

Probabilistic Reasoning for Robust Plan Execution

Steve R. Schaffer, Bradley J. Clement, Steve A. Chien

Abstract.

A planning system must reason about the uncertainty of continuous variables in order to accurately project the possible system state over time. Prior approaches to planning under uncertainty reason about discrete possible outcomes but there has been little attention given to continuous possible outcomes. A method is devised for directly reasoning about the uncertainty in continuous activity duration and resource usage for planning problems. By representing random variables as parametric distributions, computing projected system state can be simplified in some cases. Common approximation and novel methods are compared for over-constrained and lightly constrained domains. The system compares a few common approximation methods for an iterative repair planner. Results show improvements in robustness over the conventional non-probabilistic representation by reducing the number of conflicts witnessed by execution. The improvement is more significant for larger problems and problems with higher resource subscription levels but diminishes as the system is allowed to accept higher risk levels.

1 Introduction

Planning systems that reason about real world events must eventually deal with the inherent uncertainty of any real world mechanism. For example, actions may take longer or consume more resources than predicted. Even if it were possible to model every variable that affected a planned set of actions, doing so is impractical for realistically sized domains. Further, practical modeling abstractions themselves also introduce uncertainty into reasoning about a system.

The way a planning system deals with uncertainty in its actions and observations is critical to how well the system is able to perform in the real world. Clearly, systems that effectively reason about uncertainty can better avoid generating plans that are likely to violate execution constraints. But effective use of uncertainty can also improve the long-term efficiency of a plan by balancing acceptable risk levels against the inefficiencies incurred to avoid those risks. Finally, knowledge of uncertainty allows the system to better assess and report on the most risky plan segments.

One historical approach to dealing with uncertainty is to assume no uncertainty at the level of planning abstraction.

To be used in a real world system, such systems are often augmented with some replanning mechanism for when predictions do not match results. One step further is to depend on an execution system to handle any variations in plan execution. Effectively, the planner itself is abstracted from any knowledge that the real world doesn't behave as predicted.

Other approaches have relied on introducing plan branching points where one of several plans may be executed depending on the actually observed world state during execution (Dearden *et al.* 2002). Another tactic is the insertion of slack padding into the schedule to absorb execution variations (Davenport *et al.* 2001).

This paper outlines one possible approach for directly reasoning about the uncertainty in action timing and resource consumption. Parametric probability distributions for these parameters are specified by a user-supplied model. The distributions are then combined during planning to determine the net probability distribution of a resource at any time point, which in turn may be integrated to yield the probability of violating any execution constraints on the resource. The key idea is to use this "probability of conflict" to score potential plans and to drive the planner's search toward low-risk actions. An output plan provides a balance between the user's risk aversion and other measures of plan optimality.

The present work deals only with durations and resource usages that can be modeled as normally distributed random variables, though the techniques are more widely applicable. To gauge the effectiveness of our probabilistic system, batch-generated plans are executed in a stochastic simulator. A comparative evaluation of our technique versus some common probabilistic approximations is provided along with an analysis of its applicability to different kinds of planning problems.

2 Approach

Planning effort is directed to repairing areas of a plan that have unacceptable levels of risk, as determined by a user-specified risk tolerance on each resource as a function of time. Risk for any one timeline segment is assessed by computing the probability that the sum of all activity reservations

(i.e. condition and effects) that potentially overlap the segment would exceed one of the modeled system resource limits. This probability of resource constraint conflict is readily derived if the resources' net probability density functions are available. Our approach for maintaining each net resource distribution is to combine individual activity resource reservations parametrically.

Each activity in the plan is considered to make uncertain resource reservations that follow a known distribution. Further, each activity can also have a duration that is similarly uncertain. (For simplicity, all activities are considered to have certain start times - an assumption that holds for directly commanded actions, but may not apply for exogenous events.) In this paper, we only consider reservations and durations that are normally distributed random variables, though in practice other parametric distributions can also be used. The parametric representation for a normal distribution is very compact, requiring only the distribution mean (μ) and standard deviation (σ). In comparison, a particle filter (Gordon *et al.* 1993), which requires a value and weight for each sample taken from a distribution. Conveniently, certain values can be also represented as normal random variables with a given μ but $\sigma = 0$.

In the case of activities that make persistent reservations on a resource, the net resource distribution for a timeline segment is the sum of all current and preceding normally distributed reservations. Fortunately, the sum Σ of i independent normal reservations n is itself a normal, with parameters $\mu_\Sigma = \sum_i \mu_{n_i}$ and $\sigma_\Sigma = \sqrt{\sum_i \sigma_{n_i}^2}$. Notice that the uncertainty of the sum is greater than any single component, indicating that resultant uncertainty grows with the number of interacting reservations.

For actions that only have a transient reservation during their duration, the same method is limited to those reservations that are concurrent. In the simple case that each concurrent activity has a certain duration, the net resource distribution is computed by adding each local reservation. In the more complex case of concurrent activities with uncertain duration, the net resource distribution itself becomes a function of time.

For an activity A with start time t_s and duration $d = d_\mu \pm d_\sigma$, consider $P[A](t)$ to be the probability that action A is executing at time t (see figure 1). As d is normally distributed, the end time t_e is also a normal, and we can express $P[A](t)$ as:

$$P[A](t) = \begin{cases} 0 & t < t_{sA} \\ 1 - \Phi_{\mu_{t_e}, \sigma_{t_e}}(t) & t \geq t_{sA} \end{cases}$$

where $\Phi_{\mu, \sigma}(x)$ is the cumulative distribution function for a normal with mean μ and standard deviation σ . Strictly, normal distributions may yield negative samples, so we must truncate only the duration distributions to $[0, \mu + 3\sigma]$.

Each of A 's resource reservations must reflect the gradual tailing off of the activity's probability. If A makes a reser-

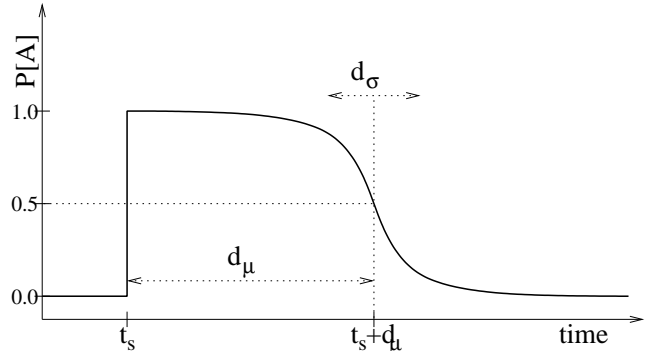


Figure 1: Probability of activity A with normally distributed duration d continuing after its start time t_s .

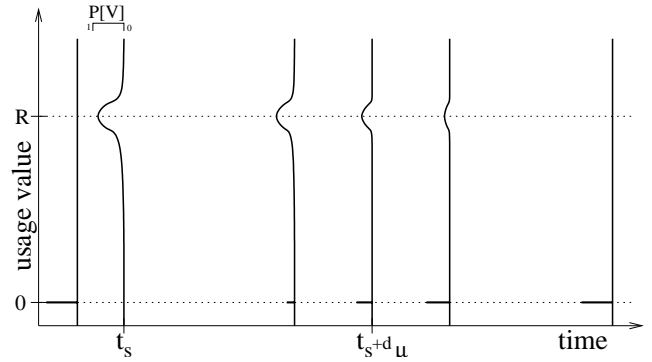


Figure 2: Transient resource usage distribution for activity A of uncertain duration, showing peaks at R when the activity is likely and 0 when the activity is unlikely.

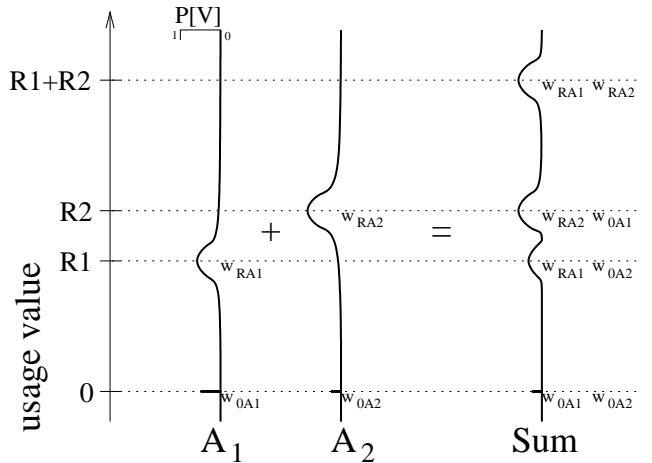


Figure 3: Computing the sum of two bimodal resource usage distributions results in a multi-modal distribution. Each resultant peak weight is the product of the component weights.

vation R when active, its effective reservation becomes a function of time, $R(t)$, as in figure 2). This distribution is bimodal: one peak at zero resource usage represents that the activity is not in effect (weighted $w_0 = 1 - P[A](t)$), and the R peak represents A 's transient reservation (weighted $w_{R_A} = P[A](t)$). The peak at zero is a scaled Dirac delta function: it integrates to w_0 , but has infinitesimal width.

The time-sensitive reservations seen in figure 2 are no longer simple normals, so the net resource distribution must also be more complex. In fact, the sum of $|A_i|$ different bimodal reservations results in a multi-modal distribution with $O(2^{|A_i|})$ distinct peaks: one for each combination of activities that could be in effect (see figure 3).

With the net resource distribution $PDF_{R(x,t)}$ in hand, Computing the probability of violating a system resource constraint during a timeline unit becomes a simple integral. For a timeline unit T with a random variable resource level R and constraints that $R \in [l_{min}, l_{max}]$, then the probability of violation is given by:

$$\begin{aligned} P[V_T](t) &= P[R(t) < l_{min}] + P[R(t) > l_{max}] \\ &= 1 - P[l_{min} \leq R(t) \leq l_{max}] \\ &= 1 - (CDF_R(l_{max}, t) - CDF_R(l_{min}, t)) \\ &= 1 - \int_{l_{min}}^{l_{max}} PDF_R(x, t) dx \end{aligned}$$

Fortunately, this integral for normal distributions amounts to a pair of constant time calls to the standard error function, $erf(x)$. For multi-modal distributions, the integral must be computed for each component normal distribution and combined according to the weight of the peaks.

In the end, $P[V_T](t)$ may still be a function of time. In this event, we consider the reportable $P[V_T]$ for a timeline unit to be the maximum instantaneous probability of violation anytime during the unit. (Such an assumption works for systems where each random value is chosen once and not sampled repeatedly from the distribution.) To avoid checking all $t \in T$, we currently only check a constant number of critical times from T , including the endpoints.

The probability of constraint violation for each timeline unit is compared to the user-specified acceptable risk level, and any violations that are more likely than the risk tolerance are flagged as plan conflicts. A planning algorithm can use the tolerance to help decide whether and where to add, order, move, or remove an activity.

2.1 Comparison Approximations

Several approximation methods were implemented for comparison against the fully probabilistic system described above. Each fits within the same planning and heuristic framework, but maintains the net resource distributions differently.

Means Only: One very natural and easy approximation method is to disregard all uncertainty in the distributions and

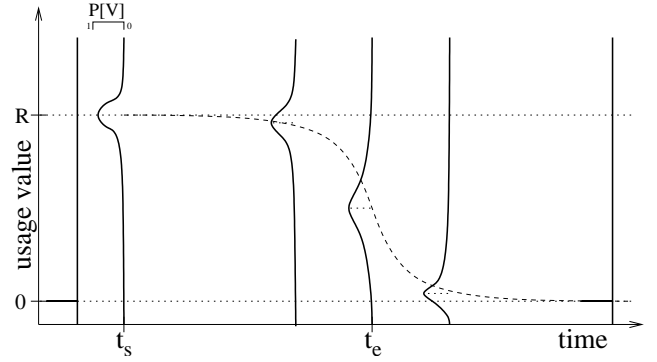


Figure 4: Single peak approximation for resource usage distribution in place of multi-modal distribution (figure 2).

instead consider only the one value of maximum likelihood. For normal distributions, this is the mean. Because durations are also estimated by the mean, there are no multi-modal distributions, and resource values can be tracked by a single value for each unit. The Means Only approximation is equivalent to default configuration of our planner wherein everything behaves as expected.

Pessimistic: Similar to the Means Only approximation, the Pessimistic approximation only tracks one value from each distribution. Instead of choosing the value of maximum likelihood, however, it chooses the “worst case” value. For a normal distribution, our pessimistic system tracks only the value $\mu + 2\sigma$ (or $\mu - 2\sigma$), and considers that to be the actual resource reservation. The choice of which direction constitutes the worst case is inherently domain dependent and must be specified.

Single Peak: A possible limiting factor of the Fully Probabilistic system described above is the $O(2^{|A|})$ peaks that emerge when summing reservations from a set of activities A , each with uncertain duration. Rather than track each of the peaks, the Single Peak approximation uses a single normal distribution as in figure 4 (compare with figure 2). This approximation forfeits accurate representation in favor of much improved time complexity. In fact, the Single Peak approximation is optimistic since it underestimates reservations.

Chebyshev Bound: The Chebyshev Bound approximation is similar to the Single Peak approximation in that both eliminate the multi-modal distributions that arise from uncertain duration. However, the Chebyshev Bound uses a more rigorous mathematical foundation for its approximation: for any random variable R , no matter the distribution, the probability of receiving a sample further than l from the distribution mean μ is given by the single-tailed version of Chebyshev’s inequality:

$$P[R - \mu \geq l] \leq \frac{\sigma^2}{\sigma^2 + l^2}$$

Because the Chebyshev Bound assumes so little about a

distribution, it is necessarily pessimistic. Like the Single Peak, Chebyshev tracks only a single mean and standard deviation, and the sum of two approximated values is taken to have the worst case standard deviation of $\sigma_{\Sigma} = \sum_i \sigma_{ni}^2$. We apply the one-sided Chebyshev inequality to the net mean and standard deviation, and report the resulting upper bound on violation probability as *the* violation probability.

3 Results

The Full Probabilistic system was evaluated against each of the comparison algorithms in two disparate planning domains. The first domain is an abstract testbed, and the second is a much more complex orbiting spacecraft domain.

For each domain, a random problem generator provided the initial schedule for the planner to repair. A iterative optimization planner was then run for a fixed number of iterations on the seed plan. The planner was augmented to use each of the full probabilistic and approximation algorithms, and an output plan was saved for each. The saved plans were then executed on a stochastic simulator that reported the number of resource constraint violations that occurred. Notably, no replanning was allowed as information became available during simulation. It would be possible to augment our experiments with more elaborate execution models (flexible time points, replanning, etc), but such was not investigated in the present work.

3.1 Abstract Domain

The abstract testbed domain has only a single resource and a series of activities that may consume or replenish that resource. The model was run with both permanent and transient resource reservations, and with different levels of reservation uncertainty. A valid solution existed for every generated problem (that is, there exists a known plan that can always execute with zero resource errors).

A comparison of the simulation error means for each approximation method is show in figure 5. As expected, the Means Only approximation stacked activities until the resource value was very close to its limit. This resulted in simulation errors when the simulated values exceeded the mean. The Pessimistic approximation only fared slightly better, likely due to a modeling deficiency: a simulation error occurs when a resource exceeds its limit or falls below zero. After a sequence of several overestimated consumers, the Pessimistic approximation replenished those reservations with twice as many underestimated replenishers. This causes the resource to fall well below zero, and an error is reported. In real systems, many resources only have one-sided constraints: it might not be an error to fall below zero power usage.

The Full Probabilistic system fared the best, consistently achieving nearly zero errors in each domain. It added an appropriate amount of both resource and schedule slack to

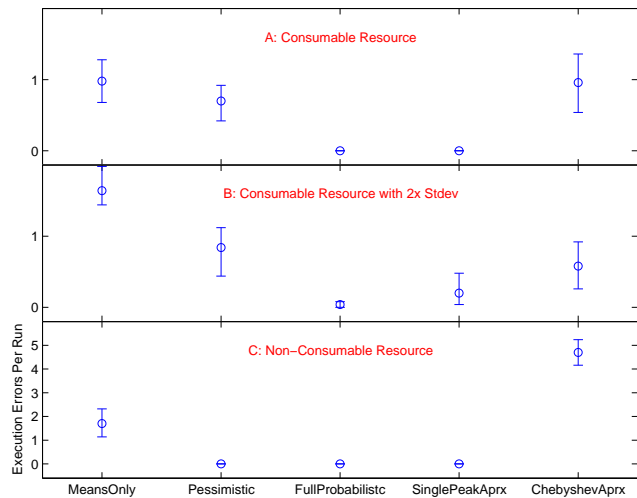


Figure 5: Execution error means for the three abstract domain variations. The 99% confidence of the mean is shown as an error bar.

accommodate the specified risk tolerance of 95%. The Single Bump approximation also performed well, only having difficulty when the resource uncertainty was doubled in (B). Notably, The Chebyshev approximation did not meet expectations: it turned out to be so very pessimistic in its distribution estimation that it failed to find good solutions, floundering with imagined conflicts.

The price of using the Fully Probabilistic system is of course computation time. For problems in which duration was not uncertain, the Fully Probabilistic system was about 10 times slower (unoptimized) than non probabilistic approaches. When duration was made uncertain, however, a vast difference appeared. Notably, the Single Peak approximation was almost 100 times faster than Full Probabilistic, on par with the non-probabilistic approaches. Systems where computational time is at a premium would likely fare well to adopt a simple Single Peak approximation and instead leave the Full Probabilistic approach for systems where execution errors are extremely high cost.

3.2 Complex Orbiter Domain

The second domain is a more realistic mock up of an orbiting spacecraft model. The probe is tasked with acquiring images during target visibility windows, processing those images in RAM, recording them to disk, and later down-linking them to a ground station. The probe has to reason about 10 resources, and has 10 different activities to complete its goals. Each activity makes reservations on multiple resource timelines. In this domain, the random problem generator does not guarantee that its problems will always have a completely valid solution (that is, the problems could be over-constrained since the planner is forbidden to shed goals).

As before, the Fully Probabilistic system achieves statisti-

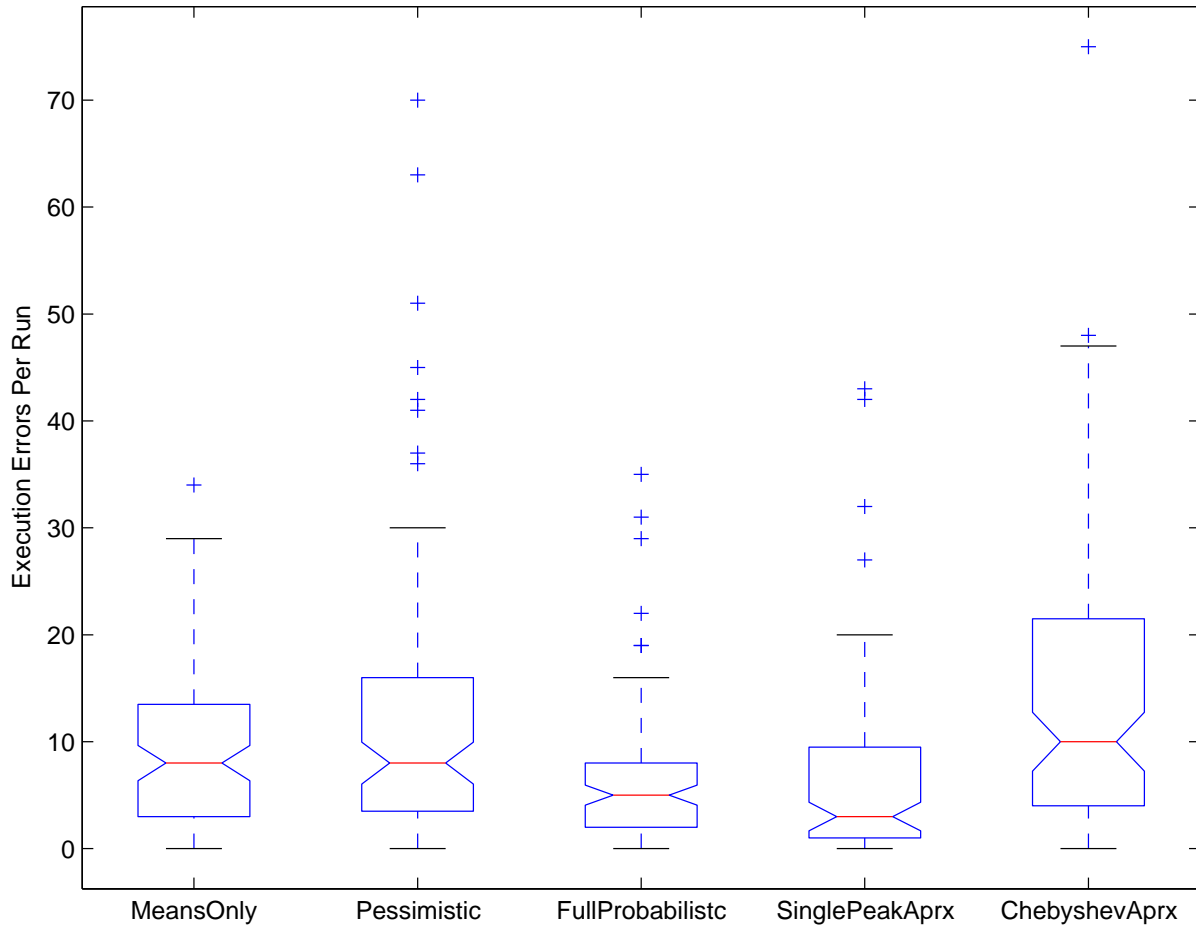


Figure 6: Execution error distribution for each reasoning system. The box plot shows the median as a horizontal line, a 95% confidence of the median as a notch, and the interquartile range as a box. The whiskers extend to encompass 1.5 more interquartile ranges, and outliers are plotted beyond that.

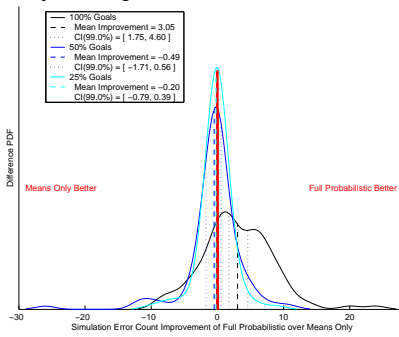


Figure 7: Execution error improvement distribution for problems of different goal densities. (The improvement is measured as the per-problem difference in errors.)

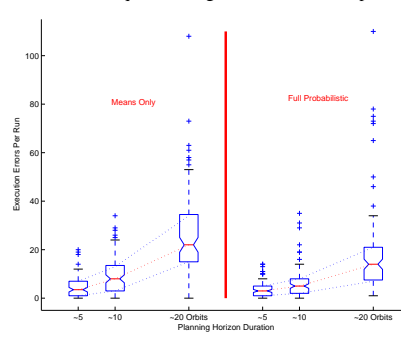


Figure 8: Execution Error Distribution for Problems of Different Sizes

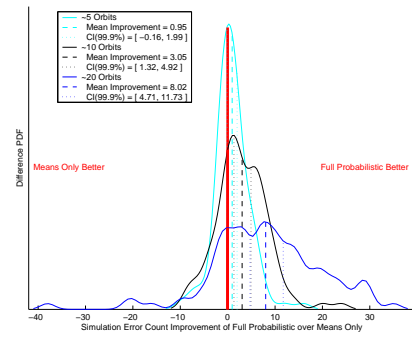


Figure 9: Execution Error Improvement Distribution for Problems of Different Sizes

cally significantly fewer simulation errors than either of the non-probabilistic systems, and generates plans on par with the Single Peak approximation's. The box plot in figure 6 conveniently shows a comparison of the error counts for each system. On a per-problem basis, the Full Probabilistic system had a mean 3.05 fewer simulation errors, with a 99.9% confidence interval of [1.35, 4.90]. The Pessimistic approximation still suffers from the double resource bound problem noted for the abstract model, but still achieves performance comparable to the Means Only approach. The overly pessimistic Chebyshev system still fares worse than the Fully Probabilistic system, but is not statistically significantly worse than either of the non-probabilistic systems. Notably, the Single Peak approximation achieves an error rate that is comparable to - perhaps even better than (confidence of 85%) - the Fully Probabilistic system. This is likely an artifact of our domain, in which activities seldom have tails that stack up into large multi-modal distributions.

Various parameters of the system were changed to evaluate the relative sensitivity of each approach. One such parameter is the user-specified risk tolerance. As expected, the payoff (in terms of reduced simulation errors) for using the Fully Probabilistic over a Means Only approach diminishes as the risk tolerance is increased. At a risk tolerance of 5%, they are distinct with 100% confidence, but even at 10% risk tolerance the statistical significance has dropped to 80%. At a risk tolerance of 50%, the Full Probabilistic system becomes essentially algorithmically equivalent to the Means Only system.

The difficulty of the problem also plays an important role in determining the Full Probabilistic system's dominance. As problem difficulty (measured as number of goals required) decreases, the Means Only approach gains on and eventually overtakes the Full Probabilistic approach in terms of simulation errors. Figure 7 shows the relevant confidence intervals.

Perhaps the most important change is that due to overall problem size. Figure 8 shows that both the probabilistic and non-probabilistic systems suffer a roughly exponential growth in simulation errors as a function of problem size. However, the slope of the Full Probabilistic system's function is significantly lower than that for Means Only. This indicates that the difference in simulation error counts will probably grow roughly exponentially as well. Figure 9 demonstrates this fact more clearly by showing the per-problem improvement distribution. At large problem sizes, the Fully Probabilistic system vastly dominates the Means Only approach, while at small problem sizes, there is hardly any difference.

4 Related Work

The idea of planning with uncertainty is not new. There are a variety of systems that incorporate probabilistic reasoning to improve plan robustness.

One of the most natural mechanisms for dealing with uncertainty in planning is with Markov Decision Processes, as is used in systems like PGraphplan from (Blum and Langford 1999). While MDP-based systems can leverage Markovian state independence to improve search strategy, one of their critical disadvantages is the lack of a simple metric resource mechanism within such a state representation.

More traditional planning systems, such as STRIPS, have been augmented with uncertainty reasoning mechanisms as in the CBURIDAN system from (Draper *et al.* 1994). Such systems generate conformant plans with a specified probability of achieving a goal state regardless of random actions taken. In this way, they are similar to our system of handling unobservable state and uncertain actions. However, like most STRIPS systems, resource constraints must be represented as propositions.

Some planning systems such as CNLP (Peot and Smith 1992) generate contingent plans where the actions to be taken are determined by observing the execution-time state of the world. Though each of the contingencies are determined in advance, this still leads to more flexible plans. Our system does not handle execution-time concerns, though it is possible that it could be extended to either generate contingent branches or facilitate dynamic re-planning.

5 Conclusions

We have described an approach for directly dealing with plan uncertainty by collecting and merging the probability distributions from action duration and resource usage. The essential idea is that by maintaining such merged distributions, a planning system can ask specific questions about the risk of violating constraints at any time. Being able to ask such questions allows the planner to better balance its risk posture against its desire to achieve goals.

We have shown that augmenting a planner with such a probabilistic reasoning system allows for plans with execution-time quality superior to that which can be obtained without directly considering uncertainty. Though the underlying structure of the planner's decisions are not changed, the more robust risk assessment afforded by a probabilistic system allow the planner to focus its decisions on the most probable errors. As problem size increases or as resources become more saturated with subscriptions, such focus becomes more important to finding plans that perform well on execution.

The fully probabilistic system makes its gains using a $O(2^n)$ algorithm, but we have also shown that a simple approximation technique that still tracks distributions can achieve comparable (and sometimes superior) results with only a $O(n)$ algorithm.

The techniques we have demonstrated are applicable to most planning problems that satisfy a few constraints. First the resource and duration distributions of actions must be known. Second, the system must have a relatively high

risk averseness for the probabilistic system to make a difference. In the current implementation, we have not handled many desirable planner capabilities such as direct temporal constraints or discrete state resources. We believe the techniques are still applicable for problems with such characteristics, albeit with some modification. Probabilistic reasoning is especially suited to problems of large size and high cost of failure.

References

- Avrim Blum and John Langford. Probabilistic planning in the graphplan framework. In *European Conference on Planning*, 1999.
- A.J. Davenport, C. Gefflot, and J.C. Beck. Slack-based techniques for robust schedules. In *Proceedings of the Sixth European Conference on Planning*, 2001.
- R. Dearden, N. Meuleau, S. Ramakrishnan, D. Smith, and R. Washington. Contingency planning for planetary rovers. In *Third International NASA Workshop on Planning and Scheduling for Space*, 2002.
- Denise Draper, Steve Hanks, and Daniel Weld. Probabilistic planning with information gathering and contingent execution. In *Proceedings of the Second International Conference on AI Planning Systems*, 1994.
- N. Gordon, D. Salmond, and A. Smith. Novel approach to nonlinear/non gaussian bayesian state estimation, 1993.
- M. Peot and D. Smith. Conditional nonlinear planning. In *Proceedings of the First International Conference on AI Planning Systems*, 1992.