

Cooperative multi-robot patrol with Bayesian learning

David Portugal¹  · Rui P. Rocha¹

Received: 24 June 2014 / Accepted: 19 September 2015 / Published online: 13 October 2015
© Springer Science+Business Media New York 2015

Abstract Patrolling indoor infrastructures with a team of cooperative mobile robots is a challenging task, which requires effective multi-agent coordination. Deterministic patrol circuits for multiple mobile robots have become popular due to their exceeding performance. However their predefined nature does not allow the system to react to changes in the system's conditions or adapt to unexpected situations such as robot failures, thus requiring recovery behaviors in such cases. In this article, a probabilistic multi-robot patrolling strategy is proposed. A team of concurrent learning agents adapt their moves to the state of the system at the time, using Bayesian decision rules and distributed intelligence. When patrolling a given site, each agent evaluates the context and adopts a reward-based learning technique that influences future moves. Extensive results obtained in simulation and real world experiments in a large indoor environment show the potential of the approach, presenting superior results to several state of the art strategies.

Keywords Distributed systems · Multi-robot patrol · Multi-agent learning · Security

1 Introduction

In the past several years, evident advances in mobile robotics have occurred and roboticists have increasingly turned

to Artificial Intelligence (AI) techniques to endow robots with perception, reasoning, planning and learning capabilities. Even though single agent solutions are still one step ahead of general multi-agent solutions, with the robustness inherited by distributed AI, multi-agent systems have been increasingly proposed, providing tools for the development of complex systems and mechanisms for coordination of the behavior of independent agents (Stone and Veloso 2000).

Considerable scientific work presented recently span across the boundaries of Robotics and AI, and in the particular case of multi-robot systems (MRS), these works are usually verified through multi-agent simulations or controlled test scenarios. However, recent attempts to employ robot teams in real world environments have become evident, e.g. as described in the works of Iocchi et al. (2011), Pippin et al. (2013), Agmon et al. (2012), and more. Successful examples of solutions that have proliferated in public places include automated guided vehicles (AGVs)¹ and the Santander Interactive Guest Assistants (SIGA),² and teams of multiple robots have been increasingly used in military and security applications, taking advantage of space distribution, parallelism, task decomposition and redundancy. In this context, there have been several advances in multi-robot patrolling and coverage, map learning, graph-exploration and networked robots (Parker 2008).

Security applications are a fundamental task with unquestionable impact on society. Combining progress witnessed at the behavior-level with the technological evolution observed in the last decades, it becomes clear that robot assistance can be a valuable resource in surveillance missions. In this

✉ David Portugal
davidbsp@isr.uc.pt; davidbspportugal@sapo.pt

Rui P. Rocha
rprocha@isr.uc.pt

¹ Institute of Systems and Robotics, University of Coimbra, 3030-290 Coimbra, Portugal

¹ <http://www.cybercars.org>.

² <http://www.ydreamsrobotics.com/projects>.

article, we propose a distributed multi-robot solution for indoor patrol. We believe that employing teams of robots for active surveillance tasks has several advantages over, for instance, a camera-based passive surveillance system. Robots are mobile and have the ability to travel in the field, collect environmental samples, act or trigger remote alarm systems and inspect places that can be hard for static cameras to capture. These capabilities free human operators from executing tedious, repetitive or monotonous tasks, e.g. mine clearing or patrolling in hazardous environments, enabling them to be occupied in nobler tasks like monitoring the system from a safer location (Murphy 2004) and taking advantage of robots' expendability.

The proposed multi-robot patrolling solution is based on Bayesian reasoning and learning agents, which can adapt to the dynamics of the system. We firstly conduct simulation experiments and then validate our results in a large real world environment. We show that the approach can be applied in different types of environment, independent of its topology; withstands failures in robotic patrol units; can be performed with heterogeneous robots; and accomplishes exceeding performance. Therefore, its potential is shown as a solution for real world MRS.

In the next section, a literature review is conducted and the contributions of the article are described. Afterwards, the multi-robot patrolling problem is defined and the performance metric is presented. Section 4 describes the distributed multi-robot patrolling strategy proposed in this article. Section 5 presents the results obtained both through simulations and hardware experiments, as well as a discussion of the results obtained. Finally, the article ends with conclusions and open issues for future research.

2 Related work

To patrol is literally defined as “the activity of going around or through an area at regular intervals for security purposes” (Vocabulary.com 2015). The multi-robot patrolling problem (MRPP) requires coordination of agents' decision-making with the ultimate goal of achieving optimal group performance, while visiting every position in the environment (or at least those that need surveillance). Additionally, it aims at monitoring infrastructures, obtaining information, searching for objects, detecting anomalies and clearing areas in order to guard the grounds from intrusion. Consequently, performing a patrolling mission with a team of any given number of autonomous and cooperative robots distributed in space represents a complex challenge. We propose to solve the area patrolling problem with a team of arbitrarily many robots, focusing on intelligent strategies for coordination of the team in order to visit all the surveillance points that need vigilance inside a target area,

as effectively as possible. Thus in this context, we consider the problem of border/perimeter patrol (Elmaliach et al. 2008; Marino et al. 2013), and adversarial patrol (Basilico et al. 2009; Agmon et al. 2011a) out of the scope of this work.

Despite its high potential utility in security applications, only recently has the MRPP been rigorously addressed. The environment to patrol is commonly abstracted through a navigation graph (cf. Fabrizi and Saffiotti 2000) and numerous works explore graph theory tools like spanning trees Fazli et al. (2013), Gabriely and Rimon (2001) or graph partitioning (Sak et al. 2008; Stranders et al. 2012) to compute minimal-cost cycles that assign efficient routes for each robot in the patrolling mission. Auctions and market-based coordination are also popular among multi-robot patrolling literature as in the case of Pippin et al. (2013) and Poulet et al. (2012), where agents bid to exchange vertices of the patrol graph to increase overall patrol performance. Numerous other concepts have already been explored, such as task allocation (Sempé and Drogoul 2003), artificial forces (Sampaio et al. 2010; Jansen and Sturtevant 2008), Gaussian processes theory (Marino et al. 2012), evolutionary algorithms (Aguirre and Taboada 2012), linear programming modeling (Keskin et al. 2012), and others.

Most contributions to the literature focus on presenting strategies to patrol any given environment with a team of mobile agents, the majority verified through simulations. Nevertheless, several important theoretical contributions to the MRPP have been presented and using the idleness criterion (cf. Sect. 3) it has been shown that the problem is NP-Hard (Chevalyere 2004; Ruan et al. 2005). Based on a topological representation of the environment and using global/centralized information, it has been proved previously that optimal patrolling could eventually be obtained for the single robot case by solving the Traveling Salesman Problem (TSP) (Chevalyere 2004), a classical computationally complex problem. As for the multi-agent situation, the problem is even more complex and optimal performance depends on the environment topology and the number of robots. However, the literature commonly reports that theoretically superior performance can be obtained either by optimal k -way graph partitioning, especially for high number of agents or graphs with unbalanced edges (Pasqualetti et al. 2012; Agmon et al. 2011b); or having all robots following the same TSP cycle, equally distributed in time and space, especially for low number of agents or balanced graphs (Chevalyere 2004; Smith and Rus 2010). Both the TSP and the graph partitioning problem are NP-hard. Therefore solving these problems is non-trivial, particularly in sparse topologies, which is the case of most real world environments. For instance, TSP cycles may not even exist in such situations (recurrent in non-complete

graphs) and the tour should be reduced to a minimum cost closed walk in the graph (Portugal et al. 2014). Additionally, these solutions lead to predefined trajectories for all robots. Commonly, these solutions cannot adapt to non-standard situations (e.g. robot failures or heterogeneous agents with different speed profiles) without specific recovery behaviors. Furthermore, such recovery behaviors need to be triggered by a centralized entity with global knowledge about the system. Another disadvantage is that deterministic patrols allow an external observer to easily apprehend the trajectory pattern. Thus, we turn to probabilistic strategies with distributed intelligence and autonomous decision-making robots. Besides being non-deterministic and much less predictable, these strategies do not have a central point of failure and can potentially scale to larger teams.

The creation of adaptive behaviors that allows agents to learn how to effectively patrol a given scenario have shown to be extremely promising. Moreover, such adaptability allows the agents to face changes in the system or in the scenario. Nevertheless, the use of such techniques is far from being straightforward. Certain works in this field have adopted machine learning methods. For instance, the pioneering approach of Santana et al. proposed to model the MRPP as a Q-learning problem in an attempt to enable automatic adaptation of the agents' strategies to the environment (Santana et al. 2004). In brief, agents have a probability of choosing an action from a finite set of actions, having the goal of maximizing a long-term performance criterion, in this case, node idleness. Two reinforcement learning techniques, using different communication schemes were implemented and compared to non-adaptive architectures. Although not always scoring the best results, adaptive strategies were generally superior to other compared strategies in most of the simulation experiments reported. The main attractive characteristics in this work are distribution (no centralized coordination is assumed) and the adaptive behavior of agents, which is usually highly desirable in this domain. Lauri and Koukam (2014) built upon the work of Santana et al. by redefining the state space and reward function associated to each agent, and were able to improve the method in specific situations, depending on graph topology and number of agents. Similarly, Ishiwaka et al. (2003) proposed reinforcement learning to predict the location of teammates as well as the movement direction to a common target. In another surveillance application scenario, Ahmadi and Stone (2006) proposed an adaptive solution for area sweeping tasks based on decentralized partitioning and learning behaviors. This was verified in a small experimental arena with two AIBO robots endowed with cameras. However, as the robots had a high field of view, the focus was not primarily on planning exhaustive surveillance routes, but more on continuously moving the agents

to positions where they were able to observe from distance.

Alternatively to reinforcement learning and adaptive behaviors some strategies have followed stochastic approaches that benefit from probabilistic decision-making to overcome the deterministic nature of classical patrolling applications. For instance, in Marier et al. (2010), the patrolling problem is cast as a multi-agent Markov decision process, where reactive and planning-based techniques are compared. The authors concluded that both perform similarly, with the latter being slightly superior in general, since it looks further ahead than the former, which is purely local. However, the reactive technique runs much faster, suggesting that a simple and computationally cheaper approach can be used in many applications, instead of more complex strategies which only perform slightly better. Similarly to reactive agents, swarm-based solutions to the patrolling problem have been studied in Yanovski et al. (2003), Baglietto et al. (2009) and Cannata and Sgorbissa (2011), where it is assumed that agents have limited computational power, communication abilities and memory storage. Interesting collective patrol behavior emerges using these type of approaches.

In this article, a new distributed and adaptive approach for multi-robot patrol is presented. 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 effective coordination between patrolling agents. Experimental results illustrate the advantages of using the proposed distributed technique, when compared to several state of the art strategies in simulation, and the well-known TSP solution in real world results.

2.1 Contributions

To summarize, the contributions of the work to the state of the art are as follows:

- We propose a new probabilistic, distributed and scalable approach to solve the MRPP, whose effectiveness is attested in the experiments conducted;
- We employ Bayesian decision-making together with adaptive learning in the context of the MRPP for the first time, as far as our knowledge goes, yielding intelligent patrolling behavior within the robot team;
- We show the advantages of using the proposed algorithm by assessing its performance and effectiveness using realistic simulations in Stage/ROS (Quigley et al. 2009), and comparing these results with those reported in Portugal and Rocha (2013a) where several patrolling approaches were tested in the same conditions;

- We implement the approach in a large indoor scenario with up to six physical robots, and compare it with the classical multi-agent TSP approach, which is theoretically optimal using one robot and nearly-optimal for generic teams of robots;
- Fault-tolerance and scalable behaviors are verified in real world experiments, as well as the possibility to use the approach in heterogeneous teams of robots.

In the past, we have shown that naive Bayesian decision models have the potential to tackle the patrolling problem, by presenting methods based on conditional probability distributions and without agent learning behaviors (Portugal and Rocha 2013b). In this work, we intend to go much further by presenting a new strategy that rectifies weaknesses previously identified. We describe a probabilistic multi-robot patrolling strategy, where a team of concurrent learning agents adapt their moves to the state of the system at the time, using Bayesian decision rules based on the robot's accumulated experience and distributed intelligence. When patrolling a given site, each agent evaluates the context and adopts a reward-based learning technique that influences future moves. Learning of the likelihood distributions involved in the robots' decisions is done online, along the mission, and robots are constantly updating their prior information, i.e. their past experience.

This article is a culmination of two preliminary works (Portugal et al. 2013a, b), yet it clearly distinguishes itself from the previous publications by presenting an extended literature survey and a more complete Bayesian model, which includes vertex lookahead (cf. Sect. 4.4) in order to solve the MRPP using multi-agent concurrent reward-based learning. In addition, the implementation of a fully distributed system is described, carrying out completely new experiments both in simulation and in a real world indoor infrastructure and discussing in detail the implementation aspects. Also, unlike the two previous works, we assess through real world experiments the adaptability, scalability and fault-tolerant nature of the proposed approach.

3 Problem definition

In this article, the problem of efficiently patrolling inside a given area with an arbitrary number of robots is studied. Agents are assumed to have an a priori representation of the environment and through a graph extraction algorithm (Portugal and Rocha 2013c), they are able to acquire an undirected, connected and metric navigation graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, which enables them to assess the topology of its surroundings. \mathcal{G} is composed of vertices $v_i \in \mathcal{V}$ and edges $e_{i,j} \in \mathcal{E}$. Each vertex represents a specific location that must be visited

regularly and each edge represents the connectivity between these locations, having a weight $|e_{i,j}|$ defined by the metric distance between v_i and v_j . $|\mathcal{V}|$ and $|\mathcal{E}|$ represent the cardinality of the set \mathcal{V} and \mathcal{E} respectively.

Informally, a good strategy is one that minimizes the time lag between two passages to the same place and for all places. Thus, the MRPP can be reduced to coordinate robots in order to visit frequently all $v_i \in \mathcal{G}$, ensuring the absence of atypical situations with regard to a predefined optimization criterion.

In order to address and compare the performance of different patrolling algorithms, it is important to establish an evaluation metric. Diverse criteria have been previously proposed to assess the effectiveness of multi-robot patrolling strategies. Typically, these are based on the idleness of the vertices (Almeida et al. 2004), the frequency of visits (Agmon et al. 2011b) or the distance traveled by agents (Iocchi et al. 2011). In this work, the first one has been considered, given that it measures the elapsed time since the last visit from any agent in the team to a specific location. The idleness metric uses time units, which is particularly intuitive, e.g. in the analysis of how long vertices have not been visited for, or comparing different patrolling strategies. In the following equations, we define important variables used in the remaining sections of the article.

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

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

wherein t_l corresponds to the last time instant when the vertex v_i was visited by any robot of the team. Consequently, the average idleness of a vertex $v_i \in \mathcal{V}$ in time step t is defined as:

$$\overline{\mathcal{I}}_{v_i}(t) = \frac{\overline{\mathcal{I}}_{v_i}(t_l) \cdot C_i + \mathcal{I}_{v_i}(t)}{C_i + 1}, \quad (2)$$

where C_i represents the number of visits to v_i . Considering now $\overline{\mathcal{I}}_{\mathcal{V}}$ as the set of the average idlenesses of all $v_i \in \mathcal{V}$, given by:

$$\overline{\mathcal{I}}_{\mathcal{V}}(t) = \{\overline{\mathcal{I}}_{v_1}(t), \dots, \overline{\mathcal{I}}_{v_i}(t), \dots, \overline{\mathcal{I}}_{v_{|\mathcal{V}|}}(t)\}, \quad (3)$$

the maximum average idleness of all vertices $\max(\overline{\mathcal{I}}_{\mathcal{V}})$ in time step t is defined as:

$$\max(\overline{\mathcal{I}}_{\mathcal{V}})(t) = \max\{\overline{\mathcal{I}}_{v_1}(t), \dots, \overline{\mathcal{I}}_{v_i}(t), \dots, \overline{\mathcal{I}}_{v_{|\mathcal{V}|}}(t)\}. \quad (4)$$

For simplicity of notation, let us omit (t) whenever timing is not relevant. Finally, in order to obtain a generalized measure, the average idleness of the graph \mathcal{G} ($\overline{\mathcal{I}}_{\mathcal{G}}$) is defined as:

$$\overline{\mathcal{I}}_{\mathcal{G}} = \frac{1}{|\mathcal{V}|} \sum_{i=1}^{|\mathcal{V}|} \overline{\mathcal{I}}_{v_i}, \tag{5}$$

A similar assumption to other works in the literature (cf. Chevaleyre 2004; Almeida et al. 2004) is taken in the beginning of the experiments, where for all $v_i \in \mathcal{V}$, $\mathcal{I}_{v_i}(0) = 0$, as if every vertex had just been visited at the beginning of the mission. Consequently, there is a transitory phase in which the $\overline{\mathcal{I}}_{\mathcal{G}}$ values tend to be low, not corresponding to the reality in steady-state, as will be seen in the results section. For this reason, the final $\overline{\mathcal{I}}_{\mathcal{G}}$ value is measured only after convergence in the stable phase.

Considering a patrol path as an array of vertices of \mathcal{G} , the multi-robot patrolling problem with an arbitrary team of R robots can be described as the problem of finding a set of R paths \mathbf{x} that visit all vertices $v_i \in \mathcal{V}$ of \mathcal{G} , with the overall team goal of minimizing $\overline{\mathcal{I}}_{\mathcal{G}}$:

$$f = \operatorname{argmin}_{\mathbf{x}} (\overline{\mathcal{I}}_{\mathcal{G}}), \tag{6}$$

By finding:

$$\mathbf{x} = \{x_1, \dots, x_r, \dots, x_R\}, \tag{7}$$

Such that:

$$\begin{aligned} x_r &= \{v_a, v_b, \dots\}, \\ v_a, v_b, \dots &\in \mathcal{V}, \\ 1 \leq r \leq R, R &\in \mathbb{N}, \end{aligned} \tag{8}$$

Subject to:

$$\forall v_i \in \mathcal{V}, \exists x_r \in \mathbf{x} : v_i \in x_r. \tag{9}$$

Note that x_r represents the patrolling path of robot r , which has an arbitrary dimension that depends on each robot’s decisions and v_a, v_b, \dots are vertices that are successively connected in \mathcal{V} . The union of all R paths should be equal to the full set of vertices \mathcal{V} , and different paths x_r, x_s may have common vertices. In this work, the route x_r of each robot is computed online by each autonomous agents in order to adapt to the system’s needs.

4 Bayesian model for multi-robot patrolling

In a previous work of the authors (Portugal and Rocha 2013b), simple preliminary Bayesian-based techniques to tackle the MRPP were studied. Even though the results obtained were satisfactory, two main drawbacks were identified: a uniform prior distribution was adopted, assuming

that all decisions were equiprobable; and the likelihood distributions were immutable, representing a fixed function of random variables.

In this work, robots are endowed with increased intelligence, since the previous Bayesian inspired models are extended with likelihood reward-based learning and continued prior update. More specifically, the model represents the decision of moving from one vertex of the graph to another. For β neighbors of the current vertex v_0 , where $\beta = \operatorname{deg}(v_0)$,³ the model is applied β times. Each decision is considered independent and the agents have the ability to choose the action which has the greatest expectation of utility, weighted by the effects of all possible actions. Consequently, each robot’s patrol route is built progressively, at each decision step, adapting to the system’s needs, i.e. aiming at minimizing $\overline{\mathcal{I}}_{\mathcal{G}}$. In the following subsections, more details on the Concurrent Bayesian Learning Strategy (CBLS) to solve the MRPP are presented, and the pseudo-code of the approach, running on each individual robot of the team, is presented in Algorithm 1.

Algorithm 1: Concurrent Bayesian Learning Strategy (CBLS).

```

//Initialization of variables
All  $\mathcal{I}_{v_i}(t_0) \leftarrow 0$ ;
All  $C_i \leftarrow 0$ ;
All  $\theta_{i,j} \leftarrow \kappa$ ;

while true do
    add ( $v_k$  to  $x_r$ ); //current vertex  $v_k$ 
    forall  $v_i \in N_G(v_k)$  and  $v_i \notin \text{intended by teammate}$  do
         $P(\text{move}_i) \propto \omega \cdot \overline{\mathcal{I}}_{v_i} + (1-\omega) \cdot \max(\overline{\mathcal{I}}_{N_G(v_i)})$ ; //prior
         $P(\theta_{k,i}|\text{move}_i) \propto \theta_{k,i}$ ; //likelihood
         $P(\text{move}_i|\theta_{k,i}) \propto P(\text{move}_i) \cdot P(\theta_{k,i}|\text{move}_i)$ ; //posterior
     $H(\text{move}|\theta) \leftarrow - \sum_{i=1}^{|N_G(v_k)|} P(\text{move}_i|\theta_{k,i}) \log_2(P(\text{move}_i|\theta_{k,i}))$ ;
     $\mathcal{H}(\text{move}|\theta) \leftarrow H(\text{move}|\theta)/\log_2(|N_G(v_k)|)$ ; //normalized entropy

    //choose neighbor of  $v_k$  with highest posterior probability
     $v_{k+1} \leftarrow \operatorname{argmax}(P(\text{move}_i|\theta_{k,i}))$ ;
    send_msg (current:  $v_k$ , next:  $v_{k+1}$ );

    while move_robot to  $v_{k+1}$  do
        read_msg (arrivals, intentions);
        update ( $\mathcal{I}_{\mathcal{V}}(t)$ );
        update ( $C_{\mathcal{V}}$ );

    //robot reached  $v_{k+1}$ 
    forall  $v_i \in N_G(v_k)$  do
        compute  $S_{i,k}(C_i, \mathcal{I}_{\mathcal{V}}(t))$ ; //eq. 19
         $\gamma_{k,i} \leftarrow S_{i,k} \cdot (1 - \mathcal{H}(\text{move}|\theta))$ ; //reward
         $\theta_{k,i} \leftarrow \theta_{k,i} + \gamma_{k,i}$ ; //arc strength
     $v_k \leftarrow v_{k+1}$ ;

```

³ The degree (or valency) of a vertex of a graph $\operatorname{deg}(v_i)$, is the number of edges incident to the vertex.

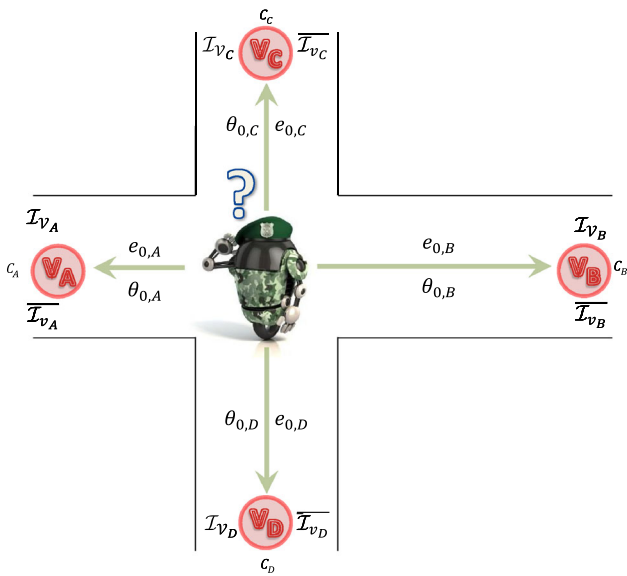


Fig. 1 Illustration of a patrolling decision instance. In this example, the robot may choose to visit one out of 4 neighbor vertices. Considering a generic neighbor v_x , I_{v_x} is the instantaneous idleness of v_x ; $\overline{I_{v_x}}$ is the average idleness of v_x ; $e_{0,x}$ is the edge that connects the current vertex v_0 and v_x , with an edge weight $|e_{0,x}|$; C_x gives the number of visits to v_x and $\theta_{0,x}$ represents the arc strength of traveling from v_0 to v_x

4.1 Distribution modeling

As stated before, when reaching a vertex v_0 of the navigation graph \mathcal{G} , each robot is faced with a decision stage, where it must decide the direction it should travel next (cf. Fig. 1). To that end, two fundamental random variables are defined. The first one is boolean and simply represents the act of moving (or not) to a neighbor vertex:

$$move_i \in \{true, false\}, \tag{10}$$

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

$$\theta_{0,i} \in \mathbb{R}; \quad 0, i \in \mathbb{N}_0. \tag{11}$$

Note that in \mathcal{G} , an edge $e_{j,k}$ represents a connection from v_j to v_k and vice versa. An edge $e_{j,k}$ has an edge cost or weight $|e_{j,k}| = |e_{k,j}|$ given by the distance between the two vertices. Thus, by definition \mathcal{G} is undirected, since $e_{j,k} = e_{k,j}$. Nevertheless, the term “arc” in this new variable is used intentionally, since it is affected by the direction of traveling. In a situation where an agent is at v_j , it will look for the suitability of traveling to v_k , given by $\theta_{j,k}$. Under those circumstances, the suitability of traveling in the opposite direction is not relevant, and $\theta_{j,k} \neq \theta_{k,j}$. As a consequence, the set of all $\theta_{i,j}$ variables, denoted as θ , has a population of $2|\mathcal{E}|$, where $|\mathcal{E}|$ is the cardinality of the set of edges \mathcal{E} of \mathcal{G} , and informally,

higher values of *arc strength* lead to the edge being traversed more often in the specified direction. Note that this variable is chosen such that the collective patrolling strategy can adapt to any environment topology.

In this work, agents compute the degree of belief (i.e. a probability) of moving to a vertex v_i , given the *arc strengths*, by applying Bayes rule:

$$P(move_i|\theta_{0,i}) = \frac{P(move_i)P(\theta_{0,i}|move_i)}{P(\theta_{0,i})}. \tag{12}$$

The posterior probability $P(move_i|\theta_{0,i})$ is estimated via Bayesian inference from the prior $P(move_i)$ and likelihood $P(\theta_{0,i}|move_i)$ distributions. The denominator term is regarded as a normalization factor (Jansen and Nielsen 2007), being often omitted for the sake of simplicity.

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

$$P(move_i) = \frac{\overline{I_{v_i}}}{\sum_{k=1}^{\beta} \overline{I_{v_k}}}, \tag{13}$$

thus decisions of moving to vertices with higher values of average idleness have intuitively higher probability. During the patrol mission, robots are continuously visiting new places and the $\overline{I_V}$ values change over time. Each agent computes these values internally by tracking its own visits to \mathcal{V} and receiving messages from other teammates when they arrive to a new vertex. In order to make an informed decision, at each decision step, the agent updates the prior information through (13), just before adopting (12) to obtain a degree of belief of moving to a neighbor vertex v_i .

In addition to the prior distribution, it is also necessary to define the likelihood through a statistical distribution to model the *arc strength* $\theta_{0,i}$. In the patrolling problem, agents must visit all $v_i \in \mathcal{G}$, thus, theoretically, assigning a uniform value for every arc would not be unreasonable. However, in such a dynamic system, where the number of visits to different locations in the environment is permanently evolving, it is usually advantageous to avoid traversing certain edges at a given time and favoring the use of others, in order to improve performance. For example, it is highly undesirable that robots move along the same edges, as this is inefficient from the collective patrolling point of view. Also, moving in the same regions causes inter-robot interference (e.g. taking longer to reach their goals if they have to avoid collisions between them). Therefore, robots should coordinate themselves properly. Furthermore, task effectiveness is strongly related to the environment topology.

Hence, in the next subsection, a reward-based learning strategy to model and continually update the likelihood distribution is proposed in order to adapt to the system’s state according to previous decisions, having a high impact on the behavior of robots and aiming at optimizing the collective performance.

4.2 Multi-agent reward-based learning

In general, reward-based learning methods are quite attractive since agents are programmed through reward and punishments without explicitly specifying how the task is to be achieved (Panait and Luke 2005). In this work, Bayesian Learning is employed to estimate the likelihood functions. Being a cooperative multi-robot task with lack of centralized control, with decentralized and distributed information and asynchronous computation, multiple simultaneous learners (one per patrolling agent) are involved.

The concept of delayed reward with a 1-step horizon model is explored. Each agent chooses an action of moving from v_0 to a neighbor v_i , based on (12). After reaching v_i , the information on its neighborhood has changed, namely the instantaneous idlenesses have been updated, i.e. $\mathcal{I}_{v_i}(t) = 0$ and $\mathcal{I}_{v_0}(t) > 0$. Through information observed after making the move, a reward-based mechanism is used to punish or benefit the arcs involved in the decision to move from v_0 to v_i . This influences upcoming moves starting in v_0 , by introducing a bias towards arcs which ought to be visited in the future.

Henceforth, the reward-based learning method is explained. When the robot decides which one of the β neighbor vertices of v_0 is going to be visited next, each neighbor v_i will have an associated degree of belief given by the posterior probability. Therefore, it is possible to calculate the entropy⁴:

$$H(\text{move}|\theta) = - \sum_{i=1}^{\beta} P(\text{move}_i|\theta_{0,i}) \log_2(P(\text{move}_i|\theta_{0,i})), \tag{14}$$

which measures the degree of uncertainty involved in the decision taken, being chosen for this reason as the basis for the punish/reward mechanism. The confidence on the decision taken is inversely proportional to the entropy H . Therefore, we use the entropy to quantify the rewards and penalties to assign. In this way, larger rewards and penalties are awarded to decisions with higher confidence (lower entropy). Note, however, that distinct v_i have different $\text{deg}(v_i)$ and, as a result, β varies for each decision instant. Therefore, the entropy is normalized to assume values in $[0, 1]$:

⁴ Entropy is a general measure for the uncertainty of a belief. When applied to a discrete random variable, it evaluates to its shortest description, being as high as the variable’s uncertainty (Rocha et al. 2005).

$$\mathcal{H}(\text{move}|\theta) = \frac{H(\text{move}|\theta)}{\log_2(\beta)}. \tag{15}$$

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

$$\gamma_{0,i} = \mathcal{S}_{i,0}(C_i, \mathcal{I}_{\mathcal{V}}(t)) \cdot (1 - \mathcal{H}(\text{move}|\theta)), \tag{16}$$

with:

$$\mathcal{S}_{i,0} \in \{-1, 0, 1\}. \tag{17}$$

$\mathcal{S}_{i,0}$ gives the reward sign, providing a quality assessment which determines whether a penalty ($\mathcal{S} = -1$), a reward ($\mathcal{S} = 1$) or a neutral reward ($\mathcal{S} = 0$) should be given. As can be seen, this piecewise function uses up-to-date information, namely the number of visits to v_i , given by C_i , and the set of current instantaneous idlenesses $\mathcal{I}_{\mathcal{V}}(t)$. The sign of \mathcal{S} is obtained using the set of heuristic rules defined below, which are checked as soon as the agent reaches v_k . For that matter, it is necessary to define firstly the normalized number of visits to any neighbor vertex v_i :

$$\zeta_i = \frac{C_i}{\text{deg}(v_i)}. \tag{18}$$

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. The rules for assigning the sign $\mathcal{S}_{i,0}$ of the rewards are given by the following piecewise function definition:

$$\mathcal{S}_{i,0}(C_i, \mathcal{I}_{\mathcal{V}}(t)) = \begin{cases} -1, & \text{if } (\beta > 1) \wedge (\text{argmax}_{j \in N_{\mathcal{G}}(v_0)} \zeta_j = i) \wedge \\ & (|\text{argmax}_{j \in N_{\mathcal{G}}(v_0)} \zeta_j| = 1) \\ -1, & \text{if } (\beta > 1) \wedge (\text{argmax}_{j \in N_{\mathcal{G}}(v_0)} \zeta_j = i) \wedge \\ & (\forall k, |\text{argmax}_{j \in N_{\mathcal{G}}(v_0)} \zeta_j| = k \rightarrow \mathcal{I}_{v_i}(t) \leq \mathcal{I}_{v_k}(t)) \\ 1, & \text{if } (\beta > 1) \wedge (\text{argmin}_{j \in N_{\mathcal{G}}(v_0)} \zeta_j = i) \wedge \\ & (|\text{argmin}_{j \in N_{\mathcal{G}}(v_0)} \zeta_j| = 1) \\ 1, & \text{if } (\beta > 1) \wedge (\text{argmin}_{j \in N_{\mathcal{G}}(v_0)} \zeta_j = i) \wedge \\ & (\forall k, |\text{argmin}_{j \in N_{\mathcal{G}}(v_0)} \zeta_j| = k \rightarrow \mathcal{I}_{v_i}(t) \geq \mathcal{I}_{v_k}(t)) \\ 0, & \text{otherwise.} \end{cases} \tag{19}$$

with:

$$\beta = \text{deg}(v_0) = |N_{\mathcal{G}}(v_0)|, \tag{20}$$

$N_{\mathcal{G}}(v_0)$ represents the open neighborhood of v_0 , i.e. the set of adjacent vertices of v_0 . As such, the assignment of the sign $\mathcal{S}_{i,0}$ respects the following criteria:

- $S_{i,0} = -1$, 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.
- $S_{i,0} = 1$, 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.
- $S_{i,0} = 0$, 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 *arc strengths* $\theta_{0,i}$ are equal to a real positive number κ :

$$\forall \theta_{0,i} \in \theta, \theta_{0,i}(t_0) = \kappa. \quad (21)$$

As the mission evolves, the agent updates $\theta_{0,i}$ every time it visits a new vertex through:

$$\theta_{0,i}(t) = \theta_{0,i}(t-1) + \gamma_{0,i}(t). \quad (22)$$

Note that the larger the value of κ is set in (21), the less immediate influence the rewards received will have on $\theta_{0,i}$. In all experimental tests conducted in this work, $\kappa = 1.0$ 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 $\theta_{0,i}$:

$$P(\theta_{0,i} | move_i) = \frac{\theta_{0,i}}{\sum_j \sum_k \theta_{j,k}}, \quad (23)$$

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

4.3 Decision-making and multi-agent coordination

Having described how agents learn their likelihood distribution, it is necessary to address agent coordination to completely characterize the CBLS solution for patrolling tasks presented in this article. Being a concurrent learning approach, each agent is adapting its behavior via its own

learning process and has no control or knowledge of how other agents behave nor their internal state, i.e. they do not know their teammates' likelihood distribution $P(\theta_{0,i} | move_i)$ and do not predict their teammates' moves. This allows the reduction of complexity of the problem, however it is necessary to guarantee the coordination of robots.

In collective missions with a common goal, multi-agent coordination 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. Thereby, the asynchronous and distributed communication system that is used to inform teammates of the current vertex v_0 is augmented with the information of the vertex v_i chosen for the next move.

Therefore, by exchanging simple messages with teammates that only contain the robot ID r , the current vertex v_0 , and the next chosen vertex v_i , each robot can update the information about the state of the system, namely the idleness values, and decide its moves taking that information into account, as well as its progressively acquired experience. When agents are close by, they can coordinate by inspecting if a teammate has already expressed intention to move to a given vertex v_i in its local neighborhood and if so, remove it from its decision.

Finally, the decision-making process of the agent consists of choosing the move from v_0 to the neighbor vertex v_j with the maximum probability among all possible decisions:

$$move_j = true : j = \underset{i \in N_G(v_0)}{\operatorname{argmax}} P(move_i | \theta_{0,i}) \quad (24)$$

4.4 Incorporating vertex look-ahead

Up to now, we have only mentioned agents that decide their moves upon knowledge of the local 1-step neighborhood. While it is true that expanding the search horizon increases the complexity of the problem, it is also true that it may lead to a higher number of correct decisions, since agents deliberate with additional information.

Therefore, we also propose to study a variation of the model presented before, by incorporating vertex look-ahead. Note however that, in this context, each agent is an independent concurrent learner and does not exchange its local beliefs about the system, only its current vertex and the intended vertex to visit in its immediate neighborhood. As a consequence, if agents plan too far ahead, their intentions may come into conflict, i.e. they may be planning their moves to inadvertently reach the same sites, which would be highly inefficient from the standpoint of the patrolling mission. The probability of such conflicts increases with the number of steps in the planning horizon. For this reason, and to manage the complexity of the system, we explore the concept of 2-

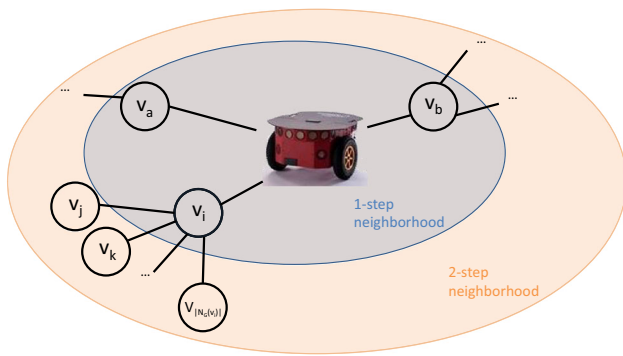


Fig. 2 Illustration of the 2-step look-ahead in the navigation graph

step look-ahead, aiming to further improve the performance of the CBLS approach.

Prior to making a decision, agents will now look not only for the average idleness of the vertices v_i in its immediate neighborhood, but also for the neighbors of these ($v_j, v_k, \dots, v_{|N_G(v_i)|}$) to find the maximum average idleness among them, as shown in Fig. 2.

Accordingly, vertex look-ahead is incorporated in the prior distribution, by modifying (13):

$$P(move_i) = \frac{\omega \cdot \overline{\mathcal{I}_{v_i}} + (1 - \omega) \cdot \max(\overline{\mathcal{I}_{N_G(v_i)}})}{\sum_{k=1}^{|V|} \omega \cdot \overline{\mathcal{I}_{v_k}} + (1 - \omega) \cdot \max(\overline{\mathcal{I}_{N_G(v_k)}})}, \quad (25)$$

where:

$$\omega \in [0, 1], \quad (26)$$

$$\max(\overline{\mathcal{I}_{N_G(v_i)}}) = \max\{\overline{\mathcal{I}_{v_j}}, \overline{\mathcal{I}_{v_k}}, \dots, \overline{\mathcal{I}_{v_{|N_G(v_i)|}}}\}, \quad (27)$$

$$v_j, v_k, \dots, v_{|N_G(v_i)|} \in N_G(v_i). \quad (28)$$

The factor ω assigns weights to the observations on the immediate neighborhood and on the 2-step neighborhood. When $\omega = 1.0$, no look-ahead is considered and (25) is equivalent to (13). The lower the ω value is, the higher the weight of observations beyond the 1-step neighborhood are. However, intuitively we should set ω for values above 0.5 (and below 1.0), because the information on the immediate 1-step neighborhood is always reliable and up-to-date when the decision is made, while the information on the 2-step neighborhood may change when the robot reaches v_i . Note also, that when reaching v_i , the robot has another decision instance and in rare cases may deliberate contrarily to the initial plan, in case the settings in its surroundings have changed in the meantime. The influence of the ω factor is analyzed in the preliminary simulation results presented in the next section, so as to establish an appropriate value to use in experiments with physical robots.

5 Results and discussion

In this section, experimental results to assess the performance of CBLS are presented and discussed. Section 5.1 reveals the outcome of simulations experiments in three environments with distinct graph connectivity, enabling to analyze the effect of ω in the mission and comparing our approach with other state of the art multi-robot patrolling strategies.

In the following subsection, experimental results with teams of physical robots in a large indoor scenario are discussed. CBLS is compared to the TSP cyclic algorithm, which finds the shortest closed walk on a graph by solving the TSP (Fazli et al. 2013), in non-complete graphs as the one presented in the experiments. This benchmarking algorithm is optimal for the single robot case and near optimal for multi-robot scenarios, being perfectly suited for a comparative analysis.

5.1 Simulation results

In the preliminary simulation experiments, the main goal is to study the effect of the ω parameter in the vertex look-ahead method and to enable comparisons with other strategies in the literature. To that end, the three environments illustrated in Fig. 3 have been used to test the approach with different team sizes of $R = \{1, 2, 4, 6, 8, 12\}$ robots. The three illustrated topologies present different algebraic connectivity or Fiedler value λ (Fiedler 1973), a well-known metric of the connectivity of a graph given by the smallest non-zero eigenvalue of the graph's Normalized Laplacian matrix. These topologies were used in Portugal and Rocha (2013a), where they were classified as: lowly (A), mildly (B) and highly (C) connected, having a Fiedler value of $\lambda_A = 0.0080$, $\lambda_B = 0.0317$ and $\lambda_C = 0.1313$, respectively. In this work, these are again adopted to enable comparative analysis against other MRPP strategies. Note, for example, that the graph of environment A (\mathcal{G}_A) has several dead-ends, i.e. vertices with degree 1. On the other hand \mathcal{G}_C is the most connected of the three, with several vertices with degree 4.

While collecting results in different scenarios, the same simulation setup was used for all strategies.

A recognized simulator with realistic modeling was chosen: the Stage 2D multi-robot simulator (Vaughan 2008). Stage considers the robot's dynamics and together with ROS (Quigley et al. 2009), adopted to program the robots, was used to implement the experiments.

The graph information of the environment is loaded by every robot in the beginning of each simulation, which then runs the described algorithm. Robots navigate safely in the environment by heading towards their goals while avoiding collisions through the use of ROS navigation stack and an adaptive Monte Carlo localization approach. Additionally, all

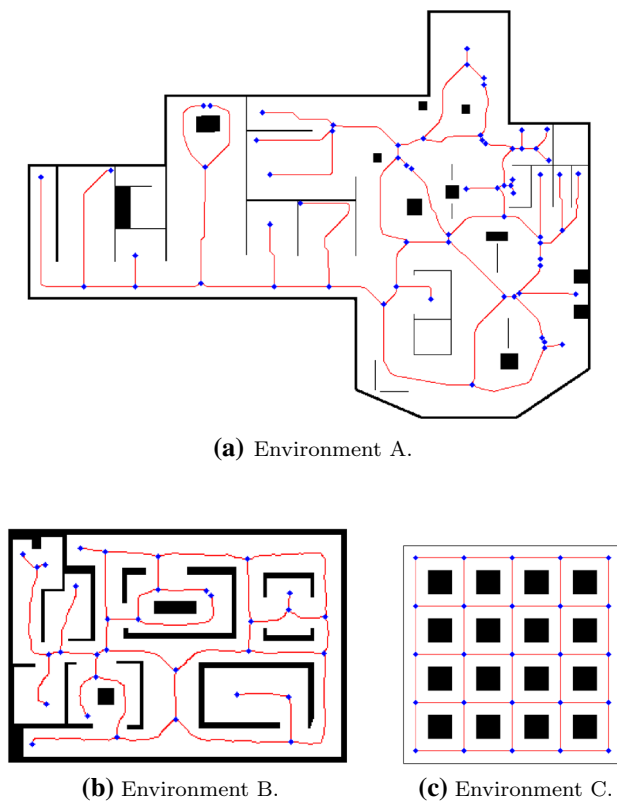


Fig. 3 Environments used in the experiments with respective topological map

robots have non-holonomic constraints and move at the same nominal speed, traveling at a maximum velocity of 0.2 m/s.

All the experiments conducted in this article were repeated three times for each setting, and they all respect the same stopping condition determined by 4 complete patrolling cycles i.e. after every $v_i \in \mathcal{G}$ has been visited at least four times. This stopping condition proved to be adequate since $\overline{\mathcal{I}}_{\mathcal{G}}$ always converged, i.e. all results were obtained in the steady-state phase of the patrolling mission, as shown by the marginal mean difference in $\overline{\mathcal{I}}_{\mathcal{G}}$ values between the third and fourth cycle of all experiments, which was around 2.5 %.

Tables 1, 2 and 3 present the mean performance results of the distributed patrolling strategy CBLs, described in this article, given by $\overline{\mathcal{I}}_{\mathcal{G}}$ in seconds, and using the environments of Fig. 3 with $\omega = 1.0$, $\omega = 3/4$ and $\omega = 2/3$. The same results are presented in more detail for each individual case in Tables 13, 14, 15, 16, 17, 18, 19, 20, 21, in Appendix. Results prove the intuition that superior performance can be obtained with $0.5 < \omega < 1.0$, seeing as, in general, the best results were obtained for $\omega = 3/4$. It is also interesting to verify that the results without look-ahead ($\omega = 1.0$) are always inferior to those obtained with vertex look-ahead. This confirms that looking further beyond the local neighborhood has the potential to increase the number of correct decisions of each agent and improve team performance. Being a complex prob-

Table 1 Final $\overline{\mathcal{I}}_{\mathcal{G}}$ values (in seconds) using CBLs in environment A with different ω

R	CBLs (map A)		
	$\omega = 1.0$	$\omega = 3/4$	$\omega = 2/3$
1	1445.45 ± 656.81	1432.21 ± 572.17	1323.55 ± 485.91
2	707.92 ± 270.61	686.23 ± 239.67	701.73 ± 242.13
4	402.28 ± 170.81	357.94 ± 124.54	371.55 ± 135.92
6	261.36 ± 123.78	231.32 ± 84.76	251.07 ± 89.83
8	188.11 ± 91.77	167.45 ± 69.00	181.53 ± 74.48
12	168.85 ± 77.41	120.91 ± 47.61	133.92 ± 68.54

Bold values represent the best values in each row

Table 2 Final $\overline{\mathcal{I}}_{\mathcal{G}}$ values (in seconds) using CBLs in Environment B with different ω

R	CBLs (map B)		
	$\omega = 1.0$	$\omega = 3/4$	$\omega = 2/3$
1	1249.45 ± 481.85	1110.66 ± 280.69	1199.97 ± 375.05
2	575.06 ± 166.58	557.00 ± 131.75	554.00 ± 130.22
4	284.88 ± 63.40	275.96 ± 65.25	283.35 ± 66.21
6	197.33 ± 53.83	192.77 ± 40.56	194.25 ± 44.14
8	143.36 ± 30.69	142.74 ± 30.49	142.50 ± 33.52
12	108.22 ± 39.85	96.48 ± 29.69	94.16 ± 41.22

Bold values represent the best values in each row

Table 3 Final $\overline{\mathcal{I}}_{\mathcal{G}}$ values (in seconds) using CBLs in Environment C with different ω

R	CBLs (map C)		
	$\omega = 1.0$	$\omega = 3/4$	$\omega = 2/3$
1	701.98 ± 63.96	684.42 ± 63.23	688.79 ± 61.61
2	359.09 ± 44.09	355.65 ± 38.09	359.38 ± 30.33
4	183.91 ± 18.76	175.40 ± 18.54	182.55 ± 17.25
6	126.62 ± 14.42	121.36 ± 10.33	125.68 ± 11.97
8	96.81 ± 12.68	90.25 ± 7.90	91.84 ± 8.13
12	75.43 ± 15.58	63.00 ± 12.27	64.19 ± 13.14

Bold values represent the best values in each row

lem, variables such as graph connectivity and team size have an important impact on the performance of every patrolling strategy. Thus, it was plainly anticipated that a certain fixed value of ω would not be better for all configurations. Nevertheless, in the experiments conducted, $\omega = 3/4$ was superior in 14 configurations out of 18. In view of this, we shall consider $\omega = 3/4$ in the experiments with physical robots.

Using findings from our previous works, Tables 4, 5 and 6 were built. Therein, tests in the same three environments and benchmarking with several state of the art patrolling approaches were conducted, and we were able to compare the performance of CBLs against 7 other state of the art

Table 4 Final $\overline{\mathcal{I}_G}$ values (in seconds) using different state of the art strategies on Environment A

R	Map A							
	CR	HCR	HPCC	CGG	MSP	GBS	SEBS	CBLs
1	1734.09	1962.42	1740.37	1717.36	1704.36	1718.93	1703.68	1432.21
2	843.93	1146.27	791.20	845.49	930.04	836.05	812.68	686.23
4	433.38	652.84	434.11	451.70	476.92	464.18	438.16	357.94
6	367.11	506.90	377.73	348.46	381.97	353.15	329.18	231.32
8	271.70	442.39	361.62	288.72	253.19	295.58	251.91	167.45
12	287.14	412.65	352.79	265.47	183.74	253.89	226.90	120.91

Table 5 Final $\overline{\mathcal{I}_G}$ values (in seconds) using different state of the art strategies on Environment B

R	Map B							
	CR	HCR	HPCC	CGG	MSP	GBS	SEBS	CBLs
1	1315.79	1283.59	1235.67	1347.30	1401.80	1267.26	1277.16	1110.66
2	675.44	654.61	670.44	675.64	749.42	708.82	671.18	557.00
4	363.46	373.45	298.77	335.45	375.15	351.19	339.93	275.96
6	238.57	273.60	254.96	234.18	248.92	275.98	230.39	192.77
8	198.90	217.38	225.44	172.39	185.28	206.19	197.03	142.74
12	172.40	255.62	212.30	143.94	–	145.89	118.73	96.48

Table 6 Final $\overline{\mathcal{I}_G}$ values (in seconds) using different state of the art strategies on Environment C

R	Map C							
	CR	HCR	HPCC	CGG	MSP	GBS	SEBS	CBLs
1	715.30	714.23	737.93	767.25	766.41	670.29	676.30	684.42
2	353.06	351.15	358.45	385.09	423.60	343.89	338.97	355.65
4	193.30	186.59	188.03	200.53	209.82	182.89	167.16	175.40
6	141.68	138.64	135.74	142.94	148.09	147.66	125.06	121.36
8	104.00	108.45	118.75	113.71	95.22	116.14	103.45	90.25
12	101.82	105.64	118.36	94.35	–	90.42	70.33	63.00

approaches,⁵ using the $\overline{\mathcal{I}_G}$ metric. For each different strategy, all the robots started in the same positions when executing a patrolling mission in a given environment and with a given team size. This was carried out to promote a fair comparison between all strategies. For further details on the various strategies and tests previously conducted, the interested reader should refer to [Portugal and Rocha \(2013a, b\)](#).

In general, CBLs clearly outperforms the other 7 state of the art approaches. This is the case even when no look-ahead is considered. The difference is apparent especially for lowly (environment A) and mildly (environment B) connected environments. In the highly connected grid environment (map C), vertices have, in general, greater degree and there are usually many alternative routes to reach a given goal. This fact makes other state of the art strategies perform better in this case than in less connected (more typical) environments. Nevertheless, CBLs presents a strong performance for smaller team sizes of 1, 2 and 4 robots; and outperforms the other strategies for

larger team sizes of 6, 8 and 12 robots, which suggests that CBLs scales better than the rest of the strategies. In fact, CBLs is superior to all other approaches in 83.33 % of the configurations tested. Additionally, these results prove that CBLs is able to adapt to all kinds of environment topologies independently of team size.

A factorial Analysis of Variance (ANOVA) was conducted aiming to examine the main effects and interactions between environments (environment A, B and C), team size (1, 2, 4, 6, 8 and 12 robots) and algorithm (CR, HCR, HPCC, CGG, MSP, GBS, SEBS and CBLs) on the performance value given by \mathcal{I}_G . A confidence level of 95 % was used to identify significant differences between factor levels. Prior analysis revealed that there were no significant outliers, i.e. no major data stood below the first quartile or above the third quartile in the dataset, and the data were normally distributed across all groups, as revealed by the Shapiro-Wilk's test ($p < 0.05$).

The analysis revealed that there is a main effect of the environment on the idleness values ($F(2, 426) = 12308.678$, $p < 0.001$, partial $\eta^2 = 0.989$), given that these values are significantly higher in environment A when compared

⁵ The source code of the patrolling approaches tested are available at: https://github.com/davidbsp/patrolling_sim.

Table 7 Cases where the \mathcal{I}_G values of CBLS algorithm were statistically significant different to the other algorithms (bold), and not statistically significant different (italic)

Environment A						Environment B						Environment C					
1	2	4	6	8	12	1	2	4	6	8	12	1	2	4	6	8	12
CR	CR	CR	CR	CR	CR	CR	CR	CR	<i>CR</i>	CR	CR	<i>CR</i>	<i>CR</i>	<i>CR</i>	<i>CR</i>	<i>CR</i>	<i>CR</i>
HCR	HCR	HCR	HCR	HCR	HCR	HCR	HCR	HCR	HCR	HCR	HCR	<i>HCR</i>	<i>HCR</i>	<i>HCR</i>	<i>HCR</i>	<i>HCR</i>	<i>HCR</i>
HP	HP	HP	HP	HP	HP	HP	HP	<i>HP</i>	HP	HP	HP	<i>HP</i>	<i>HP</i>	<i>HP</i>	<i>HP</i>	<i>HP</i>	<i>HP</i>
CC	CC	CC	CC	CC	CC	CC	CC	<i>CC</i>	CC	CC	CC	<i>CC</i>	<i>CC</i>	<i>CC</i>	<i>CC</i>	<i>CC</i>	<i>CC</i>
CGG	CGG	CGG	CGG	CGG	CGG	CGG	CGG	CGG	<i>CGG</i>	<i>CGG</i>	<i>CGG</i>	CGG	<i>CGG</i>	<i>CGG</i>	<i>CGG</i>	<i>CGG</i>	<i>CGG</i>
MSP	MSP	MSP	MSP	MSP	MSP	MSP	MSP	MSP	MSP	<i>MSP</i>	–	MSP	MSP	<i>MSP</i>	<i>MSP</i>	<i>MSP</i>	–
GBS	GBS	GBS	GBS	GBS	GBS	GBS	GBS	GBS	GBS	GBS	<i>GBS</i>	<i>GBS</i>	<i>GBS</i>	<i>GBS</i>	<i>GBS</i>	<i>GBS</i>	<i>GBS</i>
SE	SE	SE	SE	SE	SE	SE	SE	SE	<i>SE</i>	SE	<i>SE</i>	<i>SE</i>	<i>SE</i>	<i>SE</i>	<i>SE</i>	<i>SE</i>	<i>SE</i>
BS	BS	BS	BS	BS	BS	BS	BS	BS	<i>BS</i>	BS	<i>BS</i>	<i>BS</i>	<i>BS</i>	<i>BS</i>	<i>BS</i>	<i>BS</i>	<i>BS</i>

Note that even in the italic cases,, CBLS resulted in lower values most of the times (see Tables 4, 5, 6)

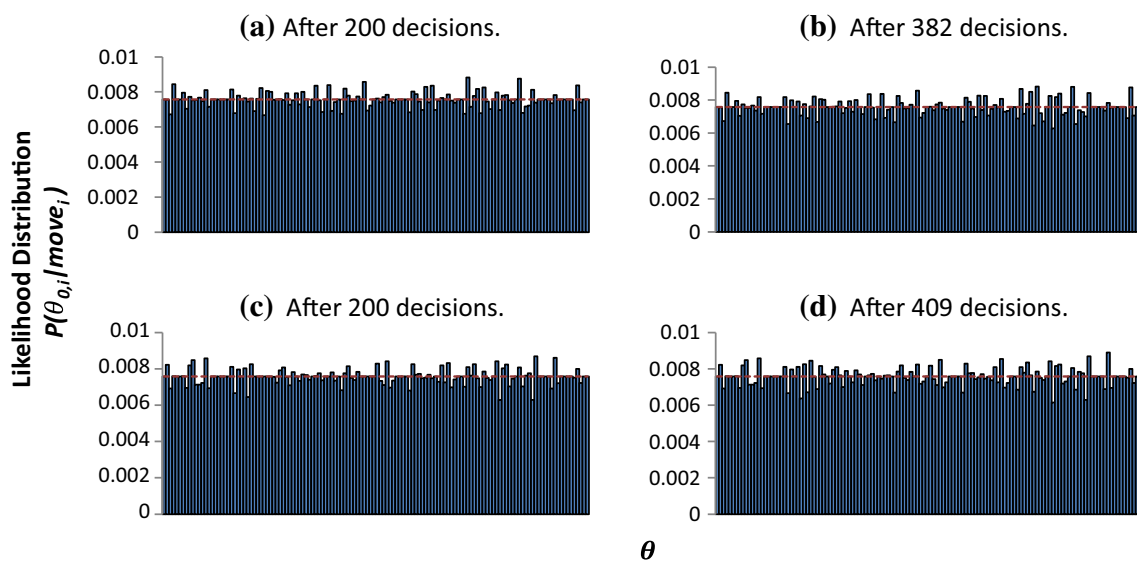


Fig. 4 Evolution of the likelihood distribution in a mission with 2 robots in Environment A, with $\omega = 0.75$. **a** robot 1 after 200 decisions; **b** robot 1 at the end of the mission; **c** robot 2 after 200 decisions; **d** robot

2 at the end of the mission. The red line represents the initial distribution when the mission started (Color figure online)

to environment B and C. Furthermore, the team size has a main effect on the idleness value ($F(5, 426) = 27051.872$, $p < 0.001$, partial $\eta^2 = 0.998$), since the idleness values decreased as more robots were used. More importantly, there was also a clear effect of the algorithm towards the idleness values ($F(7, 426) = 244.655$, $p < 0.001$, partial $\eta^2 = 0.858$), which explains the differences obtained with each of the patrolling strategy tested. Pairwise comparisons between all algorithms across all combinations of team size and environment revealed that in the case of the CBLS algorithm, the idleness values were significantly lower ($p < 0.001$) than all the other algorithms, except specific cases in Environment B and C, as shown in Table 7. In specific cases, especially in Environment C, the differences were not significant. However, descriptive statistics still show that the idleness values in CBLS was lower in most of these cases.

Moreover, there was a statistically significant interaction effect between team size and algorithm towards the idleness values ($F(14, 426) = 43151.575$, $p < 0.001$, partial $\eta^2 = 0.827$). As expected, pairwise comparisons between team size groups revealed that in all algorithms the idleness value significantly decreased as the team size increased. There was also a statistically significant interaction effect between the environment and algorithm towards the idleness values ($F(35, 426) = 13.665$, $p < 0.001$, partial $\eta^2 = 0.627$). Pairwise comparisons between environments revealed that in all algorithms the idleness value significantly decreased in environment C compared to environment A.

Looking now at the CBLS results in more detail, the evolution of the likelihood function of a patrolling mission with two robots in environment A and $\omega = 3/4$ is illustrated in Fig. 4. Note that each robot apprehends a different distribu-

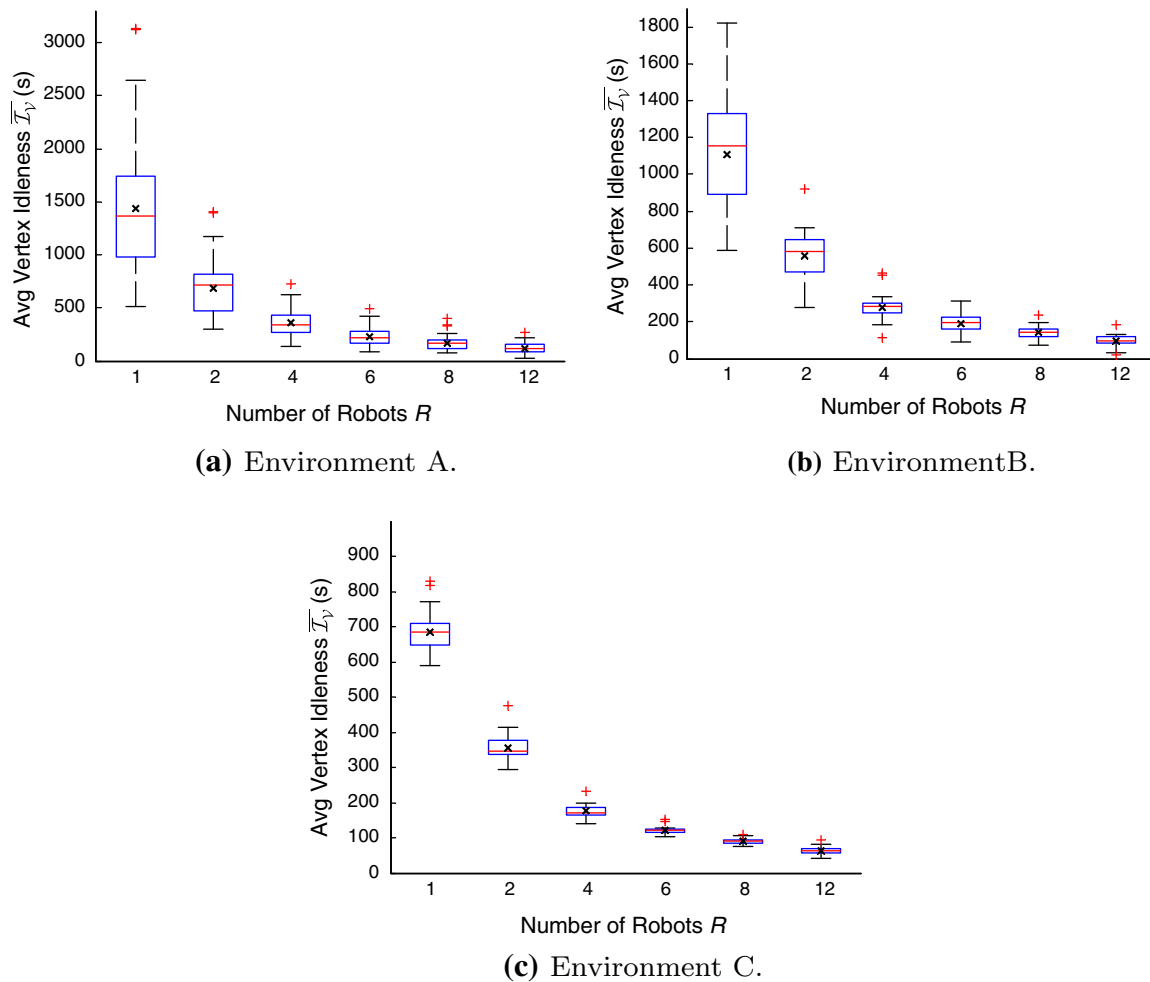


Fig. 5 Overall results running CBLS with $\omega = 3/4$ and different team sizes

tion depending on its acquired experience, and has no control or knowledge on the internal state of its teammates. 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 (represented by the red line in each chart), which comes as a consequence of robots having to visit every vertex $v_i \in \mathcal{G}$.

Moving on to the performance of the algorithm, the boxplot charts in Fig. 5 represent the $\bar{\mathcal{I}}_{\mathcal{V}}$ values, in seconds, for each tested team size on all three maps. The average value is represented by a black cross, providing a generalized measure: the average graph idleness, $\bar{\mathcal{I}}_{\mathcal{G}}$ (cf. Eq. 5). 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 $\bar{\mathcal{I}}_{\mathcal{V}}$, 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 team size. Group productivity will eventually converge with a large R .

In theory, productivity should grow during size scale-up; however, spatial limitations increase the number of times the robots meet and beyond a given R , it is argued that they will spend more time avoiding each other than effectively patrolling on their own.

Another interesting aspect illustrated in the boxplots of Fig. 5 is the fact that the median $\bar{\mathcal{I}}_{\mathcal{G}}$ is close to the mean $\bar{\mathcal{I}}_{\mathcal{G}}$ in all configurations, being usually lower. This means that the $\bar{\mathcal{I}}_{\mathcal{V}}$ values are positively skewed, i.e. most of the values are below the average, $\bar{\mathcal{I}}_{\mathcal{G}}$, and as a consequence, most outliers are above the third quartile.

On a more general note, visual inspection of the trajectories of robots using CBLS showed that the patrolling routes change over time so as to adapt to the system’s needs, making it difficult to predict the robot’s behavior, as opposed to most strategies presented in Portugal and Rocha (2013a, b). This is in fact one of the key advantages of using a distributed online approach, as the system will continuously adapt even to simple events that occur during the course of the mission, e.g. taking longer for a robot to reach a vertex due to

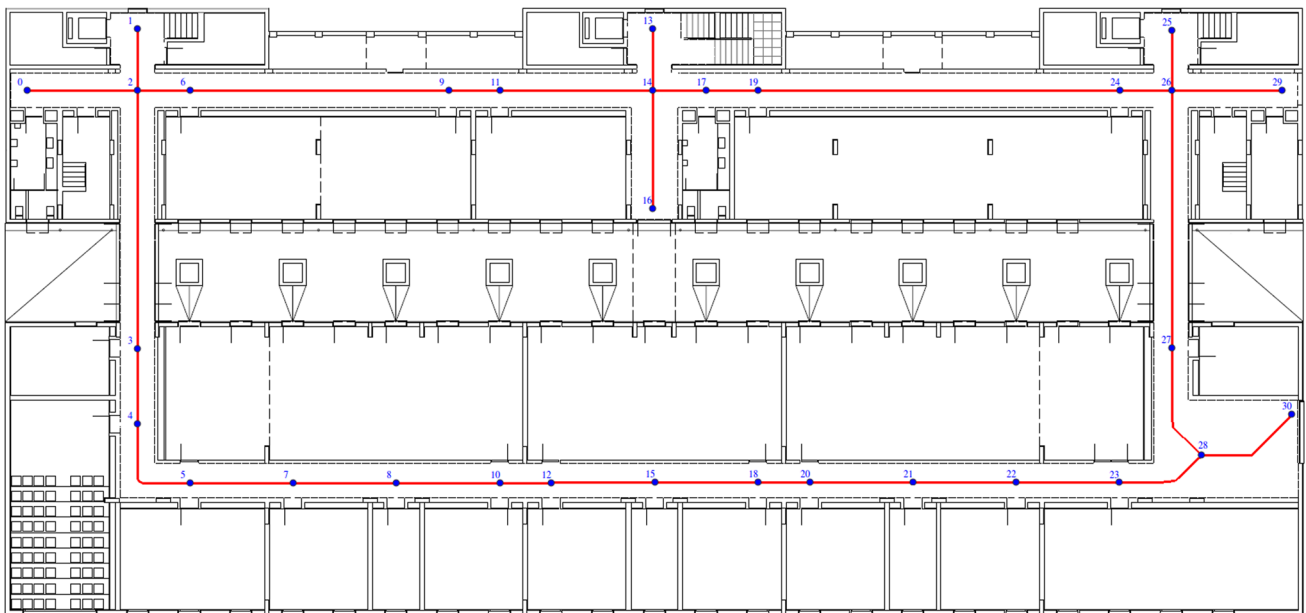


Fig. 6 Topological map of the “ISR-Floor0” environment

an unexpected obstacle, eventual navigation recovery behaviors that delay visits to the following vertices, robots traveling at different speeds (cf. Sect. 5.2.2), robot failures (cf. Sect. 5.2.3), etc. The paths of each patrolling unit will continuously change, as robots have no physical restriction to move within the whole navigation graph, in order to maintain a minimum graph idleness according to the model presented.

This phenomenon, 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.

5.2 Experiments with physical robots

Multi-Robot Patrol is mainly a practical problem and, in order for distributed intelligence systems to be useful in the real world, it is essential to go beyond simulation experiments and validate convincing solutions that prove the reliability of the proposed strategy in more demanding scenarios. In this section, the implementation of a system for multi-robot patrol in a large indoor environment is presented. Fully autonomous agents decide locally and sequentially their patrol routes according to the state of the system, as previously described, validating the CBLs distributed approach. Beyond the coordination which arises from the distributed communication of agents, it is also shown that the approach is scalable, robust to robot failures, i.e. fault-tolerant, and supports heterogeneous agents with different speed profiles.

Experiments were performed in a large indoor scenario, namely the floor 0 of the Institute of System and Robotics (ISR), in the University of Coimbra. Figure 6 shows



Fig. 7 Robots used in the experiments with the CBLs algorithm at ISR

the extracted topological map on top of the 67.85×26.15 m environment, which was obtained using the algorithm in Portugal and Rocha (2013c). The resulting topology is a non-complete, connected and sparse graph, like most real world environments.

When conducting experiments in the real world, one must overcome noisy sensor readings, localization issues and even robot failures, which are usually ignored or not precisely modeled in simulation experiments. Therefore, a team of six Pioneer-3DX robots (Robotics 2006), equipped with an Hokuyo URG-04LX-UG01 laser with 5.6 m of maximum scanning range, and a laptop on top was used, as seen in Fig. 7. Each laptop runs the ROS navigation stack using the Adaptive Monte Carlo (AMCL) algorithm for Localization as done previously in Stage simulations (cf. Sect. 5.1), being responsible for controlling the robot’s motion, which reaches

Table 8 Experiments with 1 to 6 robots using CBLS (all values in seconds)

CBLS				
R	$\overline{\mathcal{I}_G}$	$\max(\overline{\mathcal{I}_V})$	$\widetilde{\mathcal{I}_G}$	σ
1	377.10	562.49	414.80	103.31
	383.98	526.14	425.22	98.97
	392.05	580.43	431.52	106.70
2	170.16	298.77	175.33	45.52
	175.51	331.85	180.45	49.99
	183.56	343.25	185.55	50.81
3	108.71	173.24	102.94	29.72
	110.59	163.69	106.35	31.79
	122.04	240.90	118.54	39.44
4	85.37	140.68	77.31	26.26
	87.43	153.58	90.06	27.31
	90.34	168.27	83.73	29.67
5	65.79	121.86	56.62	24.25
	68.40	114.93	68.84	20.73
	69.12	136.32	63.26	23.71
6	54.67	82.03	52.14	15.10
	55.60	87.50	56.25	14.88
	60.09	116.08	54.70	21.41

Table 9 Experiments with 1 to 6 robots using TSP cycle (all values in seconds)

TSP cycle				
R	$\overline{\mathcal{I}_G}$	$\max(\overline{\mathcal{I}_V})$	$\widetilde{\mathcal{I}_G}$	σ
1	334.64	416.08	351.65	79.50
	351.89	417.60	374.47	81.53
	357.15	440.85	376.07	84.42
2	170.50	211.71	182.18	40.32
	174.76	216.65	186.67	41.17
	201.83	316.26	210.02	54.92
3	115.01	159.76	123.43	27.64
	115.60	158.21	123.22	27.78
	118.69	173.66	124.33	29.95
4	83.15	102.55	88.92	19.35
	83.87	103.51	88.95	19.46
	89.88	148.79	94.69	22.48
5	67.48	84.37	71.42	15.88
	68.88	91.50	72.57	15.64
	74.07	104.22	76.49	20.30
6	56.83	71.31	59.56	13.23
	56.86	72.41	59.33	13.43
	58.68	85.20	64.93	14.26

speeds of up to 1 m/s. As for communication, a distributed publish/subscribe mechanism has been used, due to its built-in integration in ROS. Moreover, each robot runs its own ROS master node (*roscore*), and multimaster communication is provided using the *wifi_comm*⁶ package. This means that there is no central point of failure in the system. While communication using a mobile ad hoc network deployed by the robot themselves would be possible, in these experiments we leveraged the existing WiFi infrastructure in the building.

We examine not only the average graph idleness along time, $\overline{\mathcal{I}_G}$, but also the median $\widetilde{\mathcal{I}_G}$, standard deviation σ , and the maximum average idleness of a vertex along time, $\max(\overline{\mathcal{I}_V})$. In the beginning of each test, the graph of the environment is loaded by every robot. A ROS node (i.e. a ROS application) is responsible for advertising the start of the mission and collecting results during the experiments. These results are examined in the next sections. Note that this “monitor” node does not centralize the approach nor does it give feedback to the robots whatsoever. In fact, it is merely used in this work to allow the extraction of experimental results.

Firstly, we conducted experiments with teams from one to six robots using the proposed strategy and comparing it with the TSP Cyclic Strategy. After this comparative analysis, CBLS was tested by adding one, and then, two agents to the patrolling mission with different speed profile, in order

to prove that the approach adapts well even when heterogeneous teams of robots are used. In addition, to prove its robustness, experiments which included failures in the robots at different time instants were analyzed. Finally, we carried out additional simulation experiments under the same conditions of the experiments with physical robots conducted in order to evaluate how realistic the simulation framework used throughout this work is. Whenever CBLS is adopted in these experiments, $\omega = 3/4$ is considered. The total estimated distance traveled by the robots during the course of all experiments was 50 kms ($\simeq 31$ miles).

5.2.1 Scalability and benchmarking

In this subsection, we focus on the scalability of the approach and analyze its performance when compared to the TSP Cyclic Strategy. As mentioned before, the latter is a recognized classical approach for multi-robot patrolling which finds the minimal tour that visits every vertex of the graph. This can be obtained using heuristic methods like the Chained Lin-Kernighan algorithm (Applegate et al. 2003). The approach is theoretically optimal for the single robot case, while in the multi-robot case, robots are equally spaced along the tour, and the approach provides a near optimal solution. Note also, that the deterministic route is computed a priori and offline, in contrast to our approach, where agents have the autonomy to decide their own moves online; and

⁶ Available at http://wiki.ros.org/wifi_comm.

Fig. 8 Robots positions during an experiment with 6 robots using CBLS. Implicit formation of dynamic regions

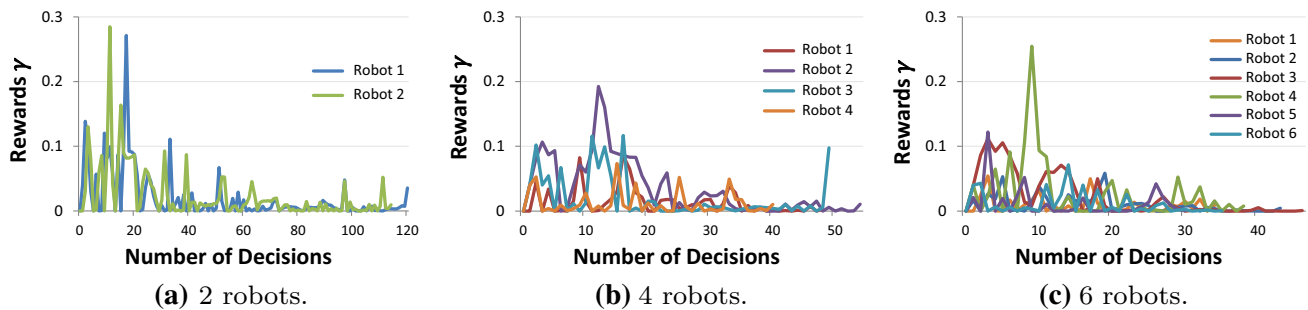
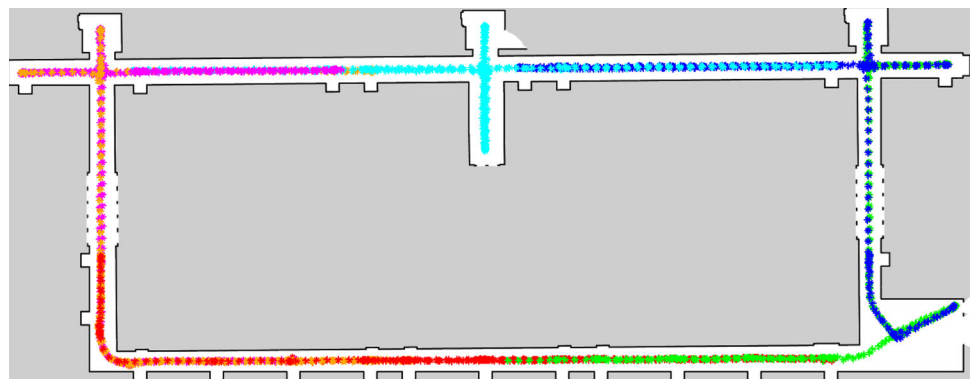


Fig. 9 Evolution of the absolute reward values along three experiments with different team size

it assumes strictly homogeneous robots without supporting different speed profiles or robot failures, as opposed to CBLS. In the particular case of the environment used in the experiments (cf. Fig. 6), the TSP cycle tour is composed of the clear rectangular cycle pattern that exists in the map, with short detours to vertices with $deg(v_i) = 1$, namely $v_i = \{0, 1, 13, 16, 25, 29, 30\}$.

Using the team of Pioneer-3DX robots, experiments with different team size from 1 to 6 robots were performed in the “ISR-Floor0” scenario. Each experiment was repeated 3 times and the results are presented in Tables 8 and 9.

An initial analysis shows that the performance values given by $\overline{\mathcal{I}_G}$ of both approaches are generally close in the configurations tested. As expected for the single robot case, the optimal TSP cycle outperforms CBLS (by $\simeq 9.5\%$). Nevertheless, considering that CBLS is running with a single learning agent limited to knowledge of its 2-step neighborhood, the result obtained is optimistic. Theoretically, having a larger horizon for the particular single robot case would improve CBLS results, as there would not be any other concurrent teammate learner in the system and, consequently, no interference with the agent’s long term plans would exist, enabling it to look further ahead without the risk of regretting its decisions.

Perhaps the most interesting aspect of these results is the performance attained by the proposed strategy with multi-robot teams. In the experiments conducted, the results

obtained with CBLS were slightly better than TSP cycle, achieving differences in performance of up to $\simeq 3.5\%$. However, one cannot say that CBLS is superior to TSP for MRS without conducting more tests. In general, both strategies perform similarly in the collected results, with TSP presenting even superior results of around $\simeq 2.5\%$ in the 4 robots situation.

The tests in teams of growing number of robots show that the approach is able to scale well, performing similarly to TSP cycle in a near optimal way. This is remarkable, considering the distributed and non-deterministic nature of the approach, as opposed to TSP cycle. A careful look at the behavior of the robots shows us that they tend to create dynamic regions where each agent patrols more often. Robots do not have any physical boundaries to restrict them. However, the fact that our model takes into consideration the teammates’ decisions results in the robots avoiding the areas of their teammates. Thus, the patrolling strategy will tend to implicitly create these regions, especially in lowly connected environments, as clearly shown in Fig. 8. As a result, there is little interference between agents.⁷ In addition, since robots only share their current and future immediate goals, the bandwidth requirements are negligible even with larger teams.

It is clear in Table 9 that by making the robots follow the same global route, TSP cycle always presents lower values of

⁷ A video of an experiment with 6 robots running CBLS is available at: <https://sites.google.com/site/davidbsp2014/videos/auro>.

Table 10 Experiments with a team of 6 physical robots using CBLS extended with one and two virtual agents (all values in seconds)

R	$\overline{\mathcal{I}_G}$	$\max(\overline{\mathcal{I}_V})$	$\widetilde{\mathcal{I}_G}$	σ
6 + 1	51.65	88.35	48.22	15.06
	52.26	89.51	52.62	15.31
	54.23	112.08	46.40	21.36
6+2	46.00	75.78	44.96	12.91
	50.87	70.76	53.12	12.35
	43.31	71.16	46.29	12.21

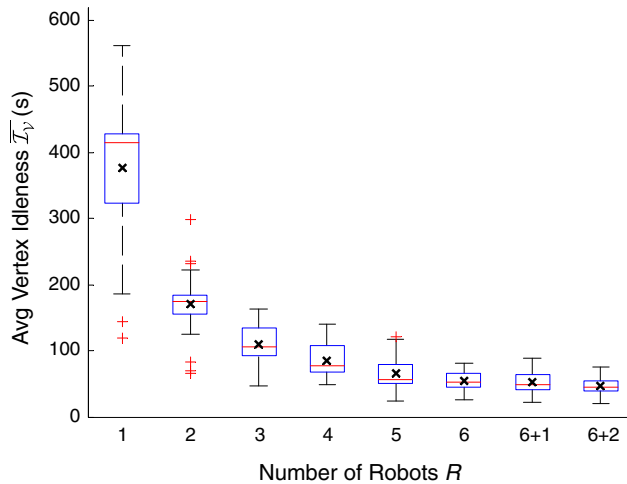


Fig. 10 Overview of the results with teams of physical robots and mixed teams of real and virtual robots, using CBLS

standard deviation, σ , when compared to CBLS, promoting more uniform visits to vertices. This is also suggested by the maximum average idleness of the vertices, $\max(\overline{\mathcal{I}_V})$, which are smaller than in CBLS.

Another interesting aspect observed in the experiments is the median value, $\widetilde{\mathcal{I}_G}$, being usually lower than the mean $\overline{\mathcal{I}_G}$ for CBLS, especially with greater team size, thus confirming the simulation results. As opposed, for TSP cycle, all $\widetilde{\mathcal{I}_G}$ values are higher than $\overline{\mathcal{I}_G}$, meaning that the distribution is negatively skewed and most of the values are above the average.

It is also worth referring another important disadvantage of TSP in general, which is the fact that the computation of a TSP cycle may become intensive or even prohibitive in large graphs.

Looking more closely at CBLS in the experiments with physical robots, a descending trend is shown by the absolute

reward values, given by the $(1 - \mathcal{H})$ factor in (16), along the experiments with different configurations. Figure 9 illustrates how these values evolve in missions with three different team sizes. Despite the occasional peaks, such values tend to decrease with the number of decisions. This is because, in general, as the system progresses, the $\overline{\mathcal{I}_V}$ 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 $\overline{\mathcal{I}_V}$ values in the neighborhood of other agents. For that reason, peaks are more observable in larger teams.

5.2.2 Heterogeneous teams

In the previous section, the number of robots R was limited to the physical robots available. However, the distributed patrolling method used supports an arbitrary high team size. In this section, the aim is to further explore the scalability of the approach and test it with teams of heterogeneous robots.

Being a distributed strategy composed of concurrent learning agents, the team should not only adapt to the system’s state but also to different robot profiles. Hence, virtual agents, running in the stage simulator, were added to the physical team, resulting in a mixed and interacting team of real and simulated robots, which communicate seamlessly. It is noteworthy that adding virtual simulated agents to the physical teams of robots was only made possible by the hardware abstraction layer of ROS and its modular structure.

Virtual agents have the same properties of the ones used in the simulation experiments. Thus, they travel slower than the physical robots. Three trials were conducted with a total of 7 agents composed by 6 physical robots and 1 simulated robot; and three more trials were performed with a team size of 8, composed by 6 physical robots and 2 simulated ones. Similarly to (locchi et al. 2011), the software layer is used unchanged both on real robots and in simulation.

Results in Table 10 show that the overall values of $\overline{\mathcal{I}_G}$, $\max(\overline{\mathcal{I}_V})$, $\widetilde{\mathcal{I}_G}$ and σ are within the expected, following the trend shown in Table 8. The additional virtual agents are integrated into the remaining team and are able to interact with teammates. In addition, their contribution to the global performance is minor, as expected. This happens not only because of the progressive decrease of individual contribu-

Table 11 Experiments with a team of 6 physical robots with one and two failures during the mission (all values in seconds)

Team size R	$\overline{\mathcal{I}_G}$	$\max(\overline{\mathcal{I}_V})$	$\widetilde{\mathcal{I}_G}$	σ	Mission time (τ)
6 (failure at 300s)	65.46	101.38	59.79	19.41	608.28
6 (failure at 150s)	68.00	107.16	68.10	17.43	642.98
6 (failure at 150s and 250s)	70.02	132.35	67.65	22.96	661.76

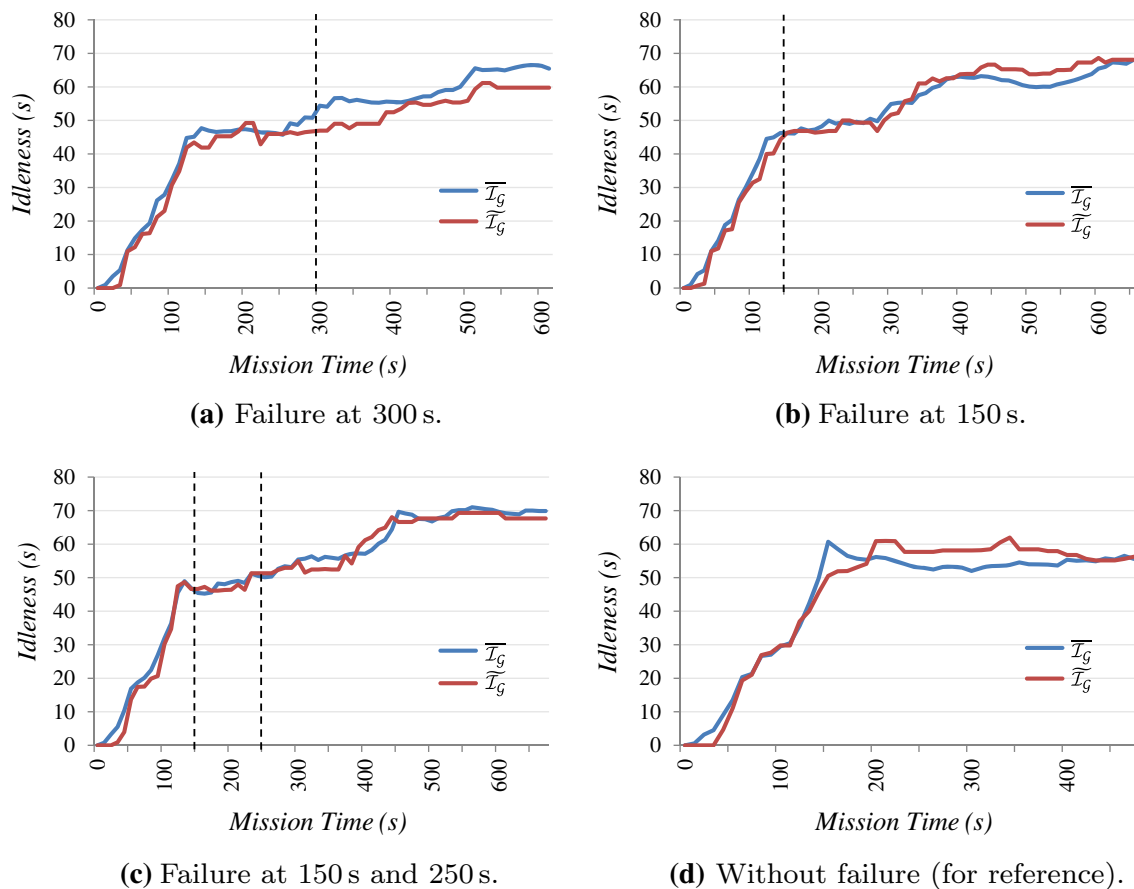


Fig. 11 Evolution of the idleness along time using CBLs with teams of six physical robots with failures. **a** Failure at 300 s. **b** Failure at 150 s. **c** Failure at 150 and 250 s. **d** Without failure (for reference)

tion of each robot as team size grows, but also due to lower speed at which these robots travel. The boxplot of Fig. 10 illustrates this trend.

It is noteworthy that incorporating robots that travel at different speeds with strategies that solve the MRPP with predefined routes, such as TSP cycle, would not be suitable because maintaining a uniform distance between each robot would not be possible unless all robots were limited to travel at the speed of the slowest robot.

5.2.3 Robustness/fault-tolerance

One of the main advantages of providing the patrol robots with means for deciding their moves in the environment is the absence of a centralized coordinator, which would represent a critical point of failure. A distributed autonomous robotic system, such as the herein presented, enables redundancy, remaining functional if some of the agents fail.

To demonstrate the robustness of the approach, three experiments using the Pioneer 3-DX robots were planned. In these experiments a robot is shut down at different instants

of time, aiming at studying the effect of the failures in the overall performance, as well as how the system evolves.

In all three tests, the team starts with six robots. In the first test, a robot is shut down 300 s into the experiment. Similarly, in the second experiment, a robot is shut down after 150 s. Finally, in the third experiment, one robot is shut down after 150 s and a second robot is shut down after 250 s. The other robots assume that a teammate has failed when no message has been received from it within 2 min.

Table 11 indicates that the results obtained in the first two experiments resemble those obtained with five robots. In these experiments, the average idleness values converged after the failure occurs. Figure 11 shows the evolution of the average vertex idleness in the three experiments. For reference, the evolution of the average vertex idleness without failure was also added (cf. Fig. 11d). In the first experiment, since approximately half of the mission was spent with six robots, the final performance was slightly better and the overall mission time τ was shorter than in the second experience, i.e. it took longer in the second test to fulfill the mission because the failure occurred earlier.

Table 12 Additional simulation experiments with 1 to 6 robots using CBLS in the “ISR-Floor0” environment (all values in seconds)

CBLS				
R	$\overline{\mathcal{I}_G}$	$\max(\overline{\mathcal{I}_V})$	$\widetilde{\mathcal{I}_G}$	σ
1	382.92	552.84	399.75	103.16
	383.31	553.49	400.05	103.28
	383.81	554.15	400.69	103.41
2	166.20	291.34	170.35	46.10
	177.13	332.76	177.83	45.06
	179.51	291.15	174.15	48.66
3	109.18	203.13	108.08	30.09
	111.68	192.94	111.53	28.38
	116.36	236.34	114.05	38.82
4	86.40	137.84	78.01	27.24
	87.75	134.10	75.64	29.00
	88.85	162.99	89.71	28.07
5	62.48	105.63	60.02	19.49
	66.31	111.70	61.92	24.83
	67.97	113.18	66.53	18.80
6	51.88	84.79	50.17	12.53
	54.30	91.50	56.41	13.32
	58.49	110.46	52.53	22.16

The two failures in the third experiment greatly influenced the final performance results in the test. However, it is also clear that extra time would be needed in order to converge to higher values, as the final average graph idleness $\overline{\mathcal{I}_G}$ was much lower than the 4 robot case. Nevertheless, one can verify that in all three cases, when the failure occurs, the values of $\overline{\mathcal{I}_G}$ and $\widetilde{\mathcal{I}_G}$ tend to increase after a while.

The evidences taken from these results show the robustness of the system, proving that it enables graceful degradation, as long as at least one robot remains operational.

Discussing now the hypothetical event of communication failures, when a message is not received by a robot, it does not update the instantaneous idleness time values and, consequently, it maintains incomplete information about the state of the system. This information becomes more incomplete with the increasing number of undelivered messages. Additionally, when robots are close to each other, if messages are not received, they may decide to patrol nearby locations and interfere with their teammates’ plans. The success of resolving such situations hugely depends on each robot’s local planner and the ability to avoid dynamic obstacles. In our work, this is taken care of by the ROS navigation stack. Therefore, we expect CBLS to be robust to communication failures and only slightly degrade its performance when the communication errors rate is moderate.

If the robots cannot communicate at all, the performance of the algorithm will drop strongly. This will occur espe-

cially in larger teams, which are much more influenced by the lack of coordination in the multi-robot system, as robots will constantly interfere with one another. In this case, robots will act greedily, will not share their intentions and will often compete for the same vertices. Performance is expected to gracefully degrade if only communication with neighbors is allowed. In this situation, robots will be able to coordinate themselves by not competing to the same goals and not interfering with teammates. Despite that, they will not have contact with agents that are further away and as a consequence they will make uninformed decisions quite often. We have conducted a similar analysis in our past work (Portugal and Rocha 2013b), where we employed an identical distributed communication protocol, despite using a different patrol strategy. Therein, we were able to verify the robustness of the multi-robot patrolling algorithm to communication failures.

5.2.4 Additional simulations in the environment used in the real world experiments

In this section, we discuss additional simulations by employing the CBLS approach described in this paper with varying team sizes in the same map used for the real world experiments. Our goal here is to verify the realism of the simulator using the same software layer that was used in the multi-robot experiments.

Similarly to Sect. 5.2.1, we have deployed teams of robots with $R = \{1, 2, 3, 4, 5, 6\}$ in the “ISR-Floor0” environment and under the same conditions described therein. However, this time we have used the stage multi-robot simulator instead of the real robots hardware. Once again, we have run three trials for each configuration. The results obtained are depicted in Table 12.

Comparing Table 12 with Table 8, it can be seen that the results drawn from simulation tests are very close to those obtained with the real robots. This resemblance translates into an average difference of 1.55 % in $\overline{\mathcal{I}_G}$ values, 5.49 % in $\max(\overline{\mathcal{I}_V})$ values, 2.75 % in $\widetilde{\mathcal{I}_G}$ values, and 4.05 % in σ values. These values are remarkably low, which suggest that these simulations can be used with a high degree of confidence to estimate the actual results with physical robots.

Nevertheless, the results extracted from the additional simulations tend to be marginally lower to those obtained with the real robots, and the variance between each different trial within the same configuration is inferior. Thus, it is our belief that multi-robot simulations in stage can provide an accurate yet slightly optimistic approximation of real world results, and the lower variance is possibly linked to real phenomena that are not fully modeled in simulations such as wheel slip, robot assembly properties or delays in processing sensor data and producing actuator commands.

Fig. 12 A Simulation in the “ISR-Floor0” environment with a team of 6 robots

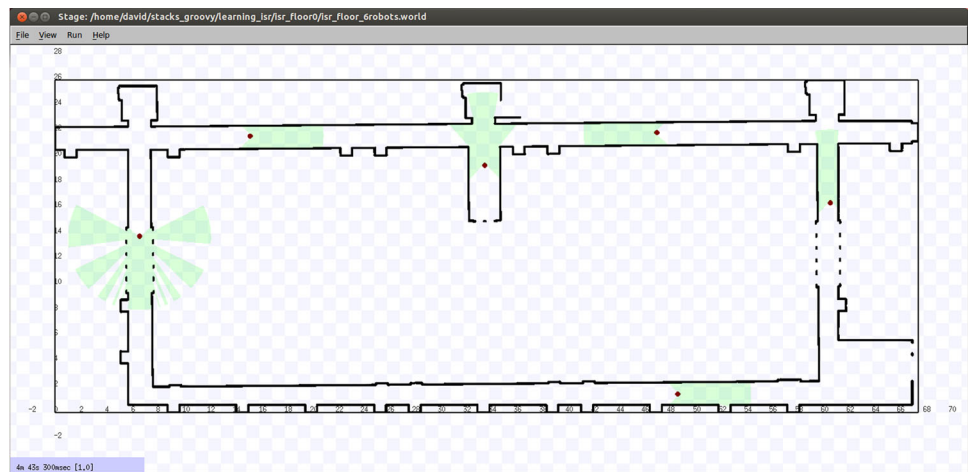


Figure 12 illustrates a snapshot of a simulation with 6 virtual pioneer robots in the environment. The results reported demonstrate how realistic the simulations are in the realm of multi-robot applications such as patrolling.

6 Conclusions

In this work, cooperative multi-agent learning has been addressed in order to solve the patrolling problem in a distributed way. 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 effective coordination in the behavior of patrolling agents. Concurrent reward-based learning has been adopted given that, in this domain, the decomposition of the problem reduces the complexity of the general mission by distributing computational load among each independent learner.

Experimental results have shown that the method is able to effectively tackle the problem, since it can deal with uncertainty and the actions are selected according not only to prior knowledge about the problem, but also the state of the system at the time, resulting in adaptive, effective and distributed cooperative patrolling. Moreover, it was shown that our Bayesian update method can be applied in different types of environments, independent of their topology, withstands failures in robotic patrol units, and accomplishes exceeding performance. In fact, when compared with other state of the art approaches through simulations, CBLS generally outperforms them independently of team size, and in the real world experiments conducted, the approach was able to obtain near optimal results, which is particularly remarkable given the limited search space considered and distributed nature of the

approach. Thus, this proves the potential of the proposed multi-robot patrolling strategy for real world applications.

In the future, it would be interesting to relax the assumption of perfect communication, testing the performance of CBLS under communication failures or allowing only local interactions between robots within a certain range. In fact, we would expect to obtain similar results as those tested in [Portugal and Rocha \(2013b\)](#), which showed that distributed patrolling approaches are robust to communication failures, performance degrades with increased ratio of failures, and limited communication range allows the team to coordinate and not compete to visit the same vertices of the graph, although keeping incomplete information about the state of the system. Finally, addressing the issue of scalability, we intend to work on estimation methods for dimensioning the size of the team of robots in a given patrolling task according to the environment topology and temporal constraints.

Acknowledgments This work was supported by a PhD scholarship (SFRH/BD/64426/2009), 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). The authors gratefully acknowledge Prof. Hélder Araújo (ISR) for conceding three robot platforms used in the experiments; Marios Belk (University of Cyprus) for his help in the statistical analysis of the results; Luís Santos, Micael S. Couceiro and Gonçalo Cabrita (ISR) for their contribution and feedback; João M. Santos, Gonçalo Augusto, João Martins, Nuno L. Ferreira, José S. Pereira, André Araújo and João B. Campos for their assistance during the experiments with real robots.

7 Appendix: Extended details of the simulations with CBLS with different ω

See Tables 13, 14, 15, 16, 17, 18, 19, 20, 21.

Table 13 Experiments using CBLs in Environment A with $\omega = 1.0$ (Extended version of Table 1)

CBLs, $\omega = 1.0$				
R	$\overline{\mathcal{I}_G}$	$\max(\overline{\mathcal{I}_V})$	$\widetilde{\mathcal{I}_G}$	σ
1	1409.80	4064.31	1748.41	652.75
	1438.08	4109.37	1769.69	651.19
	1488.47	4129.36	1803.30	666.48
2	679.83	1694.14	800.66	265.30
	686.26	1682.06	803.57	271.16
	757.67	1753.50	852.23	275.37
4	366.86	882.43	437.92	163.37
	400.51	889.05	458.52	174.76
	439.48	931.07	485.30	174.30
6	240.54	1328.39	300.35	121.31
	267.66	1375.58	325.23	125.26
	275.87	1399.79	319.15	124.77
8	172.51	541.57	201.94	88.31
	191.95	549.59	212.86	93.91
	199.88	562.83	217.96	93.09
12	159.81	488.03	199.66	77.72
	170.32	488.54	202.42	77.56
	176.42	484.81	202.39	76.95

Table 15 Experiments using CBLs in Environment A with $\omega = 2/3$ (Extended version of Table 1)

CBLs, $\omega = 2/3$				
R	$\overline{\mathcal{I}_G}$	$\max(\overline{\mathcal{I}_V})$	$\widetilde{\mathcal{I}_G}$	σ
1	1260.44	2856.95	1709.81	476.38
	1345.07	2912.23	1788.10	487.89
	1365.13	2916.27	1773.84	493.46
2	678.40	1663.71	807.96	239.29
	690.69	1661.83	794.35	232.78
	736.11	1751.91	889.34	254.32
4	369.59	806.64	441.57	133.91
	371.22	811.65	448.51	135.88
	373.83	817.76	456.50	137.97
6	249.28	531.77	242.99	89.94
	250.62	532.01	242.36	89.44
	253.30	535.53	245.33	90.11
8	180.38	474.79	231.62	73.83
	182.09	478.97	236.24	75.04
	182.12	478.06	234.86	74.56
12	126.82	410.10	168.42	67.26
	134.53	423.72	178.26	69.30
	140.41	430.74	181.47	69.06

Table 14 Experiments using CBLs in Environment A with $\omega = 3/4$ (Extended version of Table 1)

CBLs, $\omega = 1.0$				
R	$\overline{\mathcal{I}_G}$	$\max(\overline{\mathcal{I}_V})$	$\widetilde{\mathcal{I}_G}$	σ
1	1408.98	3096.21	1711.80	568.39
	1431.33	3136.20	1752.78	572.40
	1456.31	3158.46	1769.98	575.72
2	667.86	1402.55	866.45	239.24
	689.36	1387.68	840.66	237.29
	701.47	1428.39	903.10	242.48
4	325.78	726.43	435.49	124.32
	366.02	713.29	415.59	123.99
	382.02	745.25	460.85	125.31
6	218.21	467.63	197.02	82.43
	236.43	511.42	232.39	86.67
	239.32	506.76	233.20	85.18
8	159.69	390.02	156.83	67.46
	165.36	398.74	161.13	68.84
	177.30	420.30	178.12	70.69
12	117.47	269.19	153.57	49.81
	119.62	271.06	157.63	45.61
	125.64	262.38	145.01	47.41

Table 16 Experiments using CBLs in Environment B with $\omega = 1.0$ (Extended version of Table 2)

CBLs, $\omega = 1.0$				
R	$\overline{\mathcal{I}_G}$	$\max(\overline{\mathcal{I}_V})$	$\widetilde{\mathcal{I}_G}$	σ
1	1122.90	2044.08	1018.16	457.70
	1249.11	2065.41	1076.69	493.75
	1376.34	2170.45	1142.87	494.10
2	553.82	1286.68	558.61	165.76
	563.73	1206.82	563.61	162.73
	607.63	1316.28	588.17	171.25
4	263.99	448.00	259.19	61.57
	285.59	476.20	280.81	62.10
	305.05	478.35	293.22	66.53
6	170.98	381.55	167.94	51.97
	198.31	401.16	182.54	53.96
	222.70	419.39	195.62	55.56
8	132.59	220.06	133.52	30.78
	144.44	217.15	134.21	29.64
	153.06	233.74	148.90	31.65
12	101.80	233.21	102.04	38.83
	108.82	236.44	102.28	37.84
	114.03	256.90	121.09	42.88

Table 17 Experiments using CBLs in Environment B with $\omega = 3/4$ (Extended version of Table 2)

CBLs, $\omega = 3/4$				
R	$\overline{I_G}$	$\max(\overline{I_V})$	$\widetilde{I_G}$	σ
1	1069.64	1795.61	1027.41	271.18
	1105.76	1821.11	1154.11	281.00
	1156.57	1837.88	1271.98	289.89
2	534.43	916.10	557.74	132.26
	540.65	911.42	560.57	130.78
	595.91	936.18	592.55	132.21
4	247.48	451.45	275.25	64.92
	284.90	478.55	313.92	66.06
	295.50	457.61	287.12	64.77
6	172.82	301.91	182.31	39.94
	202.49	317.15	199.40	41.07
	203.00	315.29	199.47	40.67
8	128.01	229.38	135.00	29.70
	137.86	236.41	140.86	30.44
	162.35	249.12	154.95	31.33
12	86.85	174.99	92.48	28.95
	100.96	185.89	100.93	30.35
	101.63	183.82	101.10	29.77

Table 19 Experiments using CBLs in Environment C with $\omega = 1.0$ (Extended version of Table 3)

CBLs, $\omega = 1.0$				
R	$\overline{I_G}$	$\max(\overline{I_V})$	$\widetilde{I_G}$	σ
1	666.09	799.52	673.18	63.14
	712.50	804.82	709.72	60.61
	727.35	839.55	698.41	68.13
2	349.34	517.75	346.91	43.86
	351.84	510.28	342.40	43.50
	376.08	535.08	367.44	44.91
4	171.51	210.56	170.33	17.78
	182.24	214.17	178.76	18.66
	197.98	222.87	194.01	19.84
6	122.75	159.38	122.03	14.20
	127.84	161.40	125.24	14.47
	129.27	160.77	125.18	14.58
8	90.22	133.57	84.95	11.93
	99.50	142.91	97.59	13.16
	100.71	144.77	98.23	12.95
12	70.66	102.38	69.80	15.15
	75.66	103.90	69.97	14.74
	79.98	113.51	83.77	16.85

Table 18 Experiments using CBLs in Environment B with $\omega = 2/3$ (Extended version of Table 2)

CBLs, $\omega = 2/3$				
R	$\overline{I_G}$	$\max(\overline{I_V})$	$\widetilde{I_G}$	σ
1	1174.01	2120.82	1062.78	371.03
	1201.13	2126.26	1075.88	377.12
	1224.77	2147.52	1088.69	376.99
2	524.36	816.96	589.95	132.85
	554.68	792.60	563.56	130.61
	582.96	764.89	531.46	127.20
4	272.41	430.71	273.76	65.82
	280.73	439.63	278.03	64.71
	296.91	462.76	304.25	68.10
6	181.68	327.67	188.65	41.99
	199.51	351.73	213.95	45.00
	201.57	346.75	211.44	45.43
8	138.08	229.10	140.14	32.95
	142.92	234.02	145.80	33.76
	146.50	236.58	147.95	33.85
12	89.40	183.76	92.21	40.48
	94.06	188.40	95.90	41.11
	99.02	193.74	100.46	42.06

Table 20 Experiments using CBLs in Environment C with $\omega = 3/4$ (Extended version of Table 3)

CBLs, $\omega = 3/4$				
R	$\overline{I_G}$	$\max(\overline{I_V})$	$\widetilde{I_G}$	σ
1	662.66	816.26	658.82	63.03
	694.51	812.54	662.17	62.38
	696.09	857.30	729.51	64.28
2	342.27	470.40	336.87	38.02
	350.22	468.29	338.92	37.74
	374.46	484.96	362.90	38.51
4	163.58	230.44	166.70	17.03
	179.46	236.46	179.22	18.62
	183.17	228.86	167.54	19.97
6	118.34	149.03	117.21	9.92
	119.92	149.93	117.39	10.31
	125.82	155.39	126.26	10.76
8	85.86	108.84	91.81	7.78
	87.03	105.59	83.15	7.85
	97.86	112.22	94.06	8.07
12	62.55	94.62	60.87	12.25
	62.92	94.93	61.09	12.29
	63.53	95.05	61.41	12.28

Table 21 Experiments using CBLs in Environment C with $\omega = 2/3$ (Extended version of Table 3)

CBLs, $\omega = 2/3$				
R	$\overline{I_G}$	$\max(\overline{I_V})$	$\widetilde{I_G}$	σ
1	676.24	784.01	671.57	61.26
	683.51	801.12	630.69	59.98
	706.62	849.22	726.69	63.59
2	345.47	422.23	341.38	29.23
	364.69	433.50	369.60	31.86
	367.98	414.72	338.12	29.90
4	176.77	201.32	158.10	15.83
	183.43	218.17	185.09	17.54
	187.45	219.61	191.46	18.39
6	118.23	159.20	117.87	11.42
	128.31	166.27	128.68	12.15
	130.50	162.68	126.44	12.34
8	87.84	100.73	83.66	7.87
	93.54	107.79	94.87	8.50
	94.14	108.09	92.18	8.02
12	59.92	91.54	62.85	12.83
	62.75	92.31	62.32	12.42
	69.90	100.44	74.22	14.17

References

- Agmon, N., Fok, C. L., Emaliach, Y., Stone, P., Julien, C., & Vishwanath, S. (2012). On coordination in practical multi-robot patrol. In *Proceedings of the IEEE international conference on robotics and automation (ICRA 2012)* (pp. 650–656), May 14–18. Saint Paul, MN.
- Agmon, N., Urieli, D., & Stone, P. (2011). Multiagent patrol generalized to complex environmental conditions. In *Proceedings of the 25th conference on artificial intelligence (AAAI 2011)*, San Francisco, CA, August 7–11.
- Agmon, N., Kaminka, G. A., & Kraus, S. (2011). Multi-robot adversarial patrolling: Facing a full-knowledge opponent. *Journal of Artificial Intelligence Research*, 42, 887–916.
- Aguirre, O., & Taboada, H. (2012). An evolutionary game theory approach for intelligent patrolling. In *Procedia computer science* (Vol. 12, Part II, pp. 140–145). Amsterdam: Elsevier.
- Ahmadi, M., & Stone, P. (2006). A multi-robot system for continuous area sweeping tasks. In *Proceedings of the international conference on robotics and automation (ICRA 2006)*. Orlando, FL, May 15–19.
- Almeida, A., Ramalho, G., Santana, H., Tedesco, P., Menezes, T., Corruble, V., & Chevalyere, Y. (2004). Recent advances on multi-agent patrolling. In *Advances in artificial intelligence—SBIA 2004*. Lecture Notes in Computer Science (Vol. 3171, pp 474–483). Berlin: Springer.
- Applegate, D., Cook, W., & Rohe, A. (2003). Chained Lin-Kernighan for large traveling salesman problems. *INFORMS Journal on Computing*, 15, 82–92.
- Baglietto, M., Cannata, G., Capezio, F., & Sgorbissa, A. (2009). Multi-robot uniform frequency coverage of significant locations in the environment. In *Distributed autonomous robotic systems* (Vol. 8, pp. 3–14). Berlin: Springer.
- Basilico, N., Gatti, N., Rossi, T., Ceppi, S., & Amigoni, F. (2009). Extending algorithms for mobile robot patrolling in the presence of adversaries to more realistic settings. In *Proceedings of the 2009 IEEE/WIC/ACM international conference on intelligent agent technology (IAT'09)* (pp. 557–564). Milan.
- Cannata, G., & Sgorbissa, A. (2011). A minimalist algorithm for multi-robot continuous coverage. *IEEE Transactions on Robotics*, 27(2), 297–312.
- Chevalyere, Y. (2004). Theoretical analysis of the multi-agent patrolling problem. In *Proceedings of the 2004 international conference on agent intelligent technologies (IAT'04)*, (pp. 30–308). Beijing, September 20–24.
- Elmaliach, Y., Shiloni, A., & Kaminka, G. A. (2008). A realistic model of frequency-based multi-robot polyline patrolling. In *Proceedings of the 7th international joint conference on autonomous agents and multiagent systems (AAMAS 2008)*, (Vol. 1, pp. 63–70).
- Fabrizi, E., & Saffiotti, A. (2000). Extracting topology-based maps from gridmaps. In *Proceedings of the 2000 IEEE international conference on robotics and automation (ICRA'2000)* (pp. 2972–2978). San Francisco, CA, April 2000.
- Fazli, P., Davoodi, A., & Mackworth, A. K. (2013). Multi-robot repeated area coverage. *Autonomous Robots*, 34(4), 251–276. Springer Science.
- Fiedler, M. (1973). Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal*, 23(2), 298–305.
- Gabriely, Y., & Rimon, E. (2001). Spanning-tree based coverage of continuous areas by a mobile robot. *Annals of mathematics and artificial intelligence* (Vol. 31, pp. 77–98). Hingham, MA: Kluwer Academic Publishers.
- Iocchi, L., Marchetti, L., & Nardi, D. (2011). Multi-robot patrolling with coordinated behaviours in realistic environments. In *Proceedings of the international conference on intelligent robots and systems (IROS'2011)* (pp. 2796–2801).
- Ishiwaka, Y., Sato, T., & Kakazu, Y. (2003). An approach to the pursuit problem on a heterogeneous multiagent system using reinforcement learning. *Robotics and Autonomous Systems (RAS)*, 43(4), 245–256. Elsevier.
- Jansen, M., & Sturtevant, N. (2008). Direction maps for cooperative pathfinding. In *Proceedings of the 4th artificial intelligence and interactive digital entertainment conference (AAAIIDE'08)*. Stanford, CA, October 22–24.
- Jansen, F., & Nielsen, T. (2007). *Bayesian networks and decision graphs* (2nd ed.). Berlin: Springer.
- Keskin, B. B., Li, S., Steil, D., & Spiller, S. (2012). Analysis of an integrated maximum covering and patrol routing problem. *Transportation Research Part E*, 48, 215–232. Elsevier.
- Lauri, F., & Koukam, A. (2014). Robustness analysis of multi-agent patrolling strategies using reinforcement learning. In *Proceedings of the international conference on swarm intelligence based optimization (ICSIBO 2014)*, Mulhouse, May 13–14.
- Marier, J., Besse, C., & Chaib-draa, B. (2010). Solving the continuous time multiagent patrol problem. In *Proceedings of the 2010 IEEE international conference on robotics and automation (ICRA 2010)*, Anchorage, AK.
- Marino, A., Antonelli, G., Aguiar, A. P., & Pascoal, A. (2012). A new approach to multi-robot harbour patrolling: Theory and experiments. In *Proceedings of the 2012 IEEE/RSJ international conference on intelligent robots and systems (IROS 2012)*. Vilamoura, Portugal, October 7–12.
- Marino, A., Parker, L. E., Antonelli, G., & Caccavale, F. (2013). A decentralized architecture for multi-robot systems based on the null-space-behavioral control with application to multi-robot border patrolling. *Journal of Intelligent and Robotic Systems*, 71, 423–444.

- Murphy, R. (2004). Human–robot interaction in rescue robotics. *IEEE Transactions on Systems, Man and Cybernetics Part C*, 34(2), 138–153.
- Panait, L., & Luke, S. (2005). Cooperative multi-agent learning: The state of the art. *Journal of Autonomous Agents and Multi-Agent Systems*, 11(3), 387–434.
- Parker, L. E. (2008). Distributed intelligence: Overview of the field and its application in multi-robot systems. *Journal of Physical Agents*, 2(2), 5–14. Special issue on Multi-robot systems.
- Pasqualetti, F., Franchi, A., & Bullo, F. (2012). On cooperative patrolling: Optimal trajectories, complexity analysis, and approximation algorithms. *IEEE Transactions on Robotics*, 28(3), 592–606.
- Pippin, C., Christensen, H., & Weiss, L. (2013). Performance based task assignment in multi-robot patrolling. In *Proceedings of the 2013 ACM symposium on applied computing (SAC '13)* (pp. 70–76). Coimbra, March 18–22.
- Portugal, D., & Rocha, R. P. (2013). Retrieving topological information for mobile robots provided with grid maps. In *Agents and artificial intelligence*. Communications in Computer and Information Science (CCIS) series, (Vol. 358, pp. 204–217). Berlin: Springer.
- Portugal, D., Couceiro, M. S., & Rocha, R. P. (2013). Applying bayesian learning to multi-robot patrol. In *Proceedings of the 2013 international symposium on safety, security and rescue robotics (SSRR 2013)*, Linköping, Oct 21–26.
- Portugal, D., Couceiro, M., & Rocha, R. P. (2013). Concurrent Bayesian learners for multi-robot patrolling missions. In *Proceedings of the 2013 IEEE international conference on robotics and automation (ICRA 2013)*, Workshop on Towards Fully Decentralized Multi-Robot Systems: Hardware, Software and Integration, Karlsruhe, May 6–10.
- Portugal, D., Pippin, C., Rocha, R. P., & Christensen, H. (2014). Finding optimal routes for multi-robot patrolling in generic graphs. In *Proceedings of the 2014 IEEE/RSJ international conference on intelligent robots and systems (IROS 2014)*. Chicago, Sep. 14–18.
- Portugal, D., & Rocha, R. P. (2013). Multi-robot patrolling algorithms: examining performance and scalability. *Advanced Robotics Journal*, 27(5), 325–336.
- Portugal, D., & Rocha, R. P. (2013). Distributed multi-robot patrol: A scalable and fault-tolerant framework. *Robotics and Autonomous Systems (RAS)*, 61(12), 1572–1587.
- Poulet, C., Corruble, V., & Seghrouchini, A. (2012). Working as a team: Using social criteria in the timed patrolling problem. In *Proceedings of the international conference on tools with artificial intelligence (ICTAI'2012)*. Athens, GR, Nov. 7–9.
- Quigley, M., Gerkey, B., Conley, K., Faust, J., Foote, T., Leibs, J., Berger, E., Wheeler, R., & Ng, A. (2009). ROS: an open-source robot operating system. In *Proceedings of the IEEE international conference on robotics and automation (ICRA'2009)*, Workshop On Open Source Software, Kobe, May 12–17.
- Robotics, Activ Media. (2006). *Pioneer 3 operations manual, version 3*. : Mobile Robots Inc.
- Rocha, R., Dias, J., & Carvalho, A. (2005). Cooperative multi-robot systems: A study of vision-based 3-D mapping using information theory. *Robotics and Autonomous Systems (RAS)*, 53(3–4), 282–311. Elsevier.
- Ruan, S., Meirina, C., Yu, F., Pattipati, K. R., & Popp, R. L. (2005). Patrolling in a stochastic environment. In *Proceedings of the 10th international command and control research and technology symposium*, McLean, VA, June 13–16.
- Sak, T., Wainer, J., & Goldenstein, S. (2008). Probabilistic Multiagent Patrolling. In *Advances in Artificial Intelligence, SBIA 2008*. Lecture Notes in Computer Science, (Vol. 5249, pp. 124–133). Berlin: Springer.
- Sampaio, P., Ramalho, G., & Tedesco, P. (2010). The gravitational strategy for the timed patrolling. In *Proceedings of the IEEE international conference on tools with artificial intelligence (ICTAI'10)* (pp. 113–120). Arras, France, Oct. 27–29.
- Santana, H., Ramalho, G., Corruble, V., & Ratitch, B. (2004). Multi-agent patrolling with reinforcement learning. In *Proceedings of the third international joint conference on autonomous agents and multiagent systems* (Vol. 3, pp. 1122–1129). New York, NY.
- Sempé, F., & Drogoul, A. (2003). Adaptive patrol for a group of robots. In *Proceedings of the international conference on intelligent robots and systems (IROS 2003)*. Las Vegas.
- Smith, S., & Rus, D. (December 2010). Multi-robot monitoring in dynamic environments with guaranteed currency of observations. In *Proceedings of the IEEE conference on decision and control* (pp. 514–521). Atlanta, GA.
- Stone, P., & Veloso, M. (2000). Multiagent systems: A survey from a machine learning perspective. *Autonomous Robots*, 8(3), 345–383.
- Stranders, R., de Cote, E. M., Rogers, A., & Jennings, N. R. (2012). Near-optimal continuous patrolling with teams of mobile information gathering agents. In *Artificial intelligence*. Amsterdam: Elsevier.
- Vaughan, R. (2008). Massively multi-robot simulation in stage. *Journal of Swarm Intelligence*, 2(2–4), 189–208.
- Vocabulary.com Online Dictionary, September 2015. Available at: <http://www.vocabulary.com/dictionary/patrol>.
- Yanovski, V., Wagner, I. A., & Bruckstein, A. M. (2003). A distributed ant algorithm for efficiently patrolling a network. *Algorithmica*, 37, 165–186.



David Portugal completed his graduate studies at the University of Coimbra in Portugal with a MSc degree in Electrical Engineering and Computing in 2009. He then pursued a PhD degree from the same University, in the area of Robotics and Multi-Agent Systems, which he obtained in March, 2014. His main areas of expertise are cooperative robotics, multi-agent systems, simultaneous localization and mapping, search and rescue robotics, field robotics, meta-heuristics, graph theory and optimization. Currently, he is working as an Experienced Researcher in Citard Services Ltd. (Cyprus) and his research is mostly focused on human-robot interaction, cognitive modelling, sensor fusion, and the ageing well domain. He has been involved in several local and EU-funded research projects in Robotics and Ambient Assisted Living, such as CHOPIN, TIRAMISU, Social-Robot, CogniWin and GrowMeUp, and he has co-authored over 45 research articles included in international journals, conferences and scientific books.



Rui P. Rocha completed his PhD degree on May 2006, at the Faculty of Engineering of the University of Porto. Between February 2000 and May 2006, he was a Teaching Assistant at the Department of Electrical and Computer Engineering, in the Faculty of Sciences and Technology of the University of Coimbra. Currently, he is an assistant professor at the Department of Electrical and Computer Engineering and a researcher at the Institute of Systems and Robotics, in the Faculty

of Sciences and Technology University of Coimbra. His main research topics are Cooperative Multi-Robot Systems, 3-D Map Building, Distributed Architectures and Intelligent Transportation Systems. He has been involved in several FP6 and FP7 European funded projects developed in consortium for the past few years and is currently the PI of a nationally funded research project about cooperation between human teams and robotics teams.