2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
Congress Center Hamburg
Sept 28 - Oct 2, 2015. Hamburg, Germany

A decision-theoretic planning approach for multi-robot exploration and event search

Jennifer Renoux[1], Abdel-Illah Mouaddib[1], Simon Le Gloannec[2]

Abstract— **Event exploration is the process of exploring a topologically known environment to gather information about dynamic events in this environment. Using multi-robot systems for event exploration brings major challenges such as finding and communicating relevant information. This paper presents a solution to these challenges in the form of a distributed decision-theoretic model called MAPING (Multi-Agent Planning for INformation Gathering), in which each agent computes a communication and an exploration strategy by assessing the relevance of an observation for another agent. The agents use an extended belief state that contains not only their own beliefs but also approximations of other agents' beliefs. MAPING includes a forgetting mechanism to ensure that the event-exploration remains open-ended. To overcome the resolution complexity due to the extended belief state we use a method based on the well-known adopted assumption of variables independence. We evaluate our approach on different event exploration problems with varying complexity. The experimental results on simulation show the effectiveness of MAPING, its ability to scale up and its ability to face real-word applications.**

I. Introduction

Multi-robot systems have been proven useful in more and more applications, and especially in active sensing problems. In these specific kind of problems, information gathering is not only a means to reach a goal, but the goal itself. This paper discusses a specific type of exploration, referred to as event exploration. Event exploration is the process of wandering in a dynamic, topologically known environment in order to detect events, defined as changes in the environment that are not due to the robots' actions. This problem can be encountered in a lot of applications such as industrial maintenance, surveillance or search and rescue. In event exploration the robots must detect the events but also maintain good knowledge of the environment. Multi-robot exploration raises the issue of communication between the robots since a free and complete communication cannot always be assumed. Therefore, it is important to select the information to send, which should be useful for the robot receiving it.

In a previous study [1], we suggested a relevance degree to quantify the relevance of a piece of information by approximating the knowledge of the agent receiving the information. We used this relevance degree in a decentralized decision-theoretic model based on Partially Observable Decision Processes to perform multi-robot event exploration. This model was able to compute an exploration and a communication policy in order to perform effective event exploration while reducing the number of communications. Despite encouraging results, this work suffered from significant flaws and limitations. In this paper, we extend this decision-theoretic model in a complete event exploration framework, called MAPING (MultiAgent Planning for INformationg Gathering). Since localization is not our main purpose, we consider in this study that the robots' positions are exactly known.

The paper is organized as follows. Section II presents some work related to our problem and section III presents briefly the model suggested in [1]. Section IV presents the MAPING framework and section V shows its effectiveness through experiments.

II. Related work

Robotic exploration can be defined as a process that discovers unknown features in an environment by means of mobile robots [6]. When multiple robots are involved, good coordination is crucial for efficiency. In centralized systems, as in [7], a utility is usually computed for each target based on the gain expected at this target and the cost of reaching this target. The targets are then distributed among the robots. Decentralized techniques include decentralized frontier based exploration [8] and auction-based goals assignment [9]. In most cases, the communication is assumed permanent and free so that the agents can share their respective knowledge. Some studies overcome this limitation by imposing meeting points to share information [10] or trying to maintain a team connectivity [11].

The patrolling problem shares a lot of characteristics with the event exploration problem since it focuses on detecting intrusion in a known environment. Multi-agent patrolling has been widely considered. In [12], it has been considered a Travelling Salesman problem, in [13] the authors guaranteed an optimal frequency and in [14], the author considered adversarial setting in which the intruder has knowledge about the patrolling agents.

SLAM and Active SLAM are also subsets of exploration which share characteristics with event exploration as the search for information. Active SLAM integrates planning and stochastic optimization to the classic SLAM to create efficient mapping paths [15]. In [16] a partially known environment was considered and relative entropy optimization methods were used to select the optimal trajectories. In [17], the authors used Kullback-Leibler divergence to define an expected information from a policy.

[1]GREYC Laboratory, University of Caen Lower-Normandy, France {jennifer.renoux, abdel-illah.mouaddib}@unicaen.fr
[2]Airbus Defence and Space, Val de Reuil, France simon.legloannec@cassidian.com

The problems usually studied in multi-robot exploration and active SLAM consist of exploring an unknown static environment and locating fixed targets. On the contrary, event exploration focuses on finding dynamic events in a known environment. The patrolling problem also considers this kind of settings, but focuses on finding an optimal path to detect intrusions. The behavior of the robots after the intrusion is detected is rarely discussed. In event exploration, the agents' knowledge needs to be constantly updated and maintained, even on features that have been previously discovered.

III. Previous model

A. The relevance degree

The environment $\mathcal{E}$ is modeled as a set of features, each represented by a variable X_k that can take different values. The set of all possible values for all variables is written $D(\mathcal{E})$. At each time step, there is a probability $\delta \in [0,1]$ that a change occurs for at least one variable. δ is called the *dynamism degree* of the system. An agent i collects information about the variables and maintains a belief state $\mathcal{B}^{\mathcal{E}}_{i,t} = \{b^k_{i,t} \; \forall X_k \in \mathcal{E}\}$, where each $b^k_{i,t}$ is a probability distribution over the values of X_k. An observation o_p is considered as relevant for an agent i if it is new or if it confirms agent i's beliefs. To assess novelty, the Hellinger distance is used and the observation is considered new if $D_H(\mathcal{B}^{\mathcal{E}}_{i,t}||\mathcal{B}^{\mathcal{E}}_{i,t+1}) > \epsilon$ where ϵ is a fixed problem-dependent threshold, $\mathcal{B}^{\mathcal{E}}_{i,t+1}$ is the belief state updated with o_p and $D_H(\mathcal{B}^{\mathcal{E}}_{i,t}||\mathcal{B}^{\mathcal{E}}_{i,t+1})$ is the Hellinger distance between the two belief states. To assess if an observation confirms an agent's beliefs, the accuracy of the belief state before and after update are compared. The accuracy is measured using Shannon entropy and $\mathcal{B}^{\mathcal{E}}_{i,t+1}$ is said to be more precise than $\mathcal{B}^{\mathcal{E}}_{i,t}$ if $H(\mathcal{B}^{\mathcal{E}}_{i,t+1}) < H(\mathcal{B}^{\mathcal{E}}_{i,t})$. The relevance degree is then defined as follows:

Definition 1: The degree of relevance of an observation o_p for an agent i, noted $rel_i(o_p)$, is given by

$$rel_i(o_p) = \delta D_H(\mathcal{B}^{\mathcal{E}}_{i,t}||\mathcal{B}^{\mathcal{E}}_{i,t+1}) + (1-\delta)\frac{H(\mathcal{B}^{\mathcal{E}}_{i,t}) - H(\mathcal{B}^{\mathcal{E}}_{i,t+1})}{H_{max}}$$

with $\mathcal{B}^{\mathcal{E}}_{i,t+1}$ is the belief state updated with o_p, H_{max} is the maximum entropy, and δ is the dynamism degree of the system.

For more details concerning the Hellinger distance and the entropy in this context, please refer to [1].

B. The decision-theoretic model

The goal of the agents is to reduce the uncertainty over their belief state but also to converge towards common beliefs. It is impossible to ensure that the agents have a correct belief state, but one can assume that if several agents have similar belief states, they are more likely to be close to the true state - formally, to the belief state that assigns 1 to the real state and 0 to all others. To do so, we defined in [1] a POMDP with an extended belief state to model information about other agents' beliefs. The POMDP is a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{O}, \mathcal{T}, \omega, \mathcal{R}, b_0 \rangle$ where

- $\mathcal{S}$ is a set of states, corresponding to the joint instantiations of the random variables $X_k \in \mathcal{E}$.
- $\mathcal{A}$ is a set of epistemic actions
- $\mathcal{O} = \{o_k \in \mathcal{E}\}$ is the set of observations
- $\mathcal{T} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0,1]$ is the transition function
- $\omega : \mathcal{O} \times \mathcal{S} \times \mathcal{A} \to [0,1]$ is the observation function
- $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to [0,1]$ is the reward function
- b_0 is the initial extended belief state

Only two types of actions are considered: exploration actions $Exp(X_k)$ and communication actions $Comm(o_p, j)$. When an agent explores, it senses the value of a random variable and receives an observation about this random variable. When an agent communicates, it sends an observation to another agent. The POMDP uses an extended belief state which includes not only beliefs of agent i on the environment's variables, written $\mathcal{B}^{\mathcal{E}}_{i,t}$, but also beliefs of agent i on the beliefs of all other agents j, written $\mathcal{B}^{j,\mathcal{E}}_{i,t}$. The extended belief state of an agent i is thus defined as $\mathcal{B}_{i,t} = \mathcal{B}^{\mathcal{E}}_{i,t} \bigcup_{j \neq i} \mathcal{B}^{j,\mathcal{E}}_{i,t}$. This belief state is updated in three cases: (1) when the agent receives a new observation from its sensors after an exploration $-$ it updates its own beliefs $\mathcal{B}^{\mathcal{E}}_{i,t}$ (2) when the agent communicates an observation to another agent $-$ it updates its beliefs concerning the other agent's beliefs $\mathcal{B}^{j,\mathcal{E}}_{i,t}$ (3) when the agent receives an observation from another agent $-$ it updates its own beliefs $\mathcal{B}^{\mathcal{E}}_{i,t}$ and its beliefs concerning the other agent's beliefs $\mathcal{B}^{j,\mathcal{E}}_{i,t}$. All the belief updates are done using Bayesian updates [2], regardless of their source (exploration or communication).

The reward function of the POMDP is defined on the agent's beliefs and not on the system states as in classic POMDPs. The reward for exploring a variable corresponds to the expected gain of information minus an exploration cost: $R(\mathcal{B}_{i,t}, Exp(X_k)) = \sum_{s \in \mathcal{S}} \sum_{o_k \in \mathcal{O}} \mathcal{B}_{i,t}(s)\omega(o_k, s, a) rel_i(o_k) - C_{Exp(X_k)}$. The reward for communicating an observation corresponds to the evaluated gain of information for the other agent minus a communication cost: $R(\mathcal{B}_{i,t}, Comm(o_k, j)) = \sum_{s \in \mathcal{S}} \mathcal{B}_{i,t}(s)\omega(o_k, s, a) rel_j(o_k) + (D_H(\mathcal{B}^{\mathcal{E}}_{i,t}||\mathcal{B}^{j,\mathcal{E}}_{i,t}) - D_H(\mathcal{B}^{\mathcal{E}}_{i,t+1}||\mathcal{B}^{j,\mathcal{E}}_{i,t+1})) - C_{Comm(o_k,j)}$

C. Limitations

Even though the results presented in [1] support the fact that such a model is feasible and interesting, some issues and limitations have been raised. A major undesired side-effect of this approach is that agents tend to manipulate other agents' beliefs for their own benefit. Indeed, when an agent has reached a satisfactory belief state, that is to say with low entropy, the only way to gain more reward is to communicate relevant observations since exploration is not useful anymore. Therefore, the agent tends to communicate observations it considers improbable - and so potentially false - in order to degrade other agents' belief states, and then communicate observations it considers probable in order to improve the other agents' belief state again. This can be easily explained by the fact that event exploration is an open-ended task and novelty is rewarded: an improbable observation could be the result of a change in the environment. To tackle this problem agents must cast doubt on their beliefs by forgetting old

information. This forgetting mechanism is not implemented in this model: when a satisfactory belief state is reached, it is not modified unless the agent receives an observation that contradicts its beliefs.

A second limitation concerns the scalability of the problem. The belief state grows exponentially in the number of variables and agents. This leads to a major difficulty in solving the POMDP for real problems. On top of that, the reward function suggested not being convex, it is impossible to apply most of the literature techniques as [3] suggested. We exploited in [1] the specific structure of the model to transform the POMDP directly into a Belief-MDP and solve it, using some ad hoc variable separation. In this paper, we formalize and generalize this technique.

IV. The MAPING framework

When exploring a dynamic environment, the robot's beliefs can be quickly contradicted by changes in the environment. However, the robot can only detect those changes if it explores again a feature that it already explored previously, and so forgets old observations it received. To our knowledge, there is currently no work investigating this kind of forgetting mechanism in decision-theoretic frameworks for exploration. One contribution of the MAPING model consists in adding another belief update state after the one normally present in POMDPs in order to make the robot's beliefs less precise. Mathematically, it consists of applying a transformation to the robot's beliefs, which are probability distributions, in order to bring them closer to the uniform distribution in which all the values have the same probability and which represents the fact that the robot knows nothing. We call this transformation *widening*. We will first define some notations used in the widening function. We consider the features modeled as variables X_k, taking their values in their domain $D(X_k)$. The set of all variables is noted $\mathcal{X} = \{X_k\}$. The belief state is a set $\{P^1, \dots, P^{|\mathcal{X}|}\}$ of probability distributions, where P_k is a probability distribution on the set $D(X_k)$, intuitively representing the subjective estimate of the value of X_k by the robot. To widen the belief state, we need to apply a transformation f to all the probabilities of the probability distributions in the belief state.

Definition 2 (widening function): A function $f : [0,1] \to [0,1]$ is called a *widening function* for a variable X_k with domain $D(X_k)$ if it satisfies the following constraints:

$$\begin{cases} f(P_k^i) \geq 0, \forall x_i \in D(X_k) & (1) \\ \sum_{x_i \in D(X_k)} f(P_k^i) = 1 & (2) \\ f(\frac{1}{n_k}) = \frac{1}{n_k} & (3) \\ \left| f(P_k^i) - \frac{1}{n_k} \right| \leq \left| P_k^i - \frac{1}{n_k} \right|, \forall x_i \in D(X_k) & (4) \\ (P_k^i - \frac{1}{n_k})(f(P_k^i) - \frac{1}{n_k}) \geq 0, \forall x_i \in D(X_k) & (5) \end{cases}$$

where n_k denotes $|D(X_k)|$ and $P_k^i = P(X_k = x_i)$.
Constraints 1 and 2 define that the result of the transformation must be a probability distribution. Constraints 3 and 4 represent the fact that we want to widen the probability distribution until a fixed point - the uniform distribution. Those two constraints describe a contraction mapping admitting $1/n_k$ as a fixed point. Finally, Constraint 5 models that the transformation must keep the shape of the beliefs, that is to say that if a probability is greater than $1/n_k$, the result of the transformation should still be greater than or equal to $1/n_k$.

The choice of the shape of the widening function (linear, logarithmic, etc.) depends on the problem considered and the type of belief update one wants to implement. However, in most cases, beliefs should be revised linearly every time step. Except in some very specific applications, we believe that there is no reason for the robot to forget quickly recent information and then to slow down, or on the contrary to keep all information for a certain amount of time and then to suddenly forget about it. Therefore, we think that in most of the applications a linear function should be the most simple and efficient widening function to implement. That's why we will focus on this type of function in the remainder of the paper. The method to find a widening function based on another function shape remains similar to the one presented in this section.

Theorem 1: Let $\{X_k\}$ be the set of all random variables of the POMDP and $n_k = |D(X_k)|$ the number of possible values for X_k. Let $\mathcal{P}_k$ be the set of all the possible probability distributions over X_k. Let $(f_{X_k})_{X_k \in \mathcal{X}}$ be an indexed family of linear functions. Then each f_{X_k} is a widening function if and only if $\exists a_k \in [0,1]$ such that:

$$\begin{aligned} f_{X_k} : \quad & \mathcal{P}_k \to \mathcal{P}_k \\ & P_k \mapsto a_k \times P_k + \frac{1 - a_k}{n_k} \end{aligned}$$

Proof: Let us consider a linear function defined by $f_{X_k}(P_k) = a \times P_k + b$. Let us denote $P_k^i = P(X_k = x_i)$. The constraints system can be written $\forall P_k \in \mathcal{P}_k$:

$$\begin{cases} a\, P_k^i + b \geq 0, \forall x_i \in D(X_k) & (6) \\ \sum_{x_i \in D(X_k)} \left(a\, P_k^i + b\right) = 1 & (7) \\ a\, \frac{1}{n_k} + b = \frac{1}{n_k} & (8) \\ \left| a\, P_k^i + b - \frac{1}{n_k} \right| \leq \left| P_k^i - \frac{1}{n_k} \right|, \forall x_i \in D(X_k) & (9) \\ \left(a\, P_k^i + b - \frac{1}{n_k}\right)\left(P_k^i - \frac{1}{n_k}\right) \geq 0, \forall x_i \in D(X_k) & (10) \end{cases}$$

Constraint 8 enables the derivation $b = \frac{1-a}{n_k}$. By replacing it in the other constraints, contraint 7 becomes $\forall P_k \in \mathcal{P}_k$

$$\begin{aligned} & \textstyle\sum_{x_i \in D(X_k)} \left(a\, P_k^i + \frac{1-a}{n_k}\right) = 1 \\ \Leftrightarrow\ & \textstyle\sum_{x_i \in D(X_k)} \left(a\, P_k^i\right) + 1 - a = 1 \\ \Leftrightarrow\ & a \left[\textstyle\sum_{x_i \in D(X_k)} \left(P_k^i\right) - 1\right] + 1 = 1 \end{aligned}$$

P_k being a probability distribution, $\sum_{x_i \in D(X_k)} \left(P_k^i\right) = 1$. Therefore this constraint is true whatever the value of a.

Constraint 9 becomes $\forall P_k \in \mathcal{P}_k, \forall x_i \in D(X_k)$

$$\left| a\, P_k^i + \frac{1-a}{n_k} - \frac{1}{n_k} \right| \leq \left| P_k^i - \frac{1}{n_k} \right|$$
$$\Leftrightarrow \left| a \left(P_k^i - \frac{1}{n_k} \right) \right| \leq \left| P_k^i - \frac{1}{n_k} \right|$$
$$\Leftrightarrow |a| \left| P_k^i - \frac{1}{n_k} \right| \leq \left| P_k^i - \frac{1}{n_k} \right|$$
$$\Leftrightarrow |a| \leq 1$$

Constraint 10 becomes $\forall P_k \in \mathcal{P}_k, \forall x_i \in D(X_k)$

$$\left(a\, P_k^i + \frac{1-a}{n_k} - \frac{1}{n_k} \right) \left(P_k^i - \frac{1}{n_k} \right) \geq 0$$
$$\Leftrightarrow a \left(P_k^i - \frac{1}{n_k} \right) \left(P_k^i - \frac{1}{n_k} \right) \geq 0$$
$$\Leftrightarrow a \geq 0$$

Therefore we obtain $a \in [0, 1]$.
Constraint 6 becomes $a\, P_k^i + \frac{1-a}{n_k} \geq 0$. Knowing that $a \in [0, 1]$, then $\frac{1-a}{n_k} \geq 0$. Since P_k^i is a probability, $P_k^i \geq 0$. We conclude that Constraint 6 is true for $a \in [0, 1]$.
We can conclude that any f_{X_k} respecting the previous constraints is written: $f_{X_k}(P_k) = a \times P_k - \frac{1-a}{n_k}, a \in [0, 1]$ ■

In the MAPING model, when a robot receives a new observation, it first updates its belief states as described in section III. Secondly, it applies the widening function defined previously on all the elements of the probability distributions composing its belief state:

$$\mathcal{B}_{i,t+1} = \left\langle f_{X_k}(update(\mathcal{B}_{i,t}^{j,k}, o_p)) \right\rangle_{j \in \mathcal{AG}, X_k \in \mathcal{X}}$$

where $update(\mathcal{B}_{i,t}^{j,k}, o_p)$ is the result of the updating process of the belief over the variable X_k with observation o_p and f_{X_k} is the widening function.

To solve the POMDP in the MAPING model, we showed in [1] that it was possible to transform this POMDP into an equivalent Belief-MDP and to solve it using classical techniques from the literature. However, the resolution faces a big complexity problem and is not scalable. This problem is due to two factors: the continuous belief state space and the fact that the reward computation involves belief updates, which are time-consuming. Discretization techniques could be used, but still lead to a very large belief state space in order to have an acceptable precision. We used in [1] an assumption about the independence of the variables describing the environment. In this paper, we formalize this assumption and generalize it to independent groups of variables. Exploiting independence in order to simplify the resolution of MDP-based frameworks has already been suggested [4], [5]. In that sense our work is similar to those, but using different assumptions. We assume the two following independence relations: (i) Variable independence: the value of a given random variable does not depend on all the other variables in the system but only on a subset of variables. (ii) Observation independence: the probability of receiving a given observation when performing a given action depends only on the values of a given subset of random variables. To better understand those assumptions, let us consider the following scenario that will be used for experiments. A building is made up of a given number of rooms. All the workers in this building need to leave before 8:00pm. After this time, robots wander in the building to ensure that all the rooms are empty. We can consider that the probability of a room being empty is independent of the probability of that the other rooms are empty or not. Moreover, the probability of receiving a given observation ($empty$ or $not\text{-}empty$) depends only on the room the robot is checking and not on the other rooms. We agree that in some cases, assuming such independence may be a simplification of the dynamic of the system. However, we believe that the optimality lost due to this simplification is balanced by the fact that a good solution is computable in a reasonable time. To explain how to split the POMDP, we first need to define the independence of sets.

Definition 3 (Independence of sets): Two sets A and B are called independent if and only if: $\forall a \in A, \forall b \in B, P(a, b) = P(a) \times P(b)$

Let us consider $\mathcal{X}$ the set of all the random variables describing the environment and $\mathcal{O}$ the set of all the observations. We assume a partition of the set $\mathcal{X}$, written $\mathcal{P}(\mathcal{X})$, and a partition of the set $\mathcal{O}$, written $\mathcal{P}(\mathcal{O})$, such that the following holds:

(a) $\forall X, Y \in \mathcal{P}(\mathcal{X}), X \neq Y$, X and Y are independent.
(b) $\forall O \in \mathcal{P}(\mathcal{O})$, there is a unique set $X \in \mathcal{P}(\mathcal{X})$ so that $\forall o \in O, P(o|s, a) = P(o|X, a)$.

Such a decomposition is guaranteed to exist, take $P(X) = X$ and $P(O) = O$, and the finer this decomposition is, the better the algorithm works. Constraint (a) enables us to rewrite the transition function as follows: $T(s, a, s') = \prod_{X \in \mathcal{P}(\mathcal{X})} T(X, a, X')$. Constraint (b) enables us to rewrite the observation function as follows: $\omega(o, s, a) = P(o|s, a) = P(o|X, a)$, X being the set of variables that influences the probability of receiving o. Using these partitions, we can build a set of $|\mathcal{P}(\mathcal{X})|$ sub-POMDPs, each sub-POMDP being a tuple $\langle \mathcal{S}_\ell, \mathcal{A}, \mathcal{O}_\ell, \mathcal{T}_\ell, \omega_\ell, \mathcal{R}_\ell, b_{\ell,0} \rangle$ where

- $\mathcal{S}_\ell$ is all the possible joint instantiations of the variables in the set $X_\ell \in \mathcal{P}(\mathcal{X})$
- $\mathcal{A}$ is the same set of actions as in the global POMDP
- $\mathcal{O}_\ell \in \mathcal{P}(\mathcal{O})$ being the set of observations depending on the set X_j, as defined previously
- $\mathcal{T}_\ell$ is the transition function applied to variables X_ℓ
- ω_ℓ is the observation function applied to variables of X_ℓ and observations of $\mathcal{O}_\ell$
- $\mathcal{R}_\ell$ is the reward function applied to variables of X_ℓ
- $b_{\ell,0}$ is the initial belief state

Each belief state used in the sub-POMDPs, written $b_{\ell,i,t}$, is one part of the belief state of the global POMDP: $\mathcal{B}_{i,t} = \langle b_{\ell,i,t} \rangle_{\forall \ell}$.

Each of these sub-POMDPs can be solved independently by transforming it into a Belief-MDP as explained in [1]. However, in MAPING we removed the Exploration cost

C_{Exp} from the reward in order to integrate it directly when building the global policy, as presented in algorithm 1. Indeed, in robotic exploration, the cost of exploring an area depends on the distance between the robot and the target area. If we want to take into account this cost in the reward function, we need to model the robot's position as an environment variable, which increases again dramatically the number of states in the system. This combinatorial explosion can be avoided by computing the cost later. Techniques using state-space discretization can be applied since the state space is less complex and so the number of cells is tractable. While solving the belief-MDPs, we store the optimal value function $V^*(b_\ell)$ for each sub-POMDP in addition to the optimal policy π_ℓ. Then, during execution, the robot can retrieve the action to perform following the algorithm 1. In this algorithm, the robot compares the expected gain of

Algorithm 1: Getting the action to perform from local optimal policies

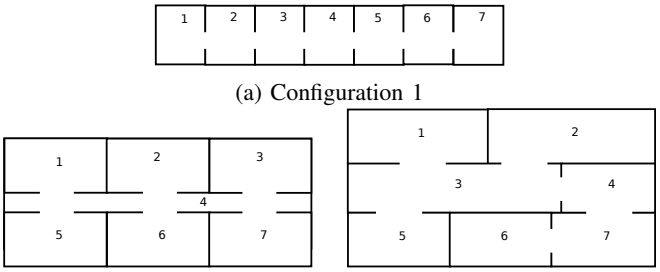
Data: local policies and associated value functions, current belief state $\mathcal{B}_{i,t}$
Result: action to execute
$V_{max} = -Infinity$;
$a_{opt} = null$;
foreach $b_{\ell,i,t}$ *composing* $\mathcal{B}_{i,t}$ **do**
  if $V_{\pi_\ell}(b_{\ell,i,t}) \geq V_{max} - C_{\pi_\ell}$ **then**
    $V_{max} = V_{\pi_\ell}(b_{\ell,i,t})$;
    $a_{opt} = \pi_\ell(b_{\ell,i,t})$;
  end
end
return a_{opt} ;

each possible action for the current belief state to select the best action. This gain corresponds to the expected value of performing the action minus the cost of performing the action. For $Explore$ actions, the cost depends on the distance between the current position of the robot and the position of the target, computed thanks to the A^* algorithm. For $Communicate$ actions, the cost depends on the bandwidth used to communicate the message. This comparison between all the V_{π_ℓ} is possible since the rewards of each POMDP are commensurable, which implies that the computed values are also commensurable. This method uses optimal local policies to build a global policy that is not proved to be optimal. However, experiments tend to show that the global policy obtained using this method is satisfactory.

The MAPING model introduced two new parameters in addition to the usual POMDP parameters: δ and a, both application-dependent. δ appears in the relevance definition and models how dynamic the system is. It is linked to the probability of change in the environment: $\delta = 1 - AverageProbabilityOfChange$. The average probability of change can be computed thanks to the transition function of the POMDP. The parameter a appears in the widening function and depends on the temporal validity of the information: the larger the temporal validity, the smaller a. Those parameters, as well as the POMDP ones, need to be tuned when applying the model to a problem. This tuning phase may be delicate and requires tests, but is mandatory for good results. However, both parameters are directly related to real parameters (the probability of change and the temporal validity), which makes them easier to manipulate.

V. Experiments

Let us consider the scenario described in section IV. A building is made up of 7 different areas, each area can be in either of two states: $empty$, meaning that no worker is in it, or $not\text{-}empty$, meaning that at least one worker is in the area. The state of an area is assumed independent of the state of the other areas and can change without intervention from the robots. The system has been tested in three different configurations of the building, presented on Figure 1, and implemented using ROS Hydro. Since our main purpose is



(a) Configuration 1

(b) Configuration 2

(c) Configuration 3

Fig. 1: The three building configurations

neither the localization nor the sensor data processing, we assumed that the robots' position are exactly known, and we implemented a noisy simulator that sends directly the high-level observations $empty$ or $not\text{-}empty$ depending on the robot's position. We considered that the simulator will send the correct observation 98% of the time. Due to the discretization of the belief state space, it was not possible practically to implement the widening technique at each time step: if the parameter a was too low, the belief state was not modified. Therefore, we decided to implement this widening only every 10 time steps. This number has been tuned through tests. We tested the system with teams of 3, 4 and 5 agents and executed 100 actions per agent.

Let's first discuss the difficulty of solving optimally the model. We used a naive discretization technique that consists of discretizing the belief state space. We considered a finite probability distribution with 11 elements ($(0,1)$; $(0.1,0.9)$; $(0.2,0.8)$; $(0.3,0.7)$; $(0.4,0.6)$; $(0.5,0.5)$; $(0.6,0.4)$; $(0.7,0.3)$; $(0.8,0.2)$; $(1,0)$) and computed the number of states in the state space. The results are presented on table I. It is clear that an optimal algorithm is unable to handle problems with more than 2 agents and 3 variables. We could choose a rougher discretization, but even with it, the problems that can be handled are hardly bigger than 3 agents and 4 variables. However, if we manage to split the POMDP in sub-POMDP, we reduce the number of variables and the problem is easier to solve. In our case since the changes can

happen in each area independently, the POMDP to solve has only one variable.

The first criteria to evaluate the performance of our system should measure the correctness of the agents' belief states. We first measured this correctness by computing the average Hellinger distance between the agents' current belief state and a perfect belief state, representing the real state of the system. Figure 2 shows the evolution of this distance for the three configurations and with $delta = 0.7$. We will present only the results obtained for the four-agent system since those obtained for three and five agents are very similar and the same conclusions can be drawn. We can note that the Hellinger distance varies a lot during the execution. These variations have two causes : (1) the widening function (2) the dynamicity of the system. The widening function artificially degrades the agents' beliefs to make them check again variables already discovered. This involves of course an increase in the Hellinger distance to the perfect state. The second cause - the system's dynamics - is also quite obvious: each time a change occurs in the environment, the perfect belief state changes, and the agents can only change their own beliefs afterward. However, these graphics show that in all configurations, the agents manage to improve quickly their knowledge of the environment after a change or a knowledge degradation.

Concerning the communication strategy, figure 3 presents the number of communications sent per emitter and per receiver for the four-agent system. We can clearly see that in

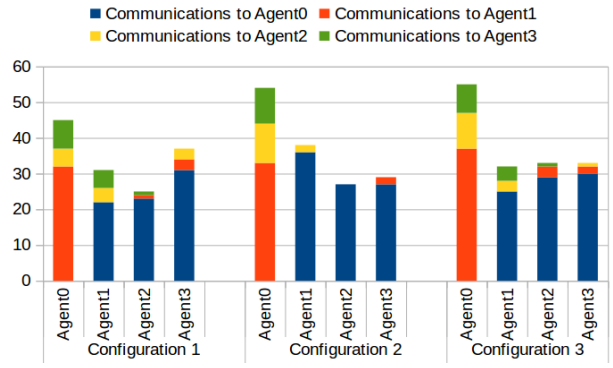


Fig. 3: Number of communications sent by each agent to any other agent with the 4-agent system

all cases, the agent 0 acts as a central point, receiving all the communication of the others and sending some information to the other agents in high-dynamics cases. This is due to algorithm 1 and the way the next action is chosen.

TABLE I: Number of states with 2 possible values per variable

	2 agents	3 agents	4 agents	5 agents
1 var.	121	1331	14641	161051
2 var.	14641	1771561	214358881	2.59×10^{10}
3 var.	1771561	2.36×10^{9}	$3,14 \times 10^{12}$	$4,18 \times 10^{15}$
4 var.	2.14×10^{7}	3.14×10^{12}	$4,59 \times 10^{16}$	$6,73 \times 10^{20}$

Indeed, if some actions have the same value, the first one in the order of their ID will be chosen. For all agents, the first communication action is communicating to agent 0 and so is chosen more often than other communication actions. This can be improved by changing algorithm 1 and select randomly the next action among those with the highest value. Figure 4 presents the evolution of the Hellinger distance to the perfect state per agent and shows that all the agents manage to improve their beliefs despite the reduced communication.

All the experiments presented have been conducted on simulation. However, the ROS implementation makes it extremely easy to connect real robots. Real tests on Pioneer 3-AT robots are ongoing but have not been included in this paper due to a lack of space.

VI. Conclusion

We presented MAPING, a model dedicated to event exploration. In this model, the agents in the system exchange relevant observations and compare their beliefs to the other agents' beliefs in order to converge toward a common belief state. In order to ensure that the exploration is open-ended, we introduce a *forgetting* step in the beliefs update process. This new step consists of applying a contraction mapping to the beliefs in order to widen them. We also presented a method based on the well-known adopted variable independence assumption to solve the MAPING model in large environments. This method splits the global into smaller sub-POMDPs to solve them independently and then to combine them again to create a satisfactory policy. Experiments showed that the MAPING model manages to do event exploration in real-type applications. However, the MAPING model may encounter some problems in case of the failure of one of the agents. New experiments should be conducted to evaluate the impact of such a failure and suggest improvements. A general improvement we considered is to make the belief update step more complex by including reputation systems. With this modification, if a robot was faulty - with a broken sensor for instance - its reputation in the system would decrease over time and the observation it transmits would less affect the agents' beliefs than observations from other robots. This would improve the robustness of the system.

References

[1] J. Renoux, A-I. Mouaddib and S. Le Gloannec, A Distributed Decision-Theoretic Model for Multiagent Active Information Gathering, in Conf. Workshop on Multi-agent Coordination in Robotic Exploration, Prague, 2014.

[2] A.R. Cassandra, L.P. Kaelbling and M.L. Littman, Acting optimally in partially observable stochastic domains, in Proc. 12th Nat. Conf. on Artificial intelligence, Seattle, 1994.

[3] M. Araya, O. Buffet, V. Thomas and F. Charpillet, A POMDP extension with belief-dependent rewards, in Advances in Neural Information Processing Systems, 2010, pp. 64-72.

[4] A. Beynier and A-I. Mouaddib, A polynomial algorithm for decentralized Markov decision processes with temporal constraints, in Proc. 4th Internat. Joint Conf. on Autonomous Agents and Multi-Agent Systems, Utrecht, The Netherlands, 2005.

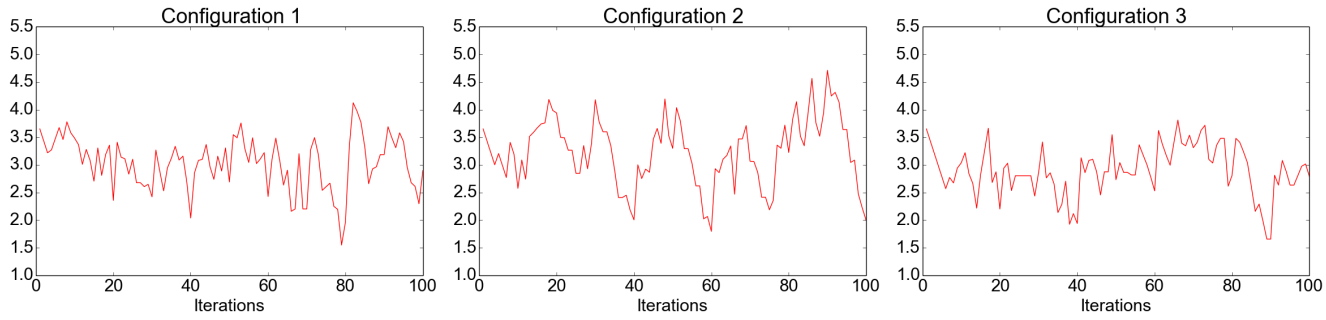


Fig. 2: Evolution of the average Hellinger distance between agents' belief state and the perfect belief state for the 4-agent system

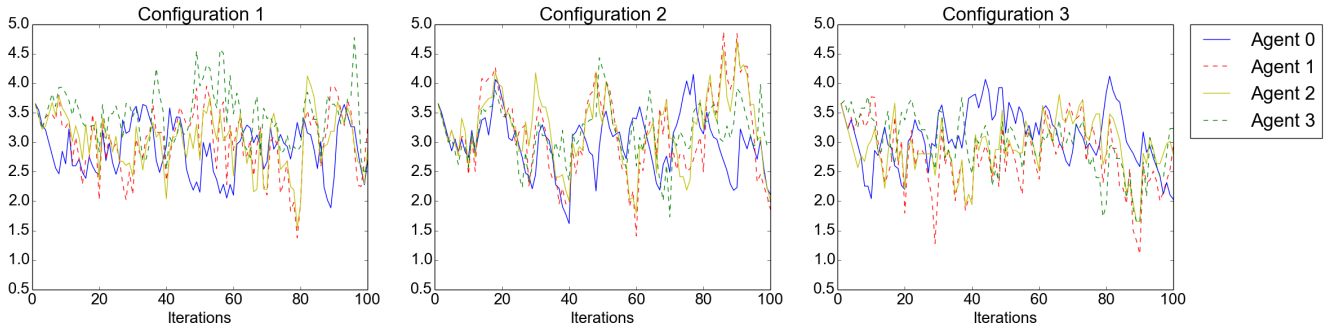


Fig. 4: Evolution of each agent's Hellinger distance between its belief state and the perfect belief state for the 4-agent system

[5] F.A. Oliehoek, M. Spaan, S. Whiteson and N. Vlassis, Exploiting locality of interaction in factored Dec-POMDPs, in Proc. 7th Internat. Joint Conference on Autonomous Agents and Multi-Agent Systems, Estoril, Portugal, 2008.

[6] F. Amigoni, N. Basilico and A.Q. Li, How much worth is coordination of mobile robots for exploration in search and rescue?, in RoboCup 2012: Robot Soccer World Cup XVI, 2013, pp 106-117.

[7] W. Burgard, M. Moors, C. Stachniss and F.E. Schneider, Coordinated multi-robot exploration, in IEEE Transactions on Robotics, 2005, pp 376-386.

[8] A. Marjovi, J.G. Nunes, L. Marques and A. de Almeida, Multi-robot exploration and fire searching, in Proc. Internat. Conf. on Intelligent Robots and Systems, St. Louis, 2009.

[9] R. Zlot, A. Stentz, M.B. Dias and S. Thayer, Multi-robot exploration controlled by a market economy,in Proc. Internat. Conf. on Robotics and Automation, Washington, 2002.

[10] J. De Hoog, S. Cameron and A. Visser, A.: Autonomous multi-robot exploration in communication-limited environments, in Proc. 11th Conference Towards Autonomous Robotic Systems, Plymouth, UK, 2010.

[11] M. Powers and T. Balch, Value-Based Communication Preservation for Mobile Robots, in Distributed Autonomous Robotic Systems, 2004.

[12] Y. Chevaleyre, Theoretical analysis of the multi-agent patrolling problem, in Proc. Internat. Conf. on Intelligent Agent Technology, Beijing, 2004.

[13] Y. Elmaliach, N. Agmon and G. A. Kaminka, Multi-robot area patrol under frequency constraints, in Annals of Mathematics and Artificial Intelligence, 2009, pp 293-320.

[14] N. Agmon, On events in multi-robot patrol in adversarial environments, in Proc. 9th Internat. Conf. on on Autonomous Agents and MultiAgent Systems, Toronto, 2010.

[15] C. Leung, S. Huang, G. Dissanayake, Active SLAM using model predictive control and attractor based exploration, in Proc. Internat. Conf. on Intelligent Robots and Systems, Beijing, 2006, pp. 5026-5031.

[16] M. Kontitsis, E.A. Theodorou and E. Todorov, Multi-robot active slam with relative entropy optimization, in Proc. American Control Conference, Washington DC, 2013, pp 2757-2764.

[17] L. Carlone, J. Du, M.K. Ng, B. Bona and M. Indri, Active SLAM and exploration with particle filters using Kullback-Leibler divergence, in Journal of Intelligent & Robotic Systems, 2013, Springer, pp 1-21.