Workshop on Towards Fully Decentralized Multi-Robot Systems: Hardware, Software and Integration
2013 IEEE International Conference on Robotics and Automation, ICRA 2013
Karlsruhe, Germany, May 6 - 10, 2013, **http://icra2013mrs.tuebingen.mpg.de/**

Concurrent Bayesian Learners for Multi-Robot Patrolling Missions

David Portugal, Micael S. Couceiro and Rui P. Rocha

***Abstract*— Distributed robot systems have been adopted lately for security purposes, such as in automatic multi-robot patrolling of infra-structures. Research has shown that deterministic patrol routes can lead to effective performance. However, they can potentially be predicted by intelligent intruders.**

This work presents a probabilistic multi-robot patrolling strategy, where each autonomous agent uses Bayesian reasoning to decide its moves in the environment. Each member of the team is a learning agent that collects information and assesses the state of the system, using concurrent reinforcement learning to influence future moves. This way, effective coordination in collective patrol is achieved, as shown by preliminary simulation experiments.

I. INTRODUCTION

Multi-robot patrolling is a cooperative task, which requires agents to coordinate their decision-making to visit every position in the environment with the ultimate goal of achieving collective optimal performance. Despite the high potential utility of such application, only recently the Multi-Robot Patrolling Problem (MRPP) has been rigorously addressed.

Contributions to the MRPP at a theoretical level have been addressed in the past and it has been shown that the problem is NP-Hard [1]. In addition, a few strategies to solve the MRPP have been presented lately [2]. Within all the strategies pursued so far, the creation of adaptive behaviors that allows agents to learn how to effectively patrol a given scenario are the more promising, because such adaptability fosters the unpredictability principle in a way that intruders may be unable to access the patrolling trajectory information.

Certain works in this field have adopted machine learning methods aiming to adapt agents' behavior. For instance, the work of Santana *et al.* proposed to model the MRPP as a Q-learning problem in an attempt to allow automatic adaptation of the agents' strategies to the environment [3].

Alternatively to reinforcement learning, some strategies have been using stochastic approaches that benefit from probabilistic decision-making to overcome the deterministic nature of classic patrolling applications. For instance, in [4] the patrolling problem is casted as a multi-agent Markov decision process. Chen and Yum [5] also formulated the problem as a Markov decision process and proposed a patrol routing strategy under a finite horizon approximation.

In this work, a new distributed and adaptive approach for multi-robot patrol is proposed. Each robot decides its local patrolling moves online, without requiring any central planner. Decision-making is based upon Bayesian reasoning on the state of the system, considering the history of visits and teammates actions, so as to promote adaptation and effective coordination between patrolling agents. Preliminary experimental results illustrate the potential of using the proposed technique.

This work was supported by PhD scholarships (SFRH/BD/64426/2009)and (SFRH/BD/73382/2010), the CHOPIN research project (PTDC/EEA-CRO/119000/2010) and by the ISR-Institute of Systems and Robotics (project PEst-C/EEI/UI0048/2011), all of them funded by the Portuguese science agency "Fundação para a Ciência e a Tecnologia" (FCT).

All authors are with the Institute of Systems and Robotics (ISR), University of Coimbra (UC), Pólo II, 3030-290 Coimbra, Portugal, e-mail: {davidbsp,micaelcouceiro,rprocha} at isr.uc.pt

II. PRELIMINARIES

Agents are assumed to have an *a priori* representation of the environment in the form of an undirected and connected navigation graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. $\mathcal{G}$ is composed of vertices $v_i \in \mathcal{V}$ and edges $e_{i,j} \in \mathcal{E}$, where each vertex represents a specific location that must be visited regularly and each edge represents the connectivity between these locations. The MRPP can be reduced to coordinate robots in order to frequently visit all $v_i \in \mathcal{G}$, ensuring the absence of atypical situations. In this work, a performance criterion based on the idleness concept [2], [6] has been considered, given that it measures the elapsed time since the last visit from any agent in the team to a specific location. Idleness is intuitive to analyze and brought into confrontation with the possibility of attacks to the system, seen as it uses time units.

The instantaneous idleness of a vertex $v_i \in \mathcal{V}$ in time step t is defined as:

$$\mathcal{I}_{v_i}(t) = t - t_l, \tag{1}$$

where t_l corresponds to the last time instant when the vertex v_i was visited by any robot of the team. Also, the average idleness of $v_i \in \mathcal{V}$ in a total time τ can be defined as:

$$\overline{\mathcal{I}_{v_i}} = \frac{1}{\tau} \sum_{t=0}^{\tau} \mathcal{I}_{v_i}(t). \tag{2}$$

Finally, in order to obtain a generalized performance measure, the average idleness of $\mathcal{G}$ ($\overline{\mathcal{I}_{\mathcal{G}}}$) is defined as:

$$\overline{\mathcal{I}_{\mathcal{G}}} = \frac{1}{|\mathcal{V}|} \sum_{i=1}^{|\mathcal{V}|} \overline{\mathcal{I}_{v_i}}, \tag{3}$$

where $|\mathcal{V}|$ represents the cardinality of the set $\mathcal{V}$.

The patrolling problem with R robots can be described as the problem of finding a set of R paths, which visit all vertices $v_i \in \mathcal{V}$ of $\mathcal{G}$, with the overall team goal of minimizing $\overline{\mathcal{I}_{\mathcal{G}}}$. Note, however, that such paths are calculated

online and locally during the mission, in order to adapt to the system's needs.

III. Concurrent Bayesian Learning Strategy

In order to solve the MRPP, a Bayesian model, which represents the decision of moving from a vertex of $\mathcal{G}$ to another, is proposed. For β neighbors of the current vertex v_0, where $\beta = deg(v_0)$[1], the model is applied β times. More details on the proposed strategy, which was named Concurrent Bayesian Learning Strategy (CBLS), are presented next.

A. Distribution Modeling

Two fundamental random variables are firstly defined to characterize the model. The first one simply represents the act of moving (or not) to a neighbor vertex:

$$move_i = \{true, false\}, \tag{4}$$

while the second one is called *arc strength* $\theta_{0,i}$, which represents the appropriateness of traveling from a vertex v_0 to a neighbor v_i using the arc that connects v_0 to v_i:

$$\theta_{0,i} \in \mathbb{R}^+. \tag{5}$$

The term "arc" is used instead of "edge" intentionally, since it implies a direction of traveling and $\theta_{j,k} \neq \theta_{k,j}$. Informally, higher values of *arc strength* lead to the edge being traversed more often in the specified direction.

Having defined these variables, agents calculate the degree of belief (*i.e.*, a probability) of moving to a vertex v_i, given the *arc strengths*, by applying Bayes rule:

$$P(move_i|\theta_{0,i}) = \frac{P(move_i)P(\theta_{0,i}|move_i)}{P(\theta_{0,i})}. \tag{6}$$

The prior $P(move_i)$ represents belief obtained from analyzing past data. In the MRPP, prior information about each vertex is encoded in the average idleness of v_i along time τ, given by $\overline{\mathcal{I}_{v_i}}$ in (2). Therefore, $P(move_i)$ is defined as:

$$P(move_i) = \frac{\overline{\mathcal{I}_{v_i}}}{\sum_{k=1}^{|\mathcal{V}|} \overline{\mathcal{I}_{v_k}}}. \tag{7}$$

During the dynamic patrol mission, agents can compute $\overline{\mathcal{I}_V}$ because they communicate to their teammates when new goals are reached. At each decision step, prior information is updated through (7), just before the adoption of (6) to obtain a degree of belief of moving to a vertex v_i.

In addition, it is necessary to define the likelihood $P(\theta_{0,i}|move_i)$ through a statistical distribution to model the *arc strength* $\theta_{0,i}$. In the MRPP, it is advantageous to avoid traversing certain edges at a given time while favoring the use of others, in order to improve performance. To this end, a reward-based learning strategy to continually update the likelihood distribution is proposed. Reinforcement learning enables adaptation to the system's state according to previous decisions and aims at optimizing the collective performance.

[1]The degree (or valency) of a vertex of a graph is the number of edges incident to the vertex.

B. Multi-Agent Reward-Based Learning

In general, reward-based learning methods are attractive since agents are programmed through reward and punishments without explicitly specifying how the task is to be achieved [7]. Herein, reinforcement learning is adopted to continuously estimate the likelihood function. Being a cooperative multi-robot task with distributed information and asynchronous computation; multiple learners are involved.

Each agent chooses an action of moving from v_0 to a neighbor v_i, based on (6). After reaching v_i, the information on its neighborhood has changed, namely the instantaneous idlenesses have been updated. Through information observed after making the move, a reward-based mechanism punishes or benefits the arcs involved in the last decision, affecting future moves starting in v_0 by biasing towards arcs which ought to be visited ahead in time.

Henceforth, the reward-based learning method is explained. When the robot decides between β different vertices v_i in its neighborhood, each v_i will have an associated posterior probability $P(move_i|\theta_{0,i})$. Therefore, it is possible to calculate the normalized entropy $\mathcal{H}$ of the decision:

$$\mathcal{H}(move|\theta) = \frac{-\sum_{i=1}^{\beta} P(move_i|\theta_{0,i})\ \log_2(P(move_i|\theta_{0,i}))}{\log_2(\beta)}. \tag{8}$$

$\mathcal{H}$ provides a measure of uncertainty, being chosen for this reason, as the basis for the reinforcement mechanism.

After deciding and moving to a given v_k, the robot will calculate rewards for each arc between v_0 and all neighbor vertices v_i (including v_k) involved in the previous decision using:

$$\gamma_{0,i} = S_{0,i}(C_i, \mathcal{I}_{v_i}(t)) \cdot (1 - \mathcal{H}(move|\theta)), \tag{9}$$

with:

$$S_{0,i} \in \{-1, 0, 1\}. \tag{10}$$

$S_{0,i}$ gives the reward sign, providing a quality assessment which determines whether a penalty ($S = -1$), a reward ($S = 1$) or no reward ($S = 0$) should be given. As it can be seen, this function uses up-to-date information, namely the number of visits to v_i, given by C_i, and the current instantaneous idleness $\mathcal{I}_{v_i}(t)$. The sign of S is obtained using a set of rules explained below, which are checked as soon as the agent reaches v_i. For that matter, it is necessary to define firstly the normalized number of visits to vertex v_i:

$$\zeta_i = \frac{C_i}{\deg(v_i)}, \tag{11}$$

This is used in the punish/reward procedure given that vertices with higher degree are naturally more visited than vertices with lower degree, being often traversed to reach isolated vertices that tend to have a lower number of visits.

Next, the rules for defining the sign of the rewards $S_{0,i}$ are explained:

- $S_{0,i} = -1$, when the degree of v_0 is higher than one ($\beta > 1$) and the normalized number of visits to v_i (ζ_i) is maximal in the neighborhood of v_0. In case there is more than one vertex with maximal ζ, a negative reward is given to the one with lower instantaneous idleness $\mathcal{I}_{v_j}(t)$ between those.
- $S_{0,i} = 1$, when the degree of v_0 is higher than one ($\beta > 1$) and the normalized number of visits to v_i (ζ_i) is minimal in the neighborhood of v_0. In case there is more than one vertex with minimal ζ, a positive reward is given to the one with higher instantaneous idleness $\mathcal{I}_{v_j}(t)$ between those.
- $S_{0,i} = 0$, in every other situation that differs from the above.

These rules guarantee that when there is more than one vertex involved in the decision, strictly one reward and one penalty are assigned.

In the beginning of the mission, when $t = t_0$, all *arcs strength* $\theta_{0,i}$ are equal to a real positive number κ:

$$\forall \theta_{0,i} \in \theta,\ \theta_{0,i}(t_0) = \kappa. \tag{12}$$

As the mission evolves, the agent updates $\theta_{0,i}$ through:

$$\theta_{0,i}(t) = \theta_{0,i}(t-1) + \gamma_{0,i}(t). \tag{13}$$

Note that the larger the value of κ is set in (12), the less immediate influence the reinforcements received will have on $\theta_{0,i}$. In the experimental tests, $\kappa = 1.0$ was used. This reward-based procedure is expected to make the values of $\theta_{0,i}$ fluctuate as time goes by, informing robots of moves which are potentially more effective, but keeping in mind that robots must visit all vertices v_i in the patrolling mission.

Finally, the learnt likelihood distribution is obtained through normalization of $\theta_{0,i}$:

$$P(\theta_{0,i}|move_i) = \frac{\theta_{0,i}}{\sum_{j=1}^{|\mathcal{E}|}\sum_{k=1}^{|\mathcal{E}|}\theta_{j,k}}, \tag{14}$$

being updated at each decision step and making use of experience acquired in the past for future decisions.

C. Decision-Making

Beyond learning individual likelihood distributions, to completely characterize CBLS it is necessary to guarantee the coordination of robots. In collective operations with a common objective, coordination between agents plays a fundamental role in the success of the mission. Particularly in this context, it is highly undesirable that agents move to the same positions. The distributed communication system used to inform teammates of the current vertex v_0 is therefore augmented with the information of the vertex v_i chosen for the next move. This way, coordination arises simply by discarding the vertex v_i from an agent's decision if another robot has expressed intention to move to it.

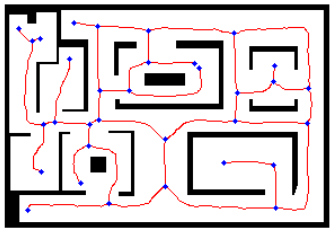


Fig. 1: Environment used in the experiments with respective topological map.

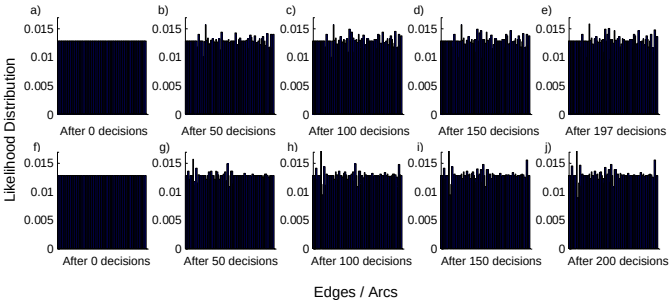


Fig. 2: Evolution of the likelihood distribution in a mission with 2 robots. a) to e) correspond to different decisions instants for robot 1; f) to j) correspond to different decisions instants for robot 2.

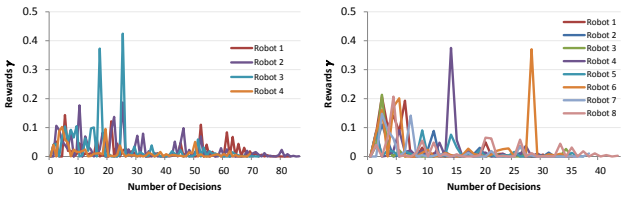
By also communicating their goals, robots can update the information about the state of the system and decide their moves based on that and their progressively acquired experience. Finally, the decision-making process consists of each agent choosing the move to the neighbor vertex with the *maximum a posteriori* (MAP) probability.

IV. Results and Discussion

In order to assess the performance of CBLS, a set of simulation experiments have been conducted. To that end, the environment illustrated in Fig. 1 has been used with different teamsizes of $R = \{1, 2, 4, 6, 8, 12\}$ robots. The Stage multi-robot simulator together with ROS were adopted to implement the experiments. Robots are endowed with navigation abilities, having non-holonomic constraints and travel at a maximum velocity of 0.2 m/s. In addition, they have localization capabilities through the use of an adaptive Monte Carlo localization approach.

All the simulations conducted respect a stopping condition which is determined by the convergence of the average graph idleness ($\overline{\mathcal{I}_{\mathcal{G}}}$) after each patrolling cycle, guaranteeing that results are obtained in the steady-state phase of the mission.

It can be seen in the histograms of Fig. 2 the evolution of the likelihood function of two agents in an example of a patrolling mission with two robots. Each robot apprehends a different distribution and has no control or knowledge on the internal state of its teammate. As expected, peaks in the histograms emerge with the increasing number of decisions. Despite that, it is also clear that values fluctuate around the initial uniform value, which comes as a consequence of



(a) 4 robots. (b) 8 robots.

Fig. 3: Evolution of the absolute reward values along two experiments with different teamsize.

robots having to visit every vertex $v_i \in \mathcal{G}$.

Another interesting aspect observed in the experiments is the descending trend shown by the absolute reward values along the different experiments, which are given by the $(1 - \mathcal{H})$ factor in (9). Fig. 3 illustrates how these values evolve in missions with 4 and 8 robots. Despite the occasional peaks that occur, such values tend to decrease with the number of decisions. This is because, as the system progresses in general, the $\overline{\mathcal{I}_\mathcal{V}}$ values of different vertices become more balanced and, as a consequence, the degree of belief in moving to distinct neighbors comes closer. In such situations, the closer the posterior probabilities are, the higher the entropy becomes, therefore the reward values descend gradually. The peaks observed are justified by situations where agents share nearby areas, temporarily perturbing the $\overline{\mathcal{I}_\mathcal{V}}$ values in the neighborhood of other agents. For that reason, peaks are more observable with greater teamsize.

Moving on to the performance of the algorithm, the boxplot chart in Fig. 4 represents the $\overline{\mathcal{I}_\mathcal{V}}$ values (in seconds) for each tested teamsize. The average value is represented by a black cross, providing a generalized measure: the average graph idleness, $\overline{\mathcal{I}_\mathcal{G}}$ (*cf.* Eq. 3). The ends of the blue boxes and the horizontal red line in between correspond to the first and third quartiles and the median values of $\overline{\mathcal{I}_\mathcal{V}}$, respectively.

As expected, the idleness values decrease when the number of robots grow. Despite the increasing performance displayed by the CBLS approach, the individual contribution of adding more robots gradually reduces with teamsize. Spatial limitations decrease productivity during size scale-up, since the number of times the robots meet increases and beyond a given R, it is argued that they will spend more time avoiding each other than effectively patrolling on their own. This aspect is common to all MRPP strategies.

Another interesting fact illustrated in the boxplot of Fig. 4 is that the median $\widetilde{\mathcal{I}_\mathcal{G}}$ is lower than the mean ($\overline{\mathcal{I}_\mathcal{G}}$) in all configurations. This means that the $\mathcal{I}_\mathcal{V}$ values are positively skewed, *i.e.*, most of the values are below the average, $\overline{\mathcal{I}_\mathcal{G}}$.

Finally, on a more general note, observing the robots using CBLS showed that prediction of patrolling routes is far from being straightforward. This stochastic behavior, together with the promising results obtained, proves the effectiveness of the approach and the potential to be applied in actual security systems with physical teams of robots.

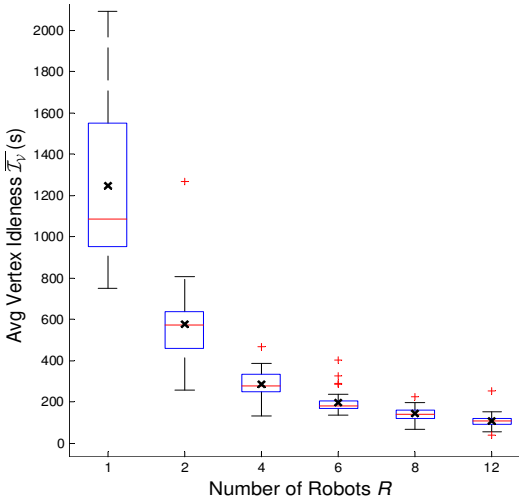


Fig. 4: Overall results running CBLS with different teamsize.

V. Conclusion

In this work, cooperative multi-agent learning has been addressed in order to solve the patrolling problem in a distributed way. Robots make use of Bayesian decision to reason on their moves so as to patrol effectively an environment, while coordinating their behaviors. The decomposition of the problem is possible, reducing the complexity of the general cooperative mission by distributing computational load among each independent learner.

Experimental results have shown that the method is able to tackle the problem, since it can deal with uncertainty and actions are selected according to prior knowledge about the problem and the state of the system at the time, resulting in adaptive, effective and distributed cooperative patrolling.

In the future, beyond testing the approach using a multi-robot system in a real-world facility, it would be interesting to relax the assumption of perfect communication, testing the performance using only local interactions between robots within range.

References

[1] F. Pasqualetti, A. Franchi and F. Bullo, "On cooperative patrolling: optimal trajectories, complexity analysis, and approximation algorithms". In *IEEE Transactions on Robotics*, 28 (3), pp. 592-606, June, 2012.

[2] D. Portugal and R.P. Rocha, "Multi-Robot Patrolling Algorithms: Examining Performance and Scalability". In *Advanced Robotics Journal*, 27 (5), pp. 325-336, March, 2013.

[3] H. Santana, G. Ramalho, V. Corruble and B. Ratitch, "Multi-Agent Patrolling with Reinforcement Learning". In Proc. of the *Int. Conf. on Aut. Agents and Multiagent Sys.*, Vol. 3, New York, 2004.

[4] J. Marier, C. Besse and B. Chaib-draa, "Solving the Continuous Time Multiagent Patrol Problem". In Proc. of the *Int. Conf. on Robotics and Automation (ICRA'10)*, Anchorage, Alaska, USA, May, 2010.

[5] X. Chen and T.S. Yum, "Patrol Districting and Routing with Security Level Functions". In Proc. of the *Int. Conf. on Systems, Man and Cybernetics (SMC'2010)*, pp. 3555-3562, Istanbul, TK, Oct. 2010.

[6] L. Iocchi, L. Marchetti and D. Nardi, "Multi-Robot Patrolling with Coordinated Behaviours in Realistic Environments". In Proc. of the *Int. Conf. on Intelligent Robots and Systems (IROS'2011)*, pp. 2796-2801, San Francisco, CA, USA, September 25-30, 2011.

[7] L. Panait and S. Luke, "Cooperative Multi-Agent Learning: The State of the Art", In *Journal of Autonomous Agents and Multi-Agent Systems*, 11(3), pp. 387-434, November 2005.