Scaling Decision Theoretic Planning

Daniel Bryce
dan.bryce@asu.edu
Department of Computer Science and Engineering
Ira A. Fulton School of Engineering
Arizona State University, Brickyard Suite 501
699 South Mill Avenue, Tempe, AZ 85281

Abstract

As classical planning branches out to consider richer models, many extensions approach decision theoretic models. Decision theory research uses models like MDPs and POMDPs which are very expressive, but can be difficult to scale. Whereas, planning research concentrates quite a bit on scalability. Our previous work and future doctoral thesis concentrates on extending the planning model toward problems characterized by partial observability, non determinism, non uniform cost, and utility.

Introduction

One of the major reasons recent planners are able to tackle problems where solutions have on the order of hundreds of actions is the advent of reachability heuristics. Reachability heuristics rely on a structure called the planning graph, originally introduced to perform search in GraphPlan [Blum and Furst, 1995]. While GraphPlan was able to outperform many of the planners at the time, it has been subsumed by a planners such as HSP [Bonet and Geffner, 1999], FF [Hoffmann and Nebel, 2001], and AltAlt [Nguyen *et al.*, 2002] that use reachability heuristics extracted from the planning graph to guide state space search.

The basic idea in a planning graph is to approximate all states at each depth of the state space progression tree by a set of literals. The approximation loses boundaries between states at a depth of the search, but keeps the literals that can be reached at each depth. The planning graph encodes a lower bound k on the number of steps needed to achieve a state with a literal l if l doesn't appear in the planning graph until layer k. While this lower bound is a useful heuristic, it can be improved by using a relaxed GraphPlan search to find actions that are needed to support the literal by back-chaining through the planning graph. This heuristic, called a relaxed plan, has been used with much success in many planners.

While we would like to use reachability heuristics in the more expressive planning models, the unmodified GraphPlan relaxation tends to break down. Our previous work, discussed in the next section, looks at extending the planning model to include partial observability, where the key challenge is generalizing reachability heuristics between sets of states (belief states). In the following section, on our current work, we discuss extensions that are needed to handle non uniform cost models for actions that are both causative and sensory. Finally, in the last section we discuss further extensions to stochastic belief states and actions and utility models.

Partial Observability

Planning problems with partially observable states can be posed as search in belief space, where search nodes are sets of states and solutions are conformant and conditional plans. We investigated using planning graph heuristics for search in belief space. Intuitively, it can be argued that the heuristic merit of a belief state depends on at least two factors–the size of the belief state (i.e., the uncertainty in the current state), and the distance of the individual states in the belief state from the goal (belief) state. The question of course is how to compute these measures and which are most effective. Generalizing classical planning heuristics that aggregate the cost of literals in a state to get a state to state distance, we can aggregate the cost of states in a belief state to get a belief state to belief state distance. We evaluated heuristics that make various assumptions about state independence, positive interaction, and negative interaction.

We [Bryce and Kambhampati, 2004] first tried a minimal extension to heuristics used in classical planning by considering heuristics from a single planning graph to guide search, which proved not to scale our planners very well. To improve the informedness of the heuristics, we tracked multiple planning graphs, each corresponding to one of the possible states in our belief. The number of planning graphs needed is exponential in the number of uncertain state literals. Hence, multiple graphs do not scale well as the size of belief states grow. The limitations in scaling involve either potentially running out of space to build planning graphs or spending too much time computing heuristics across the multiple planning graphs. Thus, we designed a new planning graph structure to addresses these limitations. The idea is to condense the multiple planning graphs to a single planning graph, called a Labelled Uncertainty Graph (LUG) [Bryce *et al.*, 2004]. Loosely speaking, this single graph unions the causal support information present in the multiple graphs and pushes the disjunction, describing sets of possible worlds, into "labels". The graph elements are the same as those present in multiple graphs, but much redundancy is avoided. For instance an action that was present in all of the multiple planning graphs would be present only once in the LUG and labelled to indicate that it is applicable in a projection from each possible world. We showed how several powerful planning graph-based heuristics from classical planning, including relaxed plans can be generalized to the case of multiple planning graphs and the LUG.

Our results showed that multiple graphs were needed but too costly, and that the LUG was able to capture the multiple graphs at a much lower cost. We found that both conformant planners and contingent planners can use these heuristics for improving scalability.

Cost Models

Currently we are working on heuristics that handle non uniform cost models when planning under partial observability. The reason cost models are interesting in partially observable problems are that planners need to be sensitive to the cost of sensing. For example, in planning medical treatment the cost of performing every test to perfectly diagnose a disease is too high, rather one may have prescribe several treatments at a lower cost, knowing at least one will work. We have taken the LUG from the previous section and defined a cost propagation procedure which allows us to extract cost aware relaxed plans.

Cost propagation on planning graphs, similar to that used in the Sapa planner [Do and Kambhampati, 2003], propagates the estimated cost of reaching literals at different times. The propagated costs enable relaxed plans to be more cost sensitive by caching the least-cost supporters for subgoals. Our situation is a bit more general because we propagate cost for a set of states (the states in a belief). The biggest challenge in our generalization is that it is possible for a literal to have different costs for every possible subset of states in our belief. Instead of tracking cost for all subsets, we partition states into fixed sets to track cost over. We propagate cost for each graph element, in terms of these sets, with a set of world group-cost tuples.

Using the cost aware heuristics in our planner, $POND$, we were able to trade off the amount acting under uncertainty versus sensing based on the cost model of the problem. We compared our planner using relaxed plans not based on cost (coverage), and our planner using relaxed plans based on cost (cost) to GPT [Bonet and Geffner, 2000], and MBP [Bertoli *et al.*, 2001], on the following domains.

Medical-Specialist: We developed an extension of the medical domain [Weld *et al.*, 1998], where in addition to staining, counting of white blood cells, and medicating, one can go to a specialist for medication and there is no chance of dying – effectively allowing conformant plans. We assigned costs as follows: c(stain) = 5, c(count_white_cells) = 10, c(inspect_stain) = X, c(analyze_white_cell_count) = X, c(medicate) = 5, and c(specialist_medicate) = 10. We generated ten problems, each with the respective number of diseases (1-10), in two sets where X = $\{15, 25\}$.

Our results in the first two columns in Figures 1, 2, and 3 show the expected cost, plan breadth, and total time for the two cost models. Extracting relaxed plans based on propagated cost, instead of coverage enables $POND$ to be more cost sensitive. The plans returned by the cost propagation method tend to branch less than coverage as the cost of sensing increases in order to reduce expected cost. Since MBP is insensitive to cost, the cost of its plans are proportionately worse as the sensor cost increases. GPT returns better plans than MBP, but tends to take significantly more time as the cost of sensing increases; this can be attributed to how their heuristic is computed by relaxing the problem to full-observability. Our heuristics measure the cost of co-achieving the goal from a set of states, whereas GPT takes the max cost of reaching the goal among the states.

Rovers: We use an adaptation of the Rovers domain from the International Planning Competition [IPC, 2003] where there are several locations with *possible* science data (images, rocks, and soil). We added sensory actions that determine the availability of scientific data and conditional actions that conformantly collect data. Our action cost model is: c(sense_visibility) = X, c(sense_rock) = Y, c(sense_soil) = Z, c(navigate) = 50, c(calibrate) = 10, c(take_image) = 20, c(communicate_data) = 40, c(sample_soil) = 30, c(sample_rock) = 60, and c(drop) = 5. The two versions have costs: (X,Y,Z) = $\{(35, 55, 45), (100, 120, 110)\}$.

The second two columns of Figures 4, 5, and 6 show the expected cost, plan breadth, and total time for the two cost models. We found that the coverage and cost based relaxed plan extraction guide POND toward similar plans, in terms of expected cost and number of branches. As sensing becomes more costly, POND is able to find plans with less branches to preserve a good expected cost. MBP, making no use of action costs, returns plans with considerably (a order of magnitude) higher expected execution costs, and does not adjust its branching as sensing cost increases. GPT fares better than MBP in terms of plan cost, but both are limited in scalability due to the weaker relaxations in their heuristics.

Future Work

With knowledge of how to propagate cost on the LUG, we are confident that we can propagate probabilistic information in a similar fashion to extract relaxed plans reflecting the stochastic dynamics of a problem. A relaxation we would like to explore is the effectiveness of using order of magnitude approximations of probability in the propagation.

We also intend to extend the methods mentioned up to this point with models that have goal utility. There exists work by our research group on handling goal utility in planning graphs [van den Briel *et al.*, 2004]. It is our intent to combine these algorithms into construction of the LUG and relaxed plan extraction.

Handling each feature of decision theoretic planning individually sheds light on the larger problem, and it is our plan to incrementally study and combine heuristic techniques for each feature to create a holistic heuristic method. With such a heuristic method for decision theoretic planning, we intend to approach problems that were previously deemed too large

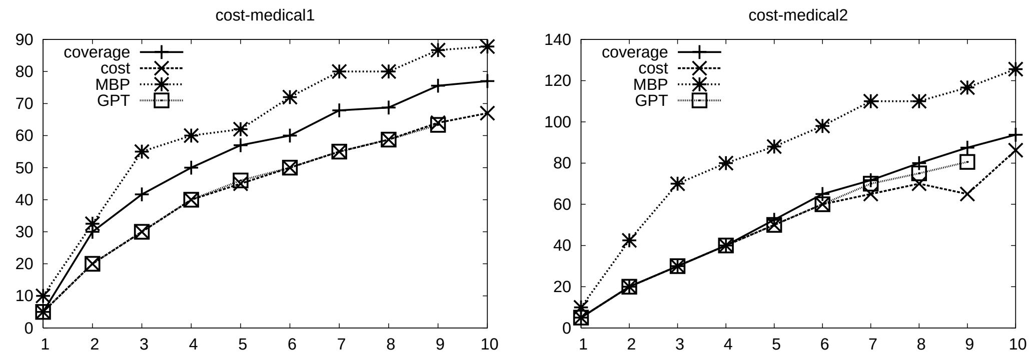

Figure 1: *Expected cost results for $POND$ (coverage and cost), MBP, and GPT for Medical-Specialist.*

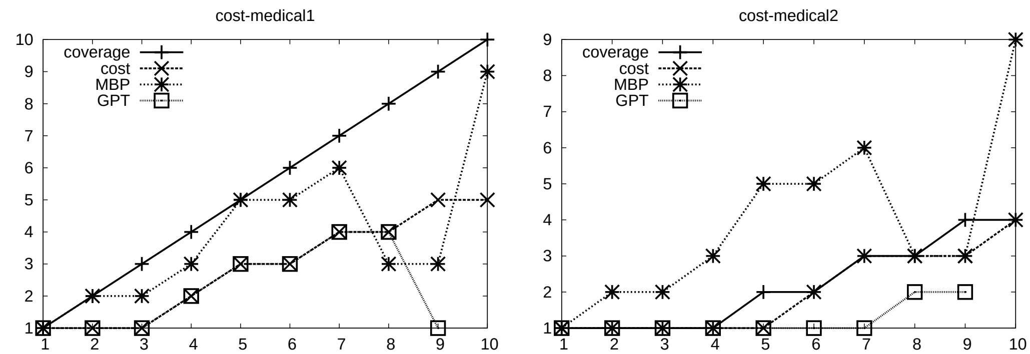

Figure 2: *Breadth (# of plan paths) results for $POND$ (coverage and cost), MBP, and GPT for Medical-Specialist.*

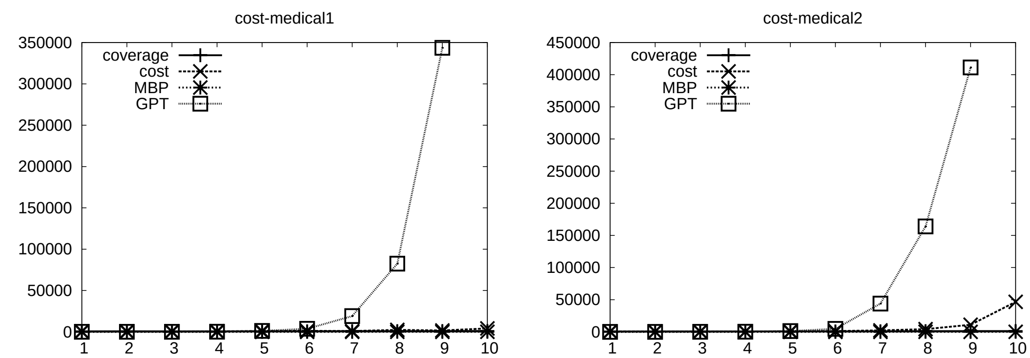

Figure 3: *Total Time(ms) results for $POND$ (coverage and cost), MBP, and GPT for Medical-Specialist.*

for traditional (optimal) solution algorithms. We note that relaxed plan heuristics are effective yet inadmissible, thus leading to non-optimal solutions. We believe that something has to give when trying to solve large problems, and we trade provable optimality for scalability.

References

Piergiorgio Bertoli, Alessandro Cimatti, Marco Roveri, and Paolo Traverso. Planning in nondeterministic domains under partial observability via symbolic model checking. In Bernhard Nebel, editor, *Proceedings of the seventeenth International Conference on Artificial Intelligence (IJCAI-01)*, pages 473–486, San Francisco, CA, August 4–10 2001. Morgan Kaufmann Publishers, Inc.

Avrim Blum and Merrick Furst. Fast planning through planning graph analysis. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI 95)*, pages 1636–1642, 1995.

Blai Bonet and Hector Geffner. Planning as heuristic search: New results. In *Proceedings of the Euoropean Conference of Planning*, pages 360–372, 1999.

Blai Bonet and Hector Geffner. Planning with incomplete information as heuristic search in belief space. In *Artificial Intelligence Planning Systems*, 2000.

Daniel Bryce and Subbarao Kambhampati. Heuristic guidance measures for conformant planning. In *Proceedings of the 14th International Conference on Automated Planning and Scheduling (ICAPS'04)*, June 2004.

Daniel Bryce, Subbarao Kambhampati, and David E. Smith. Planning in belief space with a labelled uncertainty

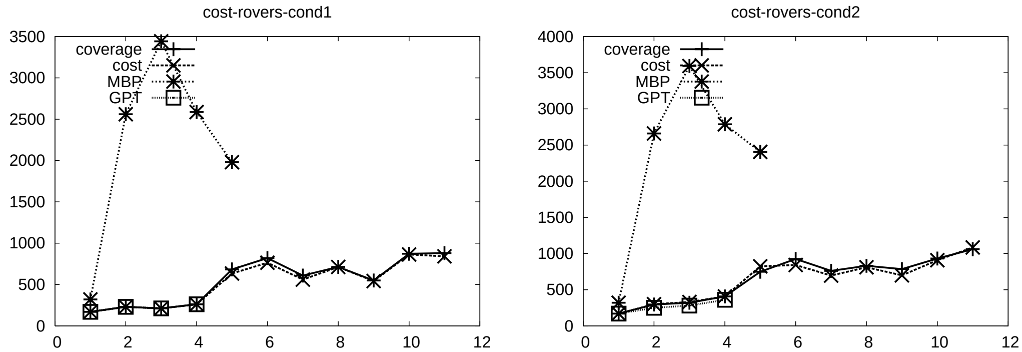

Figure 4: *Expected cost results for $POND$ (coverage and cost), MBP, and GPT for Rovers.*

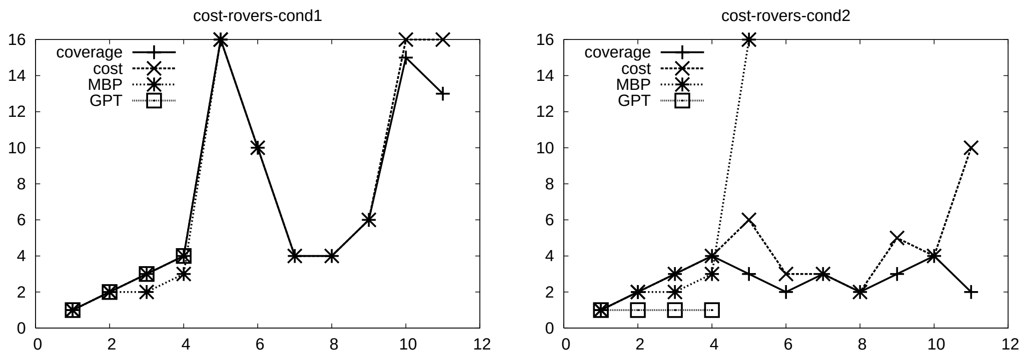

Figure 5: *Breadth (# of plan paths) results for $POND$ (coverage and cost), MBP, and GPT for Rovers.*

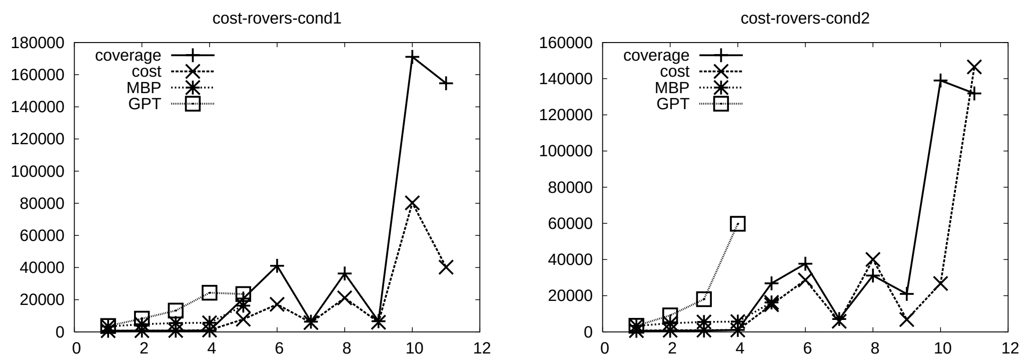

Figure 6: *Total Time(ms) results for $POND$ (coverage and cost), MBP, and GPT for Rovers.*

graph. Technical report, AAAI Workshop TR WS-04-08, 2004.

Minh Binh Do and Subbarao Kambhampati. Sapa: A scalable multi-objective heuristic metric temporal planner. *JAIR*, 2003.

Jörg Hoffmann and Bernhard Nebel. The FF planning system: Fast plan generation through heuristic search. *Journal of Artificial Intelligence Research*, 14:253–302, 2001.

Volume 20, special issue on the 3rd international planning competition. *Journal of Artificial Intelligence Research*, 2003.

XuanLong Nguyen, Subbarao Kambhampati, and Romeo Sanchez Nigenda. Planning graph as the basis for deriving heuristics for plan synthesis by state space and CSP search. *Artificial Intelligence*, 135(1-2):73–123, 2002.

Menkes van den Briel, Romeo Sanchez Nigenda, Minh Binh Do, and Subbarao Kambhampati. Effective approaches for partial satisfaction (over-subscription) planning. In *Proceedings of the AAAI conference*, 2004.

Daniel S. Weld, Corin Anderson, and David E Smith. Extending graphplan to handle uncertainty and sensing actions. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI-98)*. AAAI Press, 1998.