2010 IEEE International Conference on Robotics and Automation
Anchorage Convention District
May 3-8, 2010, Anchorage, Alaska, USA

Multirobot Coordination by Auctioning POMDPs

Matthijs T.J. Spaan, Nelson Gonçalves and João Sequeira

Abstract— **We consider the problem of task assignment and execution in multirobot systems, by proposing a procedure for bid estimation in auction protocols. Auctions are of interest to multirobot systems because they provide a flexible way to coordinate the assignment of tasks to robots. The main idea is to exploit task execution controllers that rely on the availability of value functions. These provide a natural way to obtain the bid values for a given task, compared to the heuristic and ad-hoc bid estimation procedures in common use. The Partially Observable Markov Decision Process (POMDP) framework is used to compute policies for the execution of tasks by each agent, with the task bid values obtained directly from the respective value functions. Several simulation examples are presented for an urban surveillance environment, illustrating the applicability of our ideas.**

I. Introduction

We consider the problem of the assignment and execution of tasks in multirobot systems. Auction protocols for computing the assignment of tasks are commonly used in multirobot systems [4]. The main advantages of these protocols are their robustness to individual agent failures and the reduced bandwidth requirements [6]. Another advantage is that the assignment solution is computed in a distributed manner, and thus can be used by agents with low computational resources.

A crucial challenge in auction protocols is how to estimate the value that each agent should bid for each task, given that agents must evaluate their fitness for executing a task using only locally available information. In mobile robotic applications, tasks often consist of the execution of a path [6], [4]. Thus, the bid value for each task is often a function of the path distance, the travel time or a combination of these measures [14]. In general, the fitness functions are heuristic and must be defined for each task, usually in an ad-hoc manner.

Instead, we propose to employ the value functions used in design of the controllers for the execution of each task. A value function estimates the benefit of taking a particular action in a particular state, given the long-term objective of executing a task. The fitness of an agent to execute a task, given the state of the environment, is thus obtained directly from the value functions. The main advantage is that the bid functions are not tailored for the application at hand, but instead are obtained naturally from the requirements of the tasks. In this paper, tasks are defined and solved using Partially Observable Markov Decision Processes (POMDPs) [10]. POMDPs form a general and powerful mathematical basis for planning under uncertainty, and their use in mobile robotic applications has increased in recent times [21], [20].

Auction protocols and POMDPs are complementary frameworks. The value functions, required to compute the bids in auctions, are readily available when the execution of tasks is formulated with POMDPs. The coordination among the different robots is achieved by the auction protocol because it computes the optimal task assignment given the individual fitness values. As a result, the POMDP problems have much smaller dimensions since full joint planning is not necessary. Although our focus is not on efficient POMDP solving, avoiding POMDP models that are exponentially sized in the number of robots greatly improves scalability.

We demonstrate our ideas in a simulation of an active surveillance system, illustrating the benefits of combining POMDPs and auction protocols, as well as showing the limits of centralized POMDP solutions.

The remainder of this paper is organized as follows. Section II presents an overview of the proposed approach. The POMDP framework is reviewed in Section III, and Section IV describes the auction protocol. In Section V the proposed approach for the estimation of bids is proposed. In Section VI the approach is applied to an active surveillance problem and evaluated in simulation. Finally, in Section VII the results and future work are discussed.

II. Coordination in Multirobot Systems

The problems considered in this paper are the assignment of tasks and their execution in a multirobot system. The first is formulated as the assignment of tasks with unknown arrival order. Each robot can execute only one task at the time, although it can be interrupted to begin the execution of another. It is assumed that communication and hardware failures may occur and consequently, the number of available mobile robots at any given instant is not known. The computation of the assignment solution when the order of task arrivals is known and no failures occur is NP-hard in general [9]. The available algorithms proposed for this problem often require computational resources organized in a centralized manner. For instance, in [8] the order of the tasks arrival is known and the task allocation is solved in a centralized manner.

The second problem is the synthesis of controllers for the execution of each task by the robots. Each has available a

This work was supported by the European Project FP6-2005-IST-6-045062-URUS, and was funded by Fundação para a Ciência e a Tecnologia (ISR/IST pluriannual funding) through the PIDDAC Program funds as well as by project PTDC/EEA-ACR/73266/2006. Nelson Gonçalves is working under grant SFRH/BD/23804/2005 from Fundação para a Ciência e a Tecnologia.

All authors are with Instituto de Sistemas e Robótica, Instituto Superior Técnico, Lisbon, Portugal. {mtjspaan, ngoncalves, jseq}@isr.ist.utl.pt

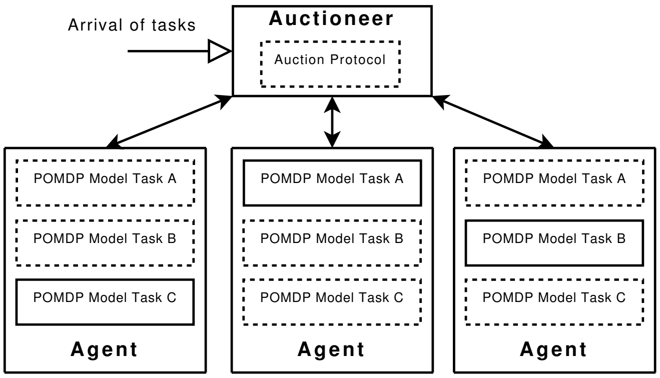


Fig. 1. Diagram of proposed solution. Each agent has a POMDP model for each task in parallel, but only one is active (indicated by a solid box).

finite, and possibly distinct, set of actions and can perceive features of interest in the environment. This problem is then formulated as computing a controller for the execution of a task by the robot. The tasks are assumed to be executed by a single agent, without explicit coordination with the others. In this way, we avoid the severe complexity penalty involved when considering the full joint planning problem (either centralized or decentralized). The coordination is achieved on a task level, by finding the optimal assignment of individual tasks to agents.

A POMDP problem is formulated and solved for each of the tasks each robot can execute. The POMDPs at each agent receive the same set of local observations, but agents do not share beliefs or other types of information. When a robot is assigned a task, the policy of the corresponding POMDP is enabled and the others disabled. That is, the actions executed by the robot are those determined only by the policy of the POMDP associated with the assigned task.

Figure II illustrates the proposed solution. The diagram is composed of a central supervisor, denoted the *auctioneer*, and a set of robot agents. The tasks to be executed by the agent are received by the auctioneer and are then assigned through an auction protocol. Although the assignment of tasks is conducted in a centralized manner, the solution can be extended to include multiple auctioneers, each responsible for a small group of agents [7]. Furthermore, the notion of a centralized task assignment is not crucial to the ideas developed in this paper. Decentralized auction techniques (e.g., [3]) that shift the burden of a centralized auctioneer to a consensus problem could be applied as well.

A related approach is the Hoplites framework [11]. However, it focuses more on tightly-coupled coordination tasks, while we target more loosely coupled scenarios. Another difference is that in the original work, Hoplites is applied to a path-planning problem. We consider more general types of tasks and also we do not plan for the joint action space (as can happen in Hoplites), to avoid an exponentially sized problem description. Related in spirit to our work, in [5] the authors propose to use combinatorial auctions for resource allocation, modeling each self-interested agent using MDPs.

III. POMDP Background

We will discuss POMDP models and solution methods, briefly introducing some general background but focusing on their application to the execution of tasks by an agent. A more elaborate POMDP model description is provided by [10], for instance.

A POMDP models the interaction of an agent with a stochastic and partially observable environment, and it provides a rich mathematical framework for acting optimally in such environments. The framework is based on the assumptions that at any time step the environment is in a state $s \in S$ and the action $a \in A$ is taken by the agent. As a result of this action, a reward $r(s, a)$ signal is received by the agent from the environment. The environment state is changed to the new state s', in accordance to a known stochastic transition model $p(s'|s, a)$. The task of an agent is defined by the reward it is given at each time step. The goal is to maximize the long-term reward signals received. After the environment transition to the new state, an observation $o \in O$ is perceived by the agent. This is conditional on the current environment state, and possibly the action executed, according to a known stochastic observation model $p(o|s', a)$.

Given the transition and observation models, the POMDP can be transformed to a belief-state MDP, where all the past information of the agent is summarized using a belief vector $b(s)$. It represents a probability distribution over S, from which a Markovian signal can be derived for the planning of actions. The initial state of the system is drawn from the initial belief b_0, which is typically included in the POMDP problem formulation. Every time the action a is taken by the agent and the observation o is obtained, the agent belief is updated by Bayes' rule; for the discrete case:

$$b_a^o(s') = \frac{p(o|s', a)}{p(o|a, b)} \sum_{s \in S} p(s'|s, a) b(s), \tag{1}$$

where $p(o|a, b) = \sum_{s' \in S} p(o|s', a) \sum_{s \in S} p(s'|s, a) b(s)$ is a normalizing constant.

In POMDP literature, a plan is called a policy $\pi(b)$ and maps beliefs to actions. A policy π can be characterized by a value function V^π which is defined as the expected future discounted reward $V^\pi(b)$ the agent can gather by following π starting from belief b:

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

where $r(b_t, \pi(b_t)) = \sum_{s \in S} r(s, \pi(b_t)) b_t(s)$ following the POMDP model as defined before, h is the planning horizon, and γ is a discount rate, $0 \leq \gamma < 1$.

The process of solving POMDPs optimally is hard, and thus algorithms that compute approximate solutions are used. Recent years have seen much progress in approximate POMDP solving which can be used in this paper, see for instance [19], [13]. Furthermore, if a value function has been computed off-line, the on-line execution of the policy it implements is computationally cheap.

IV. AUCTION PROTOCOL

The purpose of the auction protocol is to determine the POMDP policy that each agent must execute. This is equivalent, in the context of this paper, to the assignment of tasks to agents. The task generation process is assumed to be exogenous to the multirobot system. The execution of some tasks can be triggered by specific events, while others can be scheduled to be executed periodically, such as battery recharge operations. The tasks could also be executed upon request by another agent or the auctioneer. As an example, the auctioneer may directly receive event messages and locally favor the assignment of some tasks over others. The priority of each task is obtained from the specific application.

The tasks arrive at the auctioneer at any time instant, but are assigned in a bulk manner at regular intervals. It is also possible to start an auction round on demand, if for instance a high-priority task is received. The auctioneer, in order to solve the task assignment problem, is only required to know the expected discounted reward values of the POMDP task models from each agent. The auction protocol is then designed to obtain this information.

Definition 1 (Auction Protocol): The auction protocol is as follows:

1) All of the tasks are announced to the agents by the auctioneer.
2) The agents reply with their current expected discount reward $V^{\pi}(b)$ for each task. Hence, this is obtained from the solution V^{π} for the task's POMDP model, and the agent's current belief b.
3) The assignment solution is computed by the auctioneer and announced to the agents.

The main advantage of this protocol is that the auctioneer is not required to know the number of available agents or their beliefs. The approach is also robust to the failure of agents or temporary network shortages because if an agent does not offer bids, the others are still assigned tasks. Finally, the coordination of the agents for the execution of tasks is implicitly obtained through the auction protocol.

The computation of the assignment solution is performed efficiently in polynomial time using the Hungarian algorithm [2]. The bandwidth requirements are also low since only the current expected discounted reward must be reported to the auctioneer. It was shown in [7] that the protocol has low polynomial computational and communication complexities in the number of agents and tasks.

As a result, this approach can be applied to small and medium sized problems with tens or hundreds of agents and tasks. In contrast, the auction protocols described in [4] often exhibit exponential complexity. The reason is that in these protocols, agents bid on bundles of tasks instead of the single task case of our protocol.

Although the computational complexity is low, the solution is also sub-optimal because the arrival order of tasks is not known. As was shown in [12], if the arrival order is known for small bundles of tasks, the assignment solution quality is improved without significant increases on the computational and communication costs. Nevertheless the problem of computing the bid values is not considered in [12], the arrival order of the tasks is known in advance and the agents' state is known accurately. This is not the case in this paper, where the arrival order of tasks is not known and the agents only know their current state with some uncertainty.

V. POMDPS FOR BID ESTIMATION

In this work we assume that the agents do not share any information among them. The main reason is to reduce the network bandwidth and the computational requirements, since the POMDP instances are smaller. It is known that relying on perfect communication can reduce the decentralized planning problem to a centralized one [17], but the size of the centralized problem still grows exponentially in the number of agents.

Another reason is that in general the agents are not required to coordinate in order to execute tasks. Consequently, their POMDP models in general do not need to account for the beliefs and actions of other agents although it could improve overall team performance. For instance, if the planned paths of two mobile robots intersect, the collision could be avoided by sharing their beliefs. For this reason, it is assumed that robots have built-in low-level safety controllers.

Since multiple independent decision makers are present in the environment, the problem could be modeled as a decentralized POMDP (Dec-POMDP) [15]. However, given their very high complexity class, current algorithms do not scale to the types of applications we are focusing on. In our case, the coordination of the agents is obtained implicitly through the auction protocol and the auctioneer; coordination is considered on the level of task assignments vs. the level of individual agent actions, as is common in Dec-POMDPs.

The reward model is equal for all tasks, where the robot receives a single reward of 10 when it reaches the goal state. Afterwards, it is transferred to an absorbing state, in which it receives a zero reward. It leaves the absorbing state only when a new task is assigned. The value functions of all robots are normalized to $[0, 10]$ in order to allow the fitness of different robots to be compared. The absorbing state is required because otherwise the POMDP values would keep on rising after the robot would reach the goal state. This is undesirable for our approach, since we compare values between different POMDPs of the robots.

Although the value functions of the POMDPs are normalized, it is possible to define priorities for the tasks by multiplying each of the bid values by the respective task priority. Since the bid values are normalized, the result is that each bid is weighted by the respective task priority.

VI. ACTIVE SURVEILLANCE SYSTEM

The presented approach is applied, in simulation, to an active surveillance system. It is composed by a set of mobile robots, an auctioneer and a network of cameras. These are capable of detecting, with some uncertainty, the location in the environment of robots and humans. Upon the detection of

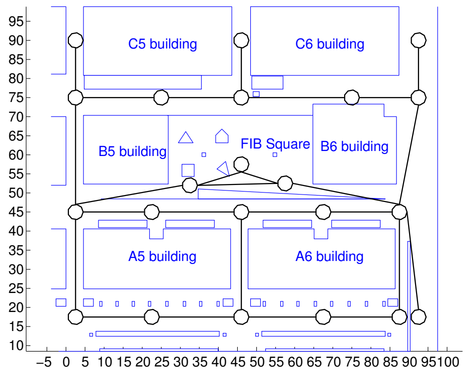


Fig. 2. Topological map of the active surveillance environment.

TABLE I
STATE VARIABLES USED BY DIFFERENT TASKS.

Task	State variables
Patrol SouthWest	Robot position
Patrol NorthWest	Robot position
Patrol NorthEast	Robot position
Patrol SouthEast	Robot position
Meet Person	Robot position, person position
Identify Person	Robot position, person position
Recharge	Robot position, battery level

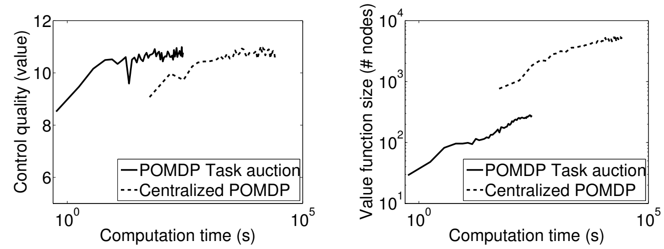
a human by the cameras, the auctioneer is notified. Note that here we present a simplified scenario, which can be extended easily to include more events (with different priorities), for instance the detection of fires.

The robots have available on-board cameras, which can recognize humans, also with some uncertainty. Each robot can obtain its localization in the environment directly from the camera network. The on-board power supply of the robots is limited and must be recharged after some time has elapsed. The tasks the mobile robots can execute are thus: (i) identifying humans, (ii) meeting a person, (iii) patrolling the environment and (iv) recharging their on-board batteries. The first two tasks are assigned only when a person was detected. In these tasks the robot must approach the desired location and use the on-board sensors either to identify a human or meet it and engage in human-robot interaction. The last two task types are assigned at regular intervals and have a low priority with respect to the first two. In this manner, if no events occur mobile robots can conduct patrols or recharge their batteries. The tasks have different priorities, for instance identifying humans is more important than the execution of a patrol.

A set of four robots were simulated (as a unicycle), three modeled after a Pioneer 3-AT robot (indicated by Pioneer A, B and C), and one after an ATRVJr robot ("AtrvJr"). The difference between the Pioneers and the AtrvJr is their maximum speed, which is respectively $0.4\frac{m}{s}$ and $1.0\frac{m}{s}$. In addition, Pioneer A has a camera with a higher resolution than those of other robots. Consequently, this robot can observe a given location from a greater distance than the others.

A topological map of the active surveillance environment is represented in Figure 2. It was obtained from the test site of the URUS project [18], at the UPC campus in Barcelona, Spain. The overall dimensions of the map are 100 by 100 meters and it was partitioned in smaller regions with their centers represented in the map.

Each of the tasks mentioned in the previous section have been modeled and approximately solved a POMDP, using Symbolic Perseus [16]. The POMDP models are represented using two-stage dynamic Bayesian networks, and the software allows for exploiting (context-specific) independence between state variables. Table I lists the different state variables for each task. We assume the surveillance cameras can localize each robot, but with a particular uncertainty. Also each robot's movement actions are subject to noise. The movement actions of the robots are subject to noise and each movement is penalized with a negative reward of -0.1. The discount rate γ is set to 0.95.



(a) Control quality.

(b) Value function size.

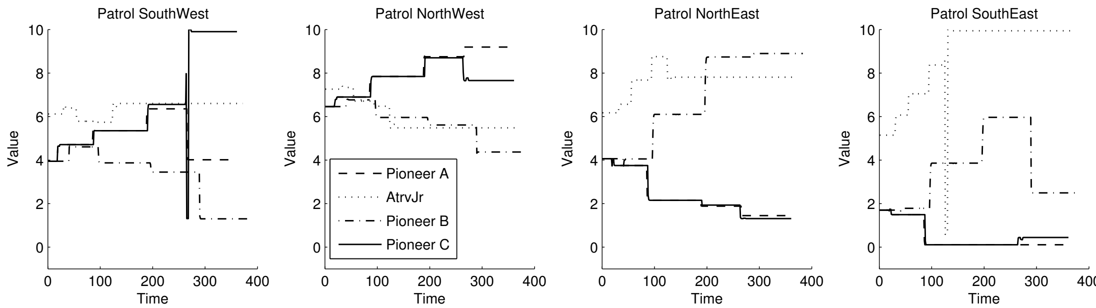
Fig. 3. Comparing POMDP task auctions to a centralized POMDP solution.

A. POMDP Auction vs. Centralized POMDP

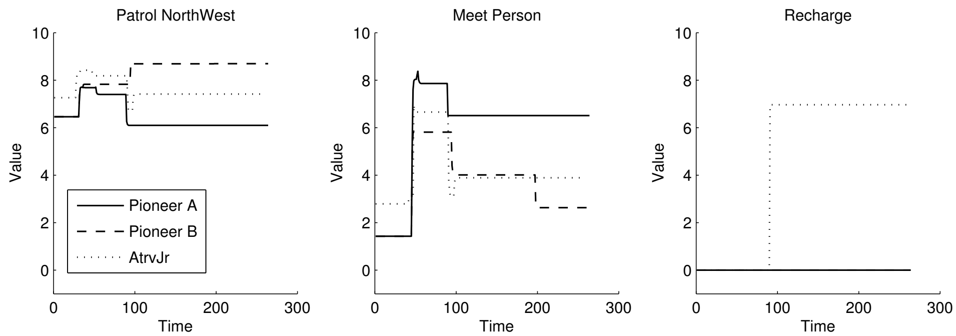
To show the advantage of auctioning individual POMDP tasks over executing a joint POMDP policy in a distributed way, we compared the performance in a scenario with two robots and two patrol tasks. A model was created for the joint task with state variables and observation models for each robot. The reward is the sum of the reward models for the individual tasks. The actions are now all possible combinations of the individual robot actions.

The centralized model was solved using the same parameters of Symbolic Perseus, for 50 iterations. The performance is compared in Figure 3 with the summed performance of the two individual tasks, denoted by "POMDP Task auction". The control quality of each value function is determined empirically by simulating the respective policy a $1,000$ times. Figure 3(a) plots the mean of the control quality for both solutions, as a function of the computation time. Both solutions reach the same control quality, but the centralized solution takes much longer to compute. In Figure 3(b) the complexity of the value functions is plotted, measured by the number of nodes in Symbolic Perseus algebraic decision diagram representation. Since the centralized model is larger, the complexity is an order of magnitude higher than the two individual POMDP tasks combined.

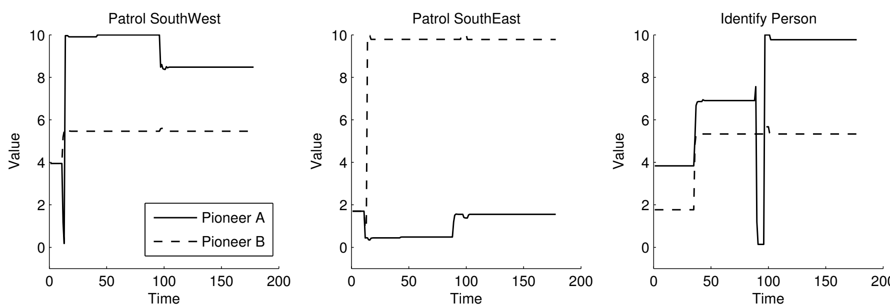
In addition to the much higher computational cost, the centralized model requires the robots to synchronize their



(a) Four Patrol tasks.



(b) A Patrol, a Meet person and a Recharge task. Two other Patrols were assigned but not shown.



(c) An Identify person task and a Patrol task.

Fig. 4. For several experiments with different sets of tasks to be assigned, we plot the POMDP value of each robot's belief over time, using the its value function for each task.

view of the state at each time step by sharing their local observations. This tightly coupled implementation requires a network with a low latency and high quality of service. On the other hand, the auction protocol has a much lower degree of coupling and does not require a high-quality communication network. Of course, for tasks that require tight coordination, e.g., two robots carrying an object jointly, a centralized solution can be hard to avoid.

B. POMDP Auction Simulation Results

In a first experiment, all of the robots were initially positioned in the center node, located at $(46, 45)$. The robots are requested four patrol tasks, one to each corner of the map. The value functions of each robot over time are plotted in Figure 4(a). They are updated as the state beliefs of the robots change while moving through the environment. An hysteresis mechanism prevented the assignment solution from changing too often. Since it is the fastest robot, the AtrvJr robot has initially the highest value for any task. It is initially assigned the "Patrol SouthEast" task, while Pioneer A gets "Patrol NorthWest", Pioneer B "Patrol NorthEast", and Pioneer C "Patrol SouthWest". The task "Patrol NorthWest" is not initially assigned to the AtrvJr because the assignment is determined by maximizing the sum of all bid values and not the individual bids.

In the second experiment, three of the robots started in the central node and the other at $(46, 75)$. Initially, the three patrol tasks are requested but a recharge task is also requested when a robots has a low battery level. Upon the detection of a person, a "Meet Person" task is requested. The obtained value functions of the robots plotted in Figure 4(b). Since the robots start with a full battery, all patrol tasks are assigned. At about 50 time units, a person was detected at $(46, 90)$ and the "Meet Person" task was requested. Since the other tasks have a lower priority, Pioneer A abandoned its patrol task and was assigned to meet the person. At about 100 time units, the AtrvJr robot while moving to the patrol task goal passed in the node containing the battery recharge station. It was then assigned the recharge task because the battery level was low and destination of the patrol task was still far.

In the last experiment, robots Pioneer A and B were initially placed at $(2.5, 17.5)$ and $(87.5, 17.5)$ respectively. Their value functions are plotted in Figure 4(c). The robots were initially requested two patrols tasks, one for each of their current locations. As a result, the robots did not move. At about 40 time units, a person was detected at node $(46, 17.5)$ and an "Identify Person' task was requested. For this task, unlike the meet person, the robot must only approach the person close enough to take a clear picture. Although A is further away from the person, it has a camera with a higher resolution. For this reason it is assigned the identify person task instead of Pioneer B.

From these experiments it is visible that the auction protocol enabled the robots to coordinate their task execution without communication of their state or beliefs. The system was also able to autonomously respond to detected events that occurred after the initial task assignment.

VII. CONCLUSIONS

We presented an approach to the assignment and execution of tasks in a multirobot system. The motivation was to illustrate the benefits for multirobot systems of mixing auction protocols with controllers based on value functions. Auction protocols enable the coordination of multiple agents in low quality networks and provide robustness to individual agent failures. In this paper, we proposed a more principled way of estimating the bid values of each agent, in lieu of the heuristic and often ad-hoc approaches in common use.

The controllers for the execution of tasks were defined using the POMDP framework. If suitable stochastic models of the environment and the agent observations are available, the synthesis problem can be formulated in a straightforward mathematical manner. The combination of the two frameworks produced a solution where the individual drawbacks are minimized. From the synthesis of controllers using POMDP task models, the values to bid are naturally obtained from the respective expected discounted rewards, and the agent's belief is already factored into this value. As a result, it is not necessary to invest additional time in the design of bid functions for each of the agents' tasks. Furthermore, as they are derived directly from the task controller, they are likely to reflect better true bid values, compared to commonly used heuristic bid functions.

The use of an auction enabled the use of smaller POMDP models than otherwise would be used if all agents and all tasks are considered simultaneously. This is because the agents coordination is implied in the use of the auction protocol and the auctioneer. Therefore, in the controller synthesis problem the other agents and tasks can be abstracted away. This is at the cost of optimality, since in practice the agents can interfere in each others' task execution.

A direction of future research is the synthesis of a controller for the auctioneer to determine the task priorities. The purpose is to maximize some performance criteria, such as the minimum assignment delay for some task types. The controller can also be used to determine which tasks to trade with other auctioneers. Finally, we plan to extend our simulations to include more events, and we intend to apply our techniques in a real-world setup [1].

REFERENCES

[1] M. Barbosa, A. Bernardino, D. Figueira, J. Gaspar, N. Gonçalves, P. U. Lima, P. Moreno, A. Pahliani, J. Santos-Victor, M. T. J. Spaan, and J. Sequeira. ISRobotNet: A testbed for sensor and robot network systems. In *Proc. of Int. Conf. on Intelligent Robots and Systems*, 2009.

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

[3] H.-L. Choi, L. Brunet, and J. How. Consensus-based decentralized auctions for robust task allocation. *IEEE Transactions on Robotics*, 25(4):912–926, Aug. 2009.

[4] M. B. Dias, R. Zlot, N. Kalra, and A. Stentz. Market-based multirobot coordination: a survey and analysis. *Proceedings of the IEEE*, 94(7):1257–1270, 2006.

[5] D. Dolgov and E. Durfee. Computationally-efficient combinatorial auctions for resource allocation in weakly-coupled MDPs. In *Proc. of Int. Conference on Autonomous Agents and Multi Agent Systems*, pages 657–664, 2005.

[6] B. P. Gerkey and M. J. Mataric. Sold!: auction methods for multi-robot coordination. *IEEE Transactions on Robotics and Automation*, 18(5):758–768, 2002.

[7] N. Gonçalves and J. Sequeira. Multirobot task assignment in active surveillance. In *Progress in Artificial Intelligence, Proceedings of the 14th EPIA*, volume 5816 of *LNCS*. Springer, 2009.

[8] H. Hanna and A.-I. Mouaddib. Task selection problem under uncertainty as decision-making. In *Proc. of Int. Conference on Autonomous Agents and Multi Agent Systems*, 2002.

[9] J. Jungwattanakit, M. Reodecha, P. Chaovalitwongse, and F. Werner. A comparison of scheduling algorithms for flexible flow shop problems with unrelated parallel machines, setup times, and dual criteria. *Computers and Operations Research*, 36(2):358–378, 2009.

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

[11] N. Kalra, D. Ferguson, and A. Stentz. Hoplites: A market-based framework for planned tight coordination in multirobot teams. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, April 2005.

[12] S. Koenig and C. Tovey. Sequential bundle-bid singlesale auction algorithms for decentralized control. In *Proc. Int. Joint Conf. on Artificial Intelligence*, pages 1359–1365, 2007.

[13] H. Kurniawati, D. Hsu, and W. Lee. SARSOP: Efficient point-based POMDP planning by approximating optimally reachable belief spaces. In *Robotics: Science and Systems*, 2008.

[14] A. R. Mosteo and L. Montano. Comparative experiments on optimization criteria and algorithms for auction based multi-robot task allocation. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 3345–3350, 2007.

[15] F. A. Oliehoek, M. T. J. Spaan, and N. Vlassis. Optimal and approximate Q-value functions for decentralized POMDPs. *Journal of Artificial Intelligence Research*, 32:289–353, 2008.

[16] P. Poupart. *Exploiting Structure to Efficiently Solve Large Scale Partially Observable Markov Decision Processes*. PhD thesis, University of Toronto, 2005.

[17] 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.

[18] A. Sanfeliu and J. Andrade-Cetto. Ubiquitous networking robotics in urban settings. In *Proceedings of the IEEE/RSJ IROS Workshop on Network Robot Systems*, 2006.

[19] M. T. J. Spaan and N. Vlassis. Perseus: Randomized point-based value iteration for POMDPs. *Journal of Artificial Intelligence Research*, 24:195–220, 2005.

[20] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, 2005.

[21] N. Vlassis, G. Gordon, and J. Pineau. Planning under uncertainty in robotics. *Robotics and Autonomous Systems*, 54(11), 2006. Special issue.