

Probabilistic Plan Management

Laura M. Hiatt

Computer Science Department
Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, PA 15213
lahiatt@cs.cmu.edu

1 Introduction

The general problem of planning and scheduling for uncertain domains is a difficult challenge. Research that focuses on constructing schedules (or scheduling policies) by reasoning with explicit models of uncertainty has produced some promising mechanisms for coping with specific types of domain uncertainties (e.g., [McKay *et al.*, 2000; Beck and Wilson, 2007]), but in general these approaches have scaling difficulties [Bryce *et al.*, 2008]. Research in robust scheduling has alternatively emphasized the use of expected models and deterministic scheduling techniques, with the goal of constructing a flexible schedule (or set of schedules) that can absorb deviations at execution time [Policella *et al.*, 2006]. These approaches are much more scalable, but either result in overly conservative schedules at the expense of performance (e.g., [Morris *et al.*, 2001]) or ignore the potential leverage that can be provided by an explicit uncertainty model.

In my thesis, I adopt a composite approach that attempts to couple the strengths of both of the above tactics. As a starting point, a deterministic schedule is assumed. The main contribution is the act of layering an uncertainty analysis on top of the deterministic schedule, identifying its weak points and strengthening them according to the probability analysis, with the overall goal of raising the schedule's expected reward. Further, using the probability analysis to drive meta-level management of the schedule provides a means to control resource usage and maximize expected reward when resources are limited. I believe that by taking advantage of the known uncertainty model while avoiding the computational overhead of probabilistic planning and scheduling, the approach ultimately will succeed in earning higher reward during execution.

2 Domain

The domain considered for this work is an oversubscribed multi-agent scheduling problem which is concerned with the collaborative execution of a joint mission by a team of agents in a highly dynamic environment. Missions are formulated as a hierarchical network of activities, with leaf nodes being the executable "methods" in the world and "tasks" being the internal nodes. Method durations are specified as discrete probability distributions. Methods also earn reward, called *quality* in this domain, which is propagated up the hierarchi-

cal network according to aggregation functions at each task. Methods may also have a specified failure distribution and, if so, an associated likelihood of producing zero quality when executed. Methods also earn zero quality if they miss a deadline as specified in the domain model.

Also part of the given domain model is an initial deterministic schedule. During the distributed execution, agents are given their portion of the initial schedule and the activity hierarchy. Agents run on separate threads, as does a simulator, and communicate relevant status information to other agents asynchronously. Agents are responsible for maintaining their own schedule, e.g. updating their schedule with the realities of execution and rescheduling as appropriate. More information about the domain can be found in [Smith *et al.*, 2007].

In this thesis, an uncertainty analysis is developed which identifies weak points in the schedule as methods which have a non-zero probability of earning zero quality, and tries to fortify those areas of the schedule. This happens centrally on the initial deterministic schedule before it is distributed to agents, as well as distributively on agents' local schedules once execution has begun. For clarity, the rest of this summary will present the work in the context of this domain; however, the ideas are in no way limited to it and can be applied to a variety of uncertain planning and scheduling domains.

3 Probabilistic Strengthening

While deterministic planners have the benefit of being more scalable than probabilistic schedulers, they make strong and often inaccurate assumptions about how execution will unfold. In oversubscribed domains such as this one, if a timeline is tightly packed and even one method exceeds its scheduled duration, there is a high risk that a later activity will violate one of its constraints (e.g. a deadline). Rescheduling during execution can effectively respond to such unexpected dynamics, as long as dependencies allow sufficient time for one or more agent schedulers to recover from problems that arise. This work focuses on making deterministic schedules more robust for execution in order to minimize the number of unrecoverable problems. To do so, a deterministic scheduler is augmented with a schedule strengthening step, which is driven by a probabilistic analysis. Schedules are strengthened both centrally on the initial schedule before it is distributed to agents, and distributively on agents' local schedules in response to the dynamics of execution.

The approach relies on an algorithm that performs a probabilistic analysis of deterministic schedules. The algorithm calculates a *probability profile* for each activity a which includes its finish time distribution $FT(a)$, its probability of earning quality $PQ(a)$, and its expected quality $EQ(a)$. The profiles are first calculated for methods and then propagated through the activity hierarchy to get a profile for each activity in the network. In the distributed case, this propagation happens in part via communication between appropriate agents. Probability profiles are stored with each activity, and are readily updated if the information associated with upstream activities changes.

Once these probability profiles are known, they are used to focus schedule strengthening. Methods with non-zero likelihoods of earning zero quality are identified, and sorted according to their expected quality loss $EQL(m)$, a number which describes how much the overall schedule's quality is expected to suffer if the method m were to earn zero quality. Relative EQL values of methods indicate the relative importance of the methods with respect to the overall schedule.

When attempting to perform a strengthening action, the most important method is considered first, followed by later methods if the first cannot be strengthened. Example strengthening actions are: (1) adding redundant back-up methods; (2) swapping methods for shorter-duration alternatives to increase schedule slack; and (3) unscheduling less important methods to increase schedule slack, potentially allowing important methods to be more likely to make their deadline. Although these are the three used in this work, the set of actions used could be adapted to suit a variety of domains. While performing strengthening, I found that applying these strengthening actions in a round-robin fashion was the most effective, stopping when no more actions increase the schedule's expected quality [Hiatt *et al.*, 2009].

Ultimately, schedule strengthening results in more robust schedules which earn higher quality during execution than their non-strengthened counterparts. Interestingly, it also has the additional benefit of hedging against an agent's scheduler going down during execution since the last schedule produced will likely be more effective in the face of unexpected outcomes. Experiments confirm this and have shown that on benchmark problems, executing schedules strengthened in this way (without rescheduling during execution) earns 31% more quality than executing the original initial schedule [Hiatt *et al.*, 2009]. Early results on using schedule strengthening in conjunction with rescheduling are similarly promising.

4 Future Work

There are two additional components of this work that I would like to address. First, the amount of communication between agents increases dramatically with the approach, as agents frequently communicate changes in their probability profiles. Second, in conjunction with receiving many more status information messages from other agents, the frequency of scheduling increases as agents often reschedule in response to receiving them.

In resource-bounded domains such as this one, these two aspects of schedule strengthening can cause problems, leading

to quality loss. Messages can be lost or delayed when bandwidth limitations are exceeded, and quality can decrease as agents become saturated with rescheduling and cannot keep up with the pace of execution. Ideally, I want to manage execution to prevent this from happening.

I believe that using probabilities to drive a meta-level management policy can address these problems. For example, when deciding whether to send a message, an agent can use probability profile information to determine whether it probably needs to communicate the message to another agent. If it is unlikely that the message will have an impact on another agent's schedule, then communication should be avoided. For instance, a message specifying that $PQ(m)$ for some method m has changed by 1% is unlikely to have much effect on other agents' schedules.

Also, when an agent does receive a message, it can factor in its current probability profiles to decide whether a reschedule is likely beneficial at that point. As an example, in the extreme case, if an incoming message does not affect an agent's probability profiles at all then it is unlikely that a reschedule would be beneficial.

Ideally, changes such as these will allow agents to intelligently control their computational resources and communication bandwidth and, with them in place, I expect to be able to fully manage plans and schedules in a probabilistic way.

References

- [Beck and Wilson, 2007] J. C. Beck and N. Wilson. Proactive algorithms for job shop scheduling with probabilistic durations. *Journal of Artificial Intelligence Research*, 28:183–232, 2007.
- [Bryce *et al.*, 2008] Daniel Bryce, Mausam, and Sungwook Yoon. Workshop on a reality check for planning and scheduling under uncertainty. <http://www.ai.sri.com/bryce/ICAPS08-workshop.html>, April 2008.
- [Hiatt *et al.*, 2009] Laura M. Hiatt, Terry L. Zimmerman, Stephen F. Smith, and Reid Simmons. Strengthening schedules through uncertainty analysis. In *Proceedings of the Twenty-First International Joint Conference on Artificial Intelligence (IJCAI-09)*, 2009.
- [McKay *et al.*, 2000] K.N. McKay, T.E. Morton, P. Ramnath, and J. Wang. Aversion dynamics' scheduling when the system changes. *Journal of Scheduling*, 3(2), 2000.
- [Morris *et al.*, 2001] Paul Morris, Nicola Muscettola, and Thierry Vidal. Dynamic control of plans with temporal uncertainty. In *Proceedings of IJCAI*, pages 494–502, 2001.
- [Policella *et al.*, 2006] Nicola Policella, Amedeo Cesta, Angela Oddi, and Stephen F. Smith. From precedence constraint posting to partial order schedules. a CSP approach to robust scheduling. *AI Communications*, 20(3):163–180, 2006.
- [Smith *et al.*, 2007] Stephen F. Smith, Anthony Gallagher, Terry Zimmerman, Laura Barbulescu, and Zachary Rubinstein. Distributed management of flexible times schedules. In *Proceedings of AAMAS-07*, May 2007.