Probabilistic Planning for Multi-Robot Systems

Christopher Amato*, Shayegan Omidshafiei†, Ali-akbar Agha-mohammadi‡, George Konidaris§,
Jonathan P. How† and Leslie P. Kaelbling¶
*Department of Computer Science, University of New Hampshire, Durham, NH 03824
†LIDS, MIT, Cambridge, MA 02139
‡Qualcomm, San Diego, CA 92121
§Departments of Computer Science and Electrical and Computer Engineering, Duke University, Durham, NC 27708
¶CSAIL, MIT, Cambridge, MA 02139

I. Introduction

Most multi-robot systems are controlled by hand-built special-purpose algorithms that are difficult to design, implement and verify. For single robots, automatic planning systems provide a flexible general-purpose strategy for constructing plans given high-level declarative domain specifications, even in the presence of substantial stochasticity and partial observability [10]. We show that this strategy can be effectively extended to multi-robot systems. Our methods allow automatic off-line construction of robust multi-robot policies that support coordinated actions. As a natural consequence of the approach, they can even generate communication strategies that exploit the domain dynamics to share critical information in service of achieving the group's overall objective.

Specifically, we are interested in problems where robots share the same objective function and each individual robot can only make noisy, partial observations of the environment. The decentralized partially observable Markov decision process (Dec-POMDP) is a general framework for representing multi-agent coordination problems. Dec-POMDPs have been widely studied in artificial intelligence as a way to address the fundamental differences in decision-making in decentralized settings [1, 5, 8]. Like the POMDP [7] model that it extends, Dec-POMDPs consider general dynamics, cost and sensor models. Any problem where multiple robots share a single overall reward or cost function can be formalized as a Dec-POMDP. As such, Dec-POMDP solvers could automatically generate control policies (including policies over when and what to communicate) for decentralized control problems, in the presence of uncertainty in outcomes, sensors and information about the other robots. Unfortunately, this generality comes at a cost: Dec-POMDPs are typically infeasible to solve except for small problems [5, 2].

One reason for the intractability of solving large Dec-POMDPs is that current approaches model problems at a low level of granularity, where each robot's actions are primitive operations lasting exactly one time step. Recent research has addressed the more realistic *MacDec-POMDP* case where each robot has *macro-actions*: temporally extended actions which may require different amounts of time to execute [2]. An alternative formulation is the Dec-POSMDP, which operates directly in belief space [9]. These models allow coordination decisions to only occur at the level of deciding which macro-actions to execute. Macro-actions are a natural model for the modular controllers (e.g., navigating to a waypoint or grasping an object) often sequenced to obtain robot behavior, bridging the gap between robotics research and Dec-POMDPs. This approach has the potential to produce high-quality general solutions for real-world heterogeneous multi-robot coordination problems by automatically generating control and communication policies, given a model.

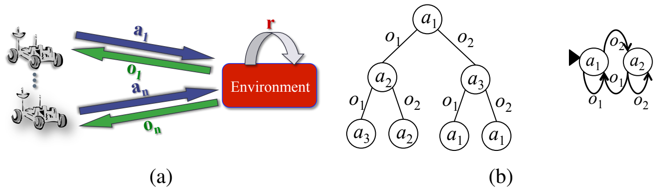


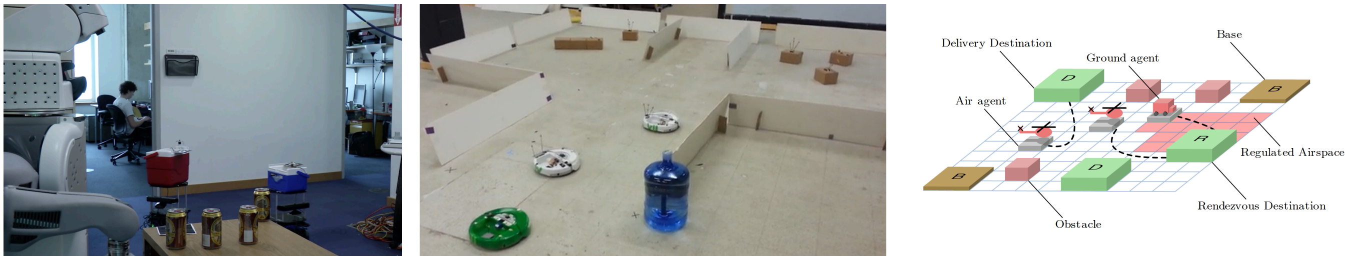
Fig. 1. Illustration of the Dec-POMDP model (a) and policies represented by trees or finite-state controllers (b).

II. MacDec-POMDPs and Dec-POSMDPs

In Dec-POMDPs (as depicted in Fig. 1(a)), multiple robots operate based on partial and local views of the world. At each step, every robot chooses an action based purely on locally observable information, resulting in an observation for each individual robot. The robots also share a single reward or cost function, making the problem cooperative, but their local views mean that execution is decentralized.

MacDec-POMDPs incorporate macro-actions into the Dec-POMDP framework, where macro-actions have defined initial conditions where they can be executed and this execution continues until some terminal condition is reached. In the MacDec-POMDP framework, it is assumed that either a low-level (Dec-POMDP) model or a simulator is available in order to evaluate solutions. As a result, MacDec-POMDPs do not explicitly model the time until completion. In contrast, Dec-POSMDPs explicitly model the distribution of time until completion. Solutions in this semi-Markov model can then be evaluated using a higher-level model (that also includes time until completion) or (again) in a simulator.

Two Dec-POMDP algorithms have been extended to the MacDec-POMDP case [2], but other extensions are possible. The resulting solution is a policy for each agent which can



(a) The bartender and waiters domain

(b) The warehouse problem

(c) The package delivery problem

Fig. 2. Experimental domains

be represented as a set of trees (see Figure 1(b)). In the Dec-POSMDP, we represent the policy as a finite-state controller for each agent (Fig. 1(b)). Discrete space search techniques can be applied to find the optimal joint policy. Greedy and probabilistic search algorithms have been successfully used for solving Dec-POSMDP finite-state controllers [9].

The MacDec-POMDP framework is a natural way to represent and generate behavior for general multi-robot systems. We assume an abstract model of the system is given in the form of macro-action representations, which include the associated policies as well as initiation and terminal conditions. These macro-actions are controllers operating in (possibly) continuous time with continuous actions and feedback, but their operation is discretized for use with the planner. Given the macro-actions and simulator, the planner then automatically generates a solution which optimizes the value function with respect to the uncertainty over outcomes, sensor information and other robots. This solution comes in the form of SMACH controllers [6] which are hierarchical state machines for use in a ROS environment.

III. Experiments

We performed comparisons with previous work on existing benchmark domains and demonstrated its effectiveness in different scenarios (Warehouse [4], Bartender and waiters [3], and Package delivery [9]). In the warehouse problem (Figure 2(b)), a team of robots is tasked with finding a set of large and small boxes in the environment and returning them to a shipping location. Here, coordination is needed not just for assigning robots to push specific boxes, but also requires that two robots push the larger box at the same time. In the bartender and waiters problem (Figure 2(a)), the waiters (Turtlebots) must find and deliver orders as quickly as possible, retrieving drinks from a bartender (PR2). In the package delivery problem (Figure 2(c)), the robots retrieve and deliver packages from base locations to delivery locations. In all problems there is stochasticity in the movements of robots and partial observability with respect to the location of the other robots and the other objects (boxes, orders and packages).

These problems are very large (consisting of over a billion discrete states or having a continuous state space), causing them to be unsolvable by previous Dec-POMDP-based approaches. We also consider cases where the robots can send communication signals to each other, but we do not define the meaning of the messages. Therefore, our planner must determine where the robots should navigate, what boxes they should push and what communication messages should be sent (if at all) at each step of the problem to optimize the solution for the team. The robots must make these decisions based solely on the information they individually receive during execution (e.g., each robot's location estimate as well as where and when boxes and other robots have been seen). Our methods outperform naive methods that do not consider uncertainty and generate optimized solutions for each problem based on the high-level domain description.

References

[1] Christopher Amato, Girish Chowdhary, Alborz Geramifard, Nazim Kemal Ure, and Mykel J. Kochenderfer. Decentralized control of partially observable Markov decision processes. In *the IEEE Conference on Decision and Control*, 2013.

[2] Christopher Amato, George D. Konidaris, and Leslie P. Kaelbling. Planning with macro-actions in decentralized POMDPs. In *the International Conference on Autonomous Agents and Multiagent Systems*, 2014.

[3] Christopher Amato, George D. Konidaris, Ariel Anders, Gabriel Cruz, Jonathan P. How, and Leslie P. Kaelbling. Policy search for multi-robot coordination under uncertainty. In *the Robotics: Science and Systems Conference*, 2015.

[4] Christopher Amato, George D. Konidaris, Gabriel Cruz, Christopher A. Maynor, Jonathan P. How, and Leslie P. Kaelbling. Planning for decentralized control of multiple robots under uncertainty. In *the International Conference on Robotics and Automation*, 2015.

[5] Daniel S. Bernstein, Robert Givan, Neil Immerman, and Shlomo Zilberstein. The complexity of decentralized control of Markov decision processes. *Mathematics of Operations Research*, 27 (4):819–840, 2002.

[6] Jonathan Bohren. SMACH. http://wiki.ros.org/smach/, 2010.

[7] Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101:1–45, 1998.

[8] Frans A. Oliehoek. Decentralized POMDPs. In Marco Wiering and Martijn van Otterlo, editors, *Reinforcement Learning: State of the Art*, volume 12 of *Adaptation, Learning, and Optimization*, pages 471–503. Springer Berlin Heidelberg, 2012.

[9] Shayegan Omidshafiei, Ali-akbar Agha-mohammadi, Christopher Amato, and Jonathan P. How. Decentralized control of partially observable Markov decision processes using belief space macro-actions. In *the International Conference on Robotics and Automation*, 2015.

[10] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005. ISBN 0262201623.