

Exploiting Belief Locality in Run-Time Decision-Theoretic Planners

William H. Turkett, Jr. and John R. Rose

Wake Forest University Computer Science Winston Salem, NC 27104 turketwh@wfu.edu	University of South Carolina Computer Science and Engineering Columbia, SC 29208 rose@cse.sc.edu
-------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------

Abstract

While Partially-Observable Markov Decision Processes have become a popular means of representing realistic planning problems, exact approaches to finding POMDP policies are extremely computationally complex. An alternative approach for control in POMDP domains is to use run-time optimization over action sequences in a dynamic decision network. While exact algorithms have to generate a policy over the entire belief space, a run-time approach only needs to reason about the reachable parts of the belief space. By combining runtime planning with caching, it is possible to exploit locality of belief states in POMDPs, allowing for significant speedups in plan generation. This paper introduces and evaluates an exact caching mechanism and a grid-based caching mechanism.

Introduction

Partially-Observable Markov Decision Processes (POMDPs) have become a popular model for representing agent planning problems due to their ability to represent action and state uncertainty as well as cost and reward functions. Optimal approaches to solving POMDPs perform complete policy generation before execution and are extremely computationally complex. Accordingly, many current researchers are investigating approximate solutions that require much less computation. While most approximation algorithms still use pre-execution policy generation, it is also possible to use run-time POMDP planning. In run-time approaches an agent predicts forward from its current belief state to the set of reachable states.

Previous research by the authors (Turkett, 2004) suggests that some POMDP domains entail a significant amount of locality in the belief states an agent will encounter during execution. This belief locality includes common entry into a small set of belief states and common entry into a few clusters of belief states where states in a cluster are highly similar. This paper presents and evaluates algorithms for exploiting both types of belief locality using a Dynamic Decision Network (DDN) run-time planning architecture.

Methodology

While the total number of representable belief states in a domain is infinite, many of these belief states can't or won't be encountered by an executing agent. This is very common in domains that are near-deterministic and informative. In these domains, agents tend to re-enter belief states. By caching previously selected actions, agents can significantly speed up reasoning as execution progresses.

Exact caching uses a hashtable for mapping string representations of factored belief states with the corresponding optimal actions. Cache checks are performed when a new action is needed. If a cache hit occurs, the corresponding action in the cache is executed. A DDN is used to select the appropriate action for the current belief state if a miss occurs (for more on DDNs, please see the text by Jensen (Jensen, 2001)). DDN-selected actions are then inserted into the cache. Cache entries are preserved over the life of the agent and LRU replacement is used for limited size caches. While DDN planning is exponential in actions and observations, cache search with a hashtable is close to O(1). Space requirements depend on the number of belief states visited. This set tends to be small if the domain is informative and near deterministic. Exact caching has no effect on rewards.

Lookup in a grid cache is implemented by using a belief similarity measure. This approach is similar to grid-based POMDP approximation (Pineau, 2003) where the closest pre-computed neighbor of a belief point is used as the basis for action selection. The current grid cache implementation defines the difference in belief states as the Manhattan distance between the two belief states (other measures could be used). If the distance between the current belief state and the closest in the cache is below a threshold set by the agent designer, the agent accepts the cached action for the nearby state as the best action for the

current state. Otherwise, the agent uses its DDN to generate an optimal action. The Manhattan distance measure requires storage and analysis of non-factored belief state keys. To prevent propagation of small errors, updates to the cache are only performed when the DDN has been used to select actions. Grid cache search requires $O(n*m)$ time complexity (n belief states with m true states represented in each). Space complexity is dependent on the set of visited belief states, a set that can grow large for noisy domains where grid-caching would be used.

Evaluation

The performance of the cache based run-time planning algorithms is presented for two problems from the POMDP literature. Previous work (Turkett, 2004) gives detailed descriptions of these problems and further results. Uniform initial beliefs over possible states are assumed.

The first problem presented is a large navigation problem, an extension of the cheese taxi problem from Pineau (Pineau, 2002). There are 300 states, seven actions, and significant initial uncertainty in location and destination. Figure 1 shows the median planning times computed over 100-action windows across 250 trials for an agent without caching and with a 500-entry exact match cache. Each trial consists of multiple actions and is complete when the agent achieves the goal for the domain. At around action 300 the median planning time for the agent falls close to zero, suggesting that the reachable belief space is not very large and is consistently re-visited.

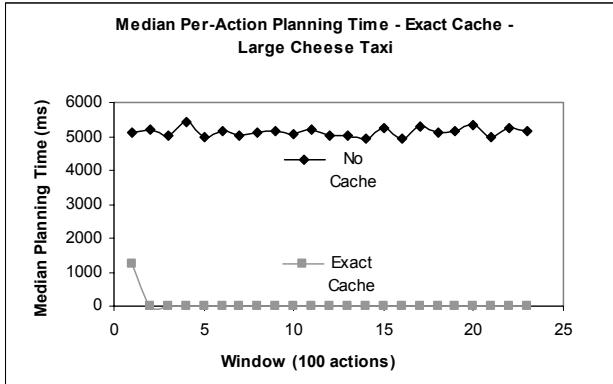


Figure 1. Median Planning Time: Large Cheese Taxi, Exact Cache

In Robot Laser Tag (Pineau, 2003), a robot player and opponent can each be in one of 29 locations. The opponent can also be tagged, leading to 870 total states. The player can perform four directional moves and tag the opponent. The opponent is unobservable and moves by backing away from the player. There are thirty observations: one indicating that the player and opponent are in the same spot and the rest for the player's position.

Figure 2 shows planning times over 250 trials for robot tag. Performance improves with the use of an exact cache and larger improvements are seen with the grid cache.

Across the tested threshold levels (labeled as epsilons in Figure 2), the steps required to tag the opponent were all within 1 of each other. A Mann-Whitney test indicates that it is not possible to say these costs are not from the same distribution.

Conclusions

Belief locality in POMDP domains can be exploited by a run-time planning agent through the introduction of caching mechanisms. Caching of previous action selections appears to perform very well on near-deterministic and informative domains. For noisier domains, grid based caches demonstrate good performance, allowing for reductions in computation time with little effect on earned reward. Improvements include pre-caching of the fully-observable MDP policy (polynomial computable). This would reduce dependence on the DDN and aid in grid-cache searching around the FOMDP belief points. A second improvement is a better search data structure for the high-dimensional grid cache.

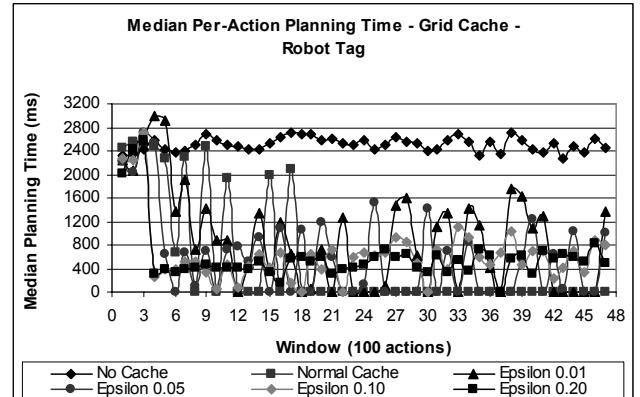


Figure 2. Median Planning Time: Robot Tag, Grid Cache

References

- Jensen, Finn. 2001. *Bayesian Networks and Decision Graphs*. New York, NY: Springer-Verlag.
- Pineau, Joelle and Thrun, Sebastian. 2002. An Integrated Approach to Hierarchy and Abstraction for POMDPs. Technical Report, CMU-RI-TR-02-21, Department of Computer Science, Carnegie Mellon University.
- Pineau, Joelle; Gordon, Geoff; and Thrun, Sebastian. 2003. Point-Based Value Iteration: An Anytime Algorithm for POMDPs. In *Proceedings of the International Joint Conference on Artificial Intelligence*.
- Turkett, Jr., William H. 2004. Robust Multiagent Plan Generation and Execution with Decision-Theoretic Planners. Ph.D. Dissertation, Department of Computer Science and Engineering, University of South Carolina.