Managing Concurrency in Temporal Planning using Planner-Scheduler Interaction: Response to Reviewers

May 22, 2008

1 Remarks

Many thanks to the reviewers for their very insightful and helpful comments on the first submitted draft of the paper. We have attempted to address all the criticisms and answer questions raised in the reviews. We believe that this has led to a much stronger and clearer paper. To simplify the task of the reviewers, we have annotated their original comments, below, with an indication of our responses.

2 Reviewer 1

> One of the difficulties that I have with the current draft of the paper is that nowhere in it will you find a statement as precise as given above, regarding what the authors view as the contribution of the paper.

We have rewritten the introduction and abstract to provide a clearer statement of the contribution of the paper.

> There are a number of problems with terminology in the paper. One example is the use of the term ”coordination.” To start with, the term is used to refer to two very different things: the interaction between a planning function and a scheduling function (p. 7, p. 45) and to refer to something approximating what Cushing et al. refer to as ”required concurrency” (everywhere else).
>
> The authors’ intended relation between coordination and concurrency is hard to determine. On page 5, they claim that required concurrency can take two forms. In their type 4 domains, ”concurrency is required in order to achieve goals by their deadlines.” In type 5 domains, ”concurrency must happen not only to achieve a goal by a deadline, but to achieve a goal at all.” But unless the authors intend that type 5 domains do not include metric time (and thus are not ”temporal”), it is not obvious that types 4 and 5 are distinct.

We agree that this created problems. “Coordination” was a term we used before the “required concurrency” expression and with a similar meaning. However, it does not seem helpful to proliferate terms, so we have rewritten section 2 and also replaced the use of “coordination” throughout.

> It is straightforward to compile deadline goals into exactly the kind of ”logical” interaction that this paper defines as coordination (type 5), such that there is a deadline or makespan action with which other actions in the plan will interact in terms of preconditions and effects. If the authors wish to claim that ”deadline-only” concurrency is strictly easier than domains where there are explicit interactions among the actions in the problem as stated, they need to provide or cite some argument to that effect.

We have added the explicit observation to the paper that deadline-only required concurrency (which was actually excluded from that term in the definition by Cushing *et al*) is easier. The reason is that this form of concurrency can be handled using compressed actions, since there is no complex interaction caused by the individual end points of the durative actions. This simplifies the planning process, even if there is some subsequent scheduling work, and even an iterative replanning process in response to scheduling failure.

> Another problem with the paper is the category confusion related to the authors’ use of the terms terms ”planning” and ”scheduling.” The authors conflate an approach or technique and a separate system, for example referring to the interaction of a ”planner” and a ”scheduler,” as opposed to presenting an approach that borrows data-structures, algorithms, representational tricks, or other techniques from either community as appropriate. In my understanding of temporal planners such as SGPlan or MIPS that do handle inherently-sequential domains, their architectures do not include separate, interacting ”planners” and ”schedulers,” but rather integrate techniques. The relevance of this point is strengthened by the fact that, as an examination of the architecture picture on p. 33 will show, the ”planner” in CRIKEY_SHE is using an STN to check envelope consistency, which the ”scheduler” is doing as a postprocessing step the same kind of computation that MIPS, SGP, and other temporal planners are (I believe) doing incrementally at each search node.

We have cleaned up our text throughout to remove this confusion. MIPS and SGPlan use less powerful forms of scheduling than the STN, but the basic point is valid: the key lies in the integration of techniques, not complete systems.

> This paper is also made more difficult to read, and even somewhat incoherent, by the confusion between an inherent property of a domain and a choice of solution approach (a ”pure planning problem” as opposed to a problem domain where data-structures and algorithms commonly described as ”planning” may prove to be effective. As pointed out in previous papers by some of the authors of the current paper, PDDL2.1 includes

constructs that can be used to represent both discrete and continuous resources. Consequently, it is possible to represent such strongly scheduling-flavored domains as a sequencing problem, in which the task is to find an ordering among activities that individually consume and replenish any of several interacting resources. This problem does not require concurrency to be computationally hard, and can be stated in terms of either optimization (minimizing makespan, e.g.) or satisfiability (does a feasible ordering exist?), which does not even require metric time. So: is this a ”pure planning” problem or not? Certainly, it belongs to a class of problems that have been studied by researchers in any of several ”scheduling communities” for quite a long time.

Of course, there *is* an inherent domain property that the authors can and do usefully refer to: inherent sequentiality versus required concurrency (for completeness, we need also to talk about ”Mixed” problem problem classes, in which some problems have required concurrency, while others do not).

The third problem with the paper related to terminology also has to do with Figure 1 on page 4 and the related discussion. The discussion of a ”spectrum” (which isn’t really a spectrum) of imprecisely described ”integrations of planning and scheduling” adds little or nothing to the paper. For example, the authors define one end of their spectrum (1p) to be ”pure planning” problems. No definition is given for this term, other than the following:

”On the left, (type 1p) are pure planning problems that contain no scheduling. These include classical planning benchmark domains.”

There are at least two problems with this as a definition. First, ”classical planning benchmark domains” does not define a language or otherwise characterize a class of planning problems. If the authors intended by this to limit type 1p to ADL, or even to STRIPS, then we come to my second objection, which is that, as described above, these languages are sufficiently expressive to define problems indistinguishable from sequencing problems.

Having read and re-read the discussion attendant on Figure 1, I am unable to elucidate a crisp definition for the categories 1p, 1s, 2a or 2b (how is a component that ”creates no consequence for either the satisfiability or the quality of the solution” actually part of the problem?), or 3b. Type 3a the authors identify as containing (and co-extensive with?) inherently sequential temporal planning problems. As argued elsewhere in this review, I do not believe that the authors have provided a meaningful differentiation between types 4 and 5. Nor have they stated whether one or both of those categories include all of, or anything other than, problem domains that require concurrency. And even that definition is somewhat imprecise: in the IJCAI-07 paper by Cushing et al. (cited in this draft), required concurrency is a feature of several different language classes, without a common

> ”minimal element” or other comprehensive defining characteristic.

We believe that these issues have been addressed by our rewriting of section 2 and a general improvement of the text throughout.

> With regard to related work, there is a long tradition of temporal planners that is totally ignored here. That is not a failing unique to this paper; since the IPC really got going, the community that has grown up around it has sometimes had a tendency to ignore earlier work, as well as work that uses languages other than than PDDL or different approaches from those for which the IPC domains are tuned. Just as a brief list of previous planning systems that could handle metric time and resources (so, very much ”integrating scheduling” as the term is used here), there was Tate’s work (Nonlin, OPlan, OPlan2, INOVA), Vere’s (Deviser), James Allen’s planner using the Interval Calculus, Forbin (Firby, Dean, and Miller), IxTeT, HSTS/Europa, Aspen, and Wilkins’ SIPE. From the scheduling community, there are quite a few systems that include some planning capability, but with a sufficiently different emphasis that while I’d like to see it mentioned as a bridging approach from the other side, its lack is more of an irritant than a flaw in the paper.

We have added a new section on related work (section 10) which we hope addresses these comments.

> Finally, I wish the authors had been more precise about the relation among envelopes, resources and required concurrency. Yes, an envelope can be used to keep track of the window over which a resource is available, but therein lies an interesting point: If the (possibly durative) action making the resource available and the (again, possibly durative) action taking it away again are not the same, then an envelope can be constructed which is not enforcing any kind of required concurrency I can see. An ordering relationship, sure.
>
> This more general view raises the possibility that the authors could actually have claimed a larger contribution than they do: they may have developed an approach that is to my knowledge the first empirically-validated means of adding efficient reasoning about resources to forward planning. That would be really neat. But in the current paper the claim is neither made nor fully supported.
>
> The more general view might also steer the authors away from statements such as ”...clearly, ending an action cannot create a window of opportunity in which something can happen.” (p 22) As an assertion about the general case, without a much more precise definition of terms, that statement is contradicted by pretty much everything I know about planning and scheduling.

We have clarified terms throughout and also tightened up the discussions of the management of resources. We believe that our modifications have addressed these issues. It is true that an envelope can be created in a situation in which no concurrency is enforced and we have clarified the circumstances in which an envelope becomes a constraint, providing a theorem (theorem 2) that demonstrates this.

> The paper takes a long time to get around to specifics. The only architecture picture appears on p. 33. Motivating examples other than the match light domain are almost entirely missing, and would help to set intuitions, for example as to the difference between SHE and more general envelopes. Figure 2 is an exception to this, in a positive way.

We have added more motivational material, but this does have the effect of extending the preamble. We have added another architecture figure earlier, showing the structure of CRIKEY itself, and other figures intended to help clarify the behaviour of envelopes and temporal relations.

> The introduction of the term ”snap-action” feels gratuitous. Snap-actions are just conventional (non-durative) classical planning actions. On the other hand, since this formalism is being set up as a strawman to be replaced by envelopes, I don’t care that much.

We spent some time discussing this. We have added a comment to explain why we think it necessary to introduce a specific term — the problem being that “action” is ambiguous when we want it to refer to durative actions sometimes and to the special actions formed by the translation of a durative action into component instantaneous actions at other times. We have switched to “instant-actions” instead of “snap-actions” because we decided that “snap-actions” might carry the wrong intuitions. We have carefully checked the consistent use of our terms.

> On p. 14, ”Formally, an envelope and its contents...” That should probably read ”INformally, ...” It is also somewhat misleading, e.g. in defining an envelope and contents as comprising ”a sequence of actions.”

We’ve dealt with this.

> The authors need to provide intuitions regarding the different definitions and functions of envelopes. SHE is easy, but the more general case is more difficult to picture without an example. For that matter, how about *starting* w. SHE and then, once that is understood, extending to the more general case(s)?

We thought about this quite a bit. We are convinced that the general case is easier to motivate from the FF-LPGP direction. We still think that the special case is easier to understand as a hobbled variant of the full version. We’ve tried to address this by strengthening the material on envelopes and on CRIKEY.

> p 15 - should the ”or” in the last line of definition 6 be ”and”? It seems like it, but I may be missing something.

Should be “or” — we’ve added an explanation of the point, since on returning to it we had to think it through again ourselves!

> p. 24 - the discussion motivating use of Floyd-Warshall is incorrect. If all that is desired truly is to compute earliest start times, Bellman-Ford can be computed STARTING AT X_0. The resulting path-lengths provide earliest start times. Floyd-Warshall is needed only if what is desired is bounds (upper, lower, or both) for all pairs of time points appearing in the STN.

We have removed this problem by simply indicating we use standard STN algorithms. In fact, we used FW for other reasons (during the STN checking we don’t necessarily have a single start point — see the explanation in the text associated with the last point) and the text got mangled in the draft.

> The claim at the beginning of Section 8.1 (p 31) regarding the ”vast majority” of action interactions is unsupported, for anything stronger than application to IPC benchmark domains, which represent among other things a systematically biased set of modeling choices.

Reworded to make a point we believe is supported.
Other minor points have all been addressed as proposed.

3 Reviewer 2

> As far as completeness is concerned, it would be good to state early in the paper (in the introduction?) that Crikey, although ’more’ complete than the planning systems mentioned in the paper, is not complete. This is only mentioned at the end of the conclusion although it is an important fact and the beginning of the paper seems to indicate that Crikey is complete. On this question of completeness,

We have clarified this.

> The claim that there is no implemented solution to the problem of temporal planning in the domains considered in the paper should be softened. For instance, the IxTeT system is able to tackle this type of problem where temporal concurrency is needed and the algorithm is complete. The system would have no problem solving the Match domain.

We have added the related work section to acknowledge this and to make the claim more precise: that our planner is the first PDDL2.1 planner able to handle required concurrency correctly.

> Section 4 introduces the $<$max and $<$min relationships but these relationships are not used later in the paper. One sees the link with the definition

> of envelopes in section 6 but these relationships are never explicitly mentioned there. Furthermore, although it is quite intuitive, there is no proof that these constraints cannot be properly handled with action compression so maybe the first sentence in the beginning of section 5 should be rephrased.

We decided that this was an important point and have added two proofs that tie these definitions firmly to our envelopes and other material. We believe this has strengthened the paper.

> Is there a risk of explosion of the number of open envelopes in Crikey? It would be good, if possible, to give an idea of the maximum number of open envelopes: is it quadratic with the number of actions currently in the schedule or more? what are the 'pathological' cases, etc.

The potential is for there to be quadratic growth and we have added text to explain this and to comment on the cases where it might happen (section 7.3.1).

> In the scheduling part, did you investigate an approach that also do some partial order lifting on the resource fluents, starting from the total order to build a partial order by removing some precedences. Something like the POS generation approached used in [2]. I think that could be interesting and could speed up the scheduling part.

We were satisfied that the simple partial order lifter worked adequately for our needs, but we agree that this is an interesting area to explore. We have added a comment to this effect and reference. Thanks for the pointer to the work.

> The explanation why Floyd-Warshall algorithm is needed to compute the earliest possible start time for each action is not clear. Why a single-source shortest path starting from X0 is not enough? Why would you need to run it from each action?

We have dealt with this (see reply to reviewer 1).

> p2: "planning with real rather than relative time". The term 'real' sounds strange (relative time is also real). Maybe you should use "quantitative" v.s. "qualitative" as in the temporal reasoning community.

We have rewritten the introduction and believe this point is addressed.

> end of p5- beginning of 6: the explanation is not clear at this point of the paper why the type of the problem changes depending on the language it is modeled in.

We have rewritten section 2 completely and we believe this point no longer applies.

> p9: in the end of the page, maybe it would be useful to give an example of the use of dummy propositional effect to ensure the precedence.

Explanation and figure added.

> p12: it is not said what P-inv means. It should be said that it is a a dummy propositional effect to ensure the precedence. Furthermore, I do not understand why P-inv needs to be in the add list of P$<->$. Is it a typo?

This has been simplified and clarified.

> p32: def16: in the formula I guess one should read an intersection instead of an inclusion.

Correct! This has been replaced.

> p33: figure 13 is quite ugly with text being intersected by the shapes.

Yes, we have replaced it.

> p55: I think it should be mentioned in section 12.1 that because of the loose coupling between planning and scheduling, optimality cannot be proved.

We have tightened up the text and believe we have dealt with this.