Decentralized Multi-Robot Cooperation with Auctioned POMDPs

Jesus Capitan
Instituto Superior Tecnico
Lisbon, Portugal
jescap@isr.ist.utl.pt

Matthijs Spaan
Delft University of Technology
Delft, The Netherlands
m.t.j.spaan@tudelft.nl

Luis Merino
Pablo de Olavide University
Seville, Spain
lmercab@upo.es

Anibal Ollero
University of Seville
Seville, Spain
aollero@us.es

Introduction

Multi-robot systems are of great interest in many applications, such as exploration, surveillance or rescue robotics. In those applications, a single robot is not able to acquire all the required information and the cooperation among multiple robots is essential. However, real scenarios present uncertain and potentially hazardous environments in which robots can experience communication constraints regarding connectivity, bandwidth and delays. Mapping the overall task into robust plans for each robot is a challenging problem.

In general, planning under uncertainty faces a scalability problem when considering multi-robot teams, as the information space scales exponentially with the number of robots. We propose to decentralize multi-robot Partially Observable Markov Decision Processes (POMDPs) while maintaining cooperation between robots by using POMDP policy auctions. The idea is to exploit the power of decision-theoretic planning methods such as POMDPs, while mitigating their complexity by lowering the dependence between individual plans. In particular, the approach solves independent POMDPs for each robot, but still fosters online cooperation during the execution phase by distributing the individual policies using auctions. Auction algorithms have been widely used for optimal multi-robot task allocation, and have also been explored in conjunction with POMDPs (Spaan, Gonçalves, and Sequeira 2010).

In addition, communication models in the multi-agent POMDP literature (Pynadath and Tambe 2002; Nair et al. 2004; Roth, Simmons, and Veloso 2005) severely mismatch with real inter-robot communication. We address this issue by exploiting a decentralized data fusion method in order to efficiently maintain a joint belief state among the robots.

This paper is an extended abstract that summarizes the main contributions of our previous work (Capitan et al. 2013). The first contribution is to emulate a multi-robot POMDP by combining individual behaviors or roles that can be represented by single-robot POMDPs. We generalize a centralized POMDP auction to assign never-ending policies (behaviors) to different robots at every step. In this novel decentralized auction, instead of tasks, POMDP policies that describe a behavior towards a common goal are distributed; robots can switch between these behaviors dynamically at each decision step. The auction determines continuously which behavior is best for each robot to cooperatively attain the goal. Since local POMDPs are solved for each robot, the inter-connection between the models is low and the approach can scale well with the number of robots.

The second key component is to efficiently maintain a joint belief state among the robots, which can serve as coordination signal. We use an existing Decentralized Data Fusion approach (Capitan et al. 2011), but in conjunction with POMDP policies for a multi-robot system. Unlike most work on POMDPs, the belief update here is separated from the decision-making process during the execution phase. This decoupling between both processes increases the robustness and reliability of real-time robotic teams.

Multi-agent Planning under Uncertainty

In the literature a wide variety of decision-theoretic models exist to deal with multi-agent systems (Seuken and Zilberstein 2008), e.g., Multi-agent POMDPs and Decentralized POMDPs. However, many of these models have severe drawbacks if applied to multi-robot scenarios. Before presenting our solution, we analyze the models available in the literature by comparing them in terms of agent interdependence and communication assumptions.

The level of interdependence between agents is determined by 1) the amount of information that an agent needs to know about other agents and 2) how coupled the final policies are. We call a system highly interdependent if a change in one of the agents' model requires re-computing policies for the others. Many models from the literature are highly interdependent, for instance Multi-agent MDPs (MMDP) (Boutilier 1996), MPOMDPs, Dec-POMDPs, and ND-POMDPs (Nair et al. 2005), and I-POMDPs (Gmytrasiewicz and Doshi 2005). Figure 1 presents a classification of existing models with respect to their interdependence and the grade of communication that is assumed.

The simplest approach is to map the global task as well as possible into a set of individual tasks, and model these as independent POMDPs (Fig. 1, bottom left). Thus, each agent can solve its own POMDP and execute its own policy without any communication. In this case, the interdependence between agents is very low, but since each agent ignores the others, the level of cooperation or even coordination is low

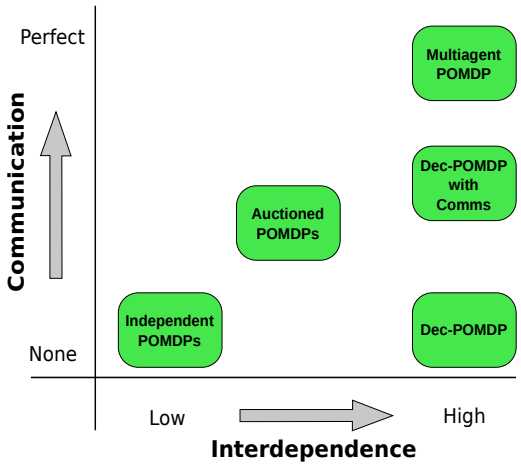

Figure 1: Classification of multi-agent POMDP approaches according to interdependence and level of communication between the agents. "Auctioned POMDPs" refers to our proposed approach.

too. Many interesting multi-agent planning problems cannot be tackled with such a loosely coupled approach. The advantage of such an approach is its relatively low computational complexity, since it only requires solving single-agent POMDPs, each of which is defined over individual action and observation spaces. Hence, scalability in the number of agents is linear, which is very low compared to other models.

On the other end of the spectrum, MPOMDPs and Dec-POMDPs solve a single decision-theoretic model for the whole team reasoning about all the actions and observations of each agent (Fig. 1, right column).

The MPOMDP model assumes perfect communication and each agent has access to joint actions and observations at every moment, whereas the Dec-POMDP model assumes no communication at all. Such models allow for tight coordination, but they present a high interdependence, since any small change in one of the agents entails a recalculation of the policy for the whole team. Furthermore, if due to imperfect communication agents do not have access to other agents' observations, the behavior of the MPOMDP model is not defined. Regarding computational complexity, an MPOMDP is a POMDP defined over the joint action and observation spaces, whose sizes grow exponentially with the number of agents.

The Dec-POMDP model, on the other hand, does not exploit communication at all, which in many scenarios could be beneficial to improve team performance. For instance, in a cooperative tracking application, it is easy to see that if a pursuer robot detects the target and informs its teammates about the target's approximate location, the other pursuers can close in on the target. Without communication, each pursuer might need to find the target by itself, which is clearly less time-efficient. Solving Dec-POMDPs optimally takes doubly-exponential time in the worst case, which severely restricts their applicability in multi-robot scenarios.

In between MPOMDPs and Dec-POMDPs there are several models in which some communication is assumed (Nair et al. 2004; Roth, Simmons, and Veloso 2005; Spaan, Oliehoek, and Vlassis 2008; Oliehoek and Spaan 2012) (Fig. 1, middle right). These models try to exploit the fact that agents actually share information, but just partially and at certain instants. Furthermore, most of them assume that communication arrives instantly.

By looking at the current state of the literature, we can conclude existing multi-agent decision-theoretic models do not take into account the requirements that multi-robot missions pose. First, a critical dependence on communication is to be avoided, but it should be exploited when available. Second, a strong coupling between individual robots is undesirable, as tightly coordinated joint actions are often hard to execute with a low probability of success.

Role-based Multi-robot POMDP

In order to address the shortcoming of existing multi-agent planning models for multi-robot scenarios, as discussed above, we present a new model that specifically takes into account multi-robot issues. In a sense, we aim to reach middle ground on both axes of Figure 1.

Many multi-robot missions can be modeled as POMDPs. If all the robots have access to joint information (actions and observations from the whole team), the problem can be modeled as a MPOMDP. The objective of the team can be encoded in a reward function that, in general, depends on joint states and joint actions, and can be seen as the addition of the local rewards for each of the n robots:

$$R(s,a) = R_1(s,a) + \cdots + R_n(s,a). \quad (1)$$

Without losing generality, the reward can be decomposed into two parts: one based only on local information R_i^{local} from each robot i; and one based on joint information R^{joint}. The local information for a robot i means its action a_i and its state s_i. In case of a factored state, each local state s_i would include the local factors that can be controlled by local actions, and the factors that are common for all the robots. Thus, the global reward can be expressed as:

$$R(s,a) = \\ R_1^{local}(s_1,a_1) + \cdots + R_n^{local}(s_n,a_n) + R^{joint}(s,a) \quad (2)$$

Apart from the *local* rewards (i.e., the rewards that robots would get if there were no others), there is the coupled term $R^{joint}(s,a)$, which models cooperation among the robots. Indeed, actions from different robots need to be considered in order to compute this reward. Even though the design of this cooperative term is very dependent on the application, in many cases, due to efficiency issues, it is common to penalize different robots repeating similar tasks. For instance, in many surveillance applications the robots should get less reward for surveying an area that is already being surveyed by another.

The previous idea is useful for many applications in which there are either limited resources that cannot be accessed simultaneously by the robots, or different roles/tasks that must be covered. Thus, the team objective in many missions (e.g.,

detecting a target or alarm) can be achieved with robots following different roles (e.g., patrol a certain area, approach the target, etc.). For instance, in smart energy grids there are providers and consumers (van der Sluis 2011); and in robotic soccer strikers and defenders (Kok, Spaan, and Vlassis 2005). Also, in active perception applications (Maza et al. 2011), where the team needs to maximize its information, it is positive to have robots following non-overlapping behaviors in order to provide richer information to the team.

We are interested in these role-based applications, but we make the following assumptions: (i) there is a finite set of non-overlapping roles/behaviors (i.e., each robot can only be playing one role at a moment); (ii) each role is a single-robot behavior that can be represented by a reward function depending on the local state of that robot. The local rewards in (2) depend on the behaviors chosen by each robot. These rewards R_i^{local} are the ones that each robot would get by acting on its own. Moreover, the cooperative term R^{joint} also depends on the assignment of the behaviors.

The idea in our role-based model is that, at each time step, the robots should select their behaviors optimally (apart from their actions) in order to maximize the expected reward of the whole team. Note that the role assignments from one step to the next one are not correlated. In the next section, we propose an approximate method to solve the role-based MPOMDP in which the policies are sub-optimal, but the computational complexity of the solution is reduced dramatically.

Decentralized Auction with POMDPs

The proposed approach builds on two mechanisms: a decentralized data fusion filter and a POMDP auction. The former allows the robots to share information and build a joint belief like in a MPOMDP, the latter is used to assign the different behaviors to the robots in a cooperative manner. In Fig. 1, our approach can be seen as in between "independent POMDPs" and MPOMDP/Dec-POMDP in terms of agent interdependence. In terms of communication requirements, our approach does not require the high-quality guarantees of other methods.

The objective is to approximate the multi-robot reward in the previous section. For that, a set of reward functions are designed to define single-robot POMDPs, which are solved separately offline. These behaviors are run online simultaneously and combined optimally to produce a joint behavior similar to the one desired for the whole team initially. In the execution phase, the best behavior for each robot is selected online with an auction algorithm where the cost or bid of assigning a policy to a robot is related to the value function of the corresponding individual POMDP. Thus, policies with a greater expected reward are more likely to be selected for each robot, which helps to maximize the global expected reward for the whole team. As explained above, the assignment of the behaviors can vary from one time step to the following in an uncorrelated manner. Note that these behaviors are not finite tasks that the robots must select and solve, but different policies to follow given the current belief. As the belief changes, the robots are allowed to switch their behaviors in order to pursue the optimal solution.

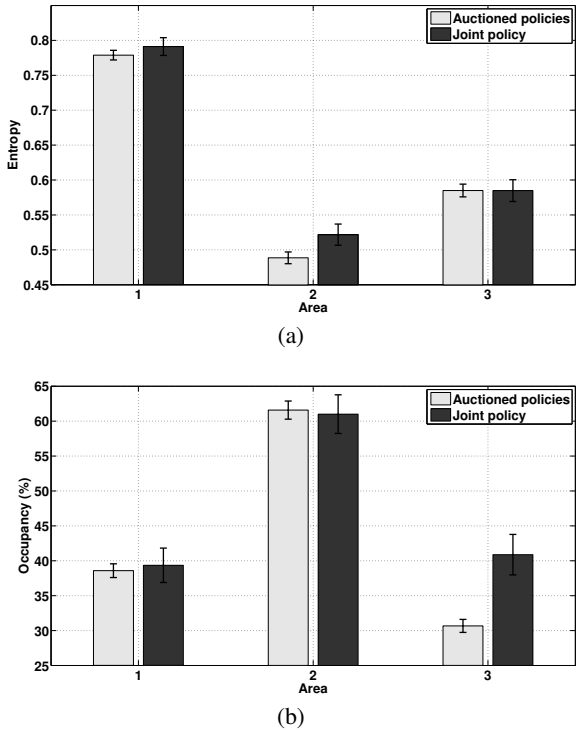

Figure 2: Average results ($\pm 3\sigma$) for simulations on environmental monitoring with two UAVs and three critical areas. Auctioned polices are compared to a joint policy. (a) Entropies of the beliefs on the contamination levels. (b) Percentages of occupancy for each critical area.

Experimental Results

Our previous work (Capitan et al. 2013) presented results for two different applications: environmental monitoring with Unmanned Aerial Vehicles (UAVs); and cooperative tracking, in which several robots have to jointly track a moving target of interest.

In the first case study there is a team of UAVs whose mission is to fly over a certain terrain in order to monitor a potential contamination that may appear. It is assumed that this contamination can only appear and propagate through a network of water flows on the terrain. Therefore, instead of surveying the whole scenario, a set of key points (areas) within that network can be extracted to evaluate the level of contamination. These points are inter-connected through water flows and the contamination can propagate among them.

We tested our approach (behaviors were based on monitoring each of the individual areas) against a joint policy for a multi-robot POMDP. The multi-robot POMDP is far from scalable, so we were only able to solve it for a simple case with 2 UAVs and 3 areas (Areas 1, 2 and 3). Actually, any variation of this small scenario considering more UAVs or areas, caused the computer to run out of memory.

We computed a single-robot policy for each behavior (5 minutes each) and a joint policy for the 2-UAV MPOMDP (14 hours). Then, we ran 1000 simulations of 100 steps (with random starting positions and no initial contamination) for

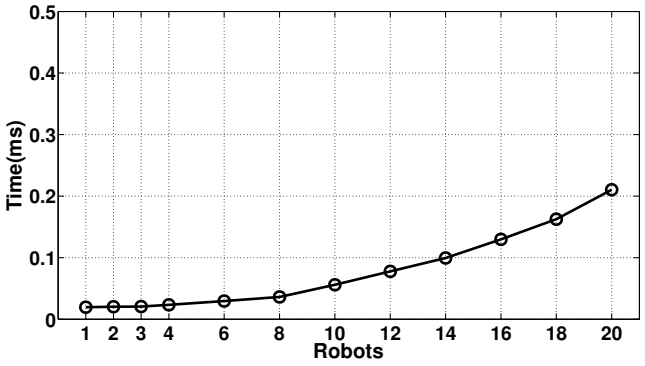

Figure 3: Average results on cooperative tracking to show scalability with multiple robots. In particular, average time spent to solve a role assignment at each robot.

our approach, and the same with the joint policy. The average values for the belief entropies and the percentage of occupancy (times visited) of each area are shown in Fig. 2. Despite the huge difference in computation time for both approaches, the results are still remarkably similar.

In the second case study, the objective is that a group of robots track a moving target estimating its position with their sensors. The idea is to obtain an estimation as accurate as possible. The robots carry bearing-only sensors, so the individual behaviors consist of pointing at the target from different angles. This fosters configurations where the robots surround the target, reducing the uncertainty on its estimated position.

Some simulations were run in order to provide empirical results about scalability. We ran our approach in the same scenario but increasing the number of robots in the team. For each experiment, 100 runs of 100 steps were carried out with the robots and the target starting at random positions. In Fig. 3, the average time spent at each robot (per iteration) to solve the role assignment is plotted. In this case, all the robots were considered within communication range in order to evaluate the worst possible case (auctions needed to be solved for the complete team). It can be seen that the execution time scales polynomially with the number of robots, but in general, our approach will only depend on the number of neighbors, which should be lower than the team size for larger scenarios. Nonetheless, even in this worst case, execution times stay at quite reasonable levels for real-time applications. Results with real multi-robot teams are also shown in (Capitan et al. 2013).

Acknowledgements

This work was partially funded by Fundação para a Ciência e a Tecnologia (ISR-IST) [Project CMU-PT/SIA/0023/2009]; the FP7 Marie Curie Actions Individual Fellowship [#275217 (FP7-PEOPLE-2010-IEF)]; the FP7 EC-SAFEMOBIL Project [288082] and the PAIS-MultiRobot (TIC-7390) Regional Project.

References

Boutilier, C. 1996. Planning, learning and coordination in multiagent decision processes. In *Proc. of the 6th Conference on Theoretical Aspects of Rationality and Knowledge*, 195–210.

Capitan, J.; Merino, L.; Caballero, F.; and Ollero, A. 2011. Decentralized delayed-state information filter (DDSIF): A new approach for cooperative decentralized tracking. *Robotics and Autonomous Systems* 59:376–388.

Capitan, J.; Spaan, M. T.; Merino, L.; and Ollero, A. 2013. Decentralized multi-robot cooperation with auctioned pomdps. *The International Journal of Robotics Research* 32(6):650–671.

Gmytrasiewicz, P. J., and Doshi, P. 2005. A framework for sequential planning in multi-agent settings. *Journal of Artificial Intelligence Research* 24:49–79.

Kok, J. R.; Spaan, M. T. J.; and Vlassis, N. 2005. Non-communicative multi-robot coordination in dynamic environments. *Robotics and Autonomous Systems* 50(2-3):99–114.

Maza, I.; Caballero, F.; Capitan, J.; de Dios, J. M.; and Ollero, A. 2011. A distributed architecture for a robotic platform with aerial sensor transportation and self-deployment capabilities. *Journal of Field Robotics* 28(3):303–328.

Nair, R.; Tambe, M.; Roth, M.; and Yokoo, M. 2004. Communication for improving policy computation in distributed POMDPs. In *Proc. AAMAS*, volume 3, 1098–1105.

Nair, R.; Varakantham, P.; Tambe, M.; and Yokoo, M. 2005. Networked distributed POMDPs: A synthesis of distributed constraint optimization and POMDPs. In *Proc. AAAI*, 133–139.

Oliehoek, F. A., and Spaan, M. T. J. 2012. Tree-based pruning for multiagent POMDPs with delayed communication. In *Proc. AAAI*, 1415–1421.

Pynadath, D. V., and Tambe, M. 2002. The communicative multiagent team decision problem: Analyzing teamwork theories and models. *Journal of Artificial Intelligence Research* 16:389–423.

Roth, M.; Simmons, R.; and Veloso, M. 2005. Decentralized communication strategies for coordinated multi-agent policies. In Schultz, A.; Parker, L.; and Schneider, F., eds., *Multi-Robot Systems: From Swarms to Intelligent Automata*, volume IV. Kluwer Academic Publishers. 93–105.

Seuken, S., and Zilberstein, S. 2008. Formal models and algorithms for decentralized decision making under uncertainty. *Autonomous Agents and Multi-Agent Systems* 17(2):190–250.

Spaan, M.; Gonçalves, N.; and Sequeira, J. 2010. Multirobot coordination by auctioning POMDPs. In *Proc. ICRA*, 1446 –1451.

Spaan, M. T. J.; Oliehoek, F. A.; and Vlassis, N. 2008. Multiagent planning under uncertainty with stochastic communication delays. In *Proc. ICAPS*, 338–345.

van der Sluis, L., ed. 2011. *Future Generation – Smartgrid Research in the Netherlands*. TU Delft Library. ISBN: 978-94-6186-008-8.