Reasoning and Learning in Real-World Systems for Long-Term Autonomy (LTA)
Papers from the AAAI 2018 Fall Symposium

A Practical Distributed Knowledge-Based Reasoning and Decision-Theoretic Planning for Multi-robot Service Systems

A.-I Mouaddib and laurent Jeanpierre
GREYC–University of Caen Normandy
abdel-illah.mouaddib, laurent.jeanpierre@unicaen.fr

Abstract

This paper presents a practical model of distributed reasoning and planning for a fleet of robots serving people in a shopping mall using distributed knowledge-based reasoning and distributed Markov Decision Process (MDP) where the environment changes frequently and the set of goals is not static. The model we present, in this paper, consists of distributed local reasoning and planning where each robot locally reasons on its perceived data (locally: onboard cameras and also from global perception system: external cameras) to update its local Knowledge Base (KB). Local KBs derive local goals and the local planners select the goal to accomplish and compute the policy to accomplish it while maintaining a coordinated behavior with the other robots by avoiding conflicts on goals. To this end, we propose a distributed market-based auction planning algorithm using a regret and opportunity costs in a distributed value function leading to augmented MDPs to coordinate the robots and to select the appropriate goals to accomplish. Our approach assumes communication between robots and external sensor and we will describe a method to minimize the dis-coordination (conflits on goals) when the communication is lacking. Experimental results on the algorithm performance and the implementation on real service robots in a shopping mall showed a very satisfying behavior as shown in the video.

1 Introduction

The claim of this paper is to present a practical model and algorithm for a multi-robot system to be deployed in a public area like a shopping mall, airport, train station, hospital, ... to assist and guide customer. In our case, we consider assistance, advertisement and security goals in a shopping mall. The characteristics of such applications are :

- The environment is highly dynamic and a long-term horizon planning could be unsuitable.
- No central planner or system could be considered.
- The set of goals is dynamic and can change during the execution.
- The nature of the application is highly distributed in terms of perception, reasoning and planning since the robots sweep as large as possible the public space to detect events to handle.
- Goals accomplishments are durative and planning for the full set of goals could be inappropriate since new high prior goals could be generated.

Reasoning and planning in such contexts make some classical approaches inappropriate. Indeed, local KB reasoners of different robots can derive different goals distributed among their Knowledge-Base and central planner should compute a plan among all distributed goals to derive an optimal goal allocation. While most central planners require a central Knowledge base [Ghallab, Nau, and Traverso, 2016], planning with distributed knowledge bases becomes out of reach of these existing algorithms [Ghallab, Nau, and Traverso, 2016]. Formalizing the planning problem where robots have their own local observations leads to some strong mathematical tools like DEC-POMDP [Bernstein, Zilberstein, and Immerman, 2000; Amato, Bernstein, and Zilberstein, 2007; Seuken and Zilberstein, 2007; Dibangoye et al., 2016]. However, DEC-POMDPs require a central planner which make their use in this context inappropriate. Considering only interactions between agents to formalize the planning problem has been considered in different POMDPs-based approaches like Networked POMDPs [Nair et al., 2003; 2005], interactive POMPDs [Sonu and Doshi, 2015] or Augmented MDPs [Matignon, Jeanpierre, and Mouaddib, 2012]. Such approaches are promising to deal with distributed planning which is more appropriate to the problems where knowledge-based are distributed. Hiowever, they can show some limits where the environment is highly dynamic and the set of goals changes frequently. To this end, we extend the approach presented in [Iocchi et al., 2016] to the multi-robot system settings where the architecture is fully decentralized and combine auctionning and MDPs to coordinate the robot policies. The combination between auctionning and (PO)MDPs is not novel and has been introduced in [Spaan, Gonçalves, and Sequeira, 2010]. However the auctionning phase is centralized while our approach is fully decentralized and the auctionning is distributed among robots.

We present an approach which allows the robots to plan in a distributed way and with a changing set of goals using a distributed market-based auction algorithm combined with a distributed value function [Matignon, Jeanpierre, and Mouaddib, 2012] allowing each robot to locally plan and select the goals to accomplish. This algorithm has been implemented and tested on real robots serving people and guiding them to their requested destination showing a very satisfying behavior as shown in the video[1].

The rest of the paper is organized as follow: section 2 describes the overall architecture and the interaction between local reasoners and planners. Section 3 presents the distributed knowledge-base reasoning and the communication with the planners. Section 4 representing the main contribution of the paper where we describe the market-based auction algorithm and the distributed value functions using opportunity cost and the regret functions. Section 5 is dedicated to the evaluation of our approach in terms of solution quality and the implementation on real robots.

2 Distributed multi-robot Architecture

The distributed architecture of the multi-robot system consists of, as depicted in Figure 1, communicative local knowlege-based reasoners, planners and executors. Each robot has each local knowledge-based module which is splitted into two parts : static part describing the semantic map of the environment (shops, restaurants, social areas in the mall for instance) and the dynamic part describing the new incoming information from sensors (onboard cameras of the robots and external cameras in the environment) and from the other modules such as the status of the robots (idle or active), their goals under accomplishment and their priorities maintaining a global partial view on the overall system. The KB is a set of logical predicates similar to the classical STRIPS-like planning language [Ghallab, Nau, and Traverso, 2016]. The KB uses a simple rule-based reasoning to derive new goals to accomplish. These new goals are communicated to the planning module which uses a maket-based auctionning algorithm which is descibed bellow. This algorithm allows the robot to select the goal to accomplish and to formalize it as an MDP as presented in [Iocchi et al., 2016] and then communicate the policy to the executor module to act. In order to compute the goal to accomplish, each robot communicates with the others the set of locally derived goals and the vector of associated optimal policy values. Each robot fuses the set of received goals with the locally derived ones. This shared information allows to each robot to use global information to plan.

3 Distributed Knowledge-based reasoning

3.1 Local KB reasoning

Each robot maintains a local KB which is split into a static KB representing the semantic map and a dynamic KB representing the events occurring in the environments or on the status of the robot. Indeed, when external or on-board sensor detect an event (a person, an object for example), the KB inserts a logical formula representing this event and executes its logical inference engine to derive new goals. These new goals are added to the existing ones and sorted according to their priority. In the shopping mall case, the security goals have higher priority than assistance goals which have high priority than advertising goals. This order allows the robots to start by allocating the goals according to their priority. The other knowledge in the dynamic part concern the information on the other robots particularly their status describing whether the robot is active and which goal is achieving or idle and its initial location when starting the execution of the goal achievement policy.

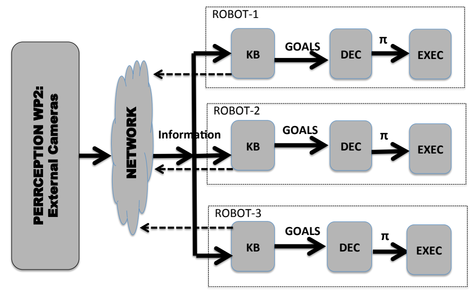

Figure 1: General multi-robot decision-making Architecture

3.2 KB update and Synchronization

The local KB for each robot should be updated by external information coming from the environments (perception) and the other robots. In Figure 2, we represent information coming from the other robots by G_i^* meaning the goal under accomplishment by the other robot and the information coming from perception allow the KB module to generate G^t a goal list at time t. These information allow the KB module to generate the new list of goals: $G^t = G^t - G_i^*$. This list is then sent to the decision module which uses a distributed matrix-based auctioning algorithm, described in the next section, to select a goal to accomplish and computes the policy to accomplish it. The decision module updates the KB with the selected goal and the exec module about the policy to execute. The exec module during the execution sends the status of the execution and the current level of priority which is necessary for considering new goals or not. The exec module updates the KB at the end of the execution and this update is sent to the other robots for their local KB update. This processing is depicted in Figure 2. At the end of the goal accomplishment, the robot switches to the idle status to consider a new auctioning step. This information is sent to the other robots which is useful when performing the distributed Matrix-based auctioning algorithm.

Communication allows robot to exchange the information concerning the status of execution and also the level of interruptibility allowing at the receipt of a list of goals to consider only robot that could accomplish the goals according to their current status and to synchronize their local KB, and to construct their local matrix.

[1] https://youtu.be/iFC6-sCL3XI

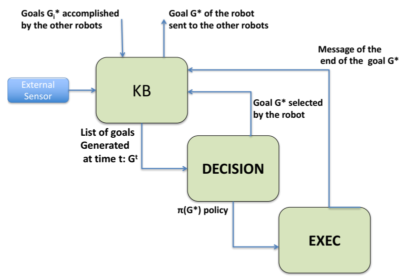

Figure 2: Communication between KB, decision and execution

It's possible that some messages contain values of some goals that the robot hasn't in its list. For these goals, the robot initializes the value of these goals in its value vector to $-V_{max}$.

3.3 Interruptibility and changing set of goals

In this section, we address the problem of changing goal sets. When the list of goals are communicated to the decision module, the robot can have different status. Indeed, the robot can be in a **idle status** or in a **active status**. When the robot is in a **idle status**, she is able to consider new goals and decides for the goals to accomplish. However, when the robot is in **active status**, the robot should consider the new goals only in the situations where the priority of one new goal is higher than the one under accomplishment.

All robots with priority lower than the priority of new goals, should consider them for a new auctioning. When the matrix-based auctioning algorithm of a robot has been performed and no goal is allocated, the robot pursues with the current policy, otherwise, the robot executes the policy of the new goal.

The priority of the robot is the one of the goal under processing. However, this priority can change during the execution since when the execution task is at its beginning stages, it's easier to cancel the execution and skip to another task rather than at its final execution stage. In our case, tasks are represented in an hierarchical structure called PRU+ as in [Iocchi et al., 2016]. Indeed, one benefit of the PRU+ structure is that goals could be accomplished partially and thus could be interrupted when the rest of the goal is not highly prior and in such case, we can enhance the PRU+ definition with the priority of levels.

4 Distributed Matrix-Based Auction Planning

4.1 General principle

The allocation of goals to the robots is performed by a distributed decision-theoretic market auction algorithm, extending the approach presented in [Spaan, Gonçalves, and Sequeira, 2010] to distributed auctioning and also using a distributed value function based on regret and opportunity cost to solve different potential conflicts on goals and robots. The proposed solution is illustrated in Figure 3 where each robot has a local auctioneer which receives the list of goals G from the local KB and sends this list to the decision-making module. This latter uses a library of task MDP models to solve for each goal in the list of goal its corresponding MDP by considering the current state of the robot. Thus the decision module of robot i solves $\{\ MDP_i(g, s_i^t) | \forall g \in G\ \}$. This allows the decision maker to compute the optimal value for each goal in the G $(v_i^{g_1}, v_i^{g_2}, \ldots, v_i^{g_k})$ for all $g_i \in G$. This vector of values is sent the local auctioneer that exchanges with the other local auctioneers. Once the local auctioneers collect all values and organize them in matrix 4. This matrix allows us to solve Equation (1). We consider in our approach the cases where this equation can have more than one solution meaning that more than one goal could be preferred by a robot and many robots can have one preferred goal leading to some conflicts situations which are not addressed in [Spaan, Gonçalves, and Sequeira, 2010].

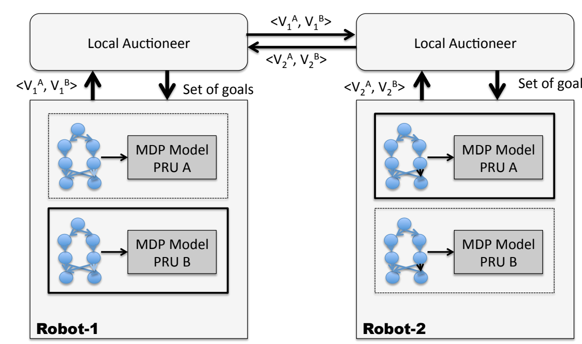

Figure 3: Each robot has an MDP for each goal represented by an acyclic PRU graph. The MDP of selected goal is active (solid box)

$$(\alpha, g^*) = argmax_{A_i, g_k} V_{A_i}(g_k) \quad (1)$$

To address these issues, we introduce regret and opportunity cost functions. The regret function measures the loss in value for a robot to accomplish a goal rather than it's preferred and the opportunity cost of accomplishing a goal by a robot measures the loss in value of the other robots because of preventing them from this goal. More formally speaking, the *Regret* of not accomplishing a goal g^* is given by the following equation :

$$regret_j(g) = V_j^{\pi^*}(g) - \max_{g' \neq g} V_j^{\pi^*}(g')$$

And the opportunity cost is defined by :

$$OC_R(g) = \max_{R' \neq R} \max_{g' \neq g} V_{R'}^{*;g'} - V_{R'}^{*;g}$$

Let S_g be the set of robots α optimizing the value of accomplishing the goal g (solutions of Equation 1), the best robot to which we allocate the goal g is the one minimizing the regret (Equation 2). If we havemany solutions, we can proceed in the same way with the other goals and so on.

Let G_r be the set of the preferred goals of robot r. The best goal selected by robot r is the one minimizing the maximum opportunity cost (Equation 3).

4.2 Market-based auction algorithm for goal allocation using distributed value functions

The distributed market-based auction algorithm, illustrated in Figure 3, consists of two steps : Matrix construction step and Market-based auction step that we described in the following:

- **Matrix construction:** each robot i maintains a matrix M_i representing the value of the optimal policy of each robot to accomplish a goal. The matrix is constructed as follows:
  1. Each robot i computes the optimal value V_i^{*,g_l} to accomplish goal g_l. Value vector $(V_i^{*,g_1}, V_i^{*,g_2}, \ldots, V_i^{*,g_k})$ represents the values of robot i optimal policies accomplishing goals in the list. This vector represents the line i of the matrix.
  2. Each robot constructs locally this vector, communicates it to its local auctioneer and this latter sends its value vector to the others, allowing them to complete their matrix
  3. Each robot i (local auctioneer) has thus a matrix 4.
- **Distributed Matrix-based auctioning:** Each robot proceeds as depicted in 4 to the following steps:
  1. Fo each each line L of the matrix, compute $\max_j V_l^{*,g_j}$ corresponding to the best goal to accomplish for the robot L (its bid).
  2. If there is only one goal g_L^* maximizing the value of the robot L, this means that there is only a unique preferred goal for this robot. However, we should check that this robot is the preferred one for this goal. To this end, we check in the column g_L^* that there is no value $V_i^{*,g_L^*} \geq V_L^{*,g_L^*}$. In such case, the goal g_L^* is assigned to the robot L and ligne L and column g_L^* are removed from the matrix and we repeat the same process for the matrix until all goals have been assigned or all robots have an assigned goal.
  3. If there is more than one maximum value existing in columns or lines, we proceed as follows :
     (a) **Processing columns:** this means that there is a conflict between robots R on the same goals g. We assign the goal to the robot having the minimum regret ;

$$\text{argmin}_{robot \in R} \max_{g' \neq g} V_{robot}^{*,g'} - V_{robot}^{*,g} \quad (2)$$

     Then we remove the adequate column and line.

     (b) **Processing lines:** this means that there are more than one goal G preferred by a robot R. In tis situation, we assign the goal with the minimum opportunity cost ;

$$\text{argmin}_{g \in G} OC_R(g) \quad (3)$$

     Then we remove the adequate column and line.

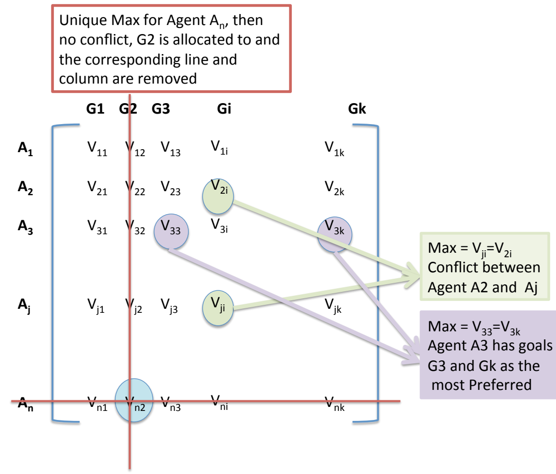

Figure 4: Matrix allocation processing

Proposition 1 *Matrix-based auctioning algorithm is based on solving $|G|$ MDPs with the same state space but with different goal states. The complexity of Matrix-based auctioning algorithm is $O(|G| \cdot |S|^2 \cdot |A|)$*

Proof Matrix-based auctioning algorithm solves one MDP for each goal of the set of goals $|G|$ with the same state space but with different goal states. The complexity of solving and MDP with value iteration algorithme is $O(|S|^2 \cdot |A|)$.

Proposition 2 *The cost of communication in Distributed Matrix-based Auctioning algorithm is in O(n^2) while in auctioning POMDP is in $O(3 \cdot n)$.*

Proof In Auctioning POMDP, each robot communicates with the central auctioning module by receiving the set of goals and sending its bid. The auctioning module sends n message to announce the goals, n bid messages are received from robots and sends n messages to robots for the auctioning result. In Distributed Matrix-based Auctioning algorithm, robots exchanges the goals and their bids as values of their optimal policies to accomplish these goals (individually). These message exchanges consist of $n - 1$ messages sent by each robot (n robots). Thus the overall number of exchanged messages is $n \cdot (n - 1)$. Thus the complexity is O(n^2).

4.3 The overall algorithm

1. The status of the Robot i is *Idle* :
   (a) Proceed to the message processing and produce the list of goals and sends local messages to the other robots;
   (b) Compute for each goal g in the list the value of the optimal policy V_g^* and put $Matrix(i, g) = V_g^*$;
   (c) Send the vector value to the decision module;
   (d) The decision module selects the best goal $g^* =$ Market_Based auction(i,g) (section 4.2) and sends the message to the KB;
   (e) The robot constructs the policy $\pi(g^*)$;

(f) The robot sends to the EXEC module and it changes its status to *active*;

2. The status of the robot i is *active*:

   (a) Transforms the policy in a PNP (Petri-Net Plan) as described in [Iocchi et al., 2016];

   (b) Execute the PNP and sends at each step of the PNP, the level of interruptibility;

   (c) At the end of the execution, sends the message to the local KB and changes the status to *idle*;

5 Dis-coordination minimization with lack of communication

When communication is not available two aspects have to be considered: (1) external sensor cannot communicate with robots and only onboard sensor of robots can be used leading to the lack of global perception and (2) robots cannot communicate and thus cannot exchange values of goals for auctioning. In order to overcome these limitations, we propose a simple policy estimation algorithm allowing each robot i to estimate the policy of each other robot $\pi_{j \neq i}$ without communication. We consider that the policy π_k^τ followed by robot k is the policy computed from the $MDP_k(s_k^\tau, G^\tau)$ where s_k^τ is the initial state of robot k and G^τ is the set of goals at time τ by selecting the policy maximizing the expected value of accomplishing one goal. In order to estimate the policy $\pi_{j \neq i}^t$ followed by each robot $j \neq i$ at time t, we need to estimate the policies followed by the robots during the interval $[t', t]$. However, during this interval one robot k can accomplish more than one goal and thus has followed a sequence of policies, noted $\mathcal{S}_k^\pi$ starting with $\pi_k^{t'}$ which is the policy of the $MDP_k(s_k^{t'}, G^{t'})$ where $(s_k^{t'}, G^{t'})$ are given by the local KB. However, to compute the next policies we need to derive the new MDP by deriving the new start state and the new set of goals. To do so, we approximate the new start state of k by s_k^τ as the most likely state can be reached by the policy $\pi_k^{t'}$ and that the new set of goal $G^\tau = G^{t'} - goal_{\pi_k^{t'}}$ where $goal_{\pi_k^{t'}}$ is the goal accomplished by $\pi_k^{t'}$ and finishing at time τ. More formally :

$$s_k^\tau = \max_{s_k'} P(s_k' | \pi_k^{t'}, s_k^{t'})$$

When s_k^τ and G^τ are derived, we can compute the next policy π_k^τ of $\mathcal{S}_k^\pi$ from the $MDP_k(s_k^\tau, G^\tau)$. We repeat the same processing until $\tau \geq t$. The last policy of $\mathcal{S}_k^\pi$, noted π_k^τ allows to estimate the s_k^τ and thus we compute the policy π_k^t of the $MDP_k(s_k^\tau, G^t)$ with the current set of goals. This principle is repeated for all robots and thus we get an approximate joint policy where each robot is assumed accomplishing a goal in G^t. Robot i compute a policy π_i^t of the $MDP_i(s_i^t, G^t - \bigcup_{k \neq i} goal_{\pi_k^t})$.

Proposition 3 *The complexity of approximating the policy of the other robots during an interval of time is* $O(|S|^3 \cdot |A| \cdot |G|^2)$ *where S and A are respectively the state and action spaces and G the set of goals at the lost of communication.*

Proof Solving an MDP with a goal and with an initial state using value iterationis O($|A| \cdot |S|^2$) [Puterman, 1994]. When we extend this to a set of goals where we should select the best goal, the complexity becomes $|G| \cdot |S| \cdot |A|^2$. To consider this problem for any initial state is in $|G| \cdot |S|^3 \cdot |A|$ because we repeat the same problem for each state and extending this problem for an interval of time is at most the resolution of all goals and thus the complexity becomes $|G|^2 \cdot |S|^3 \cdot |A|$. When we solve this problem for any state space and any set of goals, then for any robot we need just to know its initial state and which set of goals. Thus the complexity remains O($|G|^2 \cdot |S|^3 \cdot |A|$).

6 Empirical evaluation

We developed experiments where we consider our application of the shopping mall with 3 robots and a dozen of goals to accomplish. We compare our algorithm with the baseline algorithm of solving a DEC-MDP with a central planner as presented in [Hanna and Mouaddib, 2002] and an auctioning MDP approach using the principle presented in [Spaan, Gonçalves, and Sequeira, 2010]. We used different situations where goals are not located at the same place, for example for goals "guiding to shop" we consider different shops.

- We compare the computation time of each method: DEC-MDP, auctioning MDP and Decentralized Market-based Auctioning algorithm.
- We compare the values of the three approaches
- We compare the performance of these approaches when the set of goals changes by adding two more goals to the current list.
- We also compare the performance of these approaches when the communication is broken.

6.1 First results on performance comparison

The first results on computation time for three robots show that the computation time of DEC-MDP is higher than the others as expected which is explained by the complexity of DEC-MDP known to be NEXT-hard while our approach (MBAA) and Auctioning MDP have comparable computation time and MBAA is little bit higher due to the cost of communication.

The expected value of AMDP and MBA are the sum of the policy values of different goals accomplished by the three robots while for the DEC-MDP we get a value of a joint policy for the three robots. We observed that AMDP and MBAA are not far from a centralized planning approach representing more than 88% which is a satisfying performance. However, our approach outperforms the AMDP because it solves better the conflicts. We note that when conflicts doesn't occur, AMDP and MBAA obtain the same value.

To get more significant results more experiments are needed with more scenarii. Actually, we are continuing our evaluation.

6.2 Robustness to the changes of the goal set

The first experiment consists in starting with a list of 6 goals and during the execution of the policies we add two new goals. For this experiment, DECMDP approach considers the goals only when she finishes the accomplishment of the current goal and then consider the two new goals. The DECMDP plans from scratch for 7 goals (5 remaining goals plus the two new goals). In addition to that, the system should wait during the execution time which depends on the goal under execution (and the location of the shop). The AMDP finishes the execution of the current goal and considers the new one as DECMDP approach. This situation occurs in MBAA only when the priorities of the goals are lower than the one under execution otherwise, the executed is interrupted and the more prior goals are considered. A situation we observe during the experiment is when AMDP, DECMDP and MPAA are executing an advertisement goal and a new assistance a person goal arrive, the person should wait more than 3mn, 2mn and 30s respectively with DECMDP, AMDP and MPAA before being considered. These durations doesn't consider the execution time needed to meet the person (moving to the person).

6.3 Robustness to communication

DECMDP with no communication with the central planner they cannot work since they never receive their policy. AMDP needs communication with the central auctioneer and thus no solution is possible. However, MBAA can work with degraded more where only local planning and estimation of the other robots situations are performed. We use an experiment with 3 robots and 12 goals with two classes of configurations: Configuration A considers that goals scattered in the mall and configuration B considers goals in a narrow space. For configuration A, the approximation works well and only one conflict is observed (one dis-coordination) at the accomplishment of the two last goals where two robots head the same destination. However, in configuration B, we observe 4 conflicts (30%) where robots select the same goals. More deeper experiments in different situations are needed and let for future work.

6.4 Implementation on real-robots

The experiment on real robots as depicted in Figure 5 have been developed in assistance mission of visitors of our Lab where people can interact with the robot to ask for the office of a professor or administrative staff and then the robot guide him to the requested destination. When the robot accomplishes a goal, she comes back to the welcoming point to serve a new visitor depending on its location and the location of the visitor. Figure 5 shows that the same robot doesn't serve at the same point but according to their locations, the robot selects the appropriate welcoming point showing a very satisfying behavior. A video of this scenario is available at `https://youtu.be/iFC6-sCL3XI`.

7 Conclusion and discussion

We presented a practical approach allowing a fleet of robots to reason on local information and plan their activities in a coordinated way while activities and information are distributed in the environment. We develop a distributed Market-based auctioning method robust to the changes of the goal set and to the communication unavailability showing a very satisfying behavior. This method combines decision-theoretic planning

Figure 5: Figure showing different steps of robots serving visitors

by computing MDP policy for goals and bidding with values of these policies to coordinate their activities. Our method enriches the auctioning POMDP technique [Spaan, Gonçalves, and Sequeira, 2010] by considering distributed local auctioneers but also to use distributed value function based on regret and opportunity costs to solve some specific costs. Hoplites [Kalra, Ferguson, and Stentz, 2005] addresses similar problem but it is limited to path planning tasks in the opposite to ours. The dis-coordination minimization technique shows satisfying results and future work will be concerned by this aspect to deepen this problem and propose more efficient approach. Our approach has been successefuly used on real robots showing a very convincing behavior. We will develop more experiments to better evaluate the performance.

Aknowledgments:

This work has been developed in the COACHES project of the CHIST-ERA program supported by the French National Agency (ANR).

References

[Amato, Bernstein, and Zilberstein, 2007] Amato, C.; Bernstein, D. S.; and Zilberstein, S. 2007. Optimizing memory-bounded controllers for decentralized pomdps. In *UAI 2007, Proceedings of the Twenty-Third Conference on Uncertainty in Artificial Intelligence, Vancouver, BC, Canada, July 19-22, 2007*, 1–8.

[Bernstein, Zilberstein, and Immerman, 2000] Bernstein, D. S.; Zilberstein, S.; and Immerman, N. 2000. The complexity of decentralized control of markov decision processes. In *UAI '00: Proceedings of the 16th Conference in Uncertainty in Artificial Intelligence, Stanford University, Stanford, California, USA, June 30 - July 3, 2000*, 32–37.

[Dibangoye et al., 2016] Dibangoye, J. S.; Amato, C.; Buffet, O.; and Charpillet, F. 2016. Optimally solving decpomdps as continuous-state mdps. *J. Artif. Intell. Res.* 55:443–497.

[Ghallab, Nau, and Traverso, 2016] Ghallab, M.; Nau, D. S.; and Traverso, P. 2016. *Automated Planning and Acting*. Cambridge University Press.

[Hanna and Mouaddib, 2002] Hanna, H., and Mouaddib, A. 2002. Task selection problem under uncertainty as decision-making. In *The First International Joint Conference on Autonomous Agents & Multiagent Systems, AAMAS 2002, July 15-19, 2002, Bologna, Italy, Proceedings*, 1303–1308.

[Iocchi et al., 2016] Iocchi, L.; Jeanpierre, L.; Lazaro, M. T.; and Mouaddib, A. 2016. A practical framework for robust decision-theoretic planning and execution for service robots. In *Proceedings of the Twenty-Sixth International Conference on Automated Planning and Scheduling, ICAPS 2016, London, UK, June 12-17, 2016.*, 486–494.

[Kalra, Ferguson, and Stentz, 2005] Kalra, N.; Ferguson, D.; and Stentz, A. 2005. Hoplites: A market-based framework for planned tight coordination in multirobot teams. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation, ICRA 2005, April 18-22, 2005, Barcelona, Spain*, 1170–1177.

[Matignon, Jeanpierre, and Mouaddib, 2012] Matignon, L.; Jeanpierre, L.; and Mouaddib, A. 2012. Coordinated multi-robot exploration under communication constraints using decentralized markov decision processes. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, July 22-26, 2012, Toronto, Ontario, Canada.*

[Nair et al., 2003] Nair, R.; Tambe, M.; Yokoo, M.; Pynadath, D. V.; and Marsella, S. 2003. Taming decentralized pomdps: Towards efficient policy computation for multiagent settings. In *IJCAI-03, Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, Acapulco, Mexico, August 9-15, 2003*, 705–711.

[Nair et al., 2005] Nair, R.; Varakantham, P.; Tambe, M.; and Yokoo, M. 2005. Networked distributed pomdps: A synergy of distributed constraint optimization and pomdps. In *IJCAI-05, Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence, Edinburgh, Scotland, UK, July 30 - August 5, 2005*, 1758–1760.

[Puterman, 1994] Puterman, M. L. 1994. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. New York, NY, USA: John Wiley & Sons, Inc., 1st edition.

[Seuken and Zilberstein, 2007] Seuken, S., and Zilberstein, S. 2007. Memory-bounded dynamic programming for dec-pomdps. In *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6-12, 2007*, 2009–2015.

[Sonu and Doshi, 2015] Sonu, E., and Doshi, P. 2015. Scalable solutions of interactive pomdps using generalized and bounded policy iteration. *Autonomous Agents and Multi-Agent Systems* 29(3):455–494.

[Spaan, Gonçalves, and Sequeira, 2010] Spaan, M. T. J.; Gonçalves, N.; and Sequeira, J. 2010. Multirobot coordination by auctioning pomdps. In *IEEE International Conference on Robotics and Automation, ICRA 2010, Anchorage, Alaska, USA, 3-7 May 2010*, 1446–1451.