Decentralized Multi-Robot Cooperation with Auctioned POMDPs

Jesus Capitan, Matthijs T.J. Spaan, Luis Merino and Anibal Ollero

***Abstract*— Planning under uncertainty faces a scalability problem when considering multi-robot teams, as the information space scales exponentially with the number of robots. To address this issue, this paper proposes to decentralize multiagent Partially Observable Markov Decision Process (POMDPs) while maintaining cooperation between robots by using POMDP policy auctions. Furthermore, communication models in the multiagent POMDP literature severely mismatch with real inter-robot communication. We address this issue by applying a decentralized data fusion method in order to efficiently maintain a joint belief state among the robots. The paper focuses on a cooperative tracking application, in which several robots have to jointly track a moving target of interest. The proposed ideas are illustrated in real multi-robot experiments, showcasing the flexible and robust cooperation that our techniques can provide.**

I. Introduction

Multi-robot systems are of great interest in many robotic applications, such as surveillance or rescue robotics [9], [7]. These scenarios present uncertain and potentially hazardous environments in which robots can experience communication constraints regarding connectivity, bandwidth and delays. We propose a scheme for exploiting the power of decision-theoretic planning methods, while mitigating their complexity by lowering the dependence between individual plans. A key point in our approach is also that we relax the strict assumptions on the quality of the communication channel commonly found in the literature on multiagent planning under uncertainty [13].

Partially Observable Markov Decision Processes (POMDPs) provide a sound mathematical framework to cope with decision-making in uncertain and partially observable environments [8]. Although currently solvers exist that are able to successfully handle large state spaces, POMDPs ultimately face a scalability problem when considering planning for multi-robot teams. Popular models like Dec-POMDPs [1] remain limited to toy problems, and other models presuppose flawless, instantaneous communication [13].

In contrast, we consider fully decentralized solutions, that is, solutions that only involve local information and local communications, and which are scalable with the total number of robots. In particular, this paper proposes an approach that solves independent POMDPs for each robot but still allows online cooperation during the execution phase, by distributing the individual policies using auctions [5].

As a first contribution, we propose to emulate a multi-robot POMDP by combining individual behaviors that can be represented by single-robot POMDPs. We generalize a centralized POMDP auction [15] to assign never-ending tasks (behaviors) to different robots at every step. In this novel decentralized auction, instead of tasks, POMDP policies 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 only local POMDPs are solved, the 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 (DDF) approach [4], but in conjunction with POMDP policies. Unlike most work on POMDPs, the belief update here is separated from the decision-making process during the execution phase. This decoupling increases the robustness and reliability of real-time robotic teams.

We illustrate our method in a multi-robot tracking application, in which several robots cooperate to track a moving target as accurately as possible. In addition, our techniques are suited for a range of problems such as surveillance [7] or fire detection [9] which call for a cooperative effort of robots coordinating their individual behaviors. We demonstrate our approach in a multi-robot testbed, in a fully decentralized setup.

The paper is organized as follows: Section II summarizes POMDP models and describes the decentralized data fusion algorithms. Section III discusses current approaches in the literature for multiagent planning under uncertainty; Section IV describes the overall system and the algorithms for auctioning POMDPs in a decentralized manner; Section V presents an application in cooperative tracking with multi-robot systems; Section VI provides experimental results; and Section VII presents the conclusions and future work.

II. Background

We give a short description about POMDPs, followed by a sketch of the DDF method, as developed before [4].

A. POMDP model

Formally, a POMDP is defined by the tuple $\langle S, A, Z, T, O, R, h, \gamma \rangle$ [8]. The *state space* is the finite set

J.C. is funded by Fundação para a Ciência e a Tecnologia (ISR/IST pluriannual funding) through the PIDDAC Program funds and projects PEst-OE/EEI/LA0009/2011 and CMU-PT/SIA/0023/2009. M.S. is funded by the FP7 Marie Curie Actions Individual Fellowship #275217 (FP7-PEOPLE-2010-IEF). L.M. and A.O. are funded by the FP7 EC-SAFEMOBIL Project (288082)

J. Capitan is with Institute for Systems and Robotics, Instituto Superior Tecnico, Lisbon (Portugal) jescap@isr.ist.utl.pt. M.T.J. Spaan is with Delft University of Technology, Delft (The Netherlands) m.t.j.spaan@tudelft.nl. L. Merino is with Pablo de Olavide University, Seville (Spain) lmercab@upo.es. A. Ollero is with University of Seville, Seville (Spain) aollero@cartuja.us.es

of possible states $s \in S$; the *action space* is defined as the finite set of possible actions $a \in A$; and the *observation space* consists of the finite set of possible observations $z \in Z$. At every step, an action is taken, an observation is made and a reward is given. After performing an action a, the state transition is modeled by the conditional probability function $T(s', a, s) = p(s'|a, s)$, and the posterior observation by the conditional probability function $O(z, a, s') = p(z|a, s')$. The reward obtained at each step is $R(s, a)$, and the objective is to maximize the sum of expected rewards, or *value*, earned during h time steps. To ensure that the sum is finite when $h \to \infty$, rewards are weighted by a discount factor $\gamma \in [0, 1)$. Given that the current state is not directly observable, a probability density function $b(s)$ over the state space is maintained. This is called the *belief state*, which can be updated with a Bayesian filter starting from an initial belief b_0:

$$b'(s') = \eta O(z, a, s') \sum_{s \in S} T(s', a, s) b(s) \quad (1)$$

where η acts as a normalizing constant such that b' remains a probability distribution. The objective of a POMDP is to find a policy that maps beliefs into actions in the form $\pi(b) \to a$, so that the value is maximized. The value gathered by following π starting from belief b is called the value function: $V^\pi(b) = E\left[\sum_{t=0}^{h} \gamma^t r(b_t, \pi(b_t)) | b_0 = b\right]$, where $r(b_t, \pi(b_t)) = \sum_{s \in S} R(s, \pi(b_t)) b_t(s)$. Therefore, the optimal policy π^* is the one that maximizes that value function: $\pi^*(b) = \arg\max_{\pi} V^\pi(b)$.

When a set of N robots that share the same reward function is considered, it is straightforward to extend the previous framework. In that case, each robot i can execute an action a^i from a finite set A_i and receives an observation z^i from a finite set Z_i. The transition function $T(s', a^J, s)$ is now defined over the set of joint actions $a^J \in A_1 \times \cdots \times A_N$, and the observation function $O(z^J, a^J, s')$ relates the state to the joint action and the joint observation $z^J \in Z_1 \times \cdots \times Z_N$. The common reward signal is defined over the joint set of states and actions $R : S \times A_1 \times \cdots \times A_N \to \mathbb{R}$, and the goal is to compute an optimal joint policy $\pi^* = \{\pi_1, \cdots, \pi_N\}$.

B. Decentralized Data Fusion

In the multi-robot case, maintaining a belief over the state space according to (1) is not trivial. A centralized node with access to all information would update the belief as follows:

$$b'_{cen}(s') = \eta p(z^J|a^J, s') \sum_{s \in S} p(s'|a^J, s) b_{cen}(s). \quad (2)$$

However, if the belief estimation is decentralized and each robot i uses only its local information (action a^i and observation z^i) to obtain a local belief $b'_i(s')$, some communication must be allowed among the robots so that they can recover this centralized belief locally [13]. If the measurements obtained by each robot are conditionally independent given the state (a typical assumption in Bayesian data fusion), then $p(z^J|a^J, s') = \prod_i p(z^i|a^i, s')$. By substituting this expression in (2), and assuming that robot actions are known when predicting, robot i can locally combine beliefs received from other robots with its own, $b'_i(s')$, to recover the centralized belief: $b'_{cen}(s') \propto b'_i(s') \prod_{j \neq i} \frac{b'_j(s')}{b'_{ij}(s')}$. This equation fuses the belief in robot i with the one received from j by multiplying them. The additional term $b'_{ij}(s')$ is the common information previously exchanged by the robots, which must be removed not to count it twice. This common information can be maintained by a separate filter called a channel filter [2].

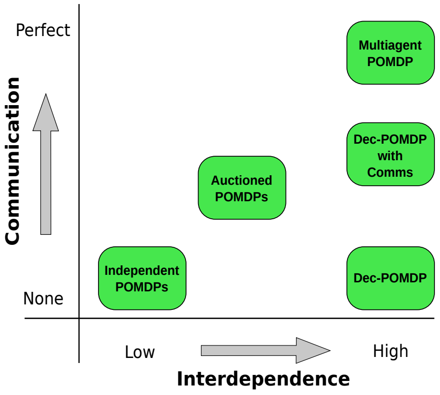


Fig. 1: Multiagent POMDP approaches according to interdependence and level of communication between the agents. “Auctioned POMDPs” refers to our approach.

This fusion scheme can be applied to different belief representations, like Gaussian representations [11], [4] or grids [2]. Robots accumulate information that share with their neighbors in the communication steps. Thus, the belief will propagate through the network, which allows, under certain conditions, the recovery of exactly the same belief as in a centralized node [4].

III. Multiagent Planning under Uncertainty

There is a wide variety of decision-theoretic models to deal with multiagent systems, such as Multiagent POMDPs [13] and Decentralized POMDPs [1], which we compare in terms of agent interdependence and communication assumptions. The level of interdependence is determined by 1) the amount of information that an agent needs to know about the others 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 the policies for the others.

Fig. 1 classifies existing models with respect to their interdependence and the grade of communication that is assumed for the agents. The simplest approach is trying to map the global task 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, there is no explicit cooperation. Many interesting multiagent planning problems cannot be addressed properly by such a loosely coupled approach.

On the other hand, MPOMDPs and Dec-POMDPs solve a single decision-theoretic model for the whole team reasoning about all the actions/observations of each agent (Fig. 1, right column). The MPOMDP model assumes perfect communication and each agent accesses joint actions/observations at every moment, whereas the Dec-POMDP model assumes no communication at all. Such models allow for tight coordination, but they exhibit a high level of interdependence, since any small change in one of the agents entails a recalculation of the policy for the whole team. Also, if due to imperfect communication agents do not have access to other agents' observations, the behavior of the MPOMDP model is not defined. The Dec-POMDP model does not exploit communication, which in many scenarios could be beneficial to improve team performance. In between MPOMDPs and Dec-POMDPs there are several models in which some communication is assumed [10], [14]. These models use the fact that agents actually share information, but just partially and at certain instants (usually without delays).

We aim to exploit the power of decision-theoretic multiagent methods, but keeping in mind the possibilities for multi-robot systems. It is relevant the fact that communication between robots is often possible, but the quality of the channel can vary. This precludes centralized solutions as well as methods requiring communication guarantees.

IV. Decentralized Auction with POMDPs

We focus on decentralized systems [11] in which: 1) There is no central entity required for the operation; 2) There is no common communication facility; that is, information cannot be broadcasted to the whole team, and only local point-to-point communications between neighbors are considered; 3) The robots do not have a global knowledge about the team topology, they only know about their local neighbors. These characteristics make the system scalable as it does not require a central node and enough bandwidth to transmit all the information to that node. Moreover, the system is more robust and flexible with respect to loss or inclusion of new robots (there is no need to know the global topology), and regarding communication issues (a failure does not compromise the whole system).

Our approach builds on two mechanisms to achieve decentralization: the DDF filter in Section II-B for sharing information between robots and a POMDP auction for decentralized behavior coordination (Section IV-A). In Fig. 1, our approach can be seen as in between "independent POMDPs" and MPOMDP/Dec-POMDP in terms of interdependence. In terms of communication requirements, our approach does not require the high-quality guarantees of the methods that enhance the Dec-POMDP model with communications.

A. Auctioning POMDP Policies

In many multi-robot missions there is a certain objective (e.g., detecting a target or alarm) and a set of behaviors or roles that the robots can follow to achieve that objective (e.g., patrol, approach, etc.). In a multi-robot POMDP this overall objective is encoded into a reward function. We propose single-robot behaviors, each of which is modeled as a POMDP with its own reward function. Then, these behaviors can be run simultaneously and combined in some optimal manner to produce a joint behavior similar to the one desired initially. Such a decomposition is reasonable in many robotic applications [9], [7], like surveillance, tracking, fire detection or robotic soccer, in which cooperation between robots playing different roles is required.

The idea is to achieve a multi-robot objective combining a set of simpler reward functions $\{R_1, \ldots, R_M\}$, with $M \neq N$ in general. Each reward function R_k represents a certain single-robot behavior that can be modeled by a POMDP policy with a value function $V_k^\pi(b)$ associated. Although the actual multi-robot objective cannot be modeled as a set of single-robot reward functions, if these policies could be assigned optimally to one or more robots, all together should lead to a cooperative behavior pursuing the global objective. The problem of determining which policy should be assigned to each robot at each step can be modeled as a task allocation [15].

A task allocation algorithm attempts to assign a set of M tasks to a team of N robots minimizing a global cost. In this case, each robot has to be assigned a sole task, which is its POMDP policy. To foster cooperation, different policies are assigned to different robots as long as possible. Given that $x_{ik} = 1$ when policy k is assigned to robot i and 0 otherwise, and c_{ik} is the cost associated with that assignment, the problem consists of minimizing the total cost $\sum_{i=1}^N \left(\sum_{k=1}^M c_{ik} x_{ik}\right)$, subject to:

$$\sum_{i=1}^N x_{ik} \leq 1, \forall k \in \mathcal{K}, \quad \sum_{k=1}^M x_{ik} = 1, \forall i \in \mathcal{I},$$
$$x_{ik} \in \{0,1\}, \quad \forall i \in \mathcal{I}, \forall k \in \mathcal{K},$$

where $\mathcal{I} = \{1, \ldots, N\}$ and $\mathcal{K} = \{1, \ldots, M\}$.

The best behavior for each robot is selected with an auction algorithm [15] where the cost or bid of assigning a policy k to a robot i is $c_{ik} = -V_k^\pi(b_i)$. Thus, policies with greater expected rewards are more likely to be selected for robots, which helps to maximize the global reward for the whole team. If $N > M$, the algorithm will leave robots with no policy assigned. Therefore, the assignment problem is repeated with these free robots until they all get a policy. In this case, some policies would be assigned to more than one robot at the same time. Algorithm 1 summarizes the decentralized auction in which the assignment problem is solved locally at each robot with the information available. Each robot i computes its own bids for the behaviors from its local belief b_i and communicates them to other neighboring robots. Then, with the bids received from other robots, a local solution for the assignment is obtained. This computation can be performed efficiently in polynomial time using the Hungarian algorithm [3]. Note that each robot only consider its neighboring peers (within communication range), what bounds the complexity of the Hungarian algorithm.

B. Discussion

The local cost matrices, and hence the local solutions for the assignments, should be the same at each robot as long as

Algorithm 1 Auctioneer Robot i (b_i)

```
for all $k \in \mathcal{K}$ do
    $c_{ik} = -V_k^{\pi}(b_i)$ {; Local bids}
    Send $c_{ik}$ to neighbors.
end for
Receive bids from neighbors.
$\mathbf{C} = \{c_{jk}\}_{j,k}$ {; Create cost matrix}
$\{x_{jk}\}_{j,k} \leftarrow Hungarian(\mathbf{C})$
return Policy selected for robot $i$.
```

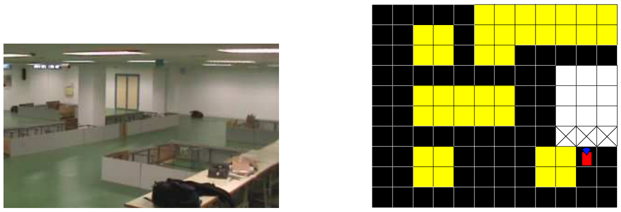
the communication is error-free and the beliefs are equal. However, for DDF systems in which the beliefs are not synchronized all the time, local cost matrices and solutions may differ among robots, leading to suboptimal assignments. A good synchronization of the beliefs is desirable to avoid these situations. In contrast, the robustness of the system is high, since information from all the robots is not required to compute each local solution. In case some communication links failed, each robot would still get a suboptimal solution with the available information from their neighbors.

The approach is completely decentralized, since the belief estimation and the decision-making are carried out without the need for a central entity. The belief estimation is computed by a DDF algorithm that is distributed along the multiple robots. Furthermore, the POMDP controllers act also separately for each robot. Despite the fact that a multi-robot POMDP for the whole team is not solved (with its computational benefits), cooperative behavior still arises in two manners. First, thanks to the information shared by the different DDF modules to achieve a fused belief (which acts as a coordination signal for policy execution); and second, by sharing the bid values for the decentralized auction, which gives an idea about the behaviors others may be performing.

V. Multi-Robot Cooperative Tracking

Target tracking problems benefit from reasoning about future steps [6]. To illustrate our approach we use an active perception approach where N robots have to track a moving target estimating its position with their sensors. Moreover, their actions are aimed at improving that estimate.

The state for each robot is composed of the target position and its own position and heading. The space is discretized into a grid, and a map of the scenario is assumed to be known. There are four possible headings for every robot: *north*, *west*, *south* or *east*. At each time step, each robot can choose between four possible actions: *stay*, *turn right*, *turn left* or *go forward*. *stay* means doing nothing; when *turning*, the robot changes its heading 90°; and when *going forward*, it moves to the cell ahead. Nonetheless, noisy transition functions for the states of the robots are considered. From one time step to the next, the target can move to any of its 8-connected (and free) cells with equal probability. Besides, the robots carry a bearing sensor that is boolean: *detected* or *non-detected*. These sensors proceed as it follows, if the target is out of its field of view (FOV), the sensor produces a *non-detected* measurement. However, when the target is within its FOV, it can be *detected* with a probability p_D.



(a) Testbed.

(b) Occupancy grid.

Fig. 2: (a) Multi-robot testbed. (b) Testbed occupancy grid (yellow cells are obstacles) and FOV for a robot (white cells). All the robots have the same FOV. If the target is in one of the cells with crosses and the heading is adequate, a high reward is obtained.

The robots aim to improve the target estimation by reducing its uncertainty. Their bearing sensors entail mainly uncertainty in depth, so pointing at the target from different angles definitely helps to reduce the uncertainty of its estimation. Therefore, cross configurations should be fostered by giving a high reward to each robot that is keeping the target within its FOV, and even higher if the robot's orientation differs from the others'.

A multi-robot POMDP is a solution far from scalable with the number of robots. Even considering just two robots and a low number of cells for the grid (~ 80), the problem becomes intractable (for the solver and the computer indicated in the experimental section). Hence, the method presented in this paper to combine individual behaviors is used.

The robots should cooperate to track the target from different directions, so each behavior could consist of following the target from a specific direction. Here, a single-robot behavior for each possible orientation is considered: $\{north, west, south, east\}$. The reward function for the policy k (R_k) gives a high reward to robot i only if the target is within its FOV and its heading h_i corresponds to the orientation of behavior k. Since the objective is tracking the target, the robots should position their sensors in the best way not to lose it. For the sensors proposed, the high reward is only obtained when the target is in one of the closest cells. The robots' FOV and their corresponding cells with high rewards are represented in Fig. 2b.

Finally, to alleviate the complexity of the belief space, Mixed Observability Markov Decision Processes (MOMDPs) [12] are considered to find the policies. The robots' positions are assumed to be observable within the POMDP, which is reasonable if the sensors for self-positioning are accurate enough for a given grid resolution.

VI. Experiments

Some experiments were conducted with the CONET testbed[1] (see Fig. 2a), that allows the user to combine simulated and real robots (Pioneer-3AT).

[1] http://www.cooperating-objects.org

TABLE I: Average results for a three-robot team.

	Error(m)	Entropy
	Auction+DDF	
Robot 0	4.07 ± 0.16	2.61 ± 0.05
Robot 1	3.95 ± 0.15	2.55 ± 0.05
Robot 2	4.18 ± 0.16	2.66 ± 0.05
	Independent+DDF	
Robot 0	6.86 ± 0.32	2.80 ± 0.05
Robot 1	6.70 ± 0.32	2.70 ± 0.05
Robot 2	6.75 ± 0.32	2.68 ± 0.05
	Auction	
Robot 0	9.74 ± 0.29	3.85 ± 0.03
Robot 1	9.41 ± 0.34	3.28 ± 0.06
Robot 2	10.46 ± 0.40	3.62 ± 0.04

A. *Experimental setup*

The map of this testbed was discretized into 2×2-meter cells and resulted in the occupancy grid of 12×10 dimensions shown in Fig. 2b, where cells representing obstacles are in yellow. For all the robots, $p_D = 0.9$ and the FOV was the one shown in Fig. 2b. For each POMDP, the high reward when the target was in one of the nearby cells of the FOV was 100, otherwise the reward was 0. The pursuer's observations were obtained by simulating sensors with the mentioned capabilities on board the robots, since the development of real detectors is out of the scope of this paper.

During all the experiments the target (another robot) followed a path unknown to the pursuers and with a random component. A path planning algorithm was used to reach the high-level goals provided by the POMDP controllers (next cell to move and robot heading), whereas a local navigation algorithm was used to safely navigate. Each robot was running a DDF filter onboard (Section II-B) and an auctioneer controller that executed Algorithm 1.

Three approaches were tested: (i) auctioned POMDPs with DDF; (ii) auctioned POMDPs without DDF; (iii) independent POMDPs with DDF. The two first approaches are based on our auction method, but in the second one, each robot only receives its local sensor readings. In the third approach, a single and independent POMDP is used for each robot and communication between the DDF modules is allowed. All the policies were obtained by solving the corresponding MOMDPs with a C++ implementation of the SARSOP algorithm [12]. The solver ran 1700 seconds for each policy in a computer with an Intel Core 2 Duo processor @2.47GHz. For the approaches (i) and (ii), a different MOMDP is solved for each heading, whereas for approach (iii), there is a single MOMDP independent of the heading.

B. *Experimental results*

First, three Pioneer-3AT were used to track another one. The robots always started at the same fixed points and the sample times were 10 seconds for the decision-making and 3 seconds for the DDF. An experiment of 15 minutes was performed for each of the three approaches above[2].

Some average results are presented in Table I. At each time step, the actual target position is compared to the estimate (cell with highest likelihood). The entropies of the belief ($\sum_{\forall cell} -p_{cell} \log(p_{cell})$) are also averaged. Our approach (Auction+DDF) reduces the entropy and the target localization error with respect to the Independent POMDPs approach, since the robots cooperate and track the target from different directions. Also, the estimation of the target position is worse for the auctioned approach in which no DDF is included. In this example, the mean errors are bounded by the resolution of the cells (2 meters).

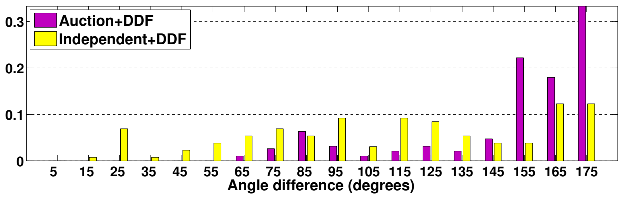


Fig. 3: Normalized histograms of the maximum angle differences between robots when the target is within FOV.

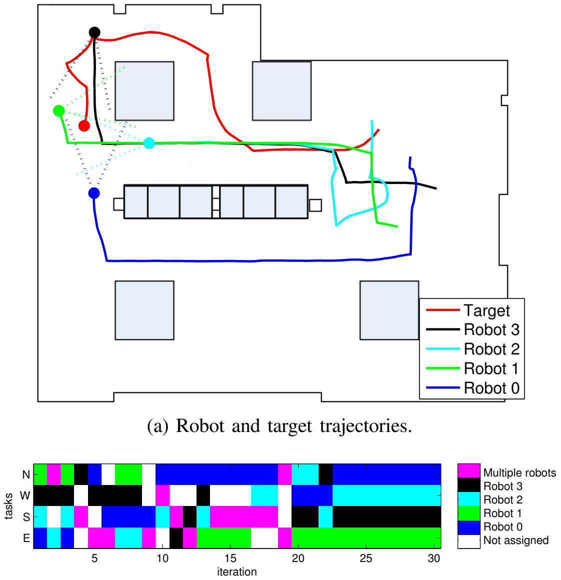
Due to the nature of the sensors, cross configurations among the pursuers allow them to reduce the uncertainty of the estimation. Fig. 3 shows how our auctioned approach fosters these configurations. A comparison with the Independent POMDPs with DDF is made in terms of angle configuration between the pursuers. Normalized histograms of the maximum angle difference between any of the pursuers every time the target is within FOV are shown. The Auction+DDF histogram presents a high peak close to 180° and a small mode in 90° (cross configurations), whereas the histogram is quite flat for the Independent POMDPs.

Second, to show the scalability and robustness of the system, a tracking experiment with a four-robot team was performed. The target was a simulated robot. We run this experiment for more than 30 minutes with the algorithms working on board the robots in a distributed way and using Wi-Fi communications. An extract of the trajectories followed by the pursuers and the target can be seen in Fig. 4a. The orientation of the pursuers at the end of the experiment has also been plotted to show how they surround the target to reduce the estimation uncertainty. When the target turns right, since they know the map, the pursuers opt for going directly to the other exit of the aisle so they can find it there.

The cooperation is depicted in Fig. 4b, which shows the policies allocated to each robot during the same time frame (each iteration takes place every 10 seconds). Due to differences in the local beliefs and different decision times for the robots, inconsistent solutions (robots with the same policy) are obtained in some occasions. In the end, due to the target path, the assignment stabilizes (after iteration 22 in Fig. 4b).

Finally, some simulations were run in the testbed to check how our auction algorithm performs when the robots do not access the same information. The experiments consisted of three simulated robots, 2 pursuers and a target, starting at the same positions in each experiment. The communication latency for the DDF modules was varied throughout the

[2] See video at http://vimeo.com/18898325.



(a) Robot and target trajectories.

(b) Policy assignment.

Fig. 4: Experiment with a four-robot team. (a) Trajectories and final orientations of the robots and the target. (b) Policies allocated to each robot during the same time frame.

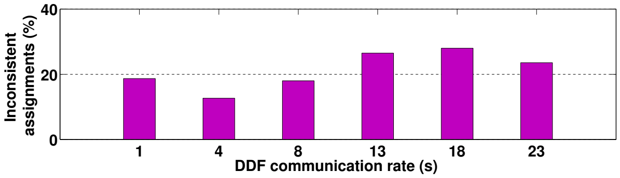


Fig. 5: Evolution of the performance of the distributed auction under variable transmission rate for the DDFs.

simulations (two simulations of 20 minutes for each latency value). The communication rate for the bid values was maintained, as the data volume is not significant for the bandwidth compared to the belief information. The percentage of inconsistent assignments for the distributed auction is shown in Fig. 5. As the communication rate for the DDFs is increased, the difference between the beliefs to which the robots have access grows too. Thus, the consistency of the assignments becomes more difficult under worse communications. However, Fig. 5 shows a graceful degradation.

VII. Conclusions

Planning-under-uncertainty techniques, such as POMDPs, face a scalability problem when considering teams of robots. Popular frameworks like Decentralized POMDPs scale poorly to many robots, unless very severe independence assumptions are applied [10]. Many of these models either do not allow robots to exploit inter-agent communication, or implicitly assume instantaneous cost-less communication (MPOMDP). We focus on scalable techniques that do not require such strict communication guarantees, which are hard to meet in multi-robot domains with unreliable wireless channels.

This paper presents an approach based on DDF and auctioning of independent POMDP-based controllers during the execution phase to generate a cooperative behavior in the team. Our approach is much more scalable than other multiagent POMDP approaches, and allows the robots to exploit imperfect communication channels, offering a trade-off between optimality and applicability. We presented as proof of concept results on a cooperative tracking application by a team of up to 4 robots. The same application cannot be solved with the current state of the art in multiagent POMDP solvers. Additionally, other multi-robot applications that can be achieved through cooperative behaviors can be modeled with this framework. For instance, the method can be used in robotic soccer (allocating the best behaviors/roles to the team depending on the current belief); or in fire-fighting applications [9]. In the future, we will investigate the exact range of multi-robot planning domains for which our approach is valuable.

References

[1] D. S. Bernstein, R. Givan, N. Immerman, and S. Zilberstein. The complexity of decentralized control of Markov decision processes. *Mathematics of Operations Research*, 27(4):819–840, 2002.

[2] F. Bourgault and H.F. Durrant-Whyte. Communication in general decentralized filters and the coordinated search strategy. In *Proc. of The 7th Int. Conf. on Information Fusion*, 2004.

[3] R. E. Burkard. Selected topics on assignment problems. *Discrete Applied Mathematics*, 123(1-3):257 – 302, 2002.

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

[5] B. P. Gerkey and M. J. Matarić. A formal analysis and taxonomy of task allocation in multi-robot systems. *International Journal of Robotics Research*, 23(9):939–954, 2004.

[6] R. He, A. Bachrach, and N. Roy. Efficient planning under uncertainty for a target-tracking micro-aerial vehicle. In *Proc. ICRA*, 2010.

[7] M. A. Hsieh, A. Cowley, J. F. Keller, L. Chaimowicz, B. Grocholsky, V. Kumar, C. J. Taylor, Y. Endo, R. C. Arkin, B. Jung, D. F. Wolf, G. S. Sukhatme, and D. C. MacKenzie. Adaptive teams of autonomous aerial and ground robots for situational awareness. *Journal of Field Robotics*, 24:991–1014, 2007.

[8] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101:99–134, 1998.

[9] L. Merino, F. Caballero, J.R. Martínez de Dios, J. Ferruz, and A. Ollero. A cooperative perception system for multiple UAVs: Application to automatic detection of forest fires. *Journal of Field Robotics*, 23:165–184, 2006.

[10] R. Nair, M. Tambe, M. Roth, and M. Yokoo. Communication for improving policy computation in distributed POMDPs. In *Proc. AAMAS*, 2004.

[11] E. Nettleton, H. Durrant-Whyte, and S. Sukkarieh. A robust architecture for decentralised data fusion. In *Proc. ICAR*, 2003.

[12] S. Ong, S. Wei Png, D. Hsu, and W. Sun Lee. POMDPs for Robotic Tasks with Mixed Observability. In *Proc. RSS*, 2009.

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

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

[15] M.T.J. Spaan, N. Gonçalves, and J. Sequeira. Multirobot coordination by auctioning POMDPs. In *Proc. ICRA*, 2010.