Universal Planning Research: A Good or Bad Idea?

Matthew L. Ginsberg

Computer Science Department
Stanford University
Stanford, California 94305

Rather than begin by discussing the points where I seem to be in disagreement with Chapman and Schoppers, let me start with something about which we seem to concur: The work on reactive plans can be broken into work on "pure" reactive plans, which specify actions for an agent to take in all situations, and "caching" reactive plans, which specify actions in only some instances and are used primarily to store the results of previous planning activity. Although Chapman would object to the use of the word "plan," the basic distinction is one that his reply appears to sanction.

Very strongly put, the argument in my initial paper boiled down to the following two claims:

1. Research on pure reactive plans is not likely to have significant impact on AI because pure reactive plans cannot be used as an architecture on which to build a cognitive agent.

2. *Current* research on caching reactive plans is not likely to have significant impact on AI because the researchers are not addressing interesting problems.

Chapman disagrees with the first of these claims, while Schoppers disputes the second (and also the first, although only halfheartedly). Let me discuss them individually.

1 Pure reactive plans

I made two arguments to support the claim that pure reactive plans will have little lasting impact on AI. They were the following:

1. In the abstract, it is extremely unlikely that the desired behavior of an intelligent artifact can be captured in a pure reactive plan, simply because there are too many possible behaviors that one might wish to capture.

2. Pure reactive plans cannot improve their performance by expending additional computational resources. This appears to be a crucial property of intelligent behavior.

Neither of the respondents addresses the second of these points at all. With regard to the first, Schoppers suggests that there is good reason to believe that the desired behavior of an intelligent agent *can* be modelled with a pure reactive plan (his answer 2, although he discounts the entire issue in answer 5), while Chapman appears to argue that although *optimal* behavior cannot be realized in a pure reactive plan, perhaps it will be possible to obtain behavior that is "good enough".

Schoppers' argument is based on the fact that we have, to date, found no problem that is provably non-polynomial in its inputs. But in light of the fact that whether $P = NP$ is an open question, this is hardly a surprise. There are *many* problems that we *expect* to be non-polynomial in their inputs – almost all of AI has this property. So I find Schoppers' argument completely unconvincing.[1]

[1] His arguments based on chess are even weaker. The chess endgames that have been solved by exhaustive analysis make up only a tiny fraction of the set of endgames generally.

Chapman's is more interesting. Essentially, his research program is to investigate *experimentally* the boundary of the set of problems that can be usefully tackled using pure reactive plans. My objection about the inability of such architectures to incrementally improve their performance stands, but the question is fundamentally an experimental one. Although I am pessimistic about the results, I applaud the efforts of Chapman and others (such as Rod Brooks) in pursuing it.

Indeed, Chapman shows in his reply that a problem that I had thought outside the scope of pure reactive planners (the fruitcake problem) in fact can be solved by them. Perhaps I should have expected this – after all, the fruitcake problem lacks the property of incrementality that I have mentioned – but the fact is that I did not and that Chapman's result is an interesting one.

2 Caching reactive plans

This brings us to caching reactive plans, which are discussed by Schoppers only. My argument here is not that caching is a bad idea, but that current *research* on caching reactive plans is poorly focussed. Schoppers seems to have missed this point entirely, and spends the bulk of his reply arguing that caching is a good idea. How could I disagree?

The argument I made in my original paper was not that caching was a bad idea, but that *research on the data structures used by the caching mechanism is a bad idea.* This is an argument that applies equally well to Schoppers' universal plans and to Nilsson's action nets.[2]

Schoppers' argument against this (his answer 8) is that the results of his research show that "a domain of 10^{12} states can be covered with 10^8 decision rules."

So what? I expect that I take about one cognitive action per minute. If we consider Schoppers' two examples of driving and making tea, it takes me about a minute to make a cup of tea and I expect that I "do something" about once a minute while driving. So my domain effectively has about 10^7 states (I have been alive for that many minutes) instead of 10^{12}. Storing 10^7 actions is not prohibitively expensive, so that reducing this number to 10^5 or so is unlikely to be of much value in practice.

But let us suppose for a moment that Schoppers is right and that we have in fact precomputed actions for many more states than my estimate of 10^7. It is still the case that in order for Schoppers' work to have any relevance *at all*, we need to have a way to generate the 10^7 or more plans that he proposes to cache; it is clear that generating these plans manually is out of the question.

Roughly speaking, then, the argument is as follows:

1. In the absence of machine generation of plans or action decisions, it is inconceivable that advances in caching technology will have a significant impact of the performance of AI systems.

[2] I would not apply it to Rosenschein and Kaelbling's work, since they appear to be more concerned with the behavioral properties of agents of this type than with underlying data types.

2. Even in the presence of machine generation of plans, it is unlikely that the number of plans generated will be large enough for advances in caching technology to be significant to the AI enterprise.

Work on caching technology is *definitely* not of use at the moment, and is unlikely to be of use in the future.

Schoppers concludes his reply with some optimistic suggestions about the eventual contributions of the approach he advocates; unfortunately, these claims are made virtually without support. Although I share his enthusiasm about the importance of planning to the AI venture, I seriously doubt that a focus on data structures will lead to significant progress. Instead, I would propose questions such as the following:

1. When should a system devote resources to the generation of plans to be cached, and which plans should be generated?
2. When should a plan be cached?
3. When is a cached plan relevant to a particular action decision?
4. Can cached plans be "debugged" in some way that extends their usefulness at runtime to situations outside those for which they were initially designed?

I would expect that progress on any of these issues – especially the last – would lead to substantial improvements in the behavior of situated automata.

Acknowledgement

I would like to thank David Smith for his comments on an early draft of this reply.