, however, these approaches typically try to maximize a certain goal type or minimize usage of a certain resource. For instance, a utility function may try to minimize the amount of fuel used in transporting objects, or may try to maximize the number of factory orders that can be filled. This type of representation is limited in that it prefers to decrease or increase the number of goals or activities of a general type, where each goal or activity is viewed as relatively equal (or interchangeable). The goal inter-dependencies required for deducing many scientific hypotheses are often much more complex since each individual goal may play a different role in the overall success of an experiment.

We can visually represent goal inter-dependencies between a set of two goals by using a graph structure where vertices represent individual goal rewards and edges represent interdependent goal rewards. For example, Figure 3 shows two goals that have individual rewards (represented by G_1 and G_2) and a combined reward (represented by R_{12}). There may also be dependencies between larger sets of goals, and thus the graph may contain hyperedges linking several goals to their combined value. Table 2 shows interdependent goal rewards for the goals introduced in Table 1. Goal combinations for goals 1-3 are given slight negative rewards to show that achieving more than one goal in this set actually has less value than just achieving one. The goal combination for goals 4-7 shows that achieving all of the goals in that set has a large bonus reward.

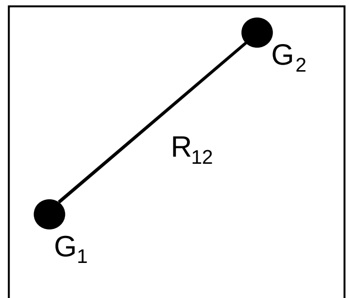

Figure 3: Two related goals

Plan Optimization for Interdependent Goals

We extended the ASPEN optimization system to support the inclusion of goal interdependencies with a planning problem description. The extension consists of two main components: an objective function to compute the value of the plan with respect to the goal interdependencies and an optimization framework for selecting goals to achieve and coordinating optimization with plan repair.

Objective Function

As is the case with most planners, the ASPEN problem specification includes a description of the goals that must be achieved to accomplish a particular problem. In

Goal Combination	Reward
<Goal 1, Goal 2>	-5
<Goal 1, Goal 3>	-5
<Goal 2, Goal 3>	-5
<Goal 4, Goal 5, Goal 6, Goal 7>	60

Table 2: Goal interdependencies and corresponding rewards

addition, ASPEN can accept a set of optional goals that, while not required, will increase the quality of the plan as more of these goals are accomplished. This is useful when the planner is given more goals than are feasible to achieve given its resource constraints. In this case, ASPEN will use an objective function to try to find a subset of goals that result in a valid, high quality plan.

Our extended version of ASPEN also takes as input a set of goal interdependencies specified as a graph of goal nodes as described in the previous section. The graph consists of a set of vertices V where each vertex corresponds to a goal that can be added to the plan, including both mandatory and optional goals, and a set of edges E. Each edge consists of a tuple of vertices: $<v_1, v_2, \ldots v_n>$. For each vertex and each edge, there is an associated weight $w_{<v1, v2, \ldots vn>}$ indicating the value that will be added to the plan if the plan includes these 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 add to the plan.

We use a simple objective function to calculate the plan quality with respect to these optional goals. Let G be the set of goals that occur in the plan. The value of plan P is then given by Equation 1. This function sums up the values of all goals that occur in the plan along with the weight for each edge for which all of the edge's vertices occur in the plan.

$$O(P) = \sum_{v \in V} o(\prec v \succ) + \sum_{\prec v_1, v_2, \ldots, v_n \succ \in E} o(\prec v_1, v_2, \ldots, v_n \succ)$$

$$o(\prec v_1, v_2, \ldots, v_n \succ) = \begin{cases} w_{\prec v_1, v_2, \ldots, v_n \succ} \text{ if } \{v_1, v_2, \ldots, v_n\} \subseteq P \\ 0 \text{ otherwise} \end{cases}$$

Equation 2: Objective function for calculating plan utility when using interdependent goals

Optimization Framework

The next step is to provide an improvement expert that can suggest what changes ASPEN should make to the plan to increase this score. Clearly, the improvement expert for interdependent goals should suggest adding more optional goals to the plan. However, adding a goal will likely result in conflicts in the plan. Therefore it is also necessary to coordinate the process of improving the plan score with ASPEN's repair process to fix conflicts in plans.

Our current approach to performing optimization for interdependent goals is randomized hill-climbing with restart. We begin by first creating a plan that achieves all of the mandatory goals. 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 the improvement expert to suggest the next optional goal to add. If there are conflicts, we perform an iteration of repair. Whenever we have a conflict free plan, if its score is the best we have seen, we record its point in the search space. At the end of the *i*th iteration, we return to the highest-valued 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.

We use a simple, greedy improvement expert to select the next goal to add. It considers all goals and picks the one that would lead to the highest score if it were added to the plan. We 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.

Evaluating ASPEN's Performance with Interdependent Goals

Our main concern in evaluating our system was to see whether or not explicitly taking into account goal interdependencies during optimization would significantly improve the quality of the plan. We expected to see some improvement over a system that did not use goal interdependencies, but were not sure if the improvement in quality would be worth a potential increase in time to produce the plans. We were also curious to see how much of an improvement would be provided by our relatively simple objective function.

Methodology

We compared our extended version of ASPEN, which we will refer to as ASPEN+IDGS (for ASPEN with InterDependent Goal Support) to two other versions of ASPEN: ASPEN+Random and ASPEN+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 optional goal to add to the plan. ASPEN+IDGS uses the objective function from Equation 1 to pick the next goal. ASPEN+Random simply selects a goal at random without considering rewards. Finally, ASPEN+SimpleReward uses an objective function that looks at rewards for individual goals without considering goal interdependencies.

We ran each system on a set of randomly generated problems from a Mars exploration domain. In this domain, a team of three rovers must collect different types of science data at various locations on the planet's surface.

The planner must decide which goals to assign to each rover, determine a sequence for each rover to use in visiting the different locations, and plan for activities such as manipulating the rover masts and communicating with earth. Generated plans must also respect resource and temporal constraints, such as not exceeding onboard memory limitations when collecting data.

The randomly generated problems varied in the number and location of the science goals. Table 3 shows the types of goals that are given to the planner along with the possible rewards for each individual goal. Note that some goals have a range of rewards in which case a specific reward is drawn randomly from this range. Each problem specification contains several mandatory panoramic images (goal type A) of different terrain areas, which always provide a base set of data on each area, and then a set of optional goals to take additional images and spectrometer measurements (goal types B, C, and D) of particular rocks in those areas. Problems could range in size from 6 to 78 different goals to examine 0 to 24 rocks in the surrounding terrain.

Goal	Reward
A: Panoramic Image of an Area (Mandatory)	20
B: Long-Range Image of a Rock	12-25
C: Close-Up Image of a Rock	7-20
D: Close-Up Spectrometer Read of a Rock	2-15

Table 3: Individual goals and rewards

The rovers are given 1 Martian day to complete these goals. Depending on the relative locations of the targets, each rover can typically handle about 10 goals in this time. With three rovers this means that most of the problems will be too large to complete and the planner will have to take into account the different goal values to determine which goals should be achieved.

Each problem description also included a randomly generated set of goal interdependencies. Although the interdependencies were randomly generated, they were based on preferences derived from our conversations with planetary geologists and represent the type of utility values considered by human experts. Table 4 shows the goal combinations used for the experiment 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 B and D). A certain percentage of the time the reward for this pair was significantly increased. Finally, for a given rock, each of the three goal combinations is removed with probability 0.5.

In selecting parameters for the randomized hill-climbing algorithm used in each planner, we decided to use 50 iterations per optimization step as it seemed to provide the best balance between allowing the planner enough time to repair goals but not so long that it would waste a lot of time if it got stuck and needed to back up to a previous plan. For ε, we selected a small value of 0.02.

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

Table 4: Goal interdependencies and rewards

Results

We generated a set of 30 problems and because there is an element of randomness both to the ASPEN iterative repair algorithm and to our optimization approach, we ran the three versions of ASPEN on each problem 5 times. The systems were run on a Sun Blade 1000 with 1 Gigabyte of RAM.

At the end of each optimization step we recorded the current plan score based on the objective function from Equation 1, the current number of goals in the plan, and the number of seconds spent during that step. Note that even though the ASPEN+Random and ASPEN+SimpleReward versions of the planner did not make use of the objective function to select goals to add, we still used that objective function to score their plans for the purpose of the experiment.

Figures 4-6 present the results from these runs. Objective function scores are compared in Figure 4, while Figures 5 and 6 compare the total number of goals achieved and the planning time used by each method. The data points in each graph are averaged over the 150 runs from each system. In each graph, the data point at optimization step 0 represents the planner performing repair on a plan containing all mandatory goals. We performed two-tailed t tests between each pair of the three systems with a Bonferroni correction. The only graph that showed significant differences among the systems was the graph of plan scores in Figure 4. ASPEN+IDGS was found to be significantly better than both ASPEN+Random and ASPEN+SimpleReward at the 0.01 confidence level. ASPEN+Random outperformed ASPEN+SimpleReward but only the data points between optimization steps 6 and 14 showed significant difference at confidence level 0.01.

Discussion

Figure 4 shows that ASPEN+IDGS outscores both ASPEN+Random and ASPEN+SimpleReward. In fact, ASPEN+IDGS showed a significant improvement over both versions at each data point. The plot of the number of goals included in each plan (Figure 5) shows that all three systems were achieving about the same number of goals. This means that ASPEN+IDGS was selecting higher quality goals. This factor is particularly important because none of the planners were able to achieve all of the goals thus it is better to achieve the higher quality subset.

It is also important to note that ASPEN+IDGS's biggest improvements in performance occur in the early

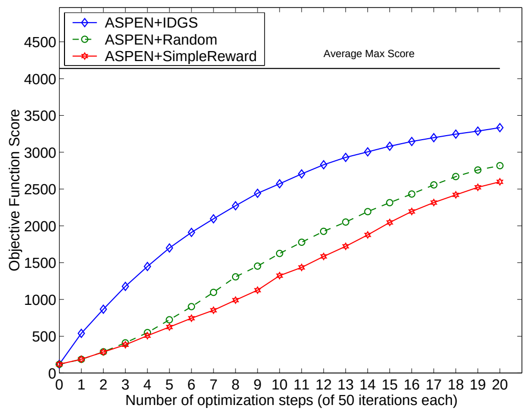

Figure 4: Objective function score

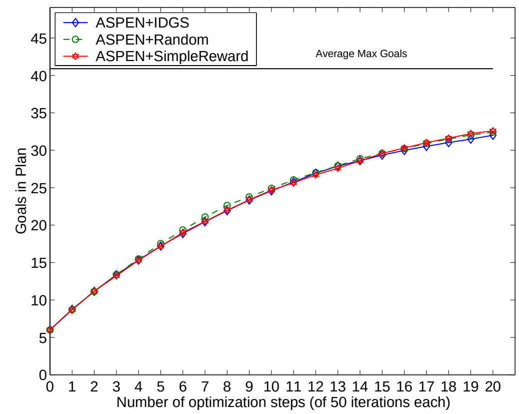

Figure 5: Number of goals achieved

optimization steps. Thus, even if the planner is capable of solving all the goals it is given but it is under tight time constraints, then using ASPEN+IDGS will allow the planner to find a much higher quality set of goals. This feature is especially important in real-world problems where planning time can be tightly bound.

The shapes of the curves reveal some interesting characteristics about each algorithm. The curve for ASPEN+IDGS rises sharply in the early optimization steps and then tapers off, while ASPEN+Random starts rising more slowly, increases in its rate of growth, and then begins to taper off at the end. Given that both planners were adding about the same number of goals to the plan at each time step, the differences in the curve shapes is a result of the way each algorithm selected goals. The sharp rise in the ASPEN+IDGS curve can be explained by the fact that ASPEN+IDGS is explicitly looking to add goals that will improve the objective function. However, as more goals are added to the plan, and therefore the rovers' resources are beginning to be stretched to their limit, making repairs to the plan becomes more difficult and the planner spends more iterations fixing problems with the plan and fewer iterations adding goals. As a result, the curve begins to level off. As can be seen in Figure 5, the number of goals added to the plan at each optimization step begins to decrease at about the same time that ASPEN+IDGS's score begins to taper off in Figure 4.

In contrast, the ASPEN+Random curve in Figure 4 begins slowly because it is randomly adding goals to the plan and, early on, it is unlikely that the interdependent goal combinations will be satisfied in the plan. However, as more goals are added, the probability of satisfying goal combinations when a new goal is added increases, and the score begins to rise more rapidly. But, just like ASPEN+IDGS, the planner begins to spend more time performing repairs and fewer goals are added to the plan causing the curve to taper off.

The fact that ASPEN+SimpleReward was the worst performer is particularly interesting. Recall that this version of the system is selecting new goals based on the each goals individual contribution to the plan. In other words, it is using the rewards from Table 3. Therefore, the planner will favor the addition of long-range images and avoid adding close-up spectrometer reads. The problem with this approach is that the goal interdependencies do not necessarily preserve the relative reward values of the individual goals. For example, although the close-up spectrometer read is the lowest rank score individually, when it is combined with a long-range image, it becomes much more valuable. However, since ASPEN+Simple-Reward typically avoids adding this goal to the plan, it does not satisfy these high-quality goal combinations. As a result, its score grows slowly and, like the other curve, tapers off in later optimization steps.

Figures 4 and 5 show that ASPEN+IDGS provides considerable benefit when the planner cannot achieve all the goals in a plan. In this case, ASPEN+IDGS selects a higher quality subset of goals than either of the two competing systems in this study. This is already advantageous, but we were also interested in whether or not ASPEN+IDGS could increase plan quality without a significant increase in planning time. The plot of each system's processing time per optimization step in Figure 6 shows ASPEN+IDGS did not significantly increase planning time.

These results show that ASPEN+IDGS provides a significant improvement in plan score over versions of the planner that do not consider goal interdependencies without a significant increase in planning time. This benefit is most 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 of its goals.

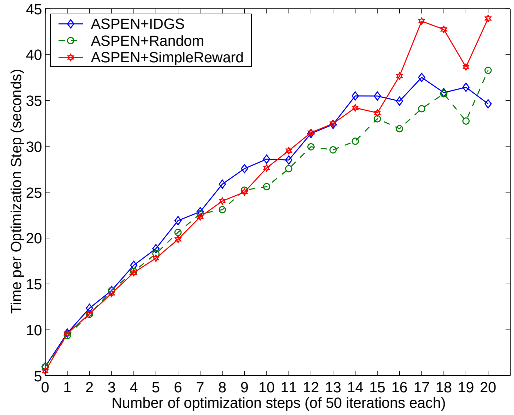

Figure 6: Plan generation time

Related Work

Other work in planning optimization has used utility models to improve on particular types of quality measures. PYRRHUS (Williamson and Hanks, 1994) extends the UCPOP partial-order planner to handle metric time, resources, and a utility model. In contrast to PYRRHUS, our approach allows for the representation of utility for specific goal combinations that can change from problem to problem.

Markov Decision Processes (MDPs) (Boutilier, et al., 1999) represent another approach to dealing with plan quality. The goal combinations used in this paper could be encoded into an MDP. However, MDPs have yet to be demonstrated on real problems of significant size in domains with time and resource constraints and it is likely that the large computational cost would be prohibitive.

Work in mixed-initiative planning allows a planner to be biased toward solutions with certain characteristics (Myers and Lee, 1999). While our work has focused on automated planning, a user could specify utility preferences to encourage certain goal combinations.

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 interdependencies is a simple type of preference structure which allows the planner to select among 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.

Conclusions

In this paper we have presented a method for utilizing interdependent goal utilities, where goal relations can be dictated by current information and can vary from problem to problem. In typical planning systems, only simple, static goal relations can be defined that remain relatively constant between problem instances. However, in many application areas, goal dependencies and their related utility metrics can dramatically change based on current information or even user preferences. To address this problem, we have implemented a new method for representing and reasoning about interdependent goals. We have also presented experimental results that show how this approach can significantly improve overall plan quality in a multi-rover application.

In future work we will 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. We also plan to enhance our current optimization algorithm to better recognize potential high-utility goal combinations. Finally, though currently this system is operated only in simulation, we intend to ultimately test its capabilities using real rovers examining actual terrain features.

Acknowledgments

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

Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement by the United States Government or the Jet Propulsion Laboratory, California Institute of Technology.

References

Boutilier, C., Dean, T. and Hanks, S. 1999. Decision-Theoretic Planning: Structural Assumptions and Computational Leverage. *Journal of Artificial Intelligence Research*, 11:1-94.

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

Estlin, T., Gray, A., Mann, T., Rabideau, G., Castano, R., Chien, S. and Mjolsness, E. 1999. An Integrated System for Multi-Rover Scientific Exploration. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence*, 613-620. Orlando, FL.

Estlin, T., Rabideau, G., Mutz, D., and Chien, S. 2000. Using Continuous Planning Techniques to Coordinate Multiple Rovers. *Electronic Transactions on Artificial Intelligence* 4:45-57.

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

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

Minton, S., and Johnston, M. 1988. Minimizing Conflicts: A Heuristic Repair Method for Constraint Satisfaction and Scheduling Problems." *Artificial Intelligence*, 58:161-205.

Myers, K., and Lee, T. 1999. Generating Qualitatively Different Plans Through Metatheoretic Biases. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence,* Orlando, FL.

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

Williamson, M., and Hanks, S. 1994. Optimal Planning with a Goal-Directed Utility Model. In *Proceedings of the Second International Conference on Artificial Intelligence Planning Systems,* Chicago, IL.

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