Information Sharing for Distributed Planning

(Extended Abstract)

Prasanna Velagapudi
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
pkv@cs.cmu.edu

ABSTRACT

Large-scale collaborative planning is critical in many domains such as sensor networks and disaster rescue. In large, heterogeneous multiagent teams, agents must work together to create plans that maximize the effectiveness of the team as a whole; coordinating actions to increase efficiency while also avoiding potentially dangerous interactions between teammates. Recognizing the utility of information to agents in teams and delivering it efficiently across a team has also been the focus of much research, with proposed approaches ranging from classic flooding, gossiping, to channel filtering. We combine these two fields of research to study the problem of information sharing mechanisms to distribute planning in both certain and uncertain environments.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Coherence and Coordination, Multiagent Systems*; G.3 [**Probability and Statistics**]: Probabilistic Algorithms; H.1.1 [**Models and Principles**]: Systems and Information Theory—*Value of Information*

General Terms

Algorithms, Performance

Keywords

distributed planning, information dissemination, multiagent

1. INTRODUCTION

Exciting applications are emerging that involve large, heterogeneous teams acting in complex environments. Examples include search and rescue, disaster response, and military surveillance. In such domains, team members must engage in cooperative planning to execute coordinated tasks. Often, there exist dramatic differences in utility depending on the outcome, e.g. when a bad sequence of actions will lead to the destruction of a robot and a good sequence of actions will lead to saving a human life.

In search and rescue situations, for example, there can be limited access to certain sections of a disaster site (e.g. a collapsed building). When teams of robotic agents are deployed into these environments, they cannot simply use their own locally optimal plans. An agent following an independent plan might inadvertently block access to a portion of the environment for some other agent, hampering the rescue operations of the team overall. However, teammates often have limited information about which, if any, team members require particular pieces of information. Thus, each team member making its own local plan needs to determine whether and where to send resulting information with only limited knowledge of who might need it and how important it is to them. At the same time, team members must also be careful about what they communicate as the volume of incoming information is typically dramatically higher than available communication bandwidth.

Much previous research has been done on the problem of cooperative planning and decision making, both with and without uncertainty. Unfortunately many of the proposed approaches do not scale to teams of hundreds of robots in constrained environments. We examine two domains, path planning and Dec-POMDPs, where *reward shaping* methods are showing promise for solving these problems for large teams.

Cite as: Information Sharing for Distributed Planning (Extended Abstract), Prasanna Velagapudi, Katia Sycara and Paul Scerri, *Proc. of 9th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, van der Hoek, Kaminka, Lespérance, Luck and Sen (eds.), May, 10–14, 2010, Toronto, Canada, pp. 1667-1668

2. DISTRIBUTED PATH PLANNING

In path planning problems, a team of agents is given some set of start locations must find collision-free paths through time and space to reach a corresponding set of goal locations on a map that may contain obstacles and varying traversal costs. One approach to this problem which has been shown effective for reasonably large teams is *prioritized planning* [3]. In this approach, robots sequentially plan paths according to a prioritization function. However, this means robots must plan paths in order, resulting in a linear increase in planning time with the number of robots. Prioritized planning is also therefore centralized, creating a potential computational and communication bottleneck, as well as a single point of failure.

However, in many domains, the strict ordering of sequential planning is likely to be unnecessarily expensive. In most cases, not all robots need to avoid all other robots. Online prioritized approaches such as [1] and [2] take advantage of this property by determining sets of robots that need to be planned sequentially by detecting interactions via local observations. We present an approach that distributes prioritized planning, allowing each robot to plan at the same time, then look for collisions between paths and require lower priority robots to replan. The intuition behind this is that if

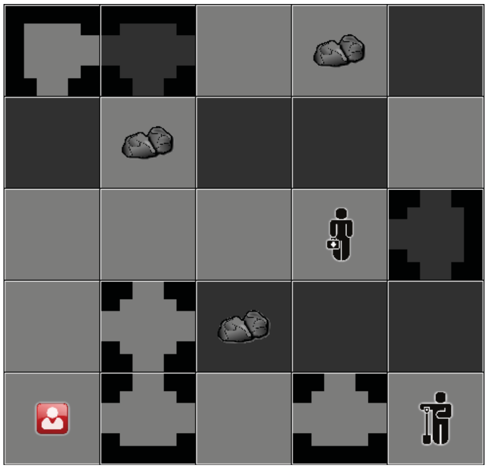

Figure 1: An abstract rescue domain where an aid agent (w/ bag) plans to reach a victim (the square symbol in lower left), possibly assisted by a debris clearing agent (w/ shovel). The aid agent may reach the victim faster if debris in its path is cleared first, but both robots will be penalized if they collide in a narrow corridor (cell w/ black borders).

robot paths are not dependent on all other paths, the number of replanning iterations might be quite low and overall planning time will be reduced because no single node needs to plan for each robot in sequence. We prove that distributing the planning in this way still converges to the same result as the centralized planner. Experimental results support the intuition, showing a dramatic reduction in the number of planning iterations. However, because the length of an iteration for the distributed algorithm is governed by the slowest planner, there is not always a large gain in overall planning time.

3. DISTRIBUTED POMDP SOLVING

Decentralized Markov Decision Problems(MDPs) and Decentralized Partially Observable MDPs accurately represent the decision making problem in domains where there exists uncertainty about the outcome of actions and dramatic differences in utility depending on the outcome, e.g. when a bad sequence of actions will lead to the destruction of a robot and a good sequence of actions will lead to rescuing a human victim. However, the computational complexity involved in solving DEC-MDP/DEC-POMDP models is NEXP, hence most approaches for solving these models have been restricted to solving decision problems for two or three agents.complexity

Recently, a model shaping approach called TREMOR was proposed in [4] to solve a sub-class of DEC-POMDPs. By exploiting dynamic locality in interactions of agents, TREMOR was able to scale to problems with ten agents. Dynamic locality in interactions assumes that interactions happen primarily in certain “coordination locales”. For example, two robots interacting only when they collide in a narrow corridor. One example of such a domain is seen in Figure 1. However, TREMOR cannot scale further as the decision problem for individual agents increases in complexity or when more agents are introduced into the environment.

We present Large-scale TREMOR (L-TREMOR), a distributed version of TREMOR that focuses computation on the most valuable interactions, to allow scale-up to hundreds of agents. The key to distributing TREMOR is being able to compute interaction values, without having to perform the exponential operation of comparing individual agent policies. In L-TREMOR, after computing the individual policy, each agent creates a list of the *coordination locales* (CLs) that have non-zero probability of occurrence and orders that list by the expected reward (or cost) of another agent being in that CL. For example, if an agent’s local policy took it into a narrow corridor with high probability and another agent being there at the same time would lead to a dramatic drop in its expected utility, that CL will appear near the top of the list. The highest value CLs are communicated to other agents who compare them against their own policy to find CLs with high value (or cost) interactions. Those are communicated back to the sending agent which uses them to shape rewards and recompute, as in TREMOR. Notice that this mechanism differs conceptually from TREMOR because instead of blindly comparing whole policies for interactions it focuses the search towards more likely and more important interactions. While this can potentially reduce solution quality by a small amount, it leads to dramatic computational and communication savings. A distributed, market-based role reallocation algorithm replaces the centralized branch and bound algorithm in TREMOR.

4. CONCLUSIONS

We investigate two approaches to scale distributed planning problems into the hundreds of agents. Preliminary work in analytical and empirical studies suggest that in these domains, using intelligent information sharing coupled with properly constructed reward shaping methods may provide competitive performance to centralized algorithms while improving scalability and reducing network and computational costs. We are currently attempting to unify these results into a generalized model of the performance of reward-shaping and intelligent information dissemination in distributed planning problems.

5. ACKNOWLEDGMENTS

I would like to thank my advisors, Paul Scerri and Katia Sycara. This research has been funded in part by the AFOSR MURI grant FA9550-08-1-0356. This material is based upon work supported under a National Science Foundation Graduate Research Fellowship.

6. REFERENCES

[1] L. Chun, Z. Zheng, and W. Chang. A decentralized approach to the conflict-free motion planning formultiple mobile robots. In *Proc. of ICRA*, 1999.

[2] C. Clark, S. Rock, and J. Latombe. Motion planning for multiple mobile robots using dynamic networks. In *Proc. of ICRA*, 2003.

[3] J. van den Berg and M. Overmars. Prioritized motion planning for multiple robots. In *Proc. of IROS*, 2005.

[4] P. Varakantham, J. Kwak, M. Taylor, J. Marecki, P. Scerri, and M. Tambe. Exploiting Coordination Locales in Distributed POMDPs via Social Model Shaping. In *Proc. of ICAPS*, 2009.