Exploration of the Robustness of Plans

Maria Fox and **Richard Howey** and **Derek Long**
Department of Computer and Information Sciences
University of Strathclyde, Glasgow, UK
Email: *firstname.lastname@cis.strath.ac.uk*

Abstract

This paper considers the problem of stochastic robustness testing for plans. As many authors have observed, unforeseen execution-time variations, both in the effects of actions and in the times at which they occur, can result in a plan failing to execute correctly even when it is valid with respect to a domain model. In this paper we contrast the validation of a plan with respect to a domain model, confirming soundness, with the validation with respect to an execution model, which we call robustness. We describe a Monte Carlo probing strategy that takes a hypothesis testing approach to confirming the robustness of a plan. An important contribution of the work is that we draw links between the robustness of plans and the notion of the "fuzzy" robustness of traces through timed hybrid automata, introduced by Gupta *et al*. We show that robustness depends on the metric used to define the set of plans that are probed, and that the most appropriate metric depends on the capabilities of the executive and the way in which it will interpret and execute the plan.

1 Introduction

Testing plan robustness is an important stage in ensuring the validity of a domain model and the reliability of the plan generation process. Several authors (Cichy *et al.* 2004; Muscettola 1994; Mahtab, Sullivan, & Williams 2004) have described validation processes followed in the development of deployed AI systems, and others have considered theoretical aspects of the development of *robust* plans (Cichy *et al.* 2004; Muscettola 1994), schedules (Davenport, Gefflot, & Beck 2001) and models (Ginsberg, Parkes, & Roy 1998). In this paper we address the problem of *stochastic robustness* to temporal and numeric variation. We are interested in confirming that a plan is robust to a high level of confidence. We investigate this by means of Monte Carlo probing.

An example of a probing strategy is described in Cichy *et al.* (2004) in their discussion of the validation processes followed in the development of the Autonomous Sciencecraft Experiment. They consider a plan to be valid if it executes correctly when minor, normally distributed, distortions are applied to the parameters of the plan. These distortions affect, for example, how long a science analysis activity takes to complete or the time at which a celestial event occurs relative to the start of an orbit. The testing strategy generates a set of distorted plans, in which the distortions are distributed normally around the nominal values of the parameters according to the domain model. The distortions model the fact that arbitrary precision, in temporal and numeric measurements, cannot be attained in practice.

Ginsberg *et al.* (1998) and Gupta *et al.* (1997) define metrics according to which solutions can be claimed to be robust to execution-time uncertainties. In Ginsberg *et al.*, a super model encodes the degree of robustness of a given model of a logical theory. Gupta *et al.* define sets of traces that are similar, with respect to some metric, to a given trace. As we will show, such metrics can be used to construct sampling sets for determining stochastic robustness of plans.

Cichy *et al.* do not explain the metric that governs the construction of their sampling sets. However, it is important to do so because the structure of the sampling set is affected by what distortions are applied, and how. For example, depending on how the plan is to be executed, the sampling set obtained by independently varying the parameters might not be the best one for exploring its robustness.

Gupta *et al.* define the notion of *robust traces* in the context of hybrid automata. A trajectory defines a robust trace, τ, through a hybrid automaton if there is a dense subset of the trajectories lying within some open tube, called a *fuzzy tube*, around τ that contains only acceptable traces. The authors define various alternative metrics that can be used in determining the open tube around a trajectory. Their work is theoretical and has not been used as a basis for practical robustness testing.

In this paper we describe a practical strategy for the stochastic determination of the robust acceptance of a plan based on the theoretical foundations established in (Gupta, Henzinger, & Jagadeesan 1997). We define several metrics that can be used to construct sampling sets for plan robustness testing, and show that the robustness of a plan can depend on the metric used. The metric should be chosen according to how the plan will be executed. If the executive will be allowed to bring actions forward, or delay them, according to execution-time conditions, then the appropriate metric will be different from the one that should be used if the timings of actions are critical.

2 Domain Models, Plans and Execution

We begin by specifying the planning context in which this work is framed.

Definition 2.1 *A Domain Model is a triple (S, A, E), where S is a set of states, A is a set of actions and E is a set of events. Each action or event, $\alpha \in A \cup E$ has a precondition $pre(\alpha)$ and defines a state-to-state mapping such that, for $s \in S$, if $s \models pre(\alpha)$ then $\alpha(s) \in S$ is the state following application of α. The distinction between actions and events is that the decision to enact an action is a choice, while, for any event $e \in E$ and state $s \in S$, if $s \models pre(e)$ then e is automatically applied.*

Timed initial literals provide an example of the kind of events that might be included in a domain model, and PDDL2.2 (Hoffmann & Edelkamp 2005) is an example of a syntax used to express such domain models.

Definition 2.2 *A Temporal Plan π, is a timed-stamped collection of actions represented as a set of triples (t, a, d), where t is the real-valued time-point relative to a nominal start time, a is the name of the action and d is the real-valued duration of the action.*

Again, plans for domains written in PDDL2.1 and PDDL2.2 are of this form. In each state visited in the execution of a plan, any events that are triggered are applied before considering the execution of actions. It is straightforward to extend this framework to include actions with duration, including actions with continuous effects (as in PDDL2.1 (Fox & Long 2003)).

3 Robustness of Plans and Traces

A plan can be validated with respect to a domain model, confirming its soundness. However, soundness is not sufficient to guarantee robustness of execution, since no executive, even under the control of highly accurate micro-controllers, can achieve arbitrary levels of accuracy in the synchronisation of actions.

One way that this problem has been handled is by the introduction of *temporal flexibility* in which actions take place not at precise timepoints but within flexible windows that allow for execution-time discrepancies (Muscettola 1994; Vidal & Ghallab 1996). These windows can be expressed as sets of temporal constraints that are subject to uncontrollable events. Determining the dynamic controllability of a set of temporal constraints (a set of constraints is DC if they can be satisfied no matter when uncontrollable events occur) has been explored and efficient algorithms have been proposed (Morris & Muscettola 2005; Tsamardinos & Pollack 2003; Morris, Muscettola, & Vidal 2001).

Gupta *et al.* identify the difficulties that arise when trajectories through hybrid automata are interpreted as defining the timing of actions and events with arbitrary precision. Again, 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 formally valid, 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 since a physical system cannot achieve the precision of execution that would avoid the invalid trajectories. This observation led the authors to consider fuzzy tubes.

By analogy with the work of Gupta *et al.*, we also consider a tube of plans around a core plan. We consider the metrics that define the tubes in the next section, but firstly we define the concept of stochastic plan robustness. By contrast with Gupta *et al.*, who require a dense set of valid traces, we want *sufficiently many* of the plans in the open tube around a core plan to be valid in order to claim that the core plan is robust. Robustness depends not only on which plans lie in the tube we consider, but also on the distribution of the variability in the plans within it. We discuss this issue in section 4 below.

Definition 3.1 *Robustness A valid plan π is robust to level p, with respect to a set of plans Π, where $\pi \in \Pi$, and a frequency distribution over Π, f, if a proportion p of the plans in Π under the distribution f are valid.*

For example, a valid plan π is 95% robust with respect to a tube, T ($\pi \in T$), and frequency distribution, f, if 95% of the plans in T under f are valid. A tube will be an open set of plans within a given distance of a core plan, according to some specified metric. The core plan is the plan returned by the planner and the structure of the tube is determined by the behaviour of the executive in attempting to execute that plan.

4 Plan Metrics and Tubes

When measuring the distance between two points there is always a specific metric in use. Depending on the metric used, two points might be different distances apart. For example, the Euclidean distance between opposite corners of a unit square is $\sqrt{2}$, while the Manhattan distance is 2.

The methods described in this paper are applicable for any metric measuring the distance between plans with the same actions. For a plan, π, let Π be the set of plans with the same actions as π and $len(\pi)$ be the length of π. The time at which the i^{th} action of π occurs is referred to as π_i. We will only consider two plans with the same actions for this discussion.

Definition 4.1 *Given two plans, π, and ψ with $\psi \in \Pi$ the* max *metric is defined by*

$$d_{max}(x, y) = \max_{1 \leq i \leq len(\pi)} |\psi_i - \pi_i|.$$

This metric measures the distance between two plans π and ψ as the maximum distance between corresponding actions. For example if π is a plan with times, $\pi_1 = 1$, $\pi_2 = 2$ and $\pi_3 = 3$ and ψ is a plan with times $\psi_1 = 1.1$, $\psi_2 = 2.3$ and $\psi_3 = 3.2$ then $d_{max}(\pi, \psi) = \max\{|0.1|, |0.3|, |0.2|\} = 0.3$. The max metric is also presented in Gupta *et al.* and is effectively the basis of the analysis performed by Cichy *et al*. It corresponds to the case in which an executive attempts to execute the actions in a plan according to absolute time. In this case, the delay of an action never affects the time at which the next action is attempted.

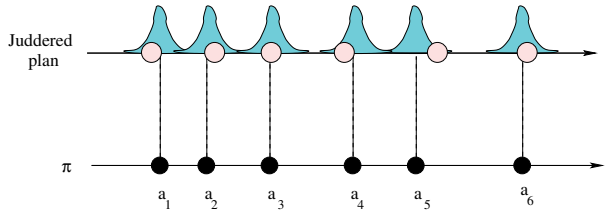

Figure 1: The max metric measures the distance between two plans as the maximum difference between corresponding pairs of actions.

Definition 4.2 *Given two plans, π, and ψ with $\psi \in \Pi$ the* cumulative *metric is defined by*
$d_{accum}(\pi, \psi) =$

$$\max\{|\psi_1 - \psi_1|, \max_{2 \leq i \leq len(\pi)} |(\psi_i - \pi_i) - (\psi_{i-1} - \pi_{i-1})|\}.$$

This metric extends the max metric by taking the accumulated time differences of previous actions into account when measuring the distance between two plans π and ψ. For example if π is a plan with times, $\pi_1 = 1$, $\pi_2 = 2$ and $\pi_3 = 3$ and ψ is a plan with times $\psi_1 = 1.1$, $\psi_2 = 2.3$ and $\psi_3 = 3.2$ then $d_{accum}(\pi, \psi) = \max\{|0.1|, |0.3 - 0.1|, |0.2 - 0.3|\} = \max\{0.1, 0.2, 0.1\} = 0.2$. It models the behaviour of an executive that executes actions at times determined relative to the preceding actions. If an action is delayed the actions following it are delayed by a concomitant amount. This metric is similar to the *sucpair* metric of Gupta *et al.*

There are two parameters that define a tube around a plan π: the metric and the *width* of the tube.

Definition 4.3 *Given a metric, d, an open* metric-tube *of width, $w > 0$, for a plan, π, is defined to be a set of plans Π within w of π measured by d. The tube, $T_d(\pi, w)$, is defined by*

$$T_d(\pi, w) = \{\eta : d(\pi, \eta) < w\}.$$

It is necessary to choose a frequency distribution to govern the sampling of the time-points of actions. We call this frequency distribution the *juddering distribution* and, for the purposes of this work we assume that the same juddering distribution can be used for all time-points. We refer to the plans that are sampled as *juddered plans*. The choice of distribution is related to how the executive interprets the time-points in a plan. One possibility is that the executive will try to execute each step as close as possible to a nominal time as defined by the plan (including any offset introduced by the cumulative metric), in which case a normal distribution of variability is appropriate.

The width of the juddering distribution is equal to the width of the tube and the nominal values lie at the centre-points of the distributions associated with each of the actions in the plan. It should be noted that the juddering distribution defines a distribution over plans in the sample space. Indeed, we restrict our attention to distributions over plans that are defined by juddering distributions.

The way in which plans within w of a given plan are sampled (and the corresponding tube constructed) is determined by the metric and the juddering distribution. For example, if the max metric is used, with a supported normal distribution, samples are obtained by independently juddering the time at which each action occurs according to a supported normal distribution around the time of that action in the original plan (a normal distribution is inappropriate as it does not stay within bounds). Alternatively, if the cumulative metric is used, with the same distribution, the timepoints cannot be juddered independently. Instead, each action start point is delayed by the sum of the delays applied so far, and is then juddered itself (again according to a supported normal distribution). The cumulative metric is shown in figure 2(a).

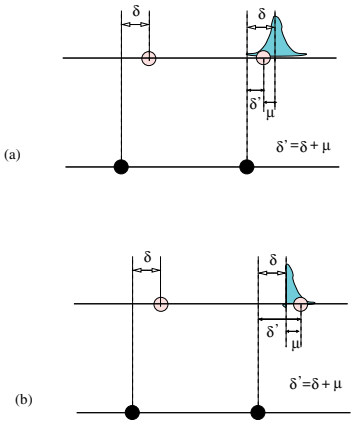

Figure 2: (a) Juddering time-points according to the accum metric. When y_i is juddered by an amount δ, y_{i+1} is juddered by $\delta + \mu$ where μ is an offset from delta sampled from the normal distribution with δ as its mean. μ might be a positive or negative offset. (b) Juddering according to the accum metric in which μ is always a positive offset.

The cumulative metric can also be used with a distribution that uses only the positive half of the supported normal curve. In this case, the delay throughout a plan is always increasing (as can be seen in figure 2(b)). This corresponds to the behaviour of an executive that never starts a new action until the propagated delay has fully elapsed after the time at which the last action finished.

We do not judder the time points at which uncontrollable events occur. We assume that these are fixed in time (for example, sunrise) and that only the behaviour of an executive can be anticipated or delayed. Furthermore, whilst physical executives are subject to inaccuracies of measurements in their interactions with the world, this cannot be said of the world itself.

5 Testing Robustness

It is obviously impossible to confirm robustness by exhaustive enumeration — tubes contain uncountably many plans. In some cases, proving robustness is possible analytically, but we are interested in a practical general approach. To achieve this, we approach the problem by Monte Carlo sampling of the tube. We randomly generate plans in the tube, using the parameters that define the tube and the distribution of plans within it. We then check the validity of these plans.

We take a hypothesis testing approach to determining whether enough executions of a juddered plan succeed for us to be able to conclude, with sufficient confidence, that the plan is robust. We contrast our approach with the Monte Carlo model-checking approach of Grosu and Smolka (2005) in which the aim is to be confident (to a certain level) that an LTL formula is modelled by a set of premises. The authors sample from the space of traces through the model, and conclude that the formula is valid if none of the sampled traces are counter-examples. The number, N, of samples that must be tested is given as

$$N = \left\lceil \frac{log\delta}{log(1-\varepsilon)} \right\rceil$$

where δ is the probability of making a Type 1 error, and ε is a bound on the error in judging the probability of there being counter-examples to be zero. As soon as a counter-example is found the experiment can be terminated. The Null Hypothesis, that the probability of seeing a counter-example is at least ε, cannot be rejected: a single counter example is sufficient to rule that the formula cannot be modelled by the premises. If no counter examples are found, then the Null Hypothesis can be rejected. If δ and ε are both 5% then $N = 59$, which is the number of trials required for the experiment.

By contrast, we want to have a certain confidence that a plan will execute correctly at least a certain proportion of the time. Let us say that we want 95% confidence that the plan will execute correctly at least 95% of the time. The probability of a Type 1 error is $\delta = 5\%$. Then $\varepsilon = 5\%$ and the Null Hypothesis is the assumption that the probability of an arbitrary trial failing is at least ε. To perform a proportion-based test we require a larger number of trials to be performed than for the yes-no test, because some failures will be tolerated. N can be calculated, given δ and ε, using Cochran's formula for proportion-based hypothesis testing (Cochran 1977). The formula is:

$$N = \left\lceil \frac{t^2(1-\varepsilon)\varepsilon}{\delta^2} \right\rceil$$

where $t = 1.96$ when 95% confidence is required, and $N = 76$ for the values of δ and ε that we have considered. At least 95% of the trials must succeed in order for us to reject the Null Hypothesis (in this case, 73 trials).

Proportion-based hypothesis testing is used when individuals are drawn, by Monte Carlo sampling, from a population to determine whether the claim that the population exhibits a certain feature can be held with sufficient confidence. It is known that some of the individuals in the population may not exhibit the feature, but the question is whether the feature occurs to a sufficient extent, with a sufficient level of confidence. The approach we have described is also used in random sampling in other contexts, such as in fish and animal populations (Lockwood & Hayes 2000).

6 Robustness, Continuous Change and Durative Actions

When a plan is interacting with continuously changing quantities its robustness can be affected by the ability of the executive to measure numeric quantities precisely. No executive can make arbitrarily precise measurements so will be unable to react with absolute accuracy to quantities reaching exact thresholds. For example, a plan that specifies that a faucet must be closed when a container holds 1.3 litres of water will not be robust to a situation in which the container holds 1.3001 litres at the time-point of the faucet-closing action. A similar problem arises if the plan specifies the effect of an action on a continuous quantity. An action to generate 15 units of power might actually generate fractionally more or less, depending on exactly when it starts and ends.

Both of these cases can be handled by juddering the continuous quantities when the conditions of any actions depending on them are checked. In other words, instead of juddering the effects of the continuous processes governing the values of the quantities (which, like events, can be seen as under the control of the world), the values are juddered only at the points of their interactions with the plan. This captures the inaccuracies of the executive whilst allowing the overall evolution of continuous change to follow the specification provided by the domain model.

A final point concerns the way in which juddering is done with respect to durative actions. Durative actions can be interpreted as encapsulating two related actions that are executed at a certain separation. Given this, it seems appropriate to judder each end point of a durative action separately. However, this can lead to a violation of the duration constraint asserting the separation of the two end points. To avoid this problem, we treat the duration as a constraint on the validity of the plan with respect to the domain model, but ignore it in the juddered plans.

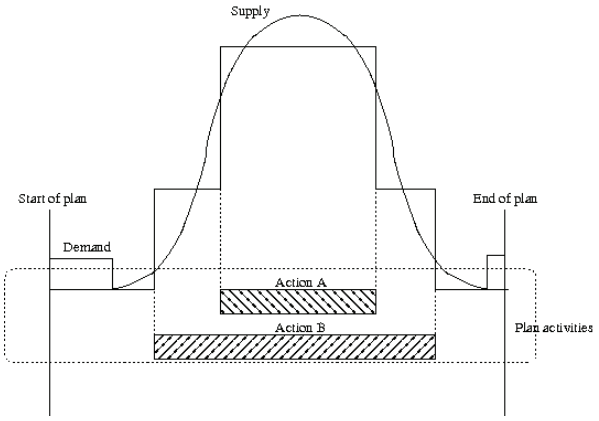

Figure 3: Concurrent durative actions drawing battery power that is continuously generated during daylight. At nightfall the battery must have sufficient power remaining to support night operations.

7 Experiments and Results

Figure 3 shows a simplified example of an operations plan generated for the defunct planetary lander, Beagle-2. Two durative actions simultaneously draw power from a battery against a backdrop of continuous power generation. At the

Watt Hrs	Max		Accum		Delay	
w = 0.2; m = 0	N	U	N	U	N	U
79.544	24	37	33	40	0*	76
79.545	0*	11	0*	0*	0*	76
79.546	0*	8	0*	0*	0*	76
79.552	0*	11	0*	0*	0*	76
79.563	0*	10	0*	0*	0*	76
w = 0.2; m = 0.02						
79.552	31	34	25	24	19	76
79.563	2*	13	0*	7	3*	76
79.8	0*	8	0*	0*	0*	76
w= 0.2; m = 0.2						
79.7	10	15	11	13	10	76
79.725	5	11	2*	4	2*	76
79.75	0*	5	0*	0*	0*	76
w = 0.02; m = 0						
79.544	0*	3*	7	23	0*	0*
79.545	0*	0*	0*	0*	0*	0*
w = 0.02; m = 0.02						
79.552	24	17	23	20	15	20
79.563	1*	2*	2*	3*	3*	3*

Figure 4: Results of robustness tests on the plan depicted in figure 5.

start and end-points of the actions the power demand increases and decreases respectively (the power demands of an action remain constant throughout execution of that action). Solar power generation begins at daybreak and continues until nightfall, rising to a peak at midday and then falling until dark. Whenever power generation exceeds demand, the excess is used to charge a battery that can power operations during the night. At nightfall the battery state of charge is checked to ensure that there is enough remaining to support night-time operations (such as the warming of instruments). If the power level is too low, the plan is invalid.

The plan may be valid, but not robust, if the power usage is high enough that temporal variations in the execution of the plan could cause the night-time power constraint to be violated. Figure 5 shows part of an actual operations plan. The critical check will be made at approximately time-point 14221, depending on the variation incurred during execution of the preceding steps. Before this time a critical point occurs at time-point 8659, when a paw operation relies on a paw movement having successfully completed immediately before its commencement. Depending on the accuracy with which the executive can measure time, the coincidence of these two actions might render the plan extremely fragile. A further critical point occurs when the communication action begins, at approximately time-point 15199. An event, or timed initial literal, opens a communication window just prior to this point, and another closes it at 17201. The communication action must start early enough to exploit the window. If there is too much variation in the execution of the earlier part of the plan, the remaining communication window might not be large enough to permit the successful execution of the communication action.

1:	(generate-solar-power)	[14219]
. . . .		
749:	(paw-move rock1_scs_closeup rock1_mbs_position)	[40]
799:	(paw-make-contact) [80]	
899:	(seq-mbs-rock rock1_mbs_position) [1890]	
. . . .		
6519:	(seq-rcg-grinding-nosample rock1_sample)	[1850]
8389:	(paw-move rock1_sample rock1_closeup)	[80]
8479:	(paw-invert down up)	[120]
8618.9:	(paw-move rock1_closeup rock1_scs_closeup)	**[40]**
8659:	(seq-scs-closeup rock1_scs_closeup)	[540]
. . . .		
12929:	(paw-move rock1_closeup rock1_xrs_position)	[40]
12979:	(paw-make-contact)	[80]
13069:	(seq-xrs-rock rock1_xrs_position)	[1820]
14221:	**(night-operations)**	[2180]
14918:	(paw-move rock1_xrs_position rock1_closeup)	[80]
15018:	(paw-move rock1_closeup rim)	[160]
15199:	**(comms)**	[1200]

Figure 5: A Beagle-2 operations plan. Critical points are highlighted in bold. There are 33 actions in total. Actions that are non-essential to this discussion are omitted.

The table in figure 4 presents results showing the conditions under which the plan in figure 5 is robust. The proportion-based hypothesis testing approach was used with $\delta = \varepsilon = 0.05$ so that 76 trials were run in each experiment. We tested the robustness of the plan to different initial power levels (expressed as numbers of Watt hours). The values in the columns are the number of times that the juddered plans failed out of 76 trials. The Null Hypothesis is that the probability that a trial will fail is at least ε. Starred values indicate that the Null Hypothesis can be rejected.

We varied the extent of temporal and numeric variability (the parameters w and m). The w parameter is the width of the fuzzy tube and m is the width of the distribution for metric juddering. We experimented with two different distributions: a supported normal distribution and a uniform distribution. In the table we refer to *Delay*: this is the *Accum* metric used with only the positive half of the distribution (either normal or uniform). We present ranges of Watt hour values that show where the boundary lies between robustness and fragility given the different parameter settings.

It can be seen that *Delay* only tolerates very small temporal and metric variation. This is because delay accumulates quickly and breaks the plan at the start of the communications operation (which depends on the event of the window opening, which is fixed in time). The *Max* metric cannot tolerate variation at the fragile mid-point at which the paw movement and activity are synchronised, and this is not affected by the power level. In all parameter combinations (except in *Delay* and the *Max* metric for large w) the default failure point is at the night-time operations check. As the initial power level is increased robustness to this check-point is increased.

8 Related Work

In this work we consider the question of whether a deterministic plan can be considered robust to temporal and numeric

variation at execution time. Our work is related to Beck and Watson's investigation into schedule robustness (2001), the Monte Carlo model-checking approach of Grosu and Smolka (2005), the theoretical work of Gupta, Henzinger and Jagadeesan (1997), and the stochastic fish population evaluation techniques discussed by, for example, Lockwood and Hayes (2000).

In the planning community probabilistic plan verification has been considered by Younes and Simmons (2002). They focus on sampling execution paths through discrete event systems in which actions have uncertain logical outcomes. Their work preceded that of Grosu and Smolka, and uses a hypothesis-testing strategy for testing CSL (Continuous Stochastic Logic) formulae. They do not address time-point or continuous quantity variation so their approach is orthogonal to ours.

The Prottle planner (Little, Aberdeen, & Thiebaux 2005) also exploits a Monte Carlo probing strategy in the planning context, but in that case it is used in order to construct plans for probabilistic domains.

The work in planning that is most similar to our own is the plan testing strategy of Cichy *et al.* (2004). Our paper has concentrated on methodology rather than on the accurate modelling of the variations that can occur in any specific domain. By contrast, Cichy *et al.* were driven by the need to validate plans for the Autonomous Sciencecraft Experiment and so focussed primarily on issues that we have so far ignored. In particular, their work accounts for the different distributions that govern the extent to which particular time points are likely to vary, whereas we have assumed that all time points are governed by the same distribution. An extension of our work to support action-dependent distributions would not be difficult. Our contributions lie in the identification of the role of metrics and in the construction of the more precise framework for the Monte Carlo testing strategy.

9 Conclusion

We have presented a stochastic strategy for determining the robustness of temporal plans. Given a metric, a width and a frequency distribution, we can determine whether a plan is robust, with respect to the metric, when the times and numeric values within it are juddered according to the distribution across the width of the tube defined by the metric. The approach we use is one of hypothesis testing, in which we seek to reject a Null Hypothesis asserting that the given plan is not robust at a particular level.

A particular contribution of this work is to adapt the notion of a fuzzy tube from the field of trace acceptance in hybrid automata, applying it to plans and observing the relationship with the behaviour of different modes of execution. By combining the use of fuzzy tubes with the Monte Carlo probing strategy we have arrived at a practical means to determine the robustness of plans.

It is important to perform robustness testing using an appropriate definition of the sampling set. The metric used to construct an appropriate sampling set depends on how the plan will be executed. If the executive refers to absolute time then the max metric might be appropriate. If actions cannot start until others have successfully completed, the max metric is too simple and something like the cumulative metric is likely to be more appropriate. Our example illustrates how robustness testing might be affected by the choice of metric and the accompanying distribution, as well as by the width of the tube from which samples are drawn.

References

Cichy, B.; Chien, S.; Schaffer, S.; Tran, D.; Rabideau, G.; and Sherwood, R. 2004. Validating the Autonomous EO-1 Science Agent. In *International Workshop on Planning and Scheduling for Space (IWPSS)*.

Cochran, W. 1977. *Sampling Techniques*. New York: Wiley.

Davenport, A.; Gefflot, C.; and Beck, J. 2001. Slack-based techniques for robust schedules. In *Proceedings of the 6th European Conference on Planning (ECP)*.

Fox, M., and Long, D. 2003. PDDL2.1: An Extension to PDDL for Expressing Temporal Planning Domains. *Journal of AI Research* 20.

Ginsberg, M.; Parkes, A.; and Roy, A. 1998. Supermodels and Robustness. In *Proc. of AAAI-98*.

Grosu, R., and Smolka, S. 2005. Monte Carlo model checking. In *Proc. 11th International Conference on Tools for the Construction and Analysis of Systems (TACAS)*, 271–286. LNCS 3440: Springer-Verlag.

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.

Hoffmann, J., and Edelkamp, S. 2005. The Deterministic Part of IPC-4: An Overview. *Journal of AI Research (JAIR)* 24:519–579.

Little, I.; Aberdeen, D.; and Thiebaux, S. 2005. Prottle: A Probabilistic Temporal Planner. In *Proc. of AAAI-05*.

Lockwood, R., and Hayes, D. 2000. Sample size for biological studies. In Schneider, J., ed., *Manual of Fisheries Survey Methods II*. Michigan Department of Natural Resources, Lansing. chapter 6.

Mahtab, T.; Sullivan, G.; and Williams, B. C. 2004. Automated Verification of Model-based Programs under Uncertainty. In *Proceedings of the 4th International Conference on Intelligent Systems Design and Application*.

Morris, P., and Muscettola, N. 2005. Temporal Dynamic Controllability Revisited. In *Proc. of AAAI*.

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.

Tsamardinos, I., and Pollack, M. 2003. Efficient solution techniques for disjunctive temporal reasoning problems. *Artificial Intelligence Journal* 151(1–2):43–89.

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 Simmons, R. 2002. Probabilistic Verification of Discrete Event Systems using Acceptance Sampling. In *Proc. 14th International Conference on Computer Aided Verification*, 223–235.