Proceedings of the Twenty-Sixth International Conference on
Automated Planning and Scheduling (ICAPS 2016)

Heuristic Guidance for Forward-Chaining Planning with Numeric Uncertainty

Liana Marinescu and **Andrew Coles**
Department of Informatics,
King's College London, UK
firstname.lastname@kcl.ac.uk

Abstract

Uncertainty hinders many interesting applications of planning – it may come in the form of sensor noise, unpredictable environments, or known limitations in problem models. In this paper we explore heuristic guidance for forward-chaining planning with continuous random variables, while ensuring a probability of plan success. We extend the Metric Relaxed Planning Graph heuristic to capture a model of uncertainty, providing better guidance in terms of heuristic estimates and dead-end detection. By tracking the accumulated error on numeric values, our heuristic is able to check if preconditions in the planning graph are achievable with a sufficient degree of confidence; it is also able to consider acting to reduce the accumulated error. Results indicate that our approach offers improvements in performance compared to prior work where a less-informed relaxation was used.

1 Introduction

Many compelling applications of planning arise from scenarios that are inherently uncertain. In some cases it is possible to adequately capture the dynamics of the world without modeling uncertainty, and thus to employ classical planning techniques. However, in many other cases it is impossible to ignore uncertainty without hindering the planner's knowledge about the world, and hence obtaining sub-par solutions.

In this paper, our focus is on finding plans for models where there is uncertainty in the outcomes of numeric effects, each governed by a continuous distribution. Here, the task is to find a plan where all the preconditions are met, and the goals are reached, with some confidence θ. This paradigm has been explored by previous work, e.g. (Beaudry, Kabanza, and Michaud 2010; Coles 2012), but *heuristic guidance* is an open challenge. A planning model without uncertainty cannot always provide reliable plans – similarly, a heuristic without a model of uncertainty cannot always provide useful guidance. A good heuristic would be better able to indicate which actions are suitable, and offer better state pruning by recognizing dead ends sooner.

We present an extension to the Metric Relaxed Planning Graph heuristic (Hoffmann 2003) that incorporates a model of uncertainty for two purposes. First, basic information about uncertainty on variables thus far is used to determine which preconditions are true in the planning graph. Second, where remedial actions are available to reduce uncertainty, the heuristic is able to include them in the relaxed plan.

To demonstrate the efficacy of this new heuristic, we present empirical results that indicate it is effective in a number of interesting domains, reducing the search effort needed to find acceptable solution plans.

2 Background

In this work, we build upon the state-progression semantics of the planner RTU (Beaudry, Kabanza, and Michaud 2010). Here, a planning problem is a tuple $\langle F, \mathbf{v}, I, G, A, \theta\rangle$ where:

- F is a set of propositional facts;
- $\mathbf{v}$ is a vector of numeric variables;
- I is the initial state: a subset of F and assignments to (some) variables in $\mathbf{v}$;
- A condition is a first-order logic formula over facts in F and Linear Normal Form (LNF) constraints on $\mathbf{v}$, each written: $(\mathbf{w}.\mathbf{v}\ \ op\ \ l)$, where $op \in \{>, \geq\}$; $l \in \Re$; and $\mathbf{w}$ is a vector of real values.
- G describes the goals: a set of conditions. Each $g \in G$ has an associated cost $c(g) \in \Re^+$ if g is not true at the end of the plan. For compulsory (hard) goals, $c(g) = \infty$.
- A is a set of actions, each $a \in A$, with:
  - $Pre(a)$: a (pre)condition on its execution;
  - $Eff^-(a)$, $Eff^+(a)$: propositions deleted (added) by a;
  - $Eff^{num}(a)$: a set of numeric variable updates that occur upon applying a. Each is of the form $\langle v\ op\ D(\mathbf{v})\rangle$ where $op \in \{\texttt{+=}, \texttt{=}\}$ and D is a (possibly deterministic) probability distribution that governs the range of outcomes of the effect. For instance, $\langle battery \mathrel{+=} \mathcal{N}(-10, 2^2)\rangle$ means 'decrease battery by an amount with mean 10 and standard deviation 2'.
- $\theta \in [0.5, 1)$ is a confidence level.

A Bayesian network (BN) is used to define the belief of each $\mathbf{v}$, and as actions are applied, the network is updated with additional variables. In a state S_i, for each $v^j \in \mathbf{v}$, a variable v_i^j is associated with the belief of v. If an action a is applied, leading to a state S_{i+1}, then for each numeric effect $\langle v^j\ \ op\ \ D(\mathbf{v})\rangle$, two random variables are added to

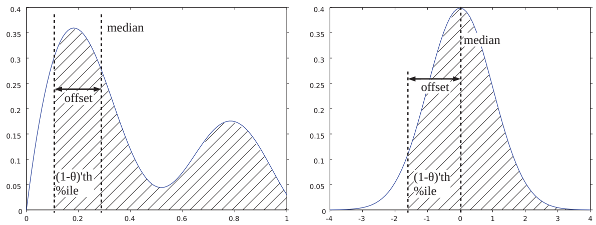


Figure 1: Possible probability distributions: Arbitrary (left) and Gaussian (right).

the network. The first of these, D^j_{i+1}, represents $D(\mathbf{v})$. The second, v^j_{i+1}, is associated with the belief of v in S_{i+1}, and it is determined by either:

- $v^j_{i+1} = v^j_i + D^j_{i+1}$, if op is +=;
- $v^j_{i+1} = D^j_{i+1}$, if op is =.

The BN is key to determining whether a plan meets the required confidence level θ. An action a is applicable in a state S_i if $Pre(a)$ is satisfied. A sequential (linear) solution is a sequence of steps $[a_0, .., a_n]$, implying a state trajectory $[I, S_0, .., S_n]$. We use the BN to ensure that with $P \geq \theta$, in a given execution of the plan, each action's preconditions are met and S_n satisfies any hard goals.

The state progression formalism of Beaudry *et al* was adopted and extended by Coles (2012) as the basis of an over-subscription planning approach. A forward-chaining planner following these semantics was used to find a single plan, onto which branches were added by making additional calls to the planner. A range of other approaches have been adopted for planning under uncertainty, such as those based on the use of Markov Decision Processes, e.g. (Meuleau et al. 2009; Mausam and Weld 2008; Rachelson et al. 2008); these approaches are particularly useful when a *policy* needs to be found. For this paper, as our contribution is on the heuristic inside a forward-chaining planner, our focus will be on planning under the semantics of RTU described above.

3 Relaxing Numeric Uncertainty

In deterministic forward-chaining numeric planning, one way to guide search is the Metric Relaxed Planning Graph (RPG) heuristic (Hoffmann 2003). This performs a forward reachability analysis that estimates the number of actions needed to reach goals by relaxing the effects of actions. For numeric state variables, this amounts to estimating reachable bounds on the values of variables, by optimistically assuming that increase effects only increase the upper bound, and decrease effects only decrease the lower bound.

When working with RTU's semantics, Coles (2012) adapted this to assume for heuristic purposes that each variable takes its *median* value. From Jensen's inequality, we know that if $\theta \geq 0.5$, this is guaranteed to be a relaxation. However, as θ becomes large, it also means the heuristic is increasingly unrealistic: a numeric condition might be true assuming variables take their median values; but not when accounting for the uncertainty in their values. In this section, we will present two strategies that improve on this:

- we incorporate the shape of the distribution on variables' values in the heuristic evaluation, rather than discarding it and using the median;
- for Gaussian distributions, we explicitly track the uncertainty of variables in the relaxed planning graph.

3.1 Heuristic Guidance with Monotonically Worsening Uncertainty

Uncertainty can affect problems in two ways: it either gets worse monotonically (error accumulates and no action can rectify it); or it may be purposefully corrected (there may be actions that reduce the error, such as recharging batteries to a fixed value, or visiting a precise weighing station).

We first discuss the case of monotonically worsening uncertainty. Outside the heuristic, each precondition is of the form $\mathbf{w}.\mathbf{v} \geq c$, and a Monte Carlo simulation is used to estimate the probability distribution of $\mathbf{w}.\mathbf{v}$. Using this distribution, we can test whether the condition is satisfied with probability θ, i.e. whether the $(1-\theta)$'th percentile of $\mathbf{w}.\mathbf{v}$ is $\geq c$. We represent this percentile as follows:

$$p_{1-\theta}(\mathbf{w}.\mathbf{v}) = median(\mathbf{w}.\mathbf{v}) - \mathit{offset}_\theta(\mathbf{w}.\mathbf{v})$$

In effect, offset_θ is the margin of error that must be tolerated, for the precondition to be true with probability θ. We illustrate the intuition behind this margin in Figure 1. The condition itself can then be rewritten:

$$median(\mathbf{w}.\mathbf{v}) \geq c + \mathit{offset}_\theta(\mathbf{w}.\mathbf{v})$$

We define that uncertainty is *monotonically increasing* if offset_θ *can never decrease*. In this case, it is still a relaxation to use the offset values when determining which preconditions are true in the heuristic – the only way to make the condition true would be to apply actions that affect the values of $\mathbf{v}$, as no actions that decrease offset_θ exist.

An illustrative example would be an autonomous car with a certain amount of fuel, which is used gradually until it runs out; refueling is not possible. The activities performed by the car (e.g. start engine, accelerate, stand still, park) each require fuel, but the amount varies non-deterministically. As the plan is constructed, uncertainty and hence offset_θ accumulates monotonically. We can thus heuristically evaluate a state by assuming offset_θ is constant, and takes its current value; this is guaranteed to be a relaxation, as offset_θ can never become smaller.

3.2 Heuristic Guidance with Gaussian Uncertainty

So far, we explained how to incorporate distributions on the left-hand side of preconditions $(\mathbf{w}.\mathbf{v})$ into heuristic computation, by using the offset_θ value to capture uncertainty information. The relaxation holds when error accumulates and cannot be lowered. However, problems may contain actions such as *recharge-batteries* or *visit-weigh-station*, which reduce uncertainty.

The challenge in these sorts of problems is to ensure the heuristic remains a relaxation. This is possible in a useful subset of domains, where the uncertainty is due to independent Gaussian-distributed effects on variables, and therefore

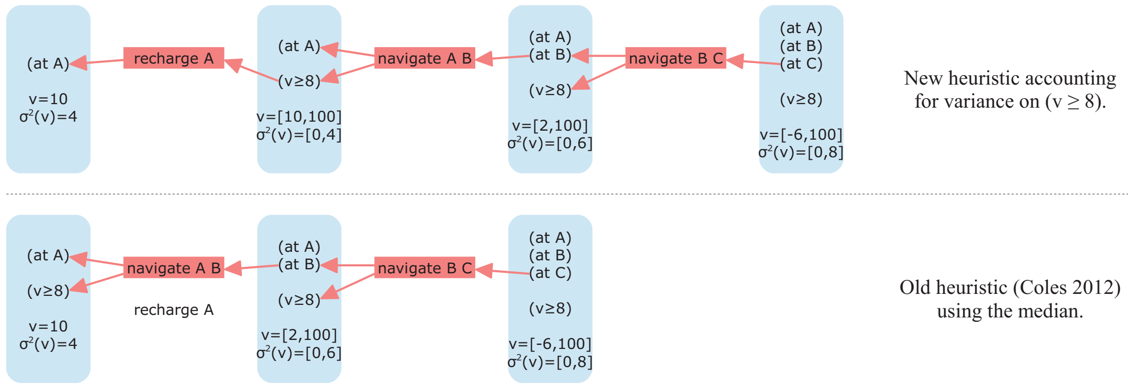


Figure 2: Example Relaxed Planning Graph, comparing the heuristic here to that in Coles (2012).

has an analytic form. We can utilize this form and extend the Metric RPG to additionally track the variance on each variable, $\sigma^2(v)$. The expansion phase, building the RPG, proceeds as follows:

- For each variable $v \in \mathbf{v}$, we track the upper and lower bound on its median value. In the first RPG layer, these are equal to the value of v in the current state S. We additionally track $\sigma^2(v)$, the variance on v. In the first RPG layer, this is the value according to the BN for S.
- In a regular RPG, if a numeric effect is applied that increases (decreases) some $v \in \mathbf{v}$, the upper (resp. lower) bound on v at the next fact layer is updated accordingly. Now, additionally, if a numeric effect decreases $\sigma^2(v)$, the lower bound on $\sigma^2(v)$ at the next fact layer is decreased[1].
- To decide which actions are applicable in each layer, we take variance into account when checking precondition satisfaction, as follows. For a precondition of the general form $\mathbf{w}.\mathbf{v} \geq c$, we can use the additive properties of Gaussians to compute the variance of $\mathbf{w}.\mathbf{v}$:

$$\sigma^2(\mathbf{w}.\mathbf{v}) = \sum_{w.v \in \mathbf{w}.\mathbf{v}} w^2.\sigma^2(v)$$

We obtain the offset using the Gaussian quantile function:

$$\mathit{offset}_\theta(\mathbf{w}.\mathbf{v}) = \sigma(\mathbf{w}.\mathbf{v}).\Phi^{-1}(\theta)$$

Hence, from Section 3.1, the precondition becomes:

$$\mathit{median}(\mathbf{w}.\mathbf{v}) \geq c + \sigma(\mathbf{w}.\mathbf{v}).\Phi^{-1}(\theta)$$

This gives us everything we need to build an RPG. We can be confident that the offset_θ values used are relaxations, because smaller values of variance result in smaller values of the Gaussian quantile function Φ^{-1}; and the semantics of the RPG guarantee we will underestimate variance.

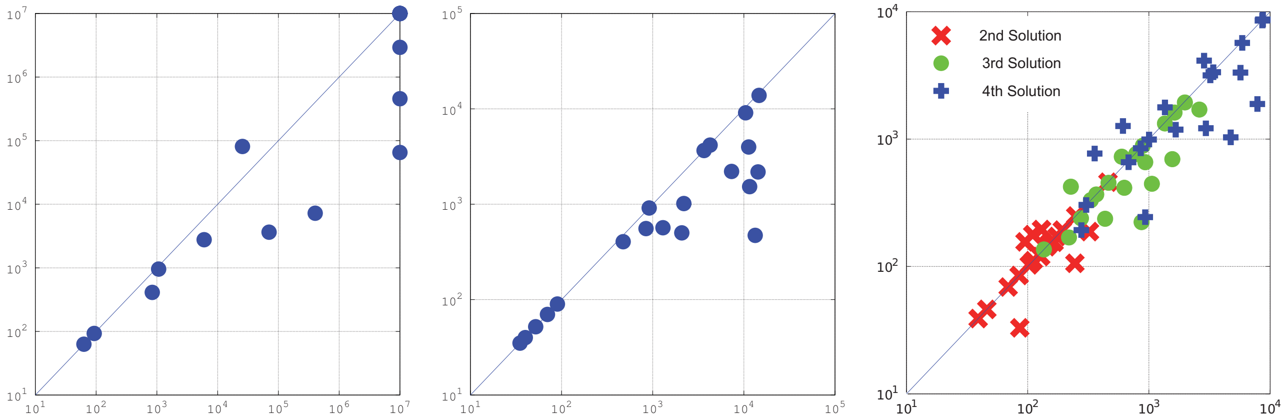
The next step is to extract a relaxed plan from the RPG; we illustrate this in Algorithm 1. The first thing to note is on lines 5 and 6, where we compute the offset_θ necessary for the condition to be met. Actions are then chosen in the standard way to attempt to meet the precondition, given this value of offset_θ. Then, if line 13 is reached and the precondition is still not true, it must mean that a decrease in variance caused it to become true at layer l (having been false at layer l-1). We now need to choose actions that decrease variance enough to achieve this. On line 15, we work out what offset_θ needs to be reduced to in order to make the precondition true; we then compute its corresponding variance on line 16. This variance can then be used to construct a new condition to be satisfied at this layer: this causes actions to be added to the relaxed plan in order to reduce variance on a later iteration of the loop.

[1] Effects increasing $\sigma^2(v)$ are ignored. If $\theta \geq 0.5$, adding more uncertainty never contributes towards preconditions becoming true, so it suffices to track only the smallest reachable values of variance.

Algorithm 1: RPG Solution Extraction

Data: RPG, a relaxed planning graph
Result: p, a relaxed plan
1 $\mathit{last} \leftarrow$ last layer index in RPG;
2 $\mathit{goals}[\mathit{last}] \leftarrow G$ *(i.e. the problem goals)*;
3 **for** $l \in [\mathit{last}..0]$ **do for** $(\mathbf{w}.\mathbf{v} \geq c) \in \mathit{goals}[l]$ **do**
4 $\quad \mathit{prev} \leftarrow$ max value of $\mathbf{w}.\mathbf{v}$ in fact layer l-1;
5 $\quad \mathit{prev\_}\sigma^2 \leftarrow$ min value of $\sigma^2(\mathbf{w}.\mathbf{v})$ in fact layer l-1;
6 $\quad \mathit{prev\_offset}_\theta \leftarrow \mathit{prev\_}\sigma.\Phi^{-1}(\theta)$;
7 $\quad$ **if** $\mathit{prev} \geq c + \mathit{prev\_offset}_\theta$ **then**
8 $\quad\quad$ add $(\mathbf{w}.\mathbf{v} \geq c)$ to $\mathit{goals}[l$-1$]$; **continue**;
9 $\quad$ **for** $(w.v) \in \mathbf{w}.\mathbf{v}$ *where* $w \neq 0$ **do**
10 $\quad\quad$ Choose actions from action layer l-1 that increase $(w.v)$;
11 $\quad\quad$ Add them to the relaxed plan and subtract their effects from c;
12 $\quad\quad$ **if** $\mathit{prev} \geq c + \mathit{prev\_offset}_\theta$ **then break**;
13 $\quad$ **if** $\mathit{prev} \geq c + \mathit{prev\_offset}_\theta$ **then**
14 $\quad\quad$ add $(\mathbf{w}.\mathbf{v} \geq c)$ to $\mathit{goals}[l$-1$]$; **continue**;
15 $\quad \mathit{max\_offset} \leftarrow \mathit{prev} - c$;
16 $\quad \mathit{max\_}\sigma^2 \leftarrow (\mathit{max\_offset}/\Phi^{-1}(\theta))^2$;
17 $\quad$ add $(-\sigma^2(\mathbf{w}.\mathbf{v}) \geq -\mathit{max\_}\sigma^2)$ to $\mathit{goals}[l]$;
18 $\quad$ add $(\mathbf{w}.\mathbf{v} \geq \mathit{prev})$ to $\mathit{goals}[l$-1$]$;

As a result of the algorithm described above, the relaxed plan now contains uncertainty-reducing actions. This makes



(a) Rovers (b) TPP (c) AUV

Figure 3: Nodes generated to solve problems in the three evaluation domains. Axes are logarithmic, comparing prior work (X axis) with the new heuristic (Y axis). The Two-Tailed Wilcoxon Signed-Rank Test confirms results are significant to $P \geq 0.95$.

for a better-informed heuristic, which is able to provide improved guidance and dead-end detection to the search, as will be demonstrated in the following section.

Figure 2 shows an example relaxed planning graph for a planetary rovers domain, where navigation uses power (v) and increases $\sigma^2(v)$. In the first layer, the only applicable action is recharge: whilst there are 10 units of power, the condition $v \geq 8$ is not true until the next action layer, as $\sigma^2(v)$ is too high to allow it to be true with confidence θ. Conversely, without our modifications to the heuristic, the RPG would effectively be offset by one layer: the fact and action layers below the dotted line would apply, and $v = 10$ would satisfy $v \geq 8$ as the value of $\sigma^2(v)$ is not taken into account when determining which numeric conditions are true. The effect is that the new relaxed plan recognises the need to recharge, whilst the old one would not. If (recharge A) was unavailable, then the modified heuristic detects a dead-end which the old one does not: there would be no way of making $v \geq 8$ true, accounting for $\sigma^2(v)$.

4 Evaluation

We evaluate on three domains: Rovers and AUV from (Coles 2012); and a variant of TPP from (Gerevini et al. 2009). In Rovers, the activities of a planetary rover are constrained by battery usage, which has Gaussian uncertainty, and the battery can only be recharged at certain locations. In TPP, the domain is modified to model the acquisition of sufficient amounts of bulk materials (e.g. coal), and trucks can visit weighing stations at some suppliers (Khajiit) to top up or shed excess load, which reduces uncertainty. AUV is an over-subscription problem where the activities of an underwater vehicle must be planned with a strict bound on total time taken, and with normally distributed activity durations. Tests were performed on 3.5GHz Core i5 machines with a limit of 4GB of memory and 1800s of CPU time. We compare our new heuristic with the metric RPG-based heuristic used by Coles (2012).

Overall, the new heuristic leads to a substantial reduction in nodes generated to solve problems, and time taken: scatterplots for nodes generated are shown in Figure 3, and the time-to-solve scatterplots have the same shape. The extra computational work (tracking variances etc.) does not adversely affect the time taken to heuristically evaluate a state. Thus, because significantly fewer states are generated, and per-state evaluation times are comparable, the performance of the planner is significantly better.

For the Rovers domain (Figure 3a), most striking are the points on the far right of the graph – these indicate problems that were previously unsolvable but can now be solved. In part, this is because the new heuristic is able to recognize many more states as being dead ends, because it does not disregard uncertainty on the battery level when evaluating preconditions. In contrast, by ignoring uncertainty, the old relaxed plans relied on moving somewhere to recharge, even though in reality uncertainty made it impossible for that move action to be applied. The new heuristic often avoids this pitfall by accounting for uncertainty to a greater extent.

In TPP (Figure 3b), all the problems could be solved by both the old and the new heuristic. However, by not accounting for uncertainty, the old heuristic can reach states in which the relaxed plan does not need to buy any more of any goods. In these states, the heuristic value is 0. As acquiring additional goods requires combinations of travel and buy actions, a substantial amount of search must be performed with no effective heuristic guidance. Unlike Rovers, there are no dead ends due to these travel actions, so this blind search will succeed, but is very time consuming – in problems furthest from the line $y = x$, the majority of nodes evaluated have an old heuristic value of 0.

AUV is an over-subscription problem: search reports a solution plan every time it finds one that solves more goals than the best so far. We are hence interested in the search effort to find progressively better solutions. Figure 3c compares the nodes generated by each configuration to find the 2nd, 3rd and 4th solutions. (These correspond to satisfying 1, 2, and 3 goals respectively.) The relaxed plans produced by the old heuristic, by ignoring uncertainty, more often use actions that there is actually no time to complete. Disregarding uncertainty is less of an impediment than in Rovers and TPP, as there is no scope for planning actions that reduce uncertainty (unlike battery charge or goods purchased, actions cannot create more time). Nonetheless, the new heuristic is generally able to find better solutions more quickly. If left to run for long enough, search with the old heuristic will tend to find solutions as good as search with the new heuristic, but loses out earlier in the search.

As a concluding remark for our results, we note that so far we assumed $\theta = 0.99$. At $\theta = 0.8$, the improvements from using the new heuristic are still noticeable, but not as substantial. By $\theta = 0.6$, which is close to the median ($\theta = 0.5$), there is no statistically significant difference between the two, as uncertainty has only a modest effect on the heuristic, or indeed search itself. This confirms that our heuristic meets our headline aim of being able to better guide the planner when the consequences of uncertainty bear a significant effect upon what is a reasonable solution plan.

5 Conclusions

In this paper, we presented a novel search heuristic that extends the Metric Relaxed Planning Graph to include information about uncertainty, in a useful subset of problems. For cases where uncertainty is monotonically increasing, we note how $offset_\theta$ values obtained from the Bayesian network for a state can be incorporated into the heuristic evaluation for that state, to better reflect the margin of error that must be allowed for when determining whether preconditions are true. For cases where actions are available to reduce uncertainty, and uncertainty is Gaussian, we detail how the variance on variables' values can be explicitly modeled in the RPG; and how RPG expansion and solution extraction can be updated to build relaxed plans that use such actions.

The promising results indicate that including information about uncertainty in this way can improve the performance of forward-chaining planning, when the aim is to find a single plan that is overwhelmingly likely to succeed. In future work we will revisit and extend the ideas of using such a planning algorithm as a kernel within a plan-with-branches approach (Coles 2012), where sensing actions at execution-time can choose which branch to take; and as an extension of work on contingent planning with discrete (rather than continuous) outcomes on actions' effects (Muise, Belle, and McIlraith 2014; Albore, Palacios, and Geffner 2009; Little, Aberdeen, and Thiébaux 2005), and work on online probabilistic planning (Yoon et al. 2010).

Acknowledgements

We extend our gratitude to Amanda Coles for her support throughout the development of this work; and to Christian Muise for his invaluable insights into non-deterministic action outcomes. Liana Marinescu's work is funded by a scholarship awarded by the Department of Informatics, King's College London.

References

Albore, A.; Palacios, H.; and Geffner, H. 2009. A translation-based approach to contingent planning. In *The 21st International Joint Conference on Artificial Intelligence*.

Beaudry, E.; Kabanza, F.; and Michaud, F. 2010. Planning with Concurrency under Resources and Time Uncertainty. In *Proceedings of the European Conference on Artificial Intelligence*.

Coles, A. J. 2012. Opportunistic Branched Plans to Maximise Utility in the Presence of Resource Uncertainty. In *Proceedings of the European Conference on Artificial Intelligence*.

Gerevini, A.; Long, D.; Haslum, P.; Saetti, A.; and Dimopoulos, Y. 2009. Deterministic Planning in the Fifth International Planning Competition: PDDL3 and Experimental Evaluation of the Planners. *Artificial Intelligence*.

Hoffmann, J. 2003. The Metric-FF Planning System: Translating Ignoring Delete Lists to Numeric State Variables. *Journal of Artificial Intelligence Research* 20:291–341.

Little, I.; Aberdeen, D.; and Thiébaux, S. 2005. Prottle: A probabilistic temporal planner. In *The 20th AAAI Conference on Artificial Intelligence*.

Mausam, and Weld, D. S. 2008. Planning with Durative Actions in Stochastic Domains. *Journal of Artificial Intelligence Research* 31:38–82.

Meuleau, N.; Benazera, E.; Brafman, R. I.; Hansen, E. A.; and Mausam. 2009. A Heuristic Search Approach to Planning with Continuous Resources in Stochastic Domains. *Journal of Artificial Intelligence Research* 34:27–59.

Muise, C.; Belle, V.; and McIlraith, S. A. 2014. Computing Contingent Plans via Fully Observable Non-Deterministic Planning. In *The 28th AAAI Conference on Artificial Intelligence*.

Rachelson, E.; Quesnel, G.; Garcia, F.; and Fabiani, P. 2008. A Simulation-Based Approach for Solving Temporal Markov Problems. In *Proceedings of the European Conference on Artificial Intelligence*.

Yoon, S.; Ruml, W.; Benton, J.; and Do, M. 2010. Improving determinization in hindsight for online probabilistic planning. In *The 20th International Conference on Automated Planning and Scheduling*.