

Utility Estimation for Contingent Execution: Preliminary Report

Richard Washington* and John L. Bresina

NASA Ames Research Center, M/S 269-2
Moffett Field, CA 94035

{rwashington, jbresina}@arc.nasa.gov

Abstract

This paper presents a method for using expected utility distributions in the execution of flexible, contingent plans. A utility distribution maps the possible start times of an action to the expected utility of the plan suffix starting with that action. The contingent plan encodes a tree of possible courses of action and includes flexible temporal constraints and resource constraints. When execution reaches a branch point, the eligible option with the highest expected utility at that point in time is selected. The utility distributions make this selection sensitive to the runtime context, yet still efficient. Our approach uses predictions of action duration uncertainty as well as expectations of resource usage and availability to determine when an action can execute and with what probability. Execution windows and probabilities inevitably change as execution proceeds, but such changes do not invalidate the cached utility distributions; thus, dynamic updating of utility information is minimized.

Introduction

The work reported here is part of a research program to develop robust, autonomous planetary rovers (Washington, *et al.*, 1999). Traditionally, spacecraft have been controlled through a time-stamped sequence of commands (Mishkin, *et al.*, 1998). The rigidity of this approach presents particular problems for rovers: since rovers interact with their environment in complex and unpredictable ways and since the environment is unknown or poorly modeled, the rover's actions are highly uncertain. We have developed a temporally flexible, contingent planning language, which enables the specification of rover actions that can adapt to the changing execution situation. The plan language is called the *Contingent Rover Language* or *CRL* (Bresina, *et al.*, 1999). CRL allows a rich specification of preconditions, maintenance conditions, and end conditions for actions. These conditions can include absolute and relative temporal constraints, resource constraints (*e.g.*, power), as well as constraints on the rover's state.

*NASA Ames contractor with Caelum Research Corp.

A contingent plan is a tree of possible courses of action; when execution reaches a branch point, the rover's onboard executive selects the eligible option with the highest expected utility. If all the actions were time-stamped, then it would suffice to precompute the expected utility for each contingent option, using classical decision theory. However, because the actions in a CRL plan can start within a flexible temporal interval, the expected utilities of the contingent options depend on the time that the branch point is reached during execution. Hence, a single utility measure is insufficient, and we need to compute a utility distribution that maps possible action start times to the expected *plan-suffix utility*, *i.e.*, the expected utility of executing the plan suffix starting with that action.

Expected plan-suffix utility depends on when actions can execute and with what probability. The time over which an action executes and the probabilities of success and failure are affected by all the constraints in the action's conditions (pre-, maintain, and end), as well as by the inherent uncertainty in action durations. As plan execution proceeds, the temporal windows for plan actions narrow, resource availability can change, and rover state can change in unpredictable ways. Such changes affect the execution time and success probabilities and, thus, the expected utilities. Note, however, that even though temporal changes can affect the probabilities of when future actions will start, the plan-suffix utility distributions of these actions do not have to be recomputed because they are conditioned on start time. Although the use of utility distributions does reduce utility recomputations, it does not eliminate them; *e.g.*, changes in resource availability can require dynamic utility updates.

In contrast to classical decision-theoretic frameworks, the uncertainty arises from an interaction of action conditions and execution time, which is uncertain because of variations in action durations. Modeling this with decision-theoretic tools would require covering the spaces of possible action times and available re-

sources. Thus, a decision-theoretic planning approach that *a priori* considers all possible decision points and pre-compiles an optimal policy is not practical.

In this paper, we present an approach for estimating the expected plan-suffix utility distribution in order to make runtime decisions regarding the best course of action to follow within a flexible, contingent plan. Our method takes into account the impact of temporal and resource constraints on possible execution trajectories and associated probabilities by using predictions of action duration uncertainty and expectations of resource usage and availability.

Plan-Suffix Utility Distributions

The utility of a plan depends on the time that each action starts, when it can execute, and its constraints. In CRL, an action may be constrained to execute within an interval of time, specified either in relative or absolute terms. In a plan with this type of temporal flexibility, the exact moment that a future action will execute cannot, in general, be predicted. We use a probability density function (PDF) to represent the probability of an action starting (or executing, or ending) at a particular time. The focus of this paper is on the ability to estimate the expected utility of a sequence of actions by propagating these PDFs from action to action. The propagation uses the temporal and resource conditions of the action to restrict the action's execution times.

The plan-suffix utility of an action is a mapping from times to values: $u(S, t)$ is the utility of starting execution of a plan suffix S at time t . The terminal case is $u(\{\}, t) = 0$. For a plan $\{a, S\}$, denoting action a followed by the plan suffix S , there are two cases, depending on whether failure of a causes general plan failure or not. Let us denote $p_{success}(t'|a, t)$ as the probability of success of a at time t' given that it started at time t , $p_{failure}(t'|a, t)$ as the probability of a 's failure at time t' , and v_a as the fixed local reward for successfully executing action a . If the plan fails when a fails,

$$u(\{a, S\}, t) = \int_{-\infty}^{\infty} p_{success}(t'|a, t) \cdot v_a \cdot u(S, t') dt'$$

If the plan continues execution when a fails,

$$u(\{a, S\}, t) = \int_{-\infty}^{\infty} [p_{success}(t'|a, t) \cdot v_a + p_{failure}(t'|a, t)] \cdot u(S, t') dt'$$

In the case of a branch point b with possible suffixes $S_b = \{S_1, \dots, S_n\}$, the plan-suffix utility $u(\{b, S_b\}, t)$ is a function of the utilities of each possible suffix:

$$u(\{b, S_b\}, t) = \sum_{S_i \in S_b} \int_{-\infty}^{\infty} p_{select}(S_i, t) \cdot u(S_i, t) dt$$

where $p_{select}(S_i, t)$ is the probability of suffix S_i being selected at time t (0 if the eligibility condition is unsatisfied). This is an average of the individual suffix utilities, weighted by the selection probabilities.

Given a planning language with a rich set of temporal, resource, and state conditions, the functions $p_{success}$ and $p_{failure}$ do not allow closed-form calculation of the plan-suffix utilities. We solve this by discretizing time into bins; the value assigned to a bin approximates the integral over a subinterval. Calculations of the integrals above become summations. The choice of bin size introduces a tradeoff between accuracy and computation cost, which we examine in the section Empirical Results.

Although the utility calculation is defined with respect to an infinite time window, the plan start time, action durations, and action conditions restrict the possible times for action execution and for transitions between actions. In this work we model only temporal and resource conditions; the time bounds we compute may be larger than the real temporal bounds because of the unmodeled conditions.

The basic approach is to propagate the temporal bounds forward in time throughout the plan, producing the temporal bounds for action execution. Those temporal bounds serve as the ranges over which the utility calculations are performed. Outside of these ranges, the plan fails. A failed plan receives the local utility of the actions that succeeded and zero utility for the remainder; failure could be penalized through a simple extension.

The temporal bounds are calculated forward in time because the current time provides the fixed point that restricts relative temporal bounds. The utilities, on the other hand, are calculated backward in time from the end(s) of the plan. The utility estimates are conditioned on the time of transitioning to an action; since they are not dependent on preceding action time PDFs, they remain valid as plan execution advances, barring changes in resource availability.

In the following sections, we describe the elements of an action and present the procedures for propagating temporal bounds and utilities in more detail.

Anatomy of an Action

In the Contingent Rover Language, each action instance includes the following information:

Start conditions. Conditions that must be true for the action to start execution.

Wait-for conditions. A subset of the start conditions for which execution can be delayed to wait for them to become true (by default, unsatisfied start conditions fail the action). Temporal start conditions are

treated as wait-for conditions, and may be absolute or relative to the previous action's end time.

Maintain conditions. Conditions that must be true throughout action execution. Failure of a maintain condition results in action failure.

End conditions. Conditions that must be true at the end of action execution. Temporal end conditions may be absolute or relative to action start time.

Duration. Action duration expressed as an expectation with a mean and standard deviation of a Gaussian distribution. Our approach would work equally well with other models of action duration.

Resource consumption. The amount of resources that the action will consume. It is expressed as an expectation with a fixed value, because we currently assume that resource consumption for a given action is a fixed quantity with no uncertainty.

Continue-on-failure flag. An indication of whether a failure of the action aborts the plan or allows execution to continue to the next action.

Resource conditions considered here are threshold conditions; *i.e.*, they ensure that enough of a given resource exists for the action to execute. The resource profile is an expectation of resource availability over time, represented by a set of temporal intervals with associated resource levels. A resource condition is checked against the availability profile to determine the intervals over which the condition is satisfied.

Temporal Interval Propagation

Each temporal aspect of an action is represented as a set of temporal intervals, and we distinguish the following temporal aspects of an action.

Transition time. The time that the execution of the previous action terminates. This is not the same as start time, since the action's preconditions may delay its execution. The transition-time intervals are the set of possible times that the previous action will transition to this action.

Start time. The time that the action's preconditions are met and it executes. The start-time intervals are the set of possible times that the action will start.

End time. The time at which the action terminates. We distinguish between successful termination and failure, due to condition violation, and determine a set of end-succeed intervals and end-fail intervals.

Execution proceeds according to the following steps:

1. If the current time is already past absolute start bounds, fail this action.

2. Execution waits until all wait-for and lower-bound temporal conditions are true (but if upper-bound temporal conditions are violated at any time, the action fails).

3. The start conditions are checked, and the action fails if any are not true.

4. The action begins execution. If any maintenance conditions fail during execution, the action fails. If the temporal upper bound is exceeded, the action fails.

5. The action ends execution. The end conditions are checked, and the action fails if any are not true.

6. Execution transitions to the next action.

As mentioned earlier, action failure either fails the plan or simply transitions to the next action, as specified within the plan (the continue-on-failure flag).

Temporal bounds and utilities are propagated to reflect the execution steps. We illustrate the temporal interval propagation by demonstrating how the various conditions affect an arbitrary transition-time PDF. The interval propagation is done simply through computations on the bounds, but since the utility computations propagate PDFs, the general case demonstrates the basics underlying both calculations.

Transition time

The possible transition times of the plan's first action is when plan execution starts; typically, this is a single time point (*e.g.*, the set time that the rover "wakes up"). For all other actions, the transition time PDF is determined from the previous action's end time PDFs, as follows. If the previous action's continue-on-failure flag is true, then the possible action transition times are the union of the possible end-succeed times and the end-fail times from the previous action. On the other hand, if the previous action's continue-on-failure flag is false, then the action's transition times are identical to the previous action's end-succeed times.

Start time

Given the possible transition times and a model of resource availability, we determine the set of temporal intervals that describes the possible action start times, along with a set of temporal intervals during which the action will fail before execution begins.

Consider an action with absolute time bounds $[lb_{abs}, ub_{abs}]$ (default $[0, \infty]$) and relative temporal bounds $[lb_{rel}, ub_{rel}]$ (default $[0, \infty]$)¹. Consider also resource wait-for conditions R_{wait} and resource start conditions R_{start} . For a given resource availability profile, each resource condition r corresponds to a set of time intervals $I_{false}(r)$ for which the resource condition is not true. We define the set of *wait intervals*:

$$I^{wait} = [-\infty, lb_{abs}] \cup \bigcup_{r \in R_{wait}} I_{false}(r).$$

¹In practice, a finite planning horizon bounds the absolute and relative time bounds; it also bounds the probability reallocation for unmodeled wait-for conditions.

We define the set of *fail intervals*:

$$I^{fail} = [ub_{abs}, \infty) \cup \bigcup_{r \in R_{start}} I^{false}(r).$$

The following rules partition the space of time; they are used to identify the possible start times and the possible fail times, given the conditions. In the rules, t is a given transition time.

1. If $t > ub_{abs}$, then the action fails at time t .
2. Else, if $t + lb_{rel} > ub_{abs}$, then the action fails at time ub_{abs} .
3. Else, if $t + lb_{rel}$ is within a wait interval I_i^{wait} , and the upper bound of the wait interval ub_{wait} is such that $ub_{wait} - t > ub_{rel}$ or $ub_{wait} > ub_{abs}$, then the action fails at time $\min(t + ub_{rel}, ub_{abs})$.
4. Else, if $t + lb_{rel}$ is within a wait interval I_i^{wait} , and the upper bound of the wait interval ub_{wait} is such that $ub_{wait} - t \leq ub_{rel}$, then the action waits until time ub_{wait} . If ub_{wait} falls within a fail interval, then the action fails at time ub_{wait} . Otherwise the action starts at time ub_{wait} .
5. Else, if $t + lb_{rel}$ is not within a wait interval I_i^{wait} , and $t + lb_{rel}$ falls within a fail interval, then the action fails at time $t + lb_{rel}$.
6. Finally, if none of the preceding conditions hold, then the action starts at time $t + lb_{rel}$.

If all of the conditions could be accurately modeled, then a transition time would map to a single start time. However, as mentioned earlier, we currently model only temporal and resource conditions. The set of unmodeled conditions adds uncertainty about the time intervals over which the sets of conditions will be true. For start conditions, this adds a fixed probability of failure to every time point. For wait-for conditions, unsatisfied preconditions move probability mass later; to reflect this, we subtract a proportion α of the probability density at each time point and allocate it uniformly to each time later within the absolute bounds; after this, the rules above apply for the modeled conditions.

End time

Here we consider how end times are calculated for an action that has its start conditions true and has started execution. The successful end time of an action is determined by its start time, duration, maintenance conditions, and end conditions. Without maintenance or end conditions, the end time PDF is determined by convolving the start time and duration PDFs; for the bounds, each start time interval $[lb_{start}, ub_{start}]$ and duration interval $[lb_{dur}, ub_{dur}]$ yields an end time interval $[lb_{start} + lb_{dur}, ub_{start} + ub_{dur}]$.² All such intervals are unioned to yield the possible end times.

²To bound the duration interval, we truncate the normal distribution at ± 2 standard deviations and at 0 and then

Maintenance conditions restrict the possible end times by defining valid execution time intervals; if execution exits a valid interval, the action fails. End conditions further restrict the successful times; if execution ends when an end condition is not true, the action will fail. The temporal end upper bounds are treated as maintenance conditions so that action execution is bounded.

An action will succeed only if the following four conditions are met:

1. It successfully begins execution.
2. Its start time falls within a valid execution interval. If not, the action will fail at that start time.
3. Its duration is such that its end time falls within the same valid execution interval. If not, the action will fail at the end of this execution interval.
4. The end time falls within a valid end interval. If not, the action fails at the end time.

Utility Propagation

Utility propagation follows the same basic rules as temporal interval propagation in terms of the effects of conditions, but it is calculated during a sweep back from the terminal actions of the plan tree. A terminal action in the plan tree has an empty plan suffix of utility 0. The plan-suffix utility is conditioned on the start time of the action: we calculate the utility of an action and its successors given a particular transition time. The plan-suffix utility composed with a PDF of possible transition times to this action yields the expected utility of the plan suffix starting with this action over the time distribution given by the PDF. Caching the utility conditioned on start times allows an efficient means of choosing the highest utility eligible contingent option.

An action's plan-suffix utility for a given transition time is computed as follows. First, the transition time is propagated to a discrete start time PDF according to the above rules for start time propagation. Second, the convolution of the start time PDF and duration PDF is computed to produce the PDFs for successful end times and failed end times according to the above rules for end time propagation. Third, the success end time PDF is composed with the local value and the plan-suffix utility of the next action to produce the plan-suffix utility for the given transition time. If the action's continue-on-failure flag is set, the failure end time PDF is also composed with the plan-suffix utility of the next action and added to the utility computed from the end time.

normalize the remaining distribution.

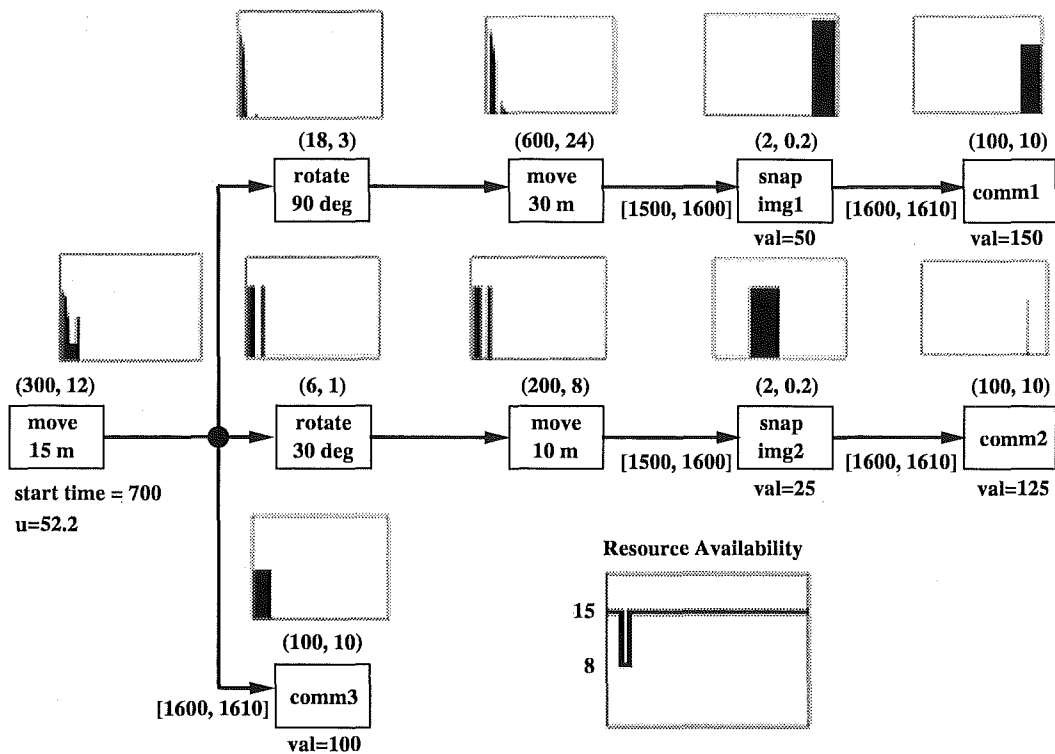


Figure 1: **Example contingent rover plan.** The (μ, σ) above an action indicate the duration's mean and standard deviation. Important start time constraints are shown in square brackets below the arrows. The local value, if nonzero, of an action is indicated below it; these values are assigned by the scientists. For a plan start time of 700, each action's plan-suffix utility distribution is plotted above it; all have x-range [955, 1605] and y-range [0, 210]. The leftmost plot is the utility of the branch point. The utility of the entire plan is 52.2. The resource availability profile has x-range [955, 1605] and a resource dip over [1000, 1025].

Empirical Results

To demonstrate our approach, we use a small plan example, which is shown in Figure 1. The plan consists of an initial traversal and then a branch point with the following three contingent options: (i) travel toward a farther, but more important science target, capture its image, and communicate the image and telemetry, (ii) travel toward a nearer, but less important science target, snap its image, and communicate the image and telemetry, or (iii) communicate telemetry.

The communication must start within the interval [1600, 1610]. If communication does not happen, then all data is lost; hence, it has a high local value (higher for more important data). Thus, the primary determinant of which option has the highest expected utility is whether there is enough time to execute the communication action. The duration uncertainty of the actions affects the probabilities of successfully completing each of the contingent options and, hence, affects the expected utility. The time that plan execution starts also affects these probabilities and utilities. For the case shown, the start time (700) falls at a time when

the first branch of the plan is likely to fail, as is clearly shown by the plan-suffix utility distributions in the figure. In addition, the power availability profile is such that it prevents motion over a small range of time; this is also reflected in the utility distributions.

The principal question that can be tested experimentally concerns the accuracy of the discrete propagation and the tradeoff of discrete bin size versus accuracy. To illustrate these experimentally, we use our example plan and a plan start time of 700 (as shown in Figure 1) and compare the utility of the entire plan when computed using bin sizes of 0.5, 1, 2, 5, 10, 20, 50, and 100; we also estimate the exact plan utility with a 100,000 trial Monte Carlo stochastic simulation. The results are shown in Figure 2. The results show increasing accuracy with decreasing bin size; the largest error is still less than 12%.

Concluding Remarks

In this paper, we presented *expected plan-suffix utility distributions*, described a method for estimating them within the context of flexible, contingent plans, and

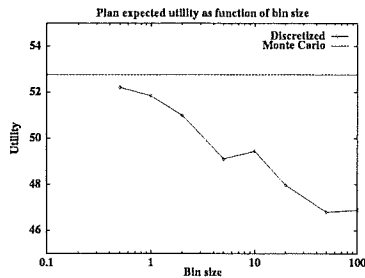


Figure 2: Tradeoff of accuracy and bin size. The reference line is the value reached through a Monte Carlo simulation. Note that the x-axis is log scale.

discussed their use for runtime decisions regarding the best course of action to take.

The approach presented in this paper attempts to minimize runtime recomputation of utility estimates. Narrowing the transition intervals of an action does not invalidate its utility distributions. Resource availability changes may affect the times over which an action's conditions are true and, thus, the probability distribution of successful execution. The plan-suffix utility of all actions before an affected action will need to be updated. Actions later than an affected action only need to be updated at newly enabled times.

In contrast to standard decision-theoretic frameworks (Pearl, 1988), uncertainty arises from an interaction of action conditions with an execution time of uncertain duration. Decision-theoretic tools would require covering the spaces of possible action times and available resources; thus, a decision-theoretic planning approach that considers all possible decision points and pre-compiles an optimal policy is not practical.

An earlier effort that propagated temporal PDFs over a plan is *Just-In-Case (JIC)* scheduling of Drummond, *et al.* (1994). The purpose was to calculate schedule break probabilities due to duration uncertainty. Unlike our rich set of action conditions, the only action constraint in the reported telescope scheduling application was a start time interval. JIC used the simplifying assumption that start time and duration PDFs were uniform distributions and that convolution produced a uniform distribution. Our discretized method is more statistically valid and could be used in JIC to increase the accuracy of its break predictions.

An alternative approach to utility estimation is to use Monte Carlo simulation onboard, choosing durations and eligible options according to their estimated probabilities. The advantage of simulation is that it is not subject to discretization errors. On the other hand, a large number of samples may be necessary to yield a good estimate of plan utility; furthermore, the length of the calculation is data-dependent (*e.g.*, to

reach a particular confidence level). We consider such an approach to be impractical for onboard use, given the computational limitations of a rover.

A number of issues are raised by this approach, and some remain for future work. The combination of plan-suffix utilities at branch points depends on the probability of choosing each sub-branch at each time. Given unmodeled conditions, this can only be estimated, but an interpretation of the conditions on each of the sub-branches can be performed to determine the expected probability of that sub-branch being eligible. If there are times for which more than one sub-branch is potentially eligible, then the resulting utility is some combination of the utility of each sub-branch at that time.

The use of discrete bins in calculating utility introduces error into the calculation; the probabilities and utilities of a precise time point are diffused over surrounding time points. As the chain of actions becomes longer, the inaccuracies grow. Smaller bin sizes minimize the error; however, the utility calculation is in the worst case $\mathcal{O}(n^3)$ for n bins. This tradeoff of accuracy versus computation time requires further study. Bin size could be scaled with the depth of the action in the plan, but this would require frequent recalculations as execution progressed through the plan.

Our approach can be extended by making more realistic modeling assumptions; *e.g.*, modeling uncertainty in resource consumption and modeling hardware failures. One possible next step is to introduce limited plan revision capabilities into the plan to handle cases where all possible plans are of low utility and are thus undesirable. Another extension would be to introduce additional sensing actions to disambiguate multiple eligible options with similar utility estimates.

References

- A. H. Mishkin, J. C. Morrison, T. T. Nguyen, H. W. Stone, B. K. Cooper, and B. H. Wilcox. 1998. Experiences with Operations and Autonomy of the Mars Pathfinder Microrover. In *Proc. IEEE Aero. Conf.*
- J. Bresina, K. Golden, D.E. Smith, and R. Washington. Increased flexibility and robustness of Mars rovers. In *Fifth Intl. Symposium on Artificial Intelligence, Robotics and Automation in Space*. Noordwijk, Netherlands. 1999.
- M. Drummond, J. Bresina, & K. Swanson. 1994. Just-in-case scheduling. In *Proc. 12th Natl. Conf. Art. Int.*
- R. Washington, K. Golden, J. Bresina, D.E. Smith, C. Anderson, and T. Smith. 1999. Autonomous Rovers for Mars Exploration. In *Proc. IEEE Aerospace Conf.*
- J. Pearl. 1988. *Probabilistic Reasoning in Intelligent Systems*. Morgan-Kaufmann.