# Exploration of the Robustness of Plans

**M. Fox** and **R. Howey** and **D. Long**
Department of Computer and Information Sciences
University of Strathclyde, Glasgow, UK

**Abstract**

This paper considers the problem of stochastic robustness testing for plans. Although plan generation systems might be proven sound the resulting plans are valid only with respect to the abstract domain model. It is well-understood that unforseen execution-time variations, both in the effects of actions and in the times at which they occur, can result in a valid plan failing to execute correctly. Other authors have investigated the stochastic validity of plans with non-deterministic action outcomes. In this paper we focus on the uncertainty that arises as a result of inaccuracies in the measurement of time and other numeric quantities. We describe a probing strategy that produces a stochastic estimate of the robustness of a temporal plan. This strategy is based on Gupta, Henzinger and Jagadeesan's (Gupta, Henzinger, & Jagadeesan 1997) notion of the "fuzzy" robustness of traces through timed hybrid automata.

## 1 Introduction

Classical planning has traditionally been concerned with construction of plans as either sequences of actions, or as partially ordered sets of actions. Researchers have explored beyond the constraints of classical planning and PDDL2.1 (Fox & Long 2003) represents a formalisation of the representation of temporal planning domains, in which plans become collections of time-stamped *durative* actions. We have shown a close relationship between PDDL2.1 and models of real-time systems based on timed hybrid automata (Fox & Long 2002). An important limitation of PDDL2.1 is that it is concerned entirely with deterministic domains in which no uncertainty is captured. Others have considered the consequences of extending PDDL2.1 to allow actions to have non-deterministic effects (Younes & Littman 2004), but we are concerned with a different form of uncertainty. In this paper we argue that there is an important reason to relax certainty about precise execution times of actions (as others have also argued — for example, see (Muscettola 1994)) even if one adopts a deterministic model of actions. We then proceed to explore the relationship between temporal uncertainty and work in timed hybrid automata (Gupta, Henzinger, & Jagadeesan 1997). We discuss the work we have done in extending our plan validation system to handle plan validation in the face of temporal uncertainty, including the implications of temporal uncertainty on plan correctness. We conclude with a discussion of the further problems of managing metric uncertainty and our progress in handling them.

## 2 Temporal Uncertainty in Planning

Although the introduction of metric time into planning makes it possible to represent and reason about far more realistic domains than with classical planning models, it introduces new problems in the relationship between planning and execution. Unlike the classical model, in which time is measured only in a relative sense, in the ordering of actions, once one has metric time, with actions assigned precise execution times, it is possible for the correctness of a plan to rely on the precise synchronisation of actions as they are performed by the executive. This is unreasonable, since no executive, even under the control of highly accurate microcontrollers, can achieve arbitrary levels of accuracy in the synchronisation of actions. This problem is only compounded when one considers that in translating plans into actions, it is inevitable that the abstractions in the domain model will fail to match precisely the reality of the world.

In the planning literature, this problem has been handled by the introduction of *temporal flexibility* in which intervals of uncertainty surround times of execution (Muscettola 1994; Vidal & Ghallab 1996). This is an attractive solution, although there has been some ambiguity about the precise semantics of the intervals: it is not always clear whether the interval indicates freedom in the choice of an executive of precisely when to execute an action or whether it indicates uncontrollable uncertainty about precisely when an action will execute. This matters a great deal, since the former intervals may be subjected to constraints to reduce their size, while the latter are presumably outside the control of the executive. Determining the dynamic controllability of a set of temporal constraints has been explored and efficient algorithms have been proposed (Morris, Muscettola, & Vidal 2001).

## 3 Robust Automata

In (Gupta, Henzinger, & Jagadeesan 1997), Gupta *et al.* also identify the difficulties that arise when trajectories through hybrid automata are interpreted as defining the timing of events with arbitrary precision. Again, the problem that is discussed is that physical interpretations of the execution of the trajectories rely on executives that cannot meet the demand for arbitrary precision. The authors observe that a trajectory in a hybrid automaton can be technically valid, according to the formal definition of validity of a trace, but can pass arbitrarily close to trajectories that are *in*valid. In such situations, the theoretical validity of the trace is of little practical value if a physical system cannot achieve the precision of execution that would avoid the invalid trajectories. The solution to the problem proposed by the authors is to identify *robust traces*. A trajectory defines a robust trace, $\tau$, through a hybrid automaton if there is a dense subset of the trajectories lying within some open tube around $\tau$ that contains only acceptable traces. The authors define various alternative metrics that can be used in determining the open tube around a trajectory and also indicate that others could be considered. Amongst these is the metric defining the distance between two traces to be the maximum of the distances between pairs of corresponding events in the two traces. We will call this metric the *max*-metric.

Although the definition of robust acceptance is a useful and intuitively appealing one, the authors do not offer any proposals for how a trajectory might be tested for this property in practice. The work described here proposes a practical strategy for the stochastic determination of plan validity based on the theoretical foundations established by Gupta *et al* (Gupta, Henzinger, & Jagadeesan 1997). In considering robustness on a stochastic basic, we are forced to consider the distribution of the trajectories that might be pursued, around the original planned trajectory. This is in contrast to the work of Gupta *et al*, which, by requiring that a dense subset of trajectories should be valid, is unconcerned with how unlikely are the possible failing trajectories around the original planned trajectory.

## 4 Robust Plan Validation

We have developed a system based on our plan validation tool, VAL (Howey, Long, & Fox 2004), which allows us to test the robustness of plans. The approach we adopt is to *probe* the plan space in the tube around the plan to be tested, using Gupta *et al.*'s *max*-metric to determine the tube we sample. The samples are identified by introducing random perturbations into the timings of the execution points of individual actions. We call this *juddering* the plan. Each such perturbation determines a new plan that can be tested using the precise deterministic testing implemented in VAL. We perform a large number of tests (a configurable value, defaulting to 1000) and then measure the proportion of successful plans. In order for the plan to be robust in an analogous way to the robust trajectories of Gupta *et al.*, the successful plans in the plan space we probe should form a dense subset. This cannot be tested empirically, so instead we report the proportion under the assumption that a plan can be considered robust if a sufficiently high percentage of the plans in the tube are valid. Although we use the *max*-metric to define the tube in which we sample, the samples are selected by applying an approximately normal distribution in generating perturbations of the times of the actions. This use of probing plan space has also been adapted to support planning under uncertainty in the work of Younes (Younes 2004). In that work, the probing allows exploration of the space generated by non-deterministic effects of actions, rather than of the space of plans in the tube around a specific plan, so the author explores a rather different direction to the one explored here.

There are some interesting questions raised in applying the probing strategy we have described. The time points that are relevant to a plan include both the times of execution of actions and also the times at which durative actions complete execution. In plans for domains that include exogenous events (as defined in PDDL+ (Fox & Long 2002)), the timing of events and the timing of actions could both be perturbed. However, we consider that the perturbations represent the inability of an executive to apply arbitrary precision in determining when to execute actions. In contrast, events model reaction of the world to the actions of the executive, and their timing is not subject to the constraints of physical limitations of an executive. For example, the event of a ball bouncing, after the executive executes the action of releasing it, will occur at a certain time after the release action without any need for a conscious reaction. One might argue that there will be slight variations in the time of flight of the ball, caused by slight variations in the air pressure, in the level of the surface the ball strikes and so on. We consider that these fluctuations are at orders of magnitude less than the accuracy of timing for most feasible executives, so they can be ignored.

## 5 Varying the Timestamps of Actions

In the family of languages based on PDDL the representation of a plan is as a list of timestamped actions. However when the plan is executed in a real world situation it is unlikely that the actions within it will be executed at *exactly* these times. Therefore we consider the possibility that these timestamps are not fixed when validating the plan, and use our probing strategy to investigate by how much the timestamps may be displaced. When a juddered plan is executed each action is executed at a time that is slightly different from the time in the original plan. Juddering ensures that, on each execution of the plan, the times of the actions will be (independently) different and we can identify the robustness of the original plan with respect to the times at which the actions are specified to occur. This approach introduces just enough temporal flexibility into the plan to guarantee a desired level of confidence in its robustness.

For each action, $a$, at time $t_a$, we execute the action in the interval $[t_a - \delta, t_a + \delta]$ for some $\delta > 0$. The chosen times of execution are random and follow a normal distribution about $t_a$. The exact nature of how the action timestamps are chosen in this interval is independent of the investigation of plan robustness. In our initial experiments into plan robustness we have used both uniformly distributed times and approximately normally distributed times within the intervals.

If a plan is not robust then it would be very useful to know where the plan is most likely to fail. This is also a consideration we are investigating. When a plan is not robust VAL reports where and when a plan is failing. See section 7 for some examples.

### 5.1 $\varepsilon$ Separation and Robust Plans

Previously, as defined in PDDL2.1, see (Fox & Long 2003), it was required that actions must be separated by a minimum value, namely $\varepsilon$, or the *tolerance value*. This was a solution to the problem of actions being so close together that the executive of the plan may not be reliable enough to ensure that these actions are executed in the correct order. Certain orderings may invalidate the plan, so ensuring that the end points of interfering actions do not coincide is very important. The solution adopted in the semantics of PDDL2.1 was the following: if two actions are within this tolerance value then they are assumed to be executed at the same time. To check that this results in a valid plan the actions are then checked, using VAL, to ensure that they are pairwise non-mutex at the coinciding end points. However there is a slight difficulty: suppose that three actions are timestamped $t_1$, $t_2$ and $t_3$ such that $t_1 < t_2 < t_3$, $t_2 - t_1 < \varepsilon$, $t_3 - t_2 < \varepsilon$ and $t_3 - t_1 > \varepsilon$ then it is unclear how to handle the interactions between the actions. Currently in VAL the first two actions are executed together and the third action escapes any mutex checks with the second action, which is clearly unsatisfactory.

With the newly proposed approach of executing many plans with varied timestamps there is no need consider any such $\varepsilon$ separation. When the timestamps of coinciding actions are juddered it can be determined whether the possible reordering of actions that occurs as a consequence invalidates the plan or not. If juddering the actions invalidates the plan then the separation between them should be increased. The size of the gap between actions will depend on how robust the plan is required to be.

The $\varepsilon$ separation approach for ensuring the robustness of a plan is inadequate when a plan contains *continuous effects*. Suppose we have a plan where all actions are separated by at least $\varepsilon$, so that when the plan is executed the actions cannot switch their order of execution. This does not guarantee the robustness of the plan. On executing the plan the time at which each action executes may differ by up to $\frac{\varepsilon}{2}$. These small changes may in turn affect the continuous effects, which may be very sensitive to the times at which actions are executed, since their effects may interact with one another. The change in values of continuous effects changes the values of PNEs for given times which may, of course, invalidate preconditions, invariants and the plan itself. The new approach of varying the timestamps of actions takes this complication into account. In fact, since continuous effects may be arbitrarily complex this is the only feasible way to ensure plans with continuous effects are robust.

### 5.2 Mutex Conditions and Robust Plans

The semantics of PDDL2.1(Fox & Long 2003) relies on mutex-checking to ensure that two actions that are executed at the same time are non-interfering so that the order of in which actions are actually applied does not change the outcome. However, with our approach of executing many plans with varied timestamps we are, indeed, checking that the order of execution of actions that are close to one another does not change the outcome of the plan. This has the same effect as checking that coinciding action end points are non-mutex. In fact, mutex conditions are effectively rendered redundant when we vary the timestamps. The timestamps of actions are varied to within certain bounds and the chances of two actions occurring at exactly the same time is very remote.

The strong mutex constraint in PDDL2.1 *guarantees* that there can be no order in which actions at a single time point might be executed in practice and interfere with one another. It is in order to support a guarantee of correctness that the mutex condition is so strict. In the sampling approach we consider here we cannot offer a guarantee of plan correctness, but only a stochastic assessment, which can, of course, be made arbitrarily close to certainty. This is significant, because, for example, it is possible to construct a family of planning domains in which a set of $n$ actions may be executed at the same time point and generate the same resulting state in every possible sequential execution of the actions but one. This means that there would be only one chance in $n!$ of the actions producing a failure, and this might well be an acceptable risk for sufficiently high $n$, even though the plan would be rejected by VAL under the mutex rules of PDDL2.1 because it is not guaranteed to execute successfully.

## 6 Statistical Analysis

Our goal is to judder the times associated with a plan and check the validity of the resulting juddered plan as often as is necessary to give an acceptable level of confidence in the robustness of the original plan. If the plan juddered and executed one thousand times we can claim to have strong evidence for the extent of its robustness. In the following we report the results we have obtained from a *binomial experiment* investigating the robustness of plans. A binomial experiment satisfies the four following conditions:

1. There must be a fixed number of trials.
2. Trials must be independent (one trial's outcome cannot affect the probabilities of other trials).
3. All outcomes of trials must be in one of two categories.
4. Probabilities must remain constant for each trial.

The number of times that a plan will succeed out of $N$ runs of the experiment (each run consisting of juddering the timestamps and executing the plan), is given by a *binomial distribution*, denoted by $B(N, p)$, where $p$ is the probability of a plan succeeding. The value of $p$ is unknown, and we wish to determine its value. It is not possible to calculate this value precisely, but we can calculate it to within certain limits. Firstly we calculate a *confidence interval*, calculated using the following formula, for the number of valid plans obtained from $N$ runs of the experiment.

$$\bar{x} \pm \frac{t_{(\frac{\alpha}{2}, N-1)} s}{\sqrt{N}}.$$

The mean of the sample is denoted by $\bar{x}$, in this case the number of valid plans, a valid plan counts 1 and an invalid plan counts 0. The value $s$ is the standard deviation of the population. This is unknown, but (due to the central limit theorem) the sample standard deviation may be used given that $N > 30$. Finally $t_{(\frac{\alpha}{2}, N-1)}$ is the upper critical value of the $t$ student distribution with $N - 1$ degrees of freedom (a number to be retrieved from a table). We set $\alpha$ equal to $0.05$ so that the level of confidence is $95\%$ which is considered to be significant. There is a $95\%$ chance of the mean lying in this interval. The more often the experiment is run the smaller the size of the interval. Simply dividing the confidence interval by $N$ gives a confidence interval for the value of $p$. Sometimes a value of $p < 1$ may be acceptable: for example, if the plan has high rewards. However, most often we will be looking for plans that never fail. For this case we can perform a different statistical test.

We want to know how sure we can be that the plan is robust if it always executes successfully when the timestamps are juddered. We can perform a hypothesis test to determine this. Suppose that we have successfully executed $N$ juddered plans. We wish to be $99\%$ certain that the plan will be valid with a probability greater than $99\%$. If that the probability of executing a plan successfully is less than or equal to $99\%$ then the probability of the result being a fluke is at most $0.99^N$. To be $99\%$ certain that the result is not a fluke we need this value to be less than $0.01$. Therefore we require, $0.99^N < 0.01$, which implies that $N \geq 459$. Similarly it can be shown that to be $99\%$ certain that the plan will be valid with a probability of at least $95\%$ then $N \geq 90$. Also, to be $95\%$ certain of a valid execution with probability of at least $99\%$ and $95\%$, we require $N \geq 299$ and $N \geq 59$ respectively. When VAL executes $N$ plans with their timestamps juddered and all are valid then it reports that you can be $99\%$ certain that the plan will have a valid execution with probability of at least some percentage. For example 1000 successful runs grants $99\%$ certainty that the plan will execute with a probability of at least $99.77\%$.

### 6.1 Calculating the Robustness of a Plan

So far we have only considered juddering action timestamps by a random amount no larger than some bound, and how likely the plan is to succeed in these circumstances. It would be useful to know, for a given plan, what is the maximum possible judder that results in valid plan execution every time. Let $v$ be the judder value. The largest value for $v$ can be calculated by searching amongst its possible values. For each given value of $v$ we check that the varied plans will always be valid with a probability of at least $95\%$ with a confidence level of $95\%$. This requires successful execution of 59 plans. In this way the value of $v$ is calculated to within a small interval, see section 7 for some examples.

## 7 Examples

### 7.1 Thermostat

Consider the temperature of a machine that is controlled by a thermostat which fluctuates over time as given in figure 1. The temperature is modelled using events and processes as specified in the description of PDDL+ (Fox & Long 2002). the details of how these are modelled in PDDL+ are omitted as it is not relevant to the current discussion. It should be noted that the juddering of action timestamps does not vary the temperature model in any way. We define the problem so that a valid plan must place a discrete action at every local maximum and minimum of the temperature curve within a limited time. This is achieved by forcing these actions to be executed for values above or below certain temperatures, and by ensuring that the actions must alternate between maxima and minima. The best plan to solve this problem is given below:

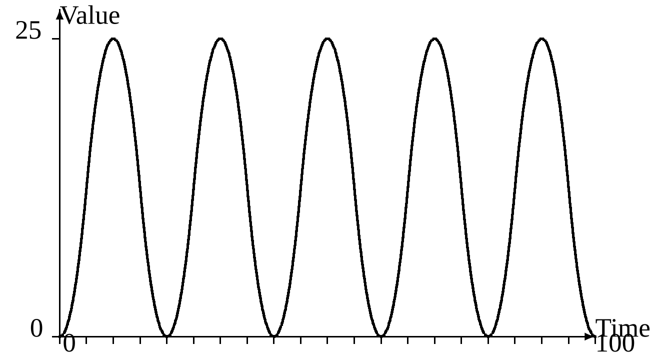

Figure 1: Graph of (temp unit).

| **Time** | **Action** |
|---|---|
| **10:** | (upper unit) |
| **20:** | (lower unit) |
| **30:** | (upper unit) |
| **40:** | (lower unit) |
| **50:** | (upper unit) |
| **60:** | (lower unit) |
| **70:** | (upper unit) |
| **80:** | (lower unit) |
| **90:** | (upper unit) |

Firstly, suppose we wish to test how the plan performs when the action timestamps can vary by up to $4$ time units. When VAL executes $100$ randomly altered plans the following results are reported:

- 12 plans are valid from 100 plans for each action timestamp $\pm$4.
- There is a 95% chance that the plan has a valid execution with probability in the range 12 $\pm$6.44724.

The plan failures are reported as follows:

| **Failures** | **Time** | **Action** |
|---|---|---|
| 22 | **10:** | (upper unit) |
| 19 | **20:** | (lower unit) |
| 6 | **30:** | (upper unit) |
| 7 | **40:** | (lower unit) |
| 12 | **50:** | (upper unit) |
| 5 | **60:** | (lower unit) |
| 13 | **70:** | (upper unit) |
| 2 | **80:** | (lower unit) |
| 2 | **90:** | (upper unit) |

As the results show, the plan is not highly robust. Each action has the same probability of failure as the other actions.

However, when a plan has failed at some point the execution stops, so the plan failures listed show the first point at which the plan fails. As a consequence the actions later in the plan are less likely to invalidate the plan because they depend on the other actions not failing first. Figure 2 shows the percentage of plans that invalidate the plan at certain times. (These points are joined by lines.) Figure 3 shows the cumulative percentage of plans that fail at certain times. The confidence interval is quite large, so to reduce the size we can perform the same test again but this time with 1000 varied plans.

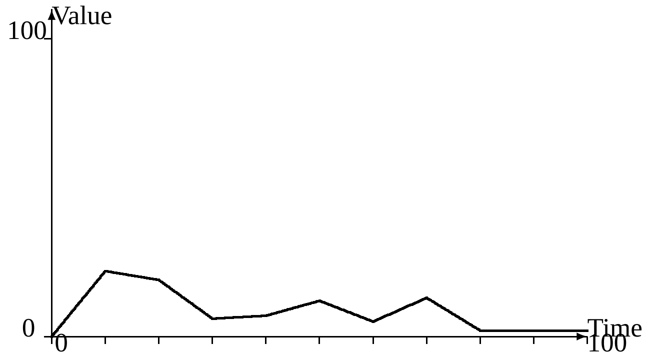


Figure 2: Percentage of plans failing at different times for 100 plans.

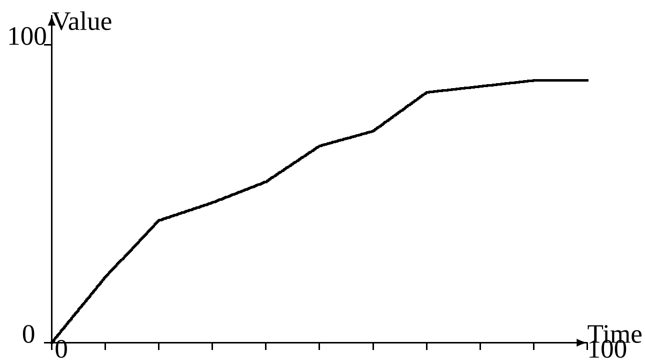


Figure 3: Cumulative percentage of plans failing by different times for 100 plans.

- 122 plans are valid from 1000 plans for each action timestamp $\pm 4$.

- There is a 95% chance that the plan has a valid execution with probability in the range 12.2 $\pm 2.03061$.

Because of the larger sample size we can be more confident that the probability of success is about $12.2\%$. The graphs showing when the plans fail, figures 4 and 5, show a smoother appearance as we would expect. The probability of a given action failing is $(1-p)^n p$, where $p$ is the probability of one of the actions failing and $n$ is the number of actions before the action in question. The graphs confirm that the plans fail following these probabilities. In more complex plans the probability of each action failing will be different and their interaction with other actions will need to be taken into account. For any plan an action can only invalidate a plan if the preceding plan has been successful, which will have a given probability.

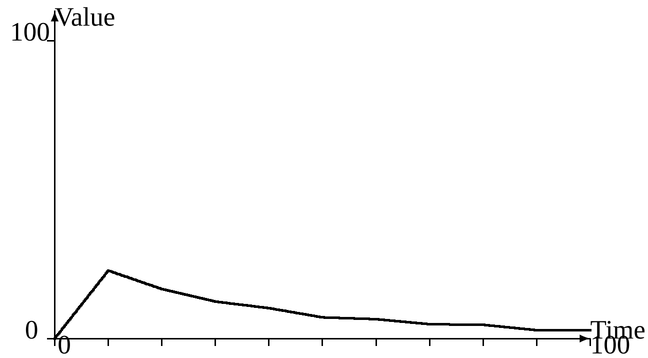


Figure 4: Percentage of plans failing at different times for 1000 plans.

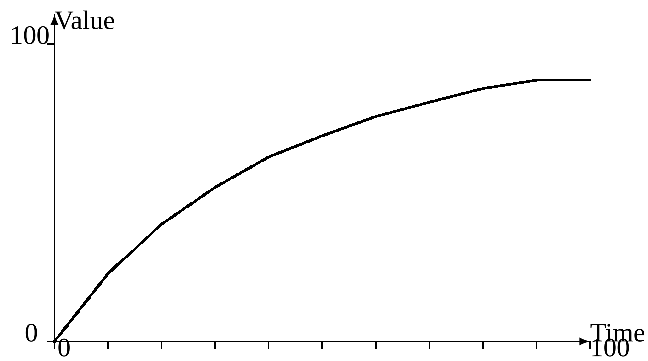


Figure 5: Cumulative percentage of plans failing by different times for 1000 plans.

If we use VAL to calculate how robust this plan is we get the following report:

- The plan has a robustness in the range $3.15918 \pm 0.00488281$.

This shows that provided that the actions do not vary by more than 3.154 (taking the most conservative bound) then the plan will be execute successfully. This value can be considered as the robustness measurement of the plan. For this example we can, in fact, calculate its robustness exactly, giving $\sqrt{10} = 3.162277...$, which is in the range calculated by VAL. In general it is not possible or feasible to calculate the robustness measurement of a plan exactly. However, using VAL, it is easy to calculate this measurement to within a small interval.

### 7.2 The Generator

As another example of calculating the robustness of a plan consider the generator example. Suppose that a generator must run continuously for 100 time units. In order to achieve this it must be refuelled whilst it is generating using two tanks of fuel. Refuelling starts quickly, but slows down to a trickle as the tank empties. If refuelling is initiated too early then the generator fuel tank will overflow. If it is initiated too late the tank will run dry. Therefore we need we refuel the generator somewhere near the mid point of the generating activity. However, the two refuelling actions must not be too close as they cannot overlap. The graph in figure 6 shows the fuel level of the generator in a robust plan for this problem. VAL reports:

- The plan has a robustness in the range $6.26465 \pm 0.00488281$.

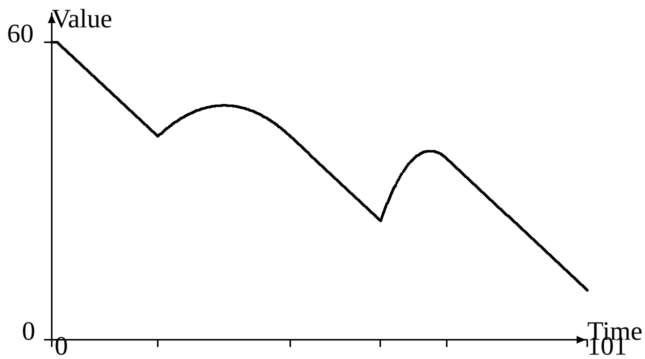


Figure 6: Graph of (fuel-level generator).

### 7.3 Robustness to Duration Variation

As well as juddering the start point of an action we can judder the action duration. This reflects the fact that actions sometimes take slightly less or more time than expected. However, the impact of this is that end points of actions can be displaced by up to twice the judder value. This can have significant impact on plan validity as illustrated in the following example.

Plans generated in the IPC3 competition, using the $\varepsilon$ tolerance value, are often not robust to variations in the action durations because they have been constructed to be as tightly packed as possible with respect to $\varepsilon$. As an example we consider a plan from the 2002 IPC produced by LPG for the zeno travel domain with time and numerics. VAL calculates the robustness of this plan as $0.000457764 \pm 0.000152588$. The plan was calculated using $\varepsilon = 0.001$, which is the minimum distance by which interfering actions should be separated. Therefore we would expect the plan to have a robustness of at least 0.0005 (since two actions may move toward one another). However, testing the plan for a variation of $\pm 0.0004$ on 1000 plans yields the result that there is a $95\%$ chance that the plan has a valid execution with probability in the range $98.6 \pm 0.728956$. This loss of robustness is directly due to the double judder phenomenon described above.

Now suppose that we wish to use this plan with a variation of $\pm 0.001$. The robustness measure is smaller than this so we do not expect the plan to always work. We wish to identify how likely the plan is to succeed and where the plan is most likely to be invalidated. If we test the plan with this variation on 1000 runs then VAL reports that 'there is a $95\%$ chance that the plan has a valid execution with probability in the range 43.1 ±3.07251.' The plan failures are reported as below:

| Failures | Time | Action |
|---|---|---|
| 157 | **0.002:** | (board person1 plane1 city0) [0.3] |
| 0 | **0.303:** | (fly plane1 city0 city1) [4.870] |
| 164 | **5.174:** | (board person3 plane1 city1) [0.3] |
| 10 | **5.175:** | (debark person1 plane1 city1) [0.6] |
| 115 | **7.196:** | (fly plane1 city1 city0) [4.87] |
| 123 | **12.067:** | (debark person3 plane1 city0) [0.6] |

Figure 7 shows a graph produced by VAL of the number of actions that fail at certain times. There is also a list of why each action failed, together with sample plan repair advice. The plan repair advice is for only one failed instance, since in general when numerical values are involved every single failure could be unique. For example the advice for the first action is:

1. 157 failures for **0.002:** (board person1 plane1 city0) [0.3]

   (a) 157 failures: The invariant condition is unsatisfied. Sample plan repair advice:

      i. *Invariant for* (board person1 plane1 city0) has its condition unsatisfied between times 0.302713 and 0.3029. The condition is satisfied on the empty set. Set (at plane1 city0) to true.

Failure of the execution of a durative action can be caused by violation of its invariant condition or failure to satisfy its precondition. This action has failed because its invariant condition has been violated. Looking at the plan it can be seen that the *(fly plane1 city0 city1)* action starts almost exactly when the boarding operation has finished. It is clear that the *board* action fails because, as a consequence of juddering, the plane has taken off before the passenger has finished boarding.

## 8 Robustness with Metric Fluents

Any plan that is intended to interact with physical processes will be subject to other sources of uncertainty than simply the times at which actions are executed. In particular, processes generate continuous change in the world that will only ever be modelled at some level of abstraction. Thus, a bath filling with water that is flowing at a constant rate can be modelled as having a volume that increases linearly. This model abstracts phenomena such as small quantities of water splashing out of the bath, minor fluctuations in the rate of flow due to unpredictable and uncontrollable additional demands on the water supply and so on. Figure 8 illustrates

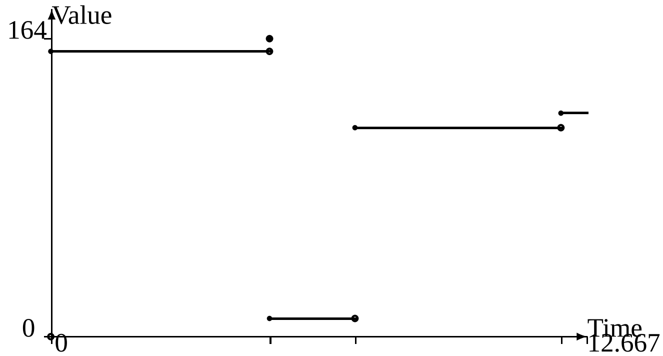


Figure 7: Number of plans failing at these times.

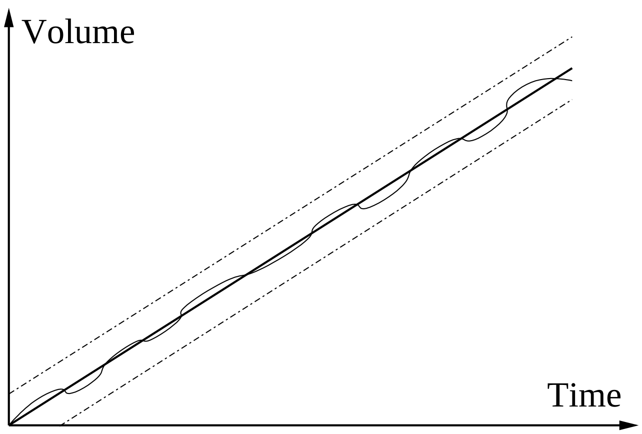


Figure 8: Graph showing water flowing into a bath.

how the volume of a bath might fluctuate from its linearly increasing estimate, as suggested by the fluctuating curve. In using the model to predict the volume of the water in the bath it would be accepted that the predicted volume would not exactly match reality to arbitrary levels of precision (nor could the real volume even be measured to arbitrary precision in order to compare it with the model). The implication of this for robust planning is that no plan can be considered robust if its correctness depends on the values predicted by its models being accurate to arbitrary degrees of accuracy. Thus, just as the times of execution of actions should be expected to judder, so also should measurements of metric fluents evolving under the influence of continuous processes.

We distinguish values that are influenced by continuous processes from values that are affected by discrete change alone. Where values increase or decrease by discrete quantities then the abstraction of the quantity into these discrete units is sufficient to ensure that the uncertainty in execution can be eliminated. Essentially, the uncertainty about the execution of actions that depend on these values is abstracted into the question of how accurately the discrete units can be measured and how appropriate these units are for the execution of actions that consume them. We may assume that continuous processes are only modelled explicitly in domains where there is a potentially significant sensitivity to threshold values and it is precisely in these cases that we want our plans to be robust to minor fluctuations in the physical processes that drive them.

### 8.1 Change, Chaos and the Butterfly Effect

In some cases, as is well known, small changes in initial conditions can lead to dramatically different evolutions of a physical system. These systems are often said to exhibit chaotic or highly non-linear behaviour. The so called “butterfly effect” is apparent in a wide range of physical phenomena. It is readily apparent that plans are extremely unlikely to be able to interact with metric fluents with this kind of behaviour in any way that is highly sensitive to the actual values of the fluents. For this reason, it will make more sense to model systems with these behaviours as abstractions that can only be managed at a coarse level. For example, we know that weather patterns have this kind of chaotic behaviour and it is therefore not reasonable to construct plans that depend on predicting precise temperatures, cloud cover or precipitation at precise times of day. Instead, we can manage abstractions that use ranges of temperatures across intervals of time, so that we can, for instance, plan what clothes to take on holiday.

If we assume that our planning models do not contain explicit models of physical processes that are highly non-linear or chaotic, then we can simplify our management of the uncertainty that can arise in handling the metric fluents that are affected by the processes. In particular, we can assume that, over time, the model is an accurate prediction of the evolution of a process, subject only to a local fluctuation in the value measured at any given instant.

### 8.2 Robust Plans with Metric Uncertainty

To test the robustness of plans to uncertainty caused by fluctuations in the behaviours of physical processes we consider only metric fluents that are subject to continuous change at points where they occur in comparison conditions. Wherever such comparisons are made as preconditions for execution of actions we apply a small judder to the value of the appropriate metric fluents before checking the condition. This process is no more complicated in the case of invariants, since the judder is treated as a constant shift in the curve governing the process for the purposes of testing the invariant across its appropriate interval. We do not propagate the effects of the judder into the use of the corresponding metric fluents for updating values in the effects of actions, which is the consequence of our assumption that all processes are sufficiently accurately modelled and sufficiently predictable to be adequately handled by the model. We also do not use judder to adjust the preconditions of events. This decision is based on the view that events represent consequences of changing processes in the world and there is no imperfection in the reaction of the world to those consequences. Of course, in some models events might be intended to represent the reactions of external agents to processes initiated by the planning agent and, in that case, it might be appropriate to apply a judder to those reactions. The question of precisely what is an appropriate way to handle events and whether to handle some events differently, remains an area for future work.

One implication of this approach is that any preconditions that require *strict equality* tests between continuously changing metric fluents and some other values will fail. We consider this to be realistic: it will only ever be possible to achieve strict equality at the level of abstraction used in measuring discrete units. Otherwise the best that can be achieved is to obtain a value lying within a particular interval.

Even though we do not judder the effects of metric updates, simply juddering the times at which actions occur is sufficient to have an impact on the behaviour of metric update processes. It is possible for these effects to be non-linear and for their propagation into the plan to have dramatic consequences for the execution. This is an aspect of considerable interest and knowing where a plan is vulnerable to non-linear effects caused by apparently minor changes in the structure of a plan would be of value in determining whether a plan is useful and how to protect it during execution.

## 9 Future Work

In future work we intend to address both an increased level of non-determinism in the metric components of the plan and the integration of our approach with the probing strategy of Younes (Younes 2004) which considers non-deterministic outcomes of actions. In terms of extending the level of metric non-determinism that we consider, we wish to address the fact that uncertainty about the time of execution of specific actions and the uncertainty in the processes that govern metric fluents are not uniform. Our current treatment assumes that they are. Our current model for introducing judder into the behaviour of durative actions assumes that the time of the final point is governed by when the action starts and our ability to measure the accuracy of its duration. In some cases the duration will be governed by a process so that a better model of the uncertainty would be to judder the end point independently of the start point.

We currently judder the plan and then validate the resulting plan, so that events are triggered according to the times at which the juddered actions occur. In cases where events are triggered by metric fluents crossing critical thresholds under the influence of continuous processes our current approach will judder the value of the metric fluents leading to a corresponding impact on the timing of events. In some cases this can lead to significant changes in the behaviour of the plan and even apparently chaotic outcomes. We are still considering how best to deal with this.

## 10 Conclusion

We have presented a stochastic strategy for determining the robustness of temporal plans to the possible variations in the timings of actions at execution. We have proposed a probing strategy which uses a juddering mechanism to sample plans within a *tube* around the original plan. The width of the tube is determined by the judder value. Using this strategy we can determine whether a plan is robust to the given judder value, and we can also determine the judder value that gives robustness to a required confidence level. We consider temporal and metric constraints to constitute a form of non-determinism because of the inaccuracy inherent in measuring properties of the physical world. We want to increase the amount of non-determinism that can be handled by our approach and to integrate our strategy with those that consider non-deterministic outcomes of actions.

## References


Fox, M., and Long, D. 2002. PDDL+ : Planning with time and metric resources. Technical report, University of Strathclyde, UK. Available at: `planning.cis.ac.uk/competition/`.

Fox, M., and Long, D. 2003. PDDL2.1: An extension to PDDL for expressing temporal planning domains. *Journal of AI Research* 20.

Gupta, V.; Henzinger, T.; and Jagadeesan, R. 1997. Robust timed automata. In *HART'97: Hybrid and Real-time Systems, LNCS 1201*, 331–345. Springer-Verlag.

Howey, R.; Long, D.; and Fox, M. 2004. VAL: Automatic plan validation, continuous effects and mixed initiative planning using PDDL. In *Proceedings of 16th IEEE International Conference on Tools with Artificial Intelligence*.

Morris, P.; Muscettola, N.; and Vidal, T. 2001. Dynamic control of plans with temporal uncertainty. In *Proceedings of IJCAI*, 494–502.

Muscettola, N. 1994. HSTS: Integrating planning and scheduling. In Zweben, M., and Fox, M., eds., *Intelligent Scheduling*. San Mateo, CA: Morgan Kaufmann. 169–212.

Vidal, T., and Ghallab, M. 1996. Constraint-based temporal management in planning: the IxTeT approach. In *Proc. of 12th European Conference on AI*.

Younes, H., and Littman, M. 2004. PPDDL1.0: An extension to PDDL for expressing planning domains with probabilistic effects. Technical report, `www.cs.cmu.edu/l̃orens/papers/ppddl.pdf`.

Younes, H. 2004. Planning and verification for stochastic processes with asynchronous events. In *Proceedings of the Nineteenth National Conference on Artificial Intelligence*, 10011002.