

# Techniques for Generating Optimal, Robust Plans when Temporal Uncertainty is Present

Janae N. Foss\*

Department of Computer Science  
Michigan Technological University  
1400 Townsend Drive  
Houghton, MI 49931  
jnfoss@mtu.edu

## Abstract

Planning under uncertainty has been well studied, but usually the uncertainty is in action outcomes. This work instead investigates uncertainty in the amount of time that actions require to execute. In addition to this temporal uncertainty, the problems being studied must have robust solution plans that are optimized based on an objective function. This thesis summary details two iterative approaches that have been used to solve these type of problems and discusses future work, including MDP approaches.

## Introduction

Uncertainty applies to several aspects of planning problems. Many planners have been built that prepare contingency plans when actions may affect the world in uncertain ways (Bresina *et al.* 2002). Less work has been done with planners that assume action durations are uncertain. One approach to dealing with this type of uncertainty is to take a pessimistic view of the world, assume a worst case scenario, and find conservative plans that are likely to execute to completion regardless of the amount time consumed by the actions in the plan. However, this approach is often undesirable as it leads to missed opportunities and slack time in the plan when actions complete quickly (Bresina *et al.* 2002). For example, assume that a Mars rover has to move from point  $a$  to point  $b$  and use either a slow, high resolution camera or a fast, low resolution camera to take an image at point  $b$ . Given that travel time is uncertain, a conservative planner may recognize that in the worst case there will not be enough time to use the high resolution camera, and thus choose to always use the low resolution camera. This plan is robust, but when the rover travels quickly the opportunity of getting a high resolution image is not realized and the rover may undesirably be left idle for some period of time. The first part of this thesis summary describes the problem in more detail, next iterative techniques for generating temporal contingency plans are given, and lastly a discussion about future research is given.

---

\*Supported by NASA Harriett G. Jenkins Pre-Doctoral Fellowship Program.  
Copyright © 2006, American Association for Artificial Intelligence (www.aaai.org). All rights reserved.

## Problem Specification

I am considering a class of problems with solutions that combine temporal uncertainty, optimality, and robustness. Each of these attributes is difficult to deal with individually and combining them makes reasoning more difficult. First, in many domains the duration of actions cannot be specified exactly. Such actions are typically represented by a closed interval specifying the lower and upper bound for the duration,  $[min-d, max-d]$ . Under this model, the actual duration required for an action is only known through observation after the action has executed. One way to deal with this uncertainty is to always assume actions take  $max-d$  and plan conservatively. However, in this class of problems solution plans are ranked by an objective function and the optimal solution is only attainable when actions complete quickly. This may lead one to try a more optimistic approach where it is assumed that actions only require  $min-d$  so that the optimal solution will always be generated, but this approach fails when the third attribute of robustness is taken into consideration. For this class of problems, it is important to have robust solutions, that is, plans that are guaranteed to run to completion regardless of the amount of time actions require to complete. Therefore, it is not good enough to simply find the optimal solution, because the optimal solution may fail when actions do not complete quickly. I am developing ways to generate plans where the optimal plan is followed when time allows but less optimal plans are also generated and followed when time constraints dictate.

## Temporal Contingency Planning

One way to create optimal plans that are also robust in the face of temporal uncertainty is to create temporal contingency plans (i.e., plans with contingency branches that are taken based on the observed time at execution). At present, I have developed two related iterative approaches for generating temporal contingency plans. They differ in that one is a greedy algorithm and the other is a hill climbing algorithm. My basic iterative approach is to first remove temporal uncertainty by assuming that each action always takes  $min-d$  amount of time to execute. Then, an optimal plan is generated by any planner that can handle durative ac-

tions and optimize based on an objective function<sup>1</sup>. Next, the optimal plan is converted to a *Simple Temporal Network (STN)* (Dechter, Meiri, & Pearl 1991). Through this conversion, the temporal uncertainty is factored into the plan. The STN is then incrementally analyzed one step at a time to find places where the plan may fail when actions require more than their minimum time to complete. After this point, the greedy and hill climbing algorithms differ. Where the greedy algorithm finds such a place, an attempt is made to generate a suboptimal contingency branch (using the same planner as above) to be executed when the offending step runs long (Foss & Onder 2005). In the hill climbing approach, an entire new plan is generated (again, using the above planner) each time a possible failure is found (Foss & Onder 2006). If the new plan differs from the initial plan only after the step where a problem occurs, a contingency branch is inserted. Otherwise, the initial (optimal) plan is replaced by the new plan.

Both the greedy and hill climbing versions of the algorithm benefit from the fact that they allow parallelism. This is especially important when deadlines are taken into consideration. Each approach has individual advantages, also. The greedy approach is faster when contingency branches can be added to repair the optimal plan. There are two related factors that contribute to this. First, the domain is modified so that the head of the plan will not be regenerated, restricting the search space. Second, because a contingency branch is shorter than a full plan, it is faster to generate it than to regenerate the entire plan as is done in the hill climbing algorithm. However, the greedy approach fails to find any solution when no contingency branches can be added to the optimal plan. Since the hill climbing approach always regenerates the whole plan, it is able to escape local minima/maxima. Also, the greedy algorithm may start with an optimal plan that is unlikely to be executed and augment this plan with very undesirable branches that are likely to be executed. In this situation the hill climbing algorithm may abandon the optimal plan and find a sub-optimal, but likely to succeed plan that would have higher utility than the branches in the greedy algorithm's plan.

### Future Work

The above described work improves on conservative planning techniques by including the most conservative plan as the least desirable contingency branch, executed only when more desirable options may cause failure. The techniques currently implemented assume a uniform distribution over the uncertain duration interval, however in real domains a normal distribution is a more likely fit (Schaffer, Clement, & Chien 2005). I plan to modify my iterative approaches to work with this type of distribution. In this context the initial assumption can be to start with a value for the duration that is likely to occur, based on the distribution. Then, op-

<sup>1</sup>Currently LPG-td(Gerevini *et al.* 2004) is being used for this step because it handles durational actions and timed initial literals (used for specifying deadlines), creates parallel plans, and considers the objective function at planning time. It has also performed well in the International Planning Competition.

portunity branches can be added for when actions complete faster than expected, and contingency branches can be added for when actions take longer than expected. This may be a good anytime approach to be applied when there is a limited amount of time available for planning.

In addition to further developing my iterative algorithms, I would also like to investigate MDP approaches to solving these problems. Unlike the iterative planning approaches, MDPs do not allow parallel actions. Even so, MDPs can be useful in this context because they naturally deal with uncertainty and take cost and rewards into account. One challenge in using MDPs to solve these type of problems is how to represent states when time is a factor. A naive approach is to include time in the state and thus have one state for each possible time increment. However, this would very quickly cause a blow-up in the size of the state space. It is likely that many states in this naive approach would be identical, only differing in time stamp. I plan to investigate ways to group states by time to reduce the number of states, without sacrificing quality in the solution policy.

Finally, I will develop a test bed of problems. In addition to the rover domain, I have been working with problems from a travel domain and an evacuation domain. In the travel domain, the goal is to travel from home to some destination within a given time constraint. In the evacuation domain, the goal is to evacuate as many people as possible within a given period of time. This is further complicated by the fact that there are intermediate deadlines for rescuing different groups of people. As such, it is easy to create problems where it is not possible to evacuate all people, resulting in oversubscribed goals (Smith 2004). Over-subscription is also an issue in the rover domain and most real world problems. I would like to develop techniques that directly address this issue by achieving more goals when actions complete quickly and only the highest priority goals, otherwise.

### References

- Bresina, J.; Dearden, R.; Meuleau, N.; Ramakrishnan, S.; Smith, D.; and Washington, R. 2002. Continuous time and resource uncertainty: A challenge for AI. In *18th Conference on Uncertainty in Artificial Intelligence*.
- Dechter, R.; Meiri, I.; and Pearl, J. 1991. Temporal constraint networks. *AI* 49:61–95.
- Foss, J., and Onder, N. 2005. Generating temporally contingent plans. In *IJCAI 2005 Workshop on Planning and Learning in A Priori Unknown or Dynamic Domains*.
- Foss, J., and Onder, N. 2006. A hill-climbing approach for planning with temporal uncertainty. In *FLAIRS 2006 Conference*. To appear.
- Gerevini, A.; Saetti, A.; Serina, I.; and Toninelli, P. 2004. Planning in PDDL2.2 domains with LPG-TD. In *International Planning Competition booklet (ICAPS-04)*.
- Schaffer, S. R.; Clement, B. J.; and Chien, S. A. 2005. Probabilistic reasoning for plan robustness. In *Proc. IJCAI 2005*.
- Smith, D. 2004. Choosing objectives in over-subscription planning. In *Proc. 14th International Conference on Automated Planning and Scheduling (ICAPS-04)*, 393–401.