Decentralized Multi-Robot Cooperation with Auctioned POMDPs

Jesús Capitán
University of Seville
Seville, Spain
jescap@cartuja.us.es

Matthijs T.J. Spaan
Inst. for Systems and Robotics
Instituto Superior Técnico
Lisbon, Portugal
mtjspaan@isr.ist.utl.pt

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

Aníbal Ollero
University of Seville
Seville, Spain
aollero@cartuja.us.es

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 multi-agent Partially Observable Markov Decision Process (POMDPs) while maintaining cooperation between robots by using POMDP policy auctions. Auctions provide a flexible way of coordinating individual policies modeled by POMDPs and have low communication requirements. Additionally, communication models in the multi-agent 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 coordination that our techniques can provide.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent systems

General Terms

Algorithms, Experimentation

Keywords

POMDP Auctions, Decentralized Data Fusion, Cooperative Tracking

1. INTRODUCTION

In many robotic applications, such as surveillance or rescue robotics, the use of multi-robot systems is of great interest [13, 10]. In those applications, a single robot is not usually 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 this paper, we propose a scheme for exploiting the power of decision-theoretic planning methods, while mitigating their complexity by lowering the dependence between individual plans. Robotic teams commonly are capable of communicating, which we exploit for maintaining a decentralized state estimate. A key point in our approach however, is that we relax the strict assumptions on the quality of the communication channel commonly found in the literature on multi-agent planning under uncertainty [19, 15, 20].

The Partially Observable Markov Decision Process model (POMDP) provides a sound mathematical framework for decision-making in uncertain and partially observable environments [12], which are commonly faced by robotic teams. Although there are POMDP solvers able to successfully handle large state spaces currently, POMDPs ultimately face a scalability problem when considering planning for multi-agent teams [21]. Popular models like Dec-POMDPs [1] or ND-POMDPs [16] remain limited to toy problems, and other models require flawless instantaneous communication [19, 15, 20].

In contrast, we consider fully decentralized solutions, that is, solutions that only involve local information and local communications, and which do not depend on the total number of agents. In particular, this paper proposes an intermediate approach that solves independent POMDPs for each agent but still allows 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 [7, 14, 24], and have also been explored in conjunction with POMDPs [22].

We propose to decompose a multi-agent POMDP by factorizing its goal into several behaviors that can be represented by single-agent POMDPs. We generalize a centralized POMDP auction [22] to auction never-ending tasks (behaviors) that can be assigned to different robots at every step without completion (unlike classic task allocation). In this

The Sixth Annual Workshop on Multiagent Sequential Decision-Making in Uncertain Domains (*MSDM-2011*), held in conjunction with *AAMAS-2011* on May 3, 2011 in Taipei, Taiwan.

novel decentralized auction, instead of tasks, policies that describe a behavior towards a common goal are distributed; agents can switch between these behaviors dynamically at each decision step, and the auction is used to determine continuously which behavior is best for each agent to cooperatively attain the goal. Since independent single-agent POMDPs are solved for every single agent, the interdependence of the model is low and the approach can scale well with the number of agents.

The second key component is to efficiently maintain a joint belief state among the robots, which can serve as coordination signal. Previous work has considered Decentralized Data Fusion (DDF) for this purpose [8, 4], but the novelty in this work is to use a DDF approach 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, which allows the agents to estimate the state with the DDF algorithm. This decoupling between both processes increases the robustness and reliability of real-time robotic teams.

We illustrate our method in a multi-robot tracking application, in which several robots have to cooperate in order to track a moving target as accurately as possible. Besides cooperative tracking, our techniques are suited for a range of problems such as surveillance [10] or forest fire detection [13] which call for a cooperative effort of robots coordinating their individual behaviors. We demonstrate our distributed POMDP approach in a multi-robot testbed, in a fully decentralized setup where each robot runs its own POMDP policy.

The paper is organized as follows: Section 2 summarizes POMDP models and describes the decentralized data fusion algorithms. Section 3 discusses current approaches in the literature for multi-agent planning under uncertainty; Section 4 describes the overall system and the algorithms for auctioning POMDPs in a decentralized manner; Section 5 presents an application for cooperative tracking with multi-robot systems; Section 6 provides experimental results; and Section 7 gives the conclusions and future work.

2. BACKGROUND

We give a short description of the POMDP model for single-agent and multi-agent planning uncertainty, followed by a method for maintaining a joint belief by multiple agents.

2.1 POMDP model

A POMDP is defined as a tuple $\langle S, A, Z, T, O, R, h, \gamma \rangle$ [12]. The *state space* is the finite set of possible states $s \in S$; the *action space*, 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. Thus, 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 total expected reward earned during h time steps. To ensure that this sum is finite when $h \to \infty$, rewards are weighted by a discount factor $\gamma \in [0, 1)$.

Given that it is not directly observable, the actual state cannot be known by the system. Instead, a probability density function $b(s)$ over the state space is maintained. This is called the *belief state* and, due to the Markov assumption, it can be updated with a Bayesian filter for every action-observation pair:

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

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 total expected reward is maximized. This expected reward gathered by following π starting from belief b is called the value function:

$$V^{\pi}(b) = E\left[\sum_{t=0}^{h} \gamma^t r(b_t, \pi(b_t)) | b_0 = b\right] \tag{2}$$

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 agents that share the same reward function is considered, it is straightforward to extend the previous framework. In that case, each agent 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 now defined over the joint set of states and actions $R : S \times A_1 \times \cdots \times A_N \to \mathbb{R}$. The goal in this case is to compute an optimal joint policy $\pi^* = \{\pi_1, \cdots, \pi_N\}$ that maximizes the expected discounted reward.

2.2 Decentralized Data Fusion

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

$$b'_{cen}(s') = \eta p(z^J | a^J, s') \sum_{s \in S} p(s' | a^J, s) b_{cen}(s) \tag{3}$$

where $a^J = \langle a^1, \cdots, a^N \rangle$ denotes the joint action and $z^J = \langle z^1, \cdots, z^N \rangle$ the joint measurement. However, if the belief estimation is decentralized and each agent i uses only its local information (action a^i and observation z^i), some communication must be allowed among the agents so that they can recover this centralized belief locally [19].

The *conditional independence* assumption of the measurements (given the state at s') is typical in Bayesian data fusion. This is reasonable when the measurements obtained by each agent do not depend on the state of the other agents. Therefore, assuming this particular independence ($p(z^J | a^J, s') = \prod_i p(z^i | a^i, s')$) and assuming that agent actions are known when predicting, it is possible to combine locally received beliefs from other agents with the one from agent i, $b'_i(s')$, to recover the centralized belief:

$$b'_{cen}(s') \propto b'_i(s') \prod_{j \neq i} \frac{b'_j(s')}{b'_{ij}(s')} \tag{4}$$

Equation 4 fuses the belief in agent i with the one received from j by multiplying them. The common information pre-

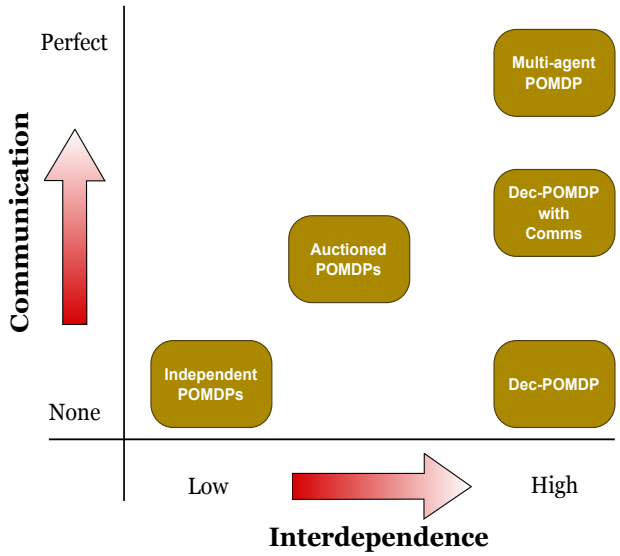


Figure 1: Classification of multi-agent POMDP approaches according to interdependence and level of communication between the agents. "Auctioned POMDPs" refers to the proposed approach.

viously exchanged by the agents $b'_{ij}(s')$ must be removed not to count it twice. This common information can be maintained by a separate filter called channel filter. Further details about the channel filter and the equation above can be seen in [2].

In previous work we have shown that if the state is dynamic, it is possible to obtain locally the same belief as in a centralized node with access to all the information available, by including delayed states in the belief [4]. This represents a common belief signal for all the agents. If only the current state is considered, some information is lost with respect to an ideal centralized fusion unless the agents communicate every time they gather new information [2].

3. MULTI-AGENT PLANNING UNDER UNCERTAINTY

In the literature a wide variety of decision-theoretic models exist to deal with multi-agent systems [21], such as Multi-agent POMDPs [19] and Decentralized POMDPs [1], which we compare 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 the other agent 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. Many models from the literature are highly interdependent, for instance Multi-agent POMDPs (MPOMDP), Dec-POMDPs, and ND-POMDPs [16].

Fig. 1 presents a possible classification of existing models with respect to their interdependence and the grade of communication that is assumed for the agents. 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 too. Many interesting multi-agent planning problems cannot be tackled adequately with 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 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 a tight cooperation, 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. 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.

In between MPOMDPs and Dec-POMDPs there are several models in which some communication is assumed [15, 20, 23]. 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.

In our work, we aim to exploit the power of decision-theoretic multi-agent methods, but keeping in mind the possibilities and constraints posed by multi-robot systems. Of particular relevance in our context is the fact that communication between robots is often possible, but the quality of the communication channel can vary. This precludes centralized solutions as well as methods requiring communication guarantees.

4. DECENTRALIZED AUCTION WITH POMDPS

As mention in the introduction of this paper, we focus on decentralized models. In particular, we follow the definition of a decentralized system given in [17]:
1) There is no central agent 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 agents 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 agents (there is no need to know the global topology), and with respect to communication issues (a failure does not compromise the whole system).

The proposed approach builds on two mechanisms for achieving decentralization: the decentralized data fusion filter we described in Section 2.2 for sharing information between agents and a POMDP auction for decentralized behavior coordination (Section 4.1). In Fig. 1, in terms of agent interdependence, our approach can be seen as in between "independent POMDPs" and MPOMDP/Dec-POMDP. 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.

4.1 Distributed POMDP with Auction

There is a wide range of missions that can be accomplished by a team of multiple robots. In all these 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 the case of a multi-agent POMDP, this overall objective is encoded into a reward function. We assume that the problem can be decomposed into different simultaneous behaviors, each of which is modeled as a POMDP with its own reward function. Then, the joint behavior produced by the distributed POMDPs should be similar to the one desired for the whole team initially. Such a decomposition is possible in many robotic applications [13, 10], like surveillance, tracking, forest fire detection or robotic soccer, in which cooperation between robots playing different roles is required.

The global multi-agent objective can be distributed into a set of simpler reward functions $\{R_1, \ldots, R_M\}$. In our case, each reward function R_k represents a certain single-agent behavior that can be modeled by a POMDP policy with a value function $V_k^\pi(b)$ associated. Then, assuming that each of the robots can execute these policies, the combination of all of them should lead to a cooperative behavior that follows 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 problem [22].

In general, a task allocation algorithm attempts to assign a set of M tasks to a team of N agents minimizing a global cost. In this case, each robot always has to be assigned a sole task, which is the POMDP policy to follow. In order to foster cooperation, different policies must be 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 can be formulated as follows:

$$\min \sum_{i=1}^{N} \left(\sum_{k=1}^{M} c_{ik} x_{ik} \right) \tag{5}$$

subject to

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

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

$$x_{ik} \in \{0, 1\}, \quad \forall i \in \mathcal{I}, \forall k \in \mathcal{K}$$

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

In order to select the best behavior for each robot, we propose an auction algorithm similar to our previous work [22]. The cost or bid of assigning a policy k to a robot i is $c_{ik} = -V_k^\pi(b_i)$. 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. In case $N > M$, the Hungarian algorithm will leave robots with no policy assigned. Therefore, the assignment problem is repeated with these free robots until they all get a policy assigned. Note that in this special case, some policies would be assigned to more than one robot at the same time.

Algorithm 1 Auctioneer Robot i (b_i)

1: **for all** $k \in \mathcal{K}$ **do**
2: $\quad c_{ik} = -V_k^\pi(b_i)$ {; Local bids}
3: $\quad$ Send c_{ik} to neighbors.
4: **end for**
5: Receive bids from neighbors.
6: $\mathbf{C} = \{c_{ik}\}_{i,k}$ {; Create cost matrix}
7: $\{x_{ik}\}_{i,k} \leftarrow Hungarian(\mathbf{C})$
8: **return** Policy selected for robot i.

Algorithm 1 summarizes a decentralized auction approach 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 problem (5) is obtained. This computation can be performed efficiently in polynomial time using the Hungarian algorithm [3].

In general, the local cost matrices, and hence the local solutions for the behavior assignment, should be the same at each robot as long as the communication is error-free and the beliefs are common. However, for DDF systems in which the local beliefs are not synchronized all the time, inconsistencies that lead to suboptimal solutions may be obtained from time to time. In an inconsistent distributed solution, due to differences in the local cost matrices, the same policy is allocated to more than one robot. Therefore, a good synchronization of the local beliefs is desirable to avoid these situations. On the other hand, 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 (subnetworks arise naturally).

In addition, there is another potential desynchronization due to the fact that the robots may not have synchronized execution time, which would lead them to make decisions at different moments and with different available information. Some previous works [5] propose consensus algorithms over this information in order to guarantee convergence for decentralized auction approaches even in case of time desynchronization. Nonetheless, the decision-making performance is still degraded. Moreover, the dynamics of the system presented here are higher, since each robot is allowed to change its policy at every step. Therefore, the establishment of a previous consensus to converge to the same distributed solution is not worthwhile.

4.2 System Overview

Fig. 2 illustrates the system elements per robot. The whole process is separated into two different modules. Each robot can execute a certain number of behaviors modeled as single-agent POMDP controllers. A DDF module is in charge of computing the belief and feeding the Auctioneer module, which then chooses the adequate POMDP controller and the associated action. Although most POMDP-based systems synchronize belief update and decision making in the same loop, here the two processes are separated. In this way some constraints that limit the flexibility and robustness of the system are avoided. For instance, commu-

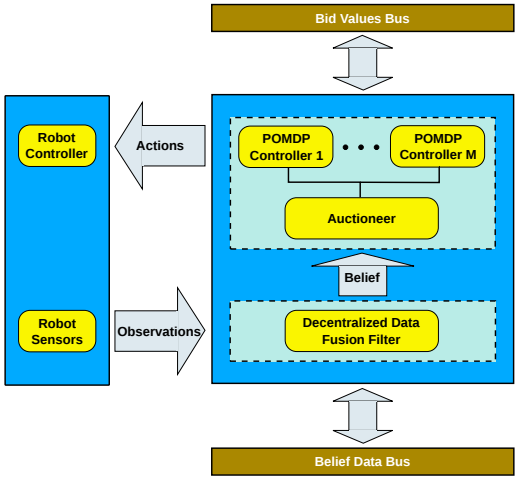


Figure 2: Functional scheme for decision-making and belief update at each robot.

nication channels and transmission rates are totally independent for both modules, which is critical in decentralized systems under possible communication failures.

The approach is totally decentralized, since the belief estimation as well as the decision-making are carried out without the need for a central entity. On the one hand, the belief estimation is computed by a DDF algorithm that is distributed along the multiple robots. On the other hand, the POMDP controllers act also separately for each robot. Despite the fact that a multi-agent POMDP for the whole team is not solved (with its computational benefits), a cooperative behavior still arises in two manners. First, thanks to the information shared by the different DDF modules in order to achieve a fused belief (which then acts as a Markovian 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.

5. MULTI-ROBOT COOPERATIVE TRACKING

In order to illustrate the proposed approach, an application for tracking a target by means of multiple robots is considered here. Target tracking is a problem in which reasoning about future steps is needed to optimize the search [9]. Besides, cooperative behaviors are particularly helpful when there are multiple robots involved in the traking. In our problem there is a moving target and a team of N robots which are the pursuers. Each robot carries a bearing-only sensor which determines whether the target is visible or not within its field of view (FOV). Thus, the objective is to find the target in the environment and localize it as well as possible.

The state for each robot is composed of the position of the target and its own position and heading. The state space is discretized into a cell 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. Besides, the target is assumed to move randomly. Therefore, the transition function for its position indicates that from one time step to the next, the target can move to any of its 8-connected cells with the same probability (only non-obstacle cells are considered in order to calculate that probability).

In addition, every sensor provides a boolean measurement: *detected* or *non-detected*. These sensors proceed as it follows, if the target is out of its 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.

The design of the reward function is crucial. Since the target must be tracked by the team, the more robots have it within their FOV, the higher reward the system should be given. Besides, given that the sensors provide bearing information, it is quite reasonable to reward cross configurations between the robots. 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, a high reward should be given for each robot that is keeping the target within its FOV, and even higher if the robot's orientation differs from the others'.

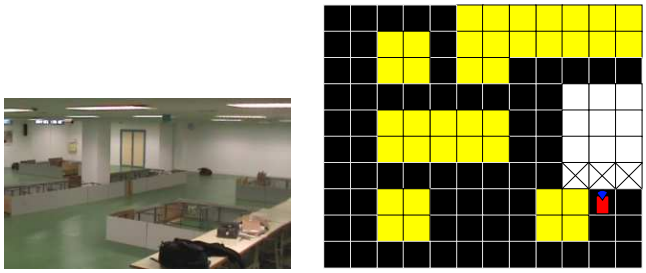
A multi-agent POMDP might be solved in order to deal with this problem. However, this solution is far from scalable with the number of robots. Actually, even considering just two robots and a reasonable number of cells for a real grid (~ 80), the problem becomes intractable (considering the solver and the computer indicated in the experimental section). Hence, the method presented in this paper to distribute the reward function is used.

The key idea is to distribute the reward function so that the same desired behavior is obtained for the whole team. In this case, the robots should track the target from different directions, so the kind of behaviors to be allocated could consist of following the target from a specific direction. For this application, four single-agent behaviors are considered, one for each possible orientation $\{north, west, south, east\}$. Therefore, 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 the robot's heading h_i corresponds to the orientation of behavior k. Since the objective is to track the target, the robots should try to get closer to the target when possible. Hence, the high reward is just obtained when the target is in one of the closest cells. The FOV for each robot and the corresponding cells with a high reward are represented in Fig. 3b.

Finally, in order to alleviate the complexity of the belief space, Mixed Observability Markov Decision Processes (MOMDPs) [18] are considered to find the policies. Hence, the robots' states are assumed to be observable within the POMDP. This is reasonable for this robotic task in which the sensors onboard allow the robots to assume their own states observable (at least for a given resolution).

6. EXPERIMENTS

Some experiments were conducted with the real testbed of the Cooperating Objects Network of Excellence (CONET) that allows the user to combine simulated robots with four real robots (Pioneer-3AT) [11]. Fig. 3a shows an illustration of the testbed. A simulated version of the testbed is also available in Player/Stage [6].



(a) Testbed. (b) Occupancy grid.

Figure 3: (a) Picture of the multi-robot testbed. (b) Testbed occupancy grid (yellow cells are obstacles) and an example of the FOV for a robot (white cells). All the robots have the same FOV. Besides, if the target is in one of the cells with crosses and the heading is adequate, a high reward is obtained.

	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

Table 1: Average results of the experiments with a three-robot team for three different approaches. The error of the estimated position of the target with respect to its actual position and the entropy of the estimated beliefs are shown.

6.1 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. 3b, where cells representing obstacles are in yellow. A team of robots was considered in order to follow a target, represented by another robot. All the pursuers presented similar perception capabilities by means of a sensor with $p_D = 0.9$ and the FOV shown in Fig. 3b. For each POMDP, the high reward when the target was in one of the close cells of the FOV was 100, otherwise the reward was 0. Note that the pursuer 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 followed a path unknown for the pursuers and with a random component. Furthermore, Player was used to command the robots, which were able to localize themselves within the map. A path planning algorithm was used to obtain the path to 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 the given path. Each robot had running onboard an estimation filter implementing the DDF scheme in Section 2.2 and an auctioneer controller that executed the algorithm in Section 4.1.

We tested the following three approaches: (i) auctioned POMDPs with DDF; (ii) auctioned POMDPs without DDF; (iii) independent POMDPs with DDF. The two first approaches are based on the auction method proposed in this paper, but in the second one, neither communication nor fusion is considered for the DDF modules. In the third approach, a single and independent POMDP is used for each robot and communication between the DDF modules is allowed. Moreover, all the policies were obtained by solving the corresponding MOMDPs with a C++ implementation of the SARSOP algorithm [18]. The solver ran 1700 seconds for each policy in a computer with an Intel Core 2 Duo processor @2.47GHz and 2.9GB. 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.

6.2 Experimental results

First, some experiments[1] were carried out in order to compare our approach with the others mentioned above. Three of the real Pioneer-3AT were used to track the remaining one, that played the target's role. In order to have similar conditions for each run, the three robots always started at the same fixed points and the sample times were the same, 10 seconds for the decision-making modules and 3 seconds for the DDF modules. An experiment of 15 minutes was performed for each of the three approaches.

Some average results with their standard deviations are presented in Table 1. At each time step, the target localization is estimated by searching the cell of the belief with a highest probability. This value is compared to the actual target position. The entropies of the belief at each step ($\sum_{\forall cell} -p_{cell} \log(p_{cell})$) are also averaged and shown. It can be seen that the approach proposed in this paper (Auction+DDF) reduces the entropy and the target localization error with respect to the Independent POMDPs approach, since the cooperation between the members of the team allows them to surround the target, pointing at it from different points of view. It can also be noticed that the estimation of the target position is worse for the auctioned approach when no DDF is included. Note that in this case, the mean errors are bounded by the resolution of the cells (2 meters). The option of Independent POMDPs without DDF resulted in very poor performance (and hence is not included), as the robots do not share any information nor coordinate their behaviors.

Due to the bearing information encoded in the sensor observations, a cross configuration among the pursuers allows them to point at the target from different points of view and reduce the uncertainty of its estimation. This cross configuration is fostered by our auctioned approach, as it can be seen in Fig. 4. This figure compares our approach to the Independent POMDPs with DDF 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 around 180° and a small mode in 90° (cross configurations), whereas the histogram is quite flat for the Independent POMDPs. This

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

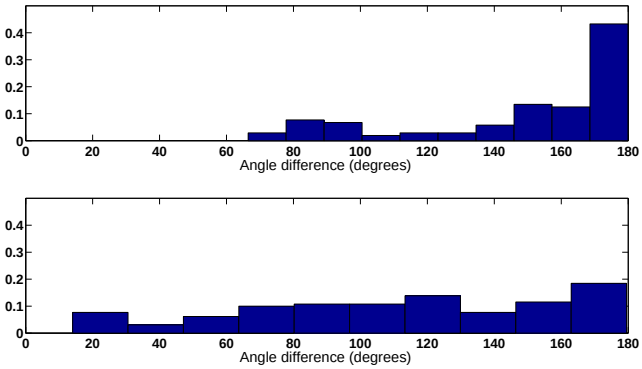


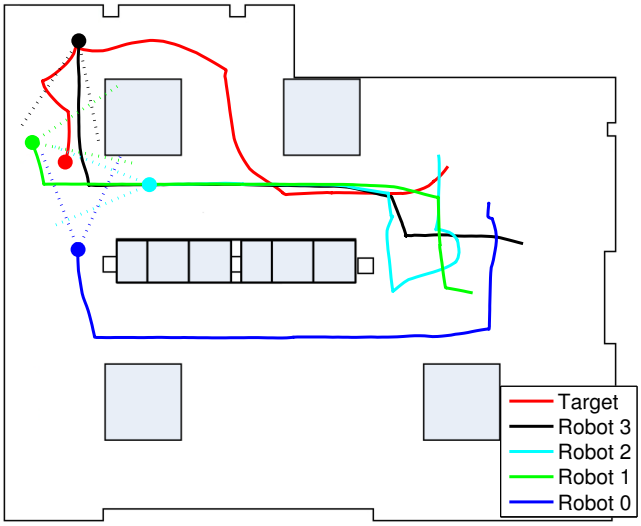
Figure 4: Normalized histograms of the maximum angle differences between the robots when the target is within the FOV of any of them. At the top Auction+DDF; at the bottom Independent POMDPs+DDF.

show that the proposed approach is more effective in reaching cross configurations.

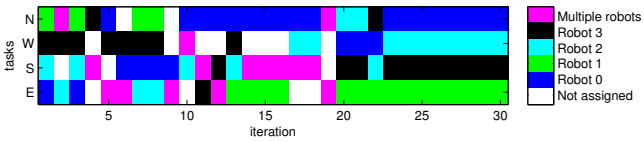
Second, to show the scalability of the system, a tracking experiment with a four-robot team was performed. In this case, the target was represented by a simulated robot. We were able to run this experiment for more than 30 minutes with the algorithms working on board the robots in a distributed way and using Wi-Fi communications, showing the robustness of the system. An extract of the trajectories followed by the pursuers and the target can be seen in Fig. 5a. 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. Note that, 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 between the members of the team is depicted in Fig. 5b, 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. However, as time passes the robots have built up a better belief regarding the target's position, and the assignment stabilizes (after iteration 22 in Fig. 5b).

Finally, some experiments in the simulated testbed were carried out in order 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 with sample times as before. However, the communication latency for the DDF modules was varied throughout the different simulations (two simulations of 20 minutes for each latency value). Even though the communication rate for the auctioneers may also have been varied, the amount of data transmitted by them (four scalar values, the bids, each robot) is not significant for the bandwidth when compared to the belief information and, hence, it can be maintained (one each 10 seconds). The performance of the distributed auction algorithm is shown in Fig. 6 by representing the percentage of inconsistent assignments. As the communication rate for the DDFs is increased, the difference between the beliefs to which the agents have access grows too. Thus, the



(a) Robot and target trajectories.



(b) Policy assignment.

Figure 5: Experiment with a four-robot team. (a) Trajectories followed by the robots and the target during the experiment. Orientations at the last time step are also shown. (b) Policies allocated to each robot during the same time frame.

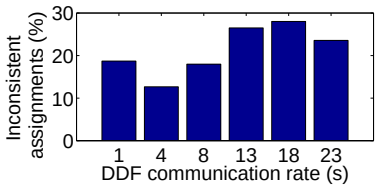


Figure 6: Evolution of the performance of the distributed auction under variable transmission rate for the DDFs.

consistency of the assignments becomes more difficult under worse communications. However, Fig. 6 shows a graceful degradation with respect to the communication rate.

7. CONCLUSIONS

Planning-under-uncertainty techniques, such as POMDPs, face a scalability problem when considering teams of robots. Popular frameworks like Dec-POMDPs scale poorly to many agents, unless very severe independence assumptions are applied (e.g., the ND-POMDP model). Furthermore, many of these models either do not allow agents 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 decentralized data fusion (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 multi-agent 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 multi-agent POMDP solvers. Besides, our framework is more general. DDF can be used for policy execution coordination in other multi-robot applications; and applications that can be achieved through cooperative behaviors can also 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 [13]. In the future, we will investigate the exact range of multi-robot planning domains for which our approach is valuable.

Acknowledgments

This work has been partially supported by CONET, the EU Cooperating Objects Network of Excellence (FP7-2007-2-224053) and the ROBAIR Project funded by the Spanish Research and Development Program (DPI2008-03847). This work was partially funded by Fundação para a Ciência e a Tecnologia (ISR/IST pluriannual funding) through the PIDDAC Program funds and was supported by project CMU-PT/SIA/0023/2009 under the Carnegie Mellon-Portugal Program.

8. 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. 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. Delayed-State Information Filter for Cooperative Decentralized Tracking. In *Proc. ICRA*, 2009.

[5] H.-L. Choi, L. Brunet, and J. How. Consensus-Based Decentralised Auctions for Robust Task Allocation. *IEEE Transactions on Robotics*, 25:912–926, 2009.

[6] B. Gerkey, R. Vaughan, and A. Howard. The Player/Stage Project: Tools for Multi-Robot and Distributed Sensor Systems. In *Proc. ICAR*, 2003.

[7] 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.

[8] B. Grocholsky, A. Makarenko, T. Kaupp, and H. F. Durrant-Whyte. *Lecture notes in Computer Science*, volume 2634, chapter Scalable Control of Decentralised Sensor Platforms. Springer, 2003.

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

[10] 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.

[11] A. Jiménez-González, J. Martínez-de Dios, and A. Ollero. An integrated testbed for heterogeneous mobile robots and other cooperating objects. In *Proc. of International Conference on Intelligent Robots and Systems*, 2010.

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

[13] L. Merino, F. Caballero, J. M. 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.

[14] A. Mosteo and L. Montano. Comparative Experiments on Optimization Criteria and Algorithms for Auction based Multi-Robot Task Allocation. In *Proc. ICRA*, 2007.

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

[16] R. Nair, P. Varakantham, M. Tambe, and M. Yokoo. Networked distributed POMDPs: A synthesis of distributed constraint optimization and POMDPs. In *Proc. AAAI*, 2005.

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

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

[19] 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.

[20] 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.

[21] S. Seuken and S. Zilberstein. Formal models and algorithms for decentralized decision making under uncertainty. *Autonomous Agents and Multi-Agent Systems*, Feb. 2008.

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

[23] M. T. J. Spaan, F. A. Oliehoek, and N. Vlassis. Multiagent planning under uncertainty with stochastic communication delays. In *Proc. ICAPS*, 2008.

[24] A. Viguria, I. Maza, and A. Ollero. S+T: An Algorithm for Distributed Multirobot Task Allocation based on Services for Improving Robot Cooperation. In *Proc. ICRA*, 2008.