Robotics and Autonomous Systems 61 (2013) 1572–1587

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

Distributed multi-robot patrol: A scalable and fault-tolerant framework

David Portugal *, Rui P. Rocha

Instituto de Sistemas e Robótica (ISR), Universidade de Coimbra (UC), 3030-290 Coimbra, Portugal

HIGHLIGHTS

- Description of two distributed, scalable and effective Patrolling strategies.
- Definition of a Bayesian-inspired formalism, which provides Patrol adaptability.
- Qualitative comparison with the state of the art through realistic simulations.
- Implementation of a system for multi-robot patrol in a real-world scenario.
- Demonstration of robustness against robot failures and communication errors.

ARTICLE INFO

Article history:
Received 20 December 2012
Received in revised form
14 June 2013
Accepted 28 June 2013
Available online 11 July 2013

Keywords:
Distributed systems
Multi-robot patrol
Scalability
Fault-tolerance
Security

ABSTRACT

This paper addresses the Multi-Robot Patrolling Problem, where agents must coordinate their actions while continuously deciding which place to move next after clearing their locations. This problem is commonly addressed using centralized planners with global knowledge and/or calculating *a priori* routes for all robots before the beginning of the mission. In this work, two distributed techniques to solve the problem are proposed. These are motivated by the need to adapt to the changes in the system at any time and the possibility to add or remove patrolling agents (*e.g.*, due to faults).

The first technique presented is greedy and aims to maximize robot's local gain. The second one is an extension of the former, which takes into account the distribution of agents in the space to reduce interference and foster scalability.

The validation of the proposed solution is preliminarily conducted through realistic simulations as well as experiments with robot platforms in a small lab scenario. Subsequently, the work is verified in a large indoor real-world environment with a team of autonomous mobile robots with scalability and fault-tolerance assessment.

1. Introduction

Over the past couple of decades, research in multi-robot systems (MRS) has witnessed progress as never before. More particularly, 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 [1].

This work addresses MRS for cooperative patrolling missions in realistic scenarios. To patrol is herein defined as "the activity of going around or through an area at regular intervals for security purposes" [2]. It requires every position in the environment, or at least the ones that need surveillance, to be regularly visited, assuring a minimum frequency for verifying the existence of intruders or other anomalies. Being monotonous and repetitive, these missions may also be dangerous (e.g., patrolling in hazardous environments). Therefore, using MRS in this context can be advantageous to secure human lives in applications like mine clearing [3], search and rescue operations [4] or surveillance [5], enabling human operators to be occupied in nobler tasks like monitoring the system from a safer location [6].

In the coverage problem, the environment is usually modeled as a grid-like map requiring the team of robots to sweep all cells of the environment. Conversely, in the area patrolling problem, it is common to abstract the environment through a topological, graph-like map and robots are expected to have improved sensing abilities, meaning that they need to visit regularly all important places in the environment without necessarily going everywhere. Thus, the Multi-Robot Patrolling Problem (MRPP) requires coordination of agents decision-making with the ultimate goal of achieving optimal group performance. Additionally, it aims at monitoring envi-

* Corresponding author. Tel.: +351 239796303.
E-mail addresses: davidbsp@isr.uc.pt (D. Portugal), rprocha@isr.uc.pt (R.P. Rocha).

0921-8890/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.robot.2013.06.011

ronments, 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.

To approach this problem, we propose two distributed solutions that make their decisions using a Bayesian-based mathematical formalism to coordinate the team of mobile robots. As will be seen, the main advantages of this framework are the adaptability to the system's needs, which results from providing the robots with suitable autonomy; the straightforwardness of implementation; and the quality of the technique when placed in comparison against other techniques, leading to a scalable and fault-tolerant solution.

In the next section, a literature review is conducted and the contributions of the paper are described. Afterwards, the MRPP is defined and the performance metric is presented in Section 3. The following section describes the two distributed multi-robot patrolling strategies proposed in this paper. Sections 5–7 present the results obtained both through simulations and hardware experiments, as well as a discussion of the facets of the problem. Finally, the article ends with conclusions and open issues for future research.

2. Related work

Research on the patrolling problem with multiple robots has focused on three different fronts: adversarial patrol (cf., [7,8]), perimeter patrol (cf., [9,10]) and area patrol (cf., [11,12]). In adversarial patrol, the team of robots assumes the existence of an intruder and the aim is to coordinate the searchers to capture the opponent as quickly as possible. On the other hand, the main focus of perimeter and area patrol is to guarantee frequent visits to strategic places in the environment for security purposes. However, in perimeter patrol, the agents only move in the boundaries of the environment, whilst in area patrol, agents conduct their tasks throughout the environment. Henceforth, the focus is mainly on the latter and the expression MRPP is used to refer to this particular problem.

Several theoretical contributions to the MRPP have already been presented [11,13,14] and it has been shown that the problem is NP-hard. Nevertheless, based on a topological representation of the environment and using global/centralized information, it is commonly accepted in the literature that optimal patrolling can be obtained if all robots follow the same TSP[1] or Hamilton cycle, equally distributed in time and space [15,16]. However, these cycles are not trivial to compute in sparse topologies (the case of most real world environments), not even existing in most cases. In addition, according to [17], "these classical optimization problems do not capture the repetitive, and hence dynamic, aspect of the patrolling problem, nor the synchronization issues that arise when a timing among the visits of certain zones is required".

Beyond these contributions, several authors have proposed distinct solutions for multi-robot coordination in patrolling missions based on a variety of concepts. Simple pioneer architectures, using agents guided to locations that have not been visited for a while, were firstly introduced in [18] and further explored in [19]. These architectures differ on the basis of their agents' capabilities, perception, communication and decision-making. Although assuming several simplifications in the simulation environment used, the authors concluded that, in some cases, simple strategies with reactive agents, even without communication capabilities, can achieve equivalent or improved performance when compared to more complex ones. Following these studies, other models have been proposed subsequently. Numerous works explore graph theory tools like spanning trees [20,21] or graph partitioning [22,23] to compute minimal-cost cycles that assign efficient routes for each robot in the patrolling mission.

Several other models have been proposed to coordinate the patrolling agents. In [24], the problem is seen in a task allocation perspective, where each robot is assigned a different region to visit. Similarly, in [25], the authors make use of group-level task allocation in an area reconnaissance scenario, where multiple robots cooperate by sharing the subcomponents of a task through task tree auctions. The concept of auctions and market-based coordination is also explored in other works, such as [26,27], where agents bid to exchange vertices of the patrol graph to increase overall patrol effectiveness. In [28], the patrolling task is modeled as a reinforcement learning problem in an attempt to allow automatic adaptation of the agents' strategies to the environment. In summary, 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. Reinforcement learning is also adopted by Ishiwaka et al. [29] to model the behavior of a team of agents, which aim at predicting the location of their teammates as well as the movement direction to a common target. In identical surveillance application scenarios, where the threat of having adversarial agents is an important issue, the problem is commonly addressed using game theory tools [8,30]. Moreover, in [31] the patrolling problem is casted as a multi-agent Markov decision process, where a reactive and a planning-based technique are compared. The authors conclude 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. Chen and Yum [32] also formulated the problem as a Markov decision process and proposed a patrol routing strategy under a finite horizon approximation. In addition, reactive agents and swarm-based solutions to the patrolling problem have been studied by [33,34], where it is assumed that agents have limited computational power, communication abilities and memory storage. Finally, other reactive-driven methods have been proposed by [35,36] using the concept of artificial forces. For a more detailed survey of multi-robot patrolling strategies, the interested reader should refer to [37].

Despite the diversity of techniques proposed, there is an evident lack of implementation using physical MRS. Only sporadic studies have gone beyond simulations. In fact, all simulators use simplifying approximations to some extent. In some cases, this may jeopardize the validity of the outcome, which departs from observations in the real world. Furthermore, the MRPP is mainly a practical problem and it is essential to validate convincing real world solutions as well as comparing different techniques.

In the past, Cabrita et al. [38] successfully employed a team of Roomba robots, which navigated through indoor corridors aiming at monitoring the environment by collecting samples of alcohol concentration and temperature. To this end, the Multilevel Subgraph Patrolling (MSP) algorithm, which partitions the environment in regions and assigns a region to each robot [39], was adopted. Similarly, Iocchi et al. [40] tested both a cyclic and a partitioning strategy, addressing coordinated robot behavior. Their experiments were validated using realistic simulations as well as employing Erratic platforms in an indoor environment. Finally, Pasqualetti et al. [17] focused on constructing tours using graph-theoretic techniques, instructing the robots to travel according to an *Equal-Time-Spacing* trajectory. Experiments were conducted in an indoor lab scenario also using Erratic mobile robots.

[1] TSP stands for the well-known Traveling Salesman Problem (a NP-hard problem).

These approaches have in common the fact that the patrolling routes for each robot were computed *a priori* by a centralized entity using global information and later passed on to the robots. Predefined routes may ensure good patrol performance in several applications, however deterministic solutions ease the task of an intruder that aims to break into the environment [7]. In contrast, we extend our previous work [41] by validating a distributed coordination approach in a large indoor real-world environment, where fully autonomous agents decide locally and sequentially their patrol routes according to the state of the system, without requiring a central planner. It is shown that agents can coordinate effectively, using distributed communication, independently of the number of robots in the team. Additionally, it is also demonstrated that the approach is robust to robot failures.

In a previous study [12], it was concluded that research should be oriented towards multi-robot patrolling strategies that minimize the effect of interference between agents in order to increase the team's scalability. Hence, preliminary Bayesian-based techniques are herein proposed to assist the agents local decision-making process according to the state of the system in their neighborhood as well as the positions of other teammates. This framework is adopted due to its proven efficiency when handling problems that deal with uncertainty [42,43]. In this work, the models proposed are simple and can easily be reproducible and expanded in the future. Also, the focus is especially put on practical experimentation and showing that simple models, as those proposed in the article can attain exceeding results in the field. To summarize, the contributions of this work to the state-of-the-art are as follows:

- description of two distributed and scalable approaches to the MRPP, whose effectiveness is attested in the experiments conducted.
- definition of a Bayesian-inspired mathematical formalism using conditional probability distributions in the context of MRPP, providing adaptability to the system and the flexibility to add and remove decision variables.
- qualitative comparison against several approaches in the state of the art, in terms of performance and scalability, showing important advantages of the proposed solutions by means of simulations using Stage/ROS [44].
- the work is initially verified with low-cost platforms in a lab scenario and an implementation of a system for multi-robot patrol in a real-world scenario is presented.
- beyond the good performance and ability to scale to larger teams, we demonstrate that the system is robust to robot failures and communication errors, and we also show that simulations conducted are realistic and present similar results to real world tests.

This paper is a culmination of two previous works [45,41], yet it distinguishes itself from these previous publications by presenting a more thorough and detailed literature survey and the development of a fully distributed system, carrying out new simulation experiments and experiments in an indoor infrastructure with detailed discussion of implementation aspects. Also novel is the assessment through real world experiments on the adaptability, scalability and fault-tolerance nature of the approach, as well as the assessment of the realism of Stage/ROS simulations and the analysis of how communication errors affect the system's performance.

3. Problem definition

In this work, the problem of efficiently patrolling a given environment with an arbitrary number of robots is studied. Agents are assumed to have an *a priori* map of the environment and through a graph extraction algorithm [46], they obtain an undirected, connected and metric navigation graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $v_i \in \mathcal{V}$ vertices and $e_{i,j} \in \mathcal{E}$ edges. 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. $|\mathcal{V}|$ represents the cardinality of the set $\mathcal{V}$ and $|\mathcal{E}|$ represents the cardinality of the set $\mathcal{E}$. Seeing as undirected graphs are assumed, then: $|\mathcal{E}| \leq \frac{|\mathcal{V}|\cdot(|\mathcal{V}|-1)}{2}$.

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 vertices of the graph, ensuring the absence of atypical situations with respect 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 access the effectiveness of multi-robot patrolling strategies. Typically, these are based on the idleness of the vertices, the frequency of visits or the distance traveled by agents [40]. In this work, the first one has been considered [12], given that it measures the elapsed time since the last visit from any agent in the team to a specific location. Idleness is intuitive to analyze and brought into confrontation with the possibility of attacks to the system, seen as it uses time units. Thus, 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:

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

where 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:

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

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:

$$\overline{\mathcal{I}_{\mathcal{V}}} = \{\overline{\mathcal{I}_{v_1}}, \ldots, \overline{\mathcal{I}_{v_i}}, \ldots, \overline{\mathcal{I}_{v_{|\mathcal{V}|}}}\}, \tag{3}$$

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

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

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 [11,18] 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, when the mission started. As a consequence, there is a transitory phase in which the $\overline{\mathcal{I}_{\mathcal{G}}}$ values tend to be low, not corresponding to the reality in a steady-state phase, 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 can be described as the problem of finding a set of paths $\mathbf{x}$ which visit all vertices $v_i \in \mathcal{V}$ of the graph $\mathcal{G}$, using an arbitrary team of R robots, with the overall team goal of minimizing $\overline{\mathcal{I}_{\mathcal{G}}}$:

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

by finding:

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

such that:

$$x_r = \{v_a, v_b, \ldots\}, \tag{8}$$

$$v_a, v_b, \ldots \in \mathcal{V},$$

$$1 \leq r \leq R, R \in \mathbb{N},$$

subject to:

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

Note that x_r represents the patrolling path of robot r, which has an arbitrary dimension that depends on each robot's decisions and $v_a, v_b, \ldots$ are generic vertices in $\mathcal{V}$, which do not imply any specific order.

In this work, instead of relying in precomputed offline routes, which is common in classical approaches, the patrolling route x_r of each robot is built online according to the state of the system. Furthermore, all robots are endowed with autonomous decision-making capabilities, being able to decide their own moves, instead of following routes that are computed by a centralized entity.

4. Distributed patrolling strategies

In this section, the proposed distributed strategies for the MRPP are presented. Based on a preliminary Bayesian-based formalism, a model was developed to support the local decision-making process of each robot when patrolling the environment. More specifically, the model represents the decision of moving from one vertex of the graph to another. Consider the degree of a vertex v_i (i.e., the number of adjacent vertices of v_i) as $deg(v_i) = \beta$. For β neighbors, the model is applied independently 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. Thus, each robot's patrol route is built progressively, at each decision step, adapting to the system's needs; i.e., aiming at minimizing the average graph idleness time ($\overline{\mathcal{I}_\mathcal{G}}$). Special focus is given in the next sections to the selection of proper statistical distributions to model the data, in order to ensure the quality of the results [47].

Additionally, it is worth noting that the MRPP is NP-hard. In fact, Chevaleyre [11] provided the first theoretical analysis of the patrol problem, using the idleness performance criterion on an undirected metric graph, similarly to the problem defined in Section 3. He was able to prove that it can be optimally solved using a TSP Tour. However, finding the optimal path in the TSP is NP-Hard [48,49]. Pasqualetti et al., which used the concept of "refresh time" as a synonym for "idleness", proved afterwards that the team refresh time problem is NP-hard (cf. Theorem II.1 in [16]), i.e., given a generic roadmap and a team of robots, finding a trajectory which minimizes team refresh time is NP-hard. This was proven through reduction from the Traveling Salesman Problem. Moreover, for the metric problem instances[2] of the TSP, there are several adequate approximation algorithms. For instance, Christofides Algorithm [50] is able to compute a tour no longer than 3/2 times the optimal in $O(|\mathcal{V}|^3)$ computation time. In addition, the Lin–Kernighan heuristic [51], typically finds tours within 5% of the optimal in $O(|\mathcal{V}|^{2.2})$ computation time.

Due to the NP-hardness nature of the MRPP, i.e., no polynomial time algorithm is known to compute an optimal solution to the problem, we attempt to solve it in a distributed way by proposing two approaches based on heuristics. The effectiveness of the approaches is shown later in Section 5, when compared with other approaches in the literature.

[2] In metric problem instances, the edge weights satisfy the triangle inequality as in the case of metric graphs.

4.1. Greedy Bayesian strategy

Greedy strategies have been successfully used in several optimization problems, where finding a global optimum in reasonable time bounds is impracticable. The idea behind such strategies is to find the locally optimal choice at each stage. Based on this concept and on Bayes rule, the Greedy Bayesian Strategy (GBS) is herein described.

After reaching a vertex v_0 of the navigation graph, each robot is faced with a decision stage, where it must decide the direction it should travel next, among all β adjacent vertices. To that end, we define two fundamental random variables. The first one represents the act of moving (or not) to a neighbor vertex v_A:

$$move(v_A) = \{true, false\}, \tag{10}$$

while the second variable used in GBS represents the Gain G_A of moving from the current vertex (v_0) to a neighbor vertex (v_A), assuming constant speed (c) as:

$$G_A(t) = c \cdot \left(\frac{\mathcal{I}_{v_A}(t) - \mathcal{I}_{v_A}(t + \Delta t)}{|e_{val}|} \right), \tag{11}$$

where $t + \Delta t$ is the arrival time in v_A, and $\Delta t = |e_{0A}|/c$. $G_A(t)$ is proportional to a difference in the idlenesses values, representing a gain that the robot expects to obtain in moving to a given vertex. Note however that $G_A(t) \geq 0$ because $\mathcal{I}_{v_A}(t + \Delta t) = 0$, when the robot reaches v_A. Wherefore (11) is equivalent to:

$$G_A(t) = c \cdot \frac{\mathcal{I}_{v_A}(t)}{|e_{val}|}. \tag{12}$$

For simplicity of notation, hereafter we write G_A instead of $G_A(t)$, since every computation is done instantaneously.

In most cases, $|e_{val}|$ takes on the value of $|e_{0A}|$, which is the distance between the two vertices, given by the weight of the edge that connects v_0 to v_A. However, constraint (13) is imposed in order to dimension $|e_{val}|$, avoiding occasional situations where robots may get trapped in local optima (i.e., repeatedly visiting vertices that are very close to each other):

$$|e_{val}| = \begin{cases} |e_{\min}|, & \text{if } \max\{e_{0A}, \ldots, e_{0\beta}\} \\ & > 2 \min\{e_{0A}, \ldots, e_{0\beta}\} \wedge |e_{0A}| < |e_{\min}| \\ |e_{0A}|, & \text{otherwise.} \end{cases} \tag{13}$$

In this work, robots update the instantaneous idleness time values online, by communicating to other robots when they reach another vertex of the navigation graph, in a distributed way. Furthermore, in GBS agents are self-interested and the routes that they take depend on the gain that they expect to obtain. Agents calculate the probability of moving to a specific vertex i given its gain, applying Bayes rule:

$$P(move(v_i)|G_i) = \frac{P(move(v_i))P(G_i|move(v_i))}{P(G_i)}, \tag{14}$$

$P(move(v_i))$ represents prior knowledge or assumptions in the problem. For example, certain vertices of the graph may require higher visit frequency than others; this situation would be codified as prior information. In this work, the prior is defined as uniform, where all decisions are equiprobable. $P(G_i|move(v_i))$, i.e., *likelihood*, is a statistical distribution modeling the gain according to the variable $move(v_i)$. The denominator term is regarded as a normalization factor [47], being often omitted for simplification purposes.

Gain (G_i) is a continuous random variable with a probability density function $f(g)$. Therefore, the probability that G_i takes on a value less than or equal to g is given by:

$$P(G_i \leq g) = \int_{-\infty}^{g} f(g)\, dg = \int_{0}^{g} f(g)\, dg = F(g), \tag{15}$$

$$\text{with: } G_i \in [0, \infty]. \tag{16}$$

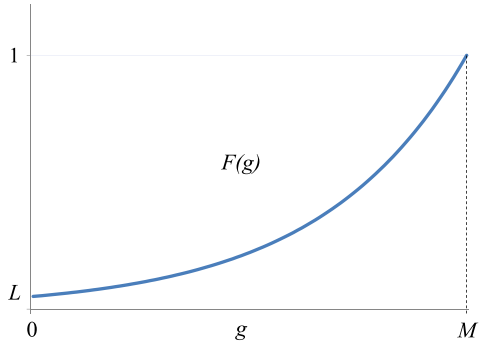


Fig. 1. Distribution function of gain, given that *move* is true.

Note that $F(g)$ is the distribution function of G_i. We define it as a monotonically increasing function, where higher values of gain become rapidly more influential on the robot's decision; therefore the distribution function follows the exponential model seen on Fig. 1:

$$F(g) = ae^{bg}; \quad a > 0, \tag{17}$$

$$\text{where: } F(0) = L \Leftrightarrow a = L, \tag{18}$$

$$\text{and: } 1 = Le^{bM} \Leftrightarrow b = \frac{\ln(1/L)}{M}. \tag{19}$$

This results in:

$$F(g) = L \cdot \exp\left(\frac{\ln(1/L)}{M} g\right), \tag{20}$$

$$\text{with: } L, M > 0 \text{ and } g < M. \tag{21}$$

L and M are constants that control the distribution function. More specifically, L is the y-intercept, which controls the probability values for lower gains and M is the gain saturation, beyond which the probability values are maximum; $F(g \geq M) = 1$. These constants are simply defined as a value close to 0 for L, e.g., 0.1 was used in the experiments; and M is calculated through (8) using an upper bound of $\mathcal{I}_{v_A}$. Finally, the probability density function $f(g)$ is obtained by differentiating $F(g)$:

$$f(g) = F'(g) = \frac{1}{M} \cdot \ln(1/L) \cdot \exp\left(\frac{\ln(1/L)}{M} g\right). \tag{22}$$

Now that the distribution model is defined, $P(move(v_i)|G_i)$ can be estimated via (14), for each vertex involved in the decision process. In Algorithm 1 a high-level pseudo-code of the GBS approach, running locally on a robot, is presented. Since the model assumes a uniform prior and considers only one likelihood function, which is fixed, the decisions taken in GBS are equivalent to moving to the adjacent vertex with maximum instantaneous idleness. However, we use the previous formalism so as to easily add a new variable to the model in the next section and denote its flexibility.

4.2. State exchange Bayesian strategy

In collective operations with a common objective, coordination between agents plays a fundamental role in the success of the mission. In the previously described strategy, robots are only interested in obtaining the best reward for themselves, neglecting the global objective of the patrolling mission by acting independently of their teammates; despite communicating every time they reach a goal in order to update the instantaneous idleness tables, they do not assist each other when making their decisions. Expected to perform well in most situations, GBS may present problems in environments where the ratio of robots per area is high, because agents will tend to compete to arrive to the same region.

Algorithm 1: Greedy Bayesian Strategy (GBS)

```
while true do
    add(v_n to x_r);   // current vertex
    write_msg_arrival_to(v_n);
    forall the v_i ∈ N_G(v_n) do
        G_i ← c((I_{v_i}(t) − I_{v_i}(t+Δt)) / |e_ni|);
        P(G_i|move(v_i)) ← L · exp((ln(1/L)/M) G_i);
        P(move(v_i)|G_i) ← P(move(v_i)) P(G_i|move(v_i)) / P(G_i);
    // Next vertex is the neighbor of the current vertex with highest
       posterior probability.
    v_{n+1} ← argmax(P(move(v_i)|G_i));
    while move_robot to v_{n+1} do
        read_msg_arrival_to(V);
        update(I_V(t));
    v_n ← v_{n+1};
```

Consequently, GBS has been extended to account for the reduction of interference between robots in the patrolling mission. Hence, in the State Exchange Bayesian Strategy (SEBS), we define Vertex state S_i as a discrete variable that represents the number of robots that *intend* to visit a given vertex v_i involved in the decision process of robot r, which is currently located in vertex v_0:

$$S_i \in \mathbb{N}_0 \cap [0, R-1], \quad R > 1. \tag{23}$$

Logically, the definition of this variable implies a mechanism for each robot to track the intentions of teammates in their neighborhood. One possibility would be to endow the robots with some kind of sensor to obtain information of the vertices in their neighborhood. Yet, another possibility seems more advantageous in this context, which is for the robots to take advantage of their distributed communication mechanism not only to send their current location in the navigation graph, but also to inform other robots where they have decided to move next. With this approach, robots are capable of computing the state directly by collecting other robot's intentions and checking the vertices involved in their decision process. This mechanism is expected to reduce interference, as it becomes less likely for two or more robots to move to the same place.

Similarly to GBS, it is necessary to define a statistical distribution to model the vertex state. The greater the number of teammates in the vicinity of a robot, it becomes increasingly unlikely for the robot to move in that direction. To describe this behavior, the following probability mass function, which uses a geometric sequence of ratio 1/2 has been defined:

$$f_{S_i}(s)_{R\to\infty} = P(S_i = s)_{R\to\infty} = \frac{1}{2^{s+1}}, \tag{24}$$

as shown in Fig. 2. This geometric sequence is used to guarantee that the total probability for all S_i equals 1:

$$\sum_{s=0}^{R-1} f_{S_i}(s) = 1. \tag{25}$$

Eq. (24) assumes that the number of robots R is unknown and can be arbitrarily high. However, since the robots communicate among themselves, it is more realistic to consider R as known and with finite values. Therefore, the following approximation to (24) is assumed:

$$f_{S_i}(s) = P(S_i = s) = \frac{2^{R-(s+1)}}{2^R - 1}; \quad R > 1, \tag{26}$$

which still holds condition (25). With the discrete probability distribution model characterized, robots can now decide moving to a specific vertex given its gain and state:

$$P(move(v_i)|G_i, S_i) \propto P(move(v_i))P(G_i|move(v_i)) \times P(S_i|move(v_i)). \tag{27}$$

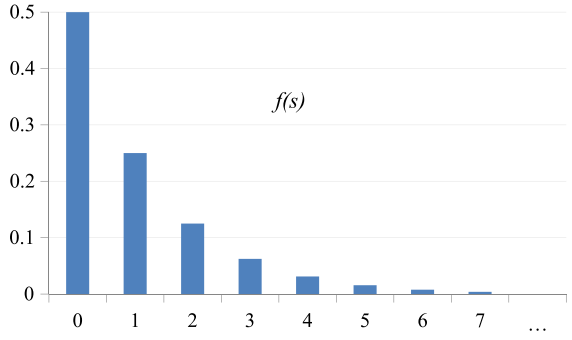


Fig. 2. Distribution function of the vertex state, given that *move* is true.

Algorithm 2: State Exchange Bayesian Strategy (SEBS)

while *true* **do**
  `add(`v_n *to* x_r`);` // current vertex
  forall the $v_i \in N_G(v_n)$ **do**
    $G_i \leftarrow c \cdot \left(\frac{\mathcal{I}_{v_i}(t)-\mathcal{I}_{v_i}(t+\Delta t)}{|e_{ni}|}\right)$;
    $P(G_i|move(v_i)) \leftarrow L \cdot \exp\left(\frac{\ln(1/L)}{M} G_i\right)$;
    $S_i \leftarrow$ `count_intentions_to(`v_i`)`;
    $P(S_i|move(v_i)) \leftarrow \frac{2^{R-(S_i+1)}}{2^R-1}$;
    $P(move(v_i)|G_i, S_i) \leftarrow \frac{P(move(v_i))P(G_i|move(v_i))P(S_i|move(v_i))}{P(G_i)P(S_i)}$;
  // Next vertex is the neighbor of the current vertex with highest posterior probability.
  $v_{n+1} \leftarrow$ `argmax(`$P(move(v_i)|G_i, S_i)$`)`;
  `write_msg_arrival_to(`v_n`)`;
  `write_msg_intention_to(`v_{n+1}`)`;
  while *move_robot to* v_{n+1} **do**
    `read_msg_arrival_and_intentions_to(`$\mathcal{V}$`)`;
    `update(`$\mathcal{I}_{\mathcal{V}}(t)$`)`;
  $v_n \leftarrow v_{n+1}$;

Algorithm 2 presents a high-level pseudo-code of the SEBS approach, which runs locally on each robot.

5. Experimental validation

5.1. Benchmarking: simulation experiments

In order to assess the performance of the two patrolling techniques proposed in this article and compare them with other techniques in the literature, simulation trials using the Stage multi-robot simulator [52] together with ROS [44] were conducted.

In these simulations, the graph information of a given environment is loaded by every robot in the beginning, which then runs one of the algorithms. The robots' dynamics is considered, as they navigate safely in the environment by heading towards their goals and avoiding collisions with walls and other robots through the use of ROS's navigation stack with simulated odometry and a probabilistic localization system, more specifically the adaptive Monte Carlo localization (AMCL) approach [53]. Note that this dynamic is implicit in both patrolling strategies, when robots move, and has a non-negligible effect on results. For all these reasons, we deem these simulation experiments to study the MRPP as fairly realistic. This is later confirmed on Section 7. In addition, all robots have the same nominal speed, reaching a maximum velocity of 0.2 m/s and communicate using a distributed publish/subscribe messaging system.

Fig. 3 presents three environment topologies with different algebraic connectivity or Fiedler value λ [54], a well-known metric of the connectivity of a graph given by the smallest non-zero eigenvalue of the graph's Normalized Laplacian matrix [12]. These topologies were used in [45], 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. While collecting results in different scenarios, the same simulation setup and initial positioning of the robots have been used.

Both GBS and SEBS were tested in all three environments with different teamsizes (1, 2, 4, 6, 8 and 12).[3] Simulations stopped when the value of the average graph idleness ($\overline{\mathcal{I}_{\mathcal{G}}}$) after each patrol cycle p converged with, at most, a 2.5% difference to the previous cycle, where each vertex has been visited at least p times. The values of the minimum edge weight threshold $|e_{\min}|$ for each graph were determined experimentally, being 3.75 m and 3 m for map A and B, respectively. For map C, since all edges have the same weight, it is not necessary to define $|e_{\min}|$, because constraint (13) does not apply. Table 1 presents the results obtained, wherein $\overline{\mathcal{I}_{\mathcal{G}}}$ was measured in seconds and was used as the performance metric.

Analyzing Table 1, both strategies have approximately the same performance when using one robot (maximum difference of 0.9%). In this case, the strategies are equivalent, because in SEBS there are no teammates to share goals and intentions to. The differences in performance are more noticeable when teamsize starts to rise, especially in teams of 6 or more robots. One explanation for this phenomenon is the growing interference between robots as teamsize R increases, which is shown in Fig. 4.

The interference between robots is measured as the overall frequency of different agents sharing nearby areas, having to avoid each other, in every experiment. Given that the interference is zero for experiments with one robot; from two robots on, GBS always presents higher levels of interference, when compared to SEBS. This happens because, occasionally, in cluttered areas robots have to compete to reach the same goal when adopting GBS.

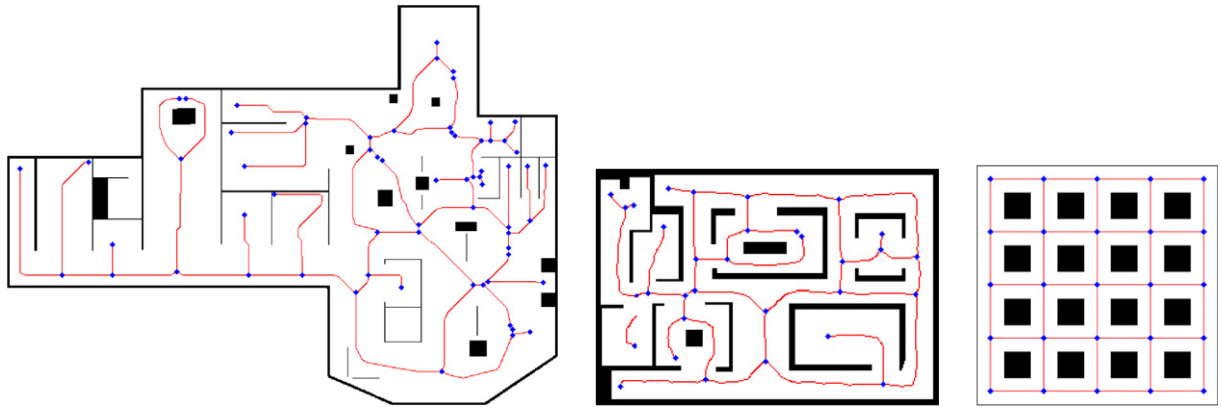
On the other hand, SEBS is an evolution of GBS, in which robots take their teammates goals and intentions into consideration when deciding their next move. This results in differences in performance for teams of 12 robots of up to 22.22%, between both algorithms. Even though the performance of SEBS is superior, as expected, it is worth noticing that being a simpler strategy, GBS requires less exchange of information and also presents interesting results.

In [45], we have conducted benchmark tests with several state-of-the-art patrolling approaches using the same three environments and the same performance metric. In brief, Conscientious Reactive (CR) [18], Heuristic Conscientious Reactive (HCR) and Heuristic Pathfinder Conscientious Cognitive (HPCC) [19] are three pioneer approaches based on distributed and reactive agents with simple behavior and no explicit communication between robots. Cyclic algorithm for Generic Graphs (CGG) and Multilevel Subgraph Patrolling (MSP) algorithm are two centralized and deterministic strategies based on [11,39], which use graph theory tools to find long cycles (CGG) or partitions (MSP) in the graph for patrolling purposes.

In fact, also including the strategies presented in [45] in the performance comparison, Tables 1–4 show that GBS is the second best strategy, performing slightly better than Conscientious Reactive (CR) and only staying behind SEBS, which is the top performing strategy tested so far. For every (*teamsize, map*) pair, SEBS performance is always in the top 3, considering the total of 7 approaches in [45] together with this work, which demonstrates the great adaptability of SEBS in each situation and the potential of employing Bayesian inspiration in the MRPP.

In terms of connectivity, even though they perform well in lowly and mildly connected environments, where generally SEBS is

[3] The simulation code is available at http://www.ros.org/wiki/patrolling_sim.



(a) Environment A. (b) Environment B. (c) Environment C.

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

Table 1
Overview of the simulation results with GBS and SEBS ($\overline{\mathcal{I}_{\mathcal{G}}}$ values in seconds).

Teamsize	Map A		Map B		Map C	
	GBS	SEBS	GBS	SEBS	GBS	SEBS
1	1718.93	1703.68	1267.26	1277.16	670.29	676.30
2	836.05	812.68	708.82	671.18	343.89	338.97
4	464.18	438.16	351.19	339.93	182.89	167.16
6	353.15	329.18	275.98	230.39	147.66	125.06
8	295.58	251.91	206.19	197.03	116.14	103.45
12	253.89	226.90	145.89	118.73	90.42	70.33

Table 2
Final $\overline{\mathcal{I}_{\mathcal{G}}}$ values (in seconds) using different state-of-the-art strategies on environment A.

Teamsize	CR	HCR	HPCC	CGG	MSP
1	1734.09	1962.42	1740.37	1717.36	1704.36
2	843.93	1146.27	791.20	845.49	930.04
4	433.38	652.84	434.11	451.70	476.92
6	367.11	506.90	377.73	348.46	381.97
8	271.70	442.39	361.62	288.72	253.19
12	287.14	412.65	352.79	265.47	183.74

Table 3
Final $\overline{\mathcal{I}_{\mathcal{G}}}$ values (in seconds) using different state-of-the-art strategies on environment B.

Teamsize	CR	HCR	HPCC	CGG	MSP
1	1315.79	1283.59	1235.67	1347.30	1401.80
2	675.44	654.61	670.44	675.64	749.42
4	363.46	373.45	298.77	335.45	375.15
6	238.57	273.60	254.96	234.18	248.92
8	198.90	217.38	225.44	172.39	185.28
12	172.40	255.62	212.30	143.94	–

Table 4
Final $\overline{\mathcal{I}_{\mathcal{G}}}$ values (in seconds) using different state-of-the-art strategies on environment C.

Teamsize	CR	HCR	HPCC	CGG	MSP
1	715.30	714.23	737.93	767.25	766.41
2	353.06	351.15	358.45	385.09	423.60
4	193.30	186.59	188.03	200.53	209.82
6	141.68	138.64	135.74	142.94	148.09
8	104.00	108.45	118.75	113.71	95.22
12	101.82	105.64	118.36	94.35	–

the leading approach; the results in the highly connected topology (map C) are excellent, outperforming by far the other approaches with the exception of the Multilevel Subgraph Patrolling (MSP) algorithm for higher teamsizes.

In order to analyze how well different strategies scale, Balch's speedup measure [55], a classical scalability metric, was calculated for each strategy:

$$\upsilon(R) = \frac{\Psi(1)/R}{\Psi(R)}, \tag{28}$$

where $\Psi(R)$ is the performance for R robots, given by $\overline{\mathcal{I}_{\mathcal{G}}}$. Fig. 5 presents a chart comparing the speedup for each strategy (including those in [45]) in map A. It can be seen that most systems enter progressively in sublinear performance ($\upsilon(R) < 1$) with teamsize, due to the more frequent existence of spatial limitations, which, in turn, increases the interference between robots causing the performance to decrease. Looking closely at these results, the two proposed strategies are less affected by teamsize, when compared to other approaches. Both perform effectively regardless of teamsize, outperforming all distributed approaches compared, which suggests that these strategies scale well, just staying behind of the MSP strategy, a centralized approach, which has particularly high performance with large teamsizes since it uses a graph partitioning scheme to assign separated patrol areas to each robot, thus drastically reducing the interference between robots. Note however, that the centralized approach adopted in MSP presents a scalability bottleneck as soon as the algorithm is no longer able to partition the graph in regions, e.g., MSP was not able to partition environment C in 12 regions, and its performance is generally inferior for smaller teamsizes.

These results show that the strategies proposed in this paper are highly scalable, when compared to other distributed strategies. Furthermore, the proposed methods are able to adapt to non-standard situations like robot failures or heterogeneous agents with different speed profiles. Incorporating robots that travel at different speeds with strategies that solve the MRPP with predefined global routes, such as CGG, would not be suitable because maintaining a uniform distance between each robot would be impossible unless all robots were limited to travel at the speed of the slower robot. Additionally, offline strategies (e.g., CGG

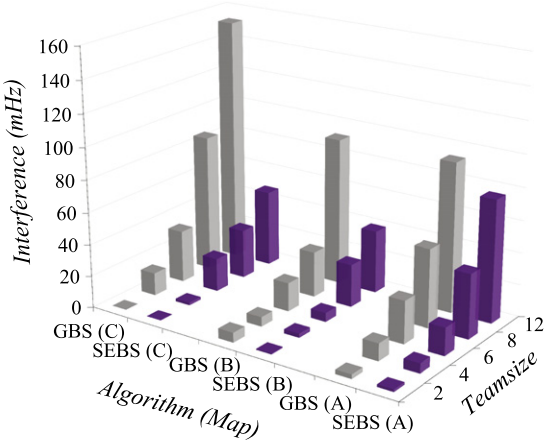


Fig. 4. Interference levels for all experiments.

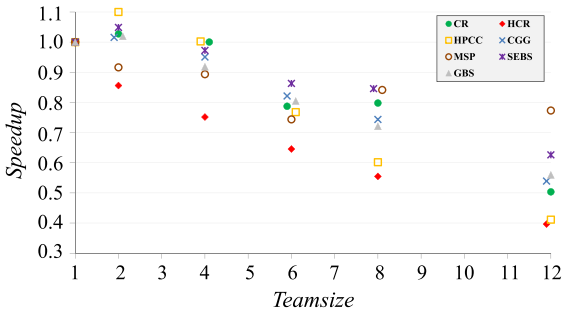


Fig. 5. Speedup comparison with other approaches in map A.

or MSP) are not able to account for robot failures unless some adaptive online behavior is provided.

5.2. Preliminary experiments in a lab scenario

Simulation experiments allow attesting and comparing empirically the performance of distinct patrolling strategies in several different scenarios and with large teams of robots, which is often not possible in the real world. However, the MRPP is mainly a practical problem and it is essential to validate physical solutions. Therefore, preliminary experiments with a team of TraxBot platforms [56,57] were conducted. These low-cost custom-made robots move at the same nominal speed and have low processing power, being equipped with an Arduino Uno board that incorporates a microcontroller ATmega 328p [58]. In addition, each robot represents a node of a self-configuring infra-structureless network, i.e., a mobile *ad hoc* network (MANET), being able to communicate with its teammates through the use of ZigBee modules, which fit on top of the Arduino board.

In these experiments, the SEBS technique is validated and it is shown that the algorithm does not need heavy computation power and does not rely on a specific communication paradigm. The lab scenario consisted of a highly connected 4×4 grid graph with 16 vertices and 24 edges, represented in the green carpet, as depicted in Figs. 6 and 7. Once again, $|e_{\min}|$ is not defined because all edges have the same weight.

Since the TraxBots have limited sensing abilities and computation power, an overhead camera facing the ground was mounted on top of the scenario at a height of around 4 m, tracking the robots' pose. The process includes background subtraction, detection of the robots' colored LEDs, position and heading calculation and image to real world coordinate transformation. The result of the system is the robot real position and heading, which is communicated through the Zigbee network. Up to three robots were deployed in the confined space, where they run the multi-robot patrolling algorithm.

One important result, which can be seen in the video of these experiments[4] and in Fig. 6, is that after the initial exploratory patrolling phase, where no vertices have been visited yet and no historic information is available, the robots tend to follow optimal TSP tours for the case of one and two robots, in this scenario. This is especially remarkable given that the robots decide their moves in an online and autonomous fashion.

As for the case of three robots, illustrated in Fig. 7, which is even more challenging, robots coordinate themselves via exchanging their intentions and reduce inter-robot interference by avoiding the same goals. As a consequence, this coordination leads to an effective patrolling scheme, where robots tend to compensate their teammates, sequentially covering regions that need to be visited. Note that, in this case, no optimal 3-way TSP tours exist; and the average number of moves, per robot, in order to complete a patrolling cycle is 5.8, which is almost optimal (the theoretical lower bound for the number of moves would be 5.33).

These preliminary results illustrate efficient coordination between robots that arise from executing the distributed algorithm. Additionally, as expected, it can be verified that the average moves per robot in each patrolling cycle decreases as teamsize grows, which is tantamount to saying that performance increases with teamsize.

6. Experiments in a real world environment

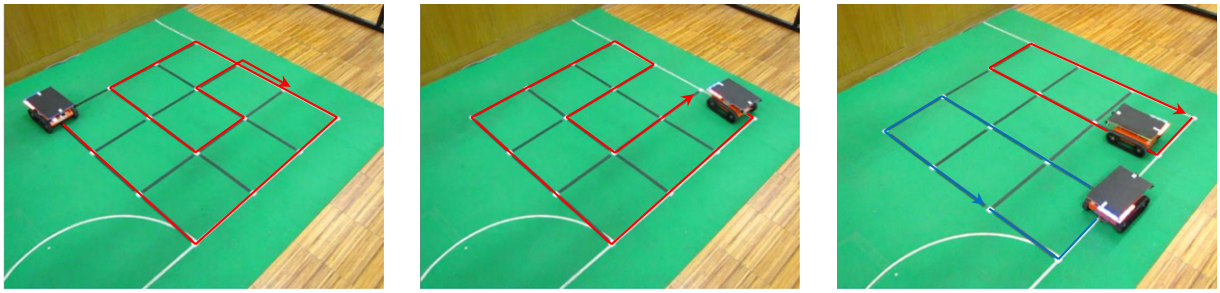
In order for distributed intelligence systems to be useful in the real world, it is necessary to go beyond lab experiments and prove the reliability of such systems in more demanding scenarios. In this section, the implementation of a system for multi-robot patrol in a real environment is presented. Aiming to fill a gap in the present state-of-the art, the SEBS distributed approach is validated in a real-world indoor scenario, where fully autonomous agents decide locally and sequentially their patrol routes according to the state of the system, as previously described. Beyond the coordination which arises from the distributed communication of agents, it is also shown that the approach is robust to robot failures, i.e., fault-tolerant.

In these experiments, not only is the average graph idleness along time, $\overline{\mathcal{I}_{\mathcal{G}}}$, examined, but also the median $\widetilde{\mathcal{I}_{\mathcal{G}}}$, standard deviation σ, and the maximum average idleness of a vertex along time, $\max(\overline{\mathcal{I}_{\mathcal{V}}})$.

Experiments were conducted in a large indoor scenario, namely the floor 0 of the Institute of System and Robotics (ISR), in the University of Coimbra, in Portugal. Fig. 8 shows a few snapshots of the corridors of the ISR and the extracted topological map on top of the 67.85×26.15 m environment, which was obtained using the algorithm in [46]. The resulting topology is a non-complete, connected and sparse graph, like most real world environments, as opposed to the graph of the laboratory scenario presented in Section 5.2.

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 three Pioneer-3DX robots [59], equipped with an Hokuyo laser in the front and a laptop on top was used, as seen in Fig. 9. Each laptop runs the ROS navigation stack using the Adaptive Monte Carlo (AMCL) algorithm for Localization as done previously in Stage simulations, being responsible for controlling the robot's motion. All robots

[4] A video of the experiments is available at: http://isr.uc.pt/~davidbsportugal/RAS.

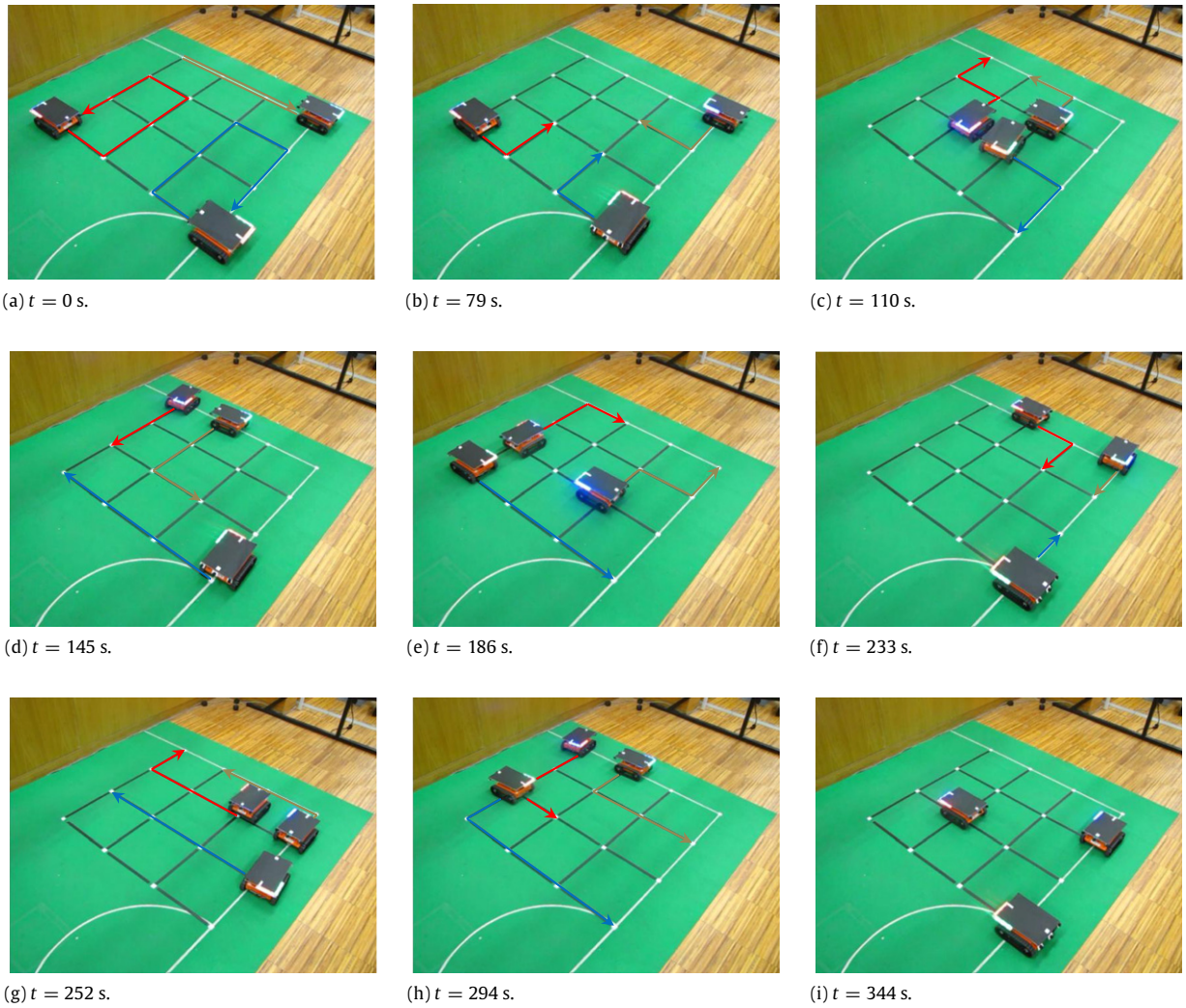


(a) One robot in the beginning of the experiment.

(b) Robot following an optimal TSP tour, at $t = 105$ s.

(c) Two robots following two-way TSP tours.

Fig. 6. Preliminary experiments with 1 and 2 robots. Arrows represent trajectories followed by robots.



(a) $t = 0$ s.

(b) $t = 79$ s.

(c) $t = 110$ s.

(d) $t = 145$ s.

(e) $t = 186$ s.

(f) $t = 233$ s.

(g) $t = 252$ s.

(h) $t = 294$ s.

(i) $t = 344$ s.

Fig. 7. Preliminary experiments with 3 robots. Arrows represent trajectories followed by robots.

have the same maximum speed of 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*). Multimaster communication is provided using the *wifi_comm*[5] package. This means that there is no central point of failure in the system.

A ROS node (i.e., a ROS application) has been programmed to announce the start of the mission and collect results during the experiments. These results are examined in the next section. Note that, this "monitor" node does not centralize the approach nor does it give feedback to the robots whatsoever. In fact, it does not even need to be running, being solely used for the two purposes referred before.

Firstly, experiments with one, two and three robots were conducted. Each experiment was repeated 3 times. Afterwards, in order to further demonstrate the scalability of the approach, virtual

[5] Available at http://www.ros.org/wiki/wifi_comm.

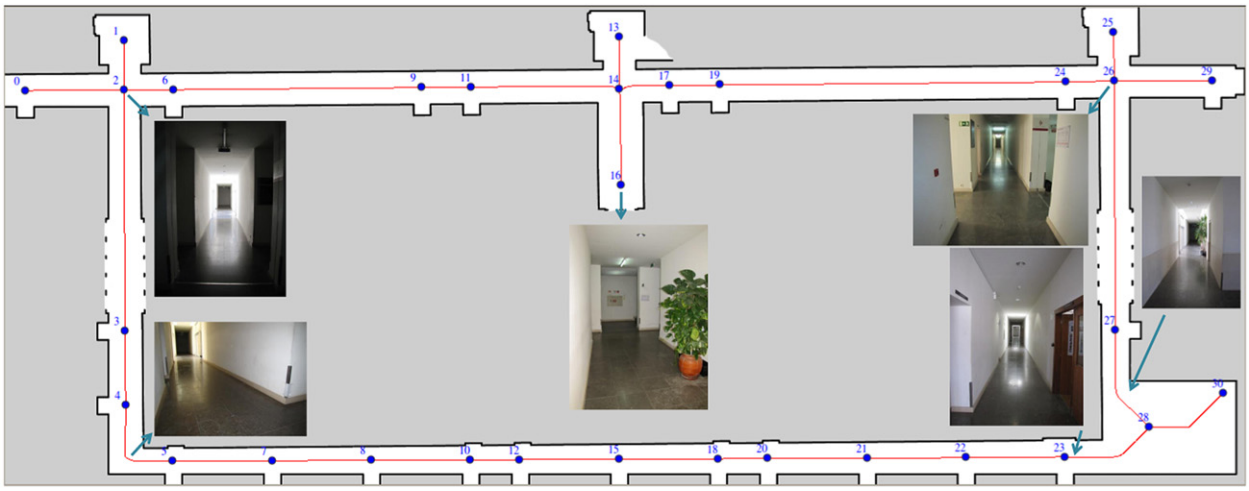


Fig. 8. Topological map of the "ISR-Floor0" environment.

Fig. 9. Robots used in the experiments.

Table 5
Experiments with 1–3 robots (all values in seconds).

Teamsize	$\overline{\mathcal{I}_\mathcal{G}}$	$\max(\overline{\mathcal{I}_\mathcal{V}})$	$\widehat{\mathcal{I}_\mathcal{G}}$	σ	τ
	336.676	412.207	370.994	78.769	1648.828
1	332.745	407.897	366.677	77.892	1631.590
	331.615	406.387	365.345	77.626	1625.550
	168.921	309.455	137.267	64.210	1237.821
2	180.761	296.085	180.293	56.064	1184.341
	170.267	328.300	146.890	62.603	1313.201
	128.875	273.670	116.269	54.893	1094.682
3	116.248	216.020	95.150	44.356	864.081
	112.954	200.030	101.923	36.066	800.121

robots were added to the team, and 3 trials with 6 agents (3 + 3) and 9 agents (3 + 6) were also conducted. Finally, to prove its robustness, experiments which included failures in the robots at different time instants are analyzed. In all experiments, we have used $|e_{\min}| = 7.5$ m.

Aiming at comparing the total time of the mission (τ) in various conditions, each experiment finishes after 4 complete patrolling cycles. This stopping condition is adequate, as the $\overline{\mathcal{I}_\mathcal{G}}$ converges in all experiments. During the course of the experiments, the total estimated distance traveled by the robots was 23 km.

Table 5 summarizes the first set of experiments using one to three robots. It can be seen that the $\overline{\mathcal{I}_\mathcal{G}}$ values, as well as the total mission time τ, decreases with teamsize, as expected. In all cases the median is fairly close to the average value, meaning that most data is divided around the mean.

A particularly interesting result is the maximum idleness, $\max(\overline{\mathcal{I}_\mathcal{V}})$, which is low for the case of 1 robot. This happens because of the existence of a main loop in the environment, which results in fairly uniform visits to all vertices of the graph, while in the cases of 2 and 3 robots, the distance to the average value increases due to robots occasionally meeting in the environment and coordinating by changing to their heading direction. Consequently, no cycles are followed in the environment and the frequency of visits becomes less balanced. This can be confirmed by the standard deviation, which is around 23% using 1 robot and 35% and 37% for a teamsize of 2 and 3 robots respectively.[6]

Fig. 10 shows the evolution of the idleness in three different experiments with 1–3 robots. It can be seen that after 4 patrolling cycles, $\overline{\mathcal{I}_\mathcal{G}}$ converges in all cases, meaning that it is no longer affected by the initial conditions, seeing as all vertices start with a null value of idleness.

6.1. Scalability

In the previous section, the number of robots R is limited to the physical robots available. However, the distributed patrolling method used supports an arbitrary high teamsize. Note however that, when $R \geq |\mathcal{V}|$, an unusual and somehow unrealistic situation occurs, where the number of robots becomes higher than the points in the environment required to be visited.

[6] A video demonstrating an experiment with 3 robots is available at: http://isr.uc.pt/~davidbsportugal/videos/RAS.

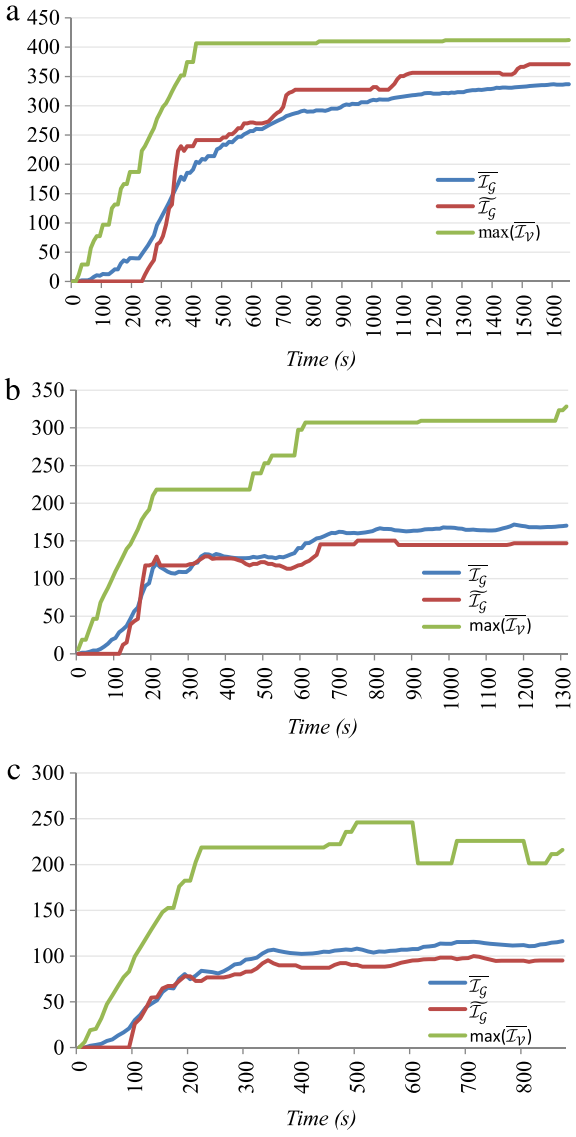


Fig. 10. Evolution of the idleness along time (a) with 1 robot, (b) with 2 robots and (c) with 3 robots.

Table 6
Experiments with 6 and 9 robots (all values in seconds).

Teamsize	$\overline{\mathcal{I}_\mathcal{G}}$	max($\overline{\mathcal{I}_\mathcal{V}}$)	$\widetilde{\mathcal{I}_\mathcal{G}}$	σ	τ
6(3 + 3)	71.097	152.625	65.483	27.130	610.500
	72.165	140.725	67.043	24.418	562.900
	77.332	150.145	72.938	27.350	600.580
9(3 + 6)	48.623	102.305	47.395	16.499	409.220
	50.239	90.580	54.157	16.083	362.320
	51.687	105.12	52.271	19.622	420.480

In order to test the approach with greater teamsize and evaluate its scalability, 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.

Three trials were conducted with a total of 6 agents composed by 3 physical robots and 3 simulated ones; and three more trials were performed with a teamsize of 9, composed by 3 physical robots and 6 simulated ones. Similarly to [40], the software layer is used unchanged both on real robots and in simulation.

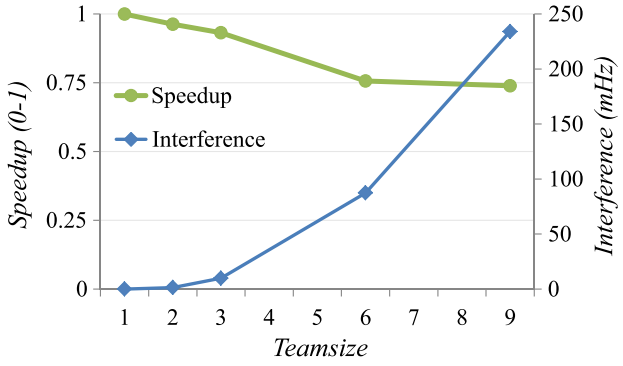


Fig. 11. Interference and speedup against teamsize.

Results in Table 6 show that the overall values of $\overline{\mathcal{I}_\mathcal{G}}$, max($\overline{\mathcal{I}_\mathcal{V}}$), $\widetilde{\mathcal{I}_\mathcal{G}}$, σ and τ are within the expected, following the trend shown in the cases of two and three robots.

Fig. 11 presents the speedup chart using different teamsizes. It can be seen that speedup and interference are negatively correlated, since the system enters progressively in sublinear performance with teamsize, due to the more frequent existence of spatial limitations, which in turn, increases the interference between robots causing the performance to decrease. These results confirm those obtained previously through simulations, proving that the SEBS technique is able to scale to high number of robots, working independently of the teamsize. In addition, it is also illustrated that the individual contribution of each robot, as teamsize grows, decreases progressively. This is, in fact, common to all MRPP approaches tested so far, however SEBS presents a smoother slope when compared to other approaches, as seen in Section 5.1.

6.2. 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 available were planned. In these experiments a robot is shutdown at different instants of time, aiming at studying the effect of the faults in the overall performance, as well as how the system evolves.

In the first experiment a robot is shutdown after 200 s from the beginning of the experiment. Similarly, in the second and third experiment, a robot is shutdown after 400 and 600 s respectively. The other robots assume that a teammate has failed when no message has been received from it in a period of 2 min.

Generally, it can be seen in Table 7 that the results obtained in the first experiment resemble those obtained with two robots, as most of the experiment is spent with only two agents, due to the failure occurring near the beginning. On the other side, the results shown in the second and third experiment are closer to those obtained using three robots, even though the performance is slightly inferior, as expected.

Analyzing now the influence of the failures in the evolution of the results, one can verify that in all three cases, when the failure occurs, the values of $\overline{\mathcal{I}_\mathcal{G}}$ and $\widetilde{\mathcal{I}_\mathcal{G}}$ increase after a while, which is particularly visible in Fig. 12(a) and (b). These results prove the robustness of the system, enabling graceful degradation, as long as one robot remains operational.

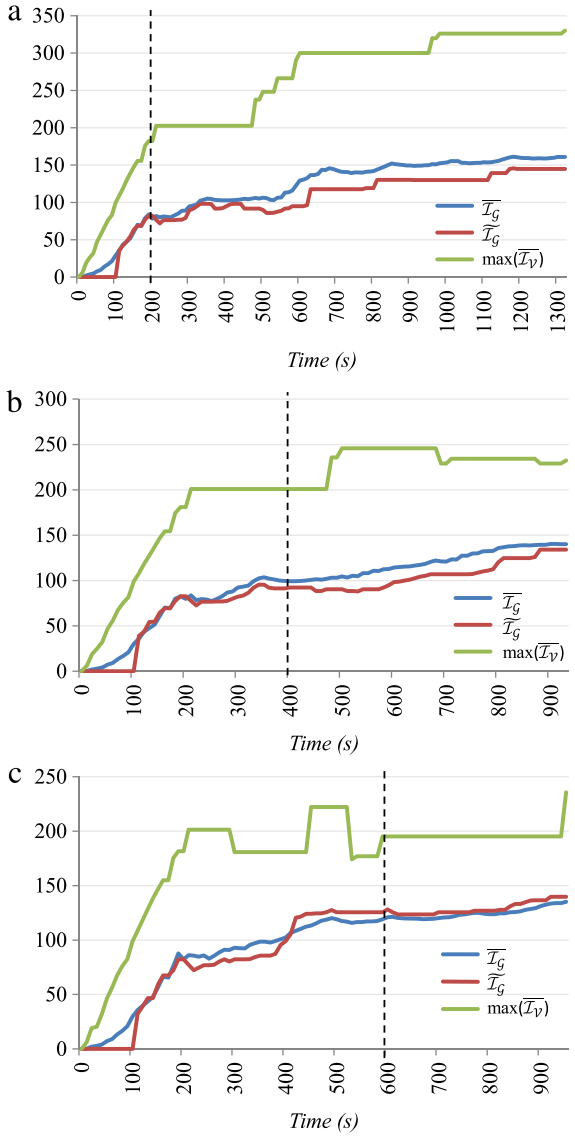


Fig. 12. Evolution of the idleness along time in experiments with robot failures. (a) Failure at 200 s. (b) Failure at 400 s. (c) Failure at 600 s.

Table 7
Experiments with 3 robots with failure of a robot in different instants of time (all values in seconds).

Failure time (s)	$\overline{\mathcal{I}_\mathcal{G}}$	max($\overline{\mathcal{I}_\mathcal{V}}$)	$\widetilde{\mathcal{I}_\mathcal{G}}$	σ	τ
200	160.975	330.225	144.846	62.825	1320.901
400	140.128	232.290	134.177	45.934	929.161
600	135.209	235.700	139.797	41.262	942.801

7. Simulation tests and evaluation of the impact of communication failures

In this section, two important aspects of this work are studied: computer simulation realism and robustness to communication failures. Having conducted experimental tests in a real world facility, it is now possible to compare the results obtained previously to simulations on the same environment. This is done in 7.1. Additionally, we seize these simulations by introducing different error rates in multi-robot communication in order to comprehend how team performance is affected. This is discussed in 7.2.

Table 8
Simulation experiments in the "ISR-Floor0" environment (all values in seconds).

Teamsize	$\overline{\mathcal{I}_\mathcal{G}}$	max($\overline{\mathcal{I}_\mathcal{V}}$)	$\widetilde{\mathcal{I}_\mathcal{G}}$	σ	τ
1	329.254	404.575	363.225	78.132	1618.3
	333.740	410.900	367.700	79.255	1643.6
	327.948	403.825	361.275	77.874	1615.3
2	160.875	326.500	147.675	59.153	1306.0
	167.190	291.000	150.457	61.256	1164.0
	170.176	312.800	149.500	61.921	1251.2
3	119.123	189.250	113.680	35.654	757.0
	113.063	201.275	107.529	34.533	805.1
	117.138	216.125	106.575	41.855	864.5
6	73.603	137.250	67.828	25.032	549.0
	71.241	130.425	67.486	24.043	521.7
	70.137	132.025	68.643	23.549	528.1
9	47.434	85.400	43.514	16.612	341.6
	46.036	80.000	44.043	14.753	320.0
	45.183	74.725	42.228	15.262	298.9

7.1. Simulation realism

In the experimental validation of the techniques (cf. Section 5), it was shown that both strategies presented in this work perform well independently of the environment topology and are able to scale to large teams. Further experiments in a lab scenario and then in a large indoor facility were made, illustrating the potential of employing these systems in the real world. In this section, the tests conducted in Section 6 are mimicked; however, simulated robots in Stage/ROS are used instead of a team of Pioneer-3DX robots. The objective is to understand how close the results drawn from simulation tests are from those obtained with the real robots, thus demonstrating how realistic the simulations are.

The performance of the physical robots can be directly compared with that obtained with simulated ones. To this end, three simulation trials with 1, 2, 3, 6 and 9 robots using SEBS were run in the "ISR-Floor0" map. The software layer remained unchanged, guaranteeing that conditions were identical in both sets of experiments. Fig. 13 illustrates a snapshot of a simulation with 9 virtual robots in the environment and in Table 8 the new simulation results are presented.

Generally, the results in Table 8 show close resemblances to those in Tables 5 and 6. In fact, the difference in performance (in terms of $\overline{\mathcal{I}_\mathcal{G}}$) between the simulated and real results is $\simeq$ 3.6%, which is remarkably low. Nevertheless, the difference in performance is more noticeable with larger teamsizes, especially with 9 robots, which suggests that agents in simulations are less affected by multi-robot interference.

Additionally, the values of $\overline{\mathcal{I}_\mathcal{G}}$, max($\overline{\mathcal{I}_\mathcal{V}}$), $\widetilde{\mathcal{I}_\mathcal{G}}$, σ and τ tend to be marginally lower in simulations and the variance between each different trial with the same configuration is inferior. Therefore, we believe that simulations can give an accurate yet slightly optimistic approximation of real-world results, and the lower variance can be associated with 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.

These results demonstrate that the simulation software considered is fairly realistic for multi-robot applications such as patrolling. Note that Stage runs at 10 Hz, which explains why the τ values only have one decimal place.

7.2. Influence of communication errors

The models proposed to solve the MRPP in Section 4 assume that agents are able to communicate seamlessly with other teammates during the course of the mission. However, this is not always the case, especially if a MANET should be maintained and robots are

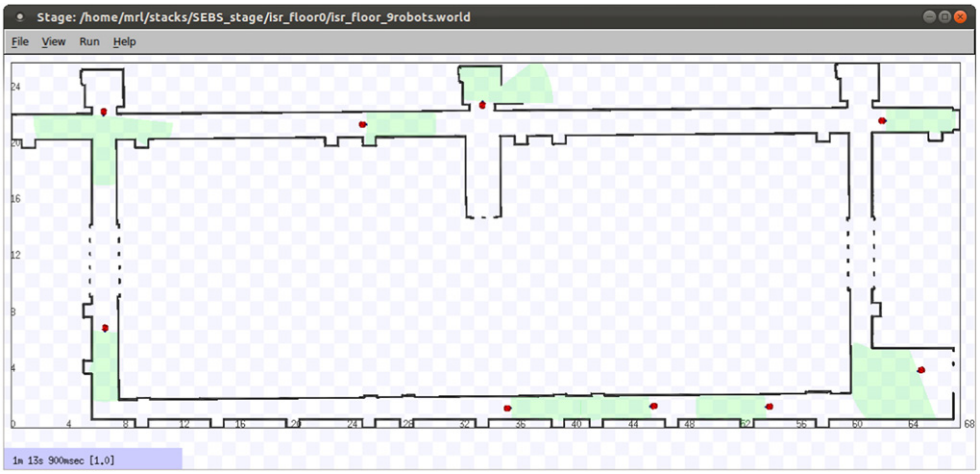


Fig. 13. Snapshot of a simulation in the "ISR-Floor0" environment with a team of 9 robots.

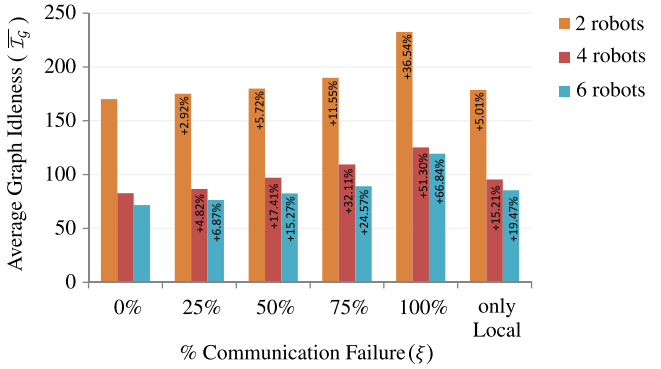


Fig. 14. Influence of communication failures in team performance.

occasionally far apart. In this section, further simulations were run in the "ISR-Floor0" map to test the robustness of the SEBS approach with different rates 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 move to the same places 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 these simulations, this is taken care of by the ROS navigation stack.

In order to simulate different rates ξ of communication failures, the robot will ignore messages with a probability equivalent to ξ. In the reported experiments, the rates considered were: $\xi = \{0\%, 25\%, 50\%, 75\%, 100\%\}$. Furthermore, the system has also been tested allowing only local communication, restricted to robots within two edges of distance in the graph $\mathcal{G}$. This is a particular situation where it is ensured that robots are able to receive all the other nearby robots' intentions and are thus able to coordinate themselves. Nevertheless, they are expected to make poor decisions as they are maintaining an incomplete information about the system.

The chart in Fig. 14 presents an overview of the simulation results with communication failures, using teamsizes of 2, 4 and 6 robots. Team performance is once again measured in terms of $\overline{\mathcal{I}_\mathcal{G}}$. The graph shows that performance gracefully degrades as ξ increases. The decrease of performance is approximately constant for the 25%, 50% and 75% cases. However, when no communication is allowed, i.e., $\xi = 100\%$, the performance of the algorithm drops strongly, especially for larger teams, which are much more influenced by the lack of coordination in the multi-robot system, as robots constantly interfere with one another. This reduction of performance, especially for greater teamsizes, is evident in the bars for $\xi = 100\%$: 36.54% for 2 robots, 51.30% for 4 robots and 66.84% for 6 robots.

Also illustrated in the rightmost side of the same figure is how performance is affected when communication is restricted to local interactions within 2 hops in $\mathcal{G}$. In this situation, robots are able to coordinate themselves by not competing to the same goals and not interfering with teammates. Despite that, they do not have contact with agents that are further away and, as a consequence, they will make uninformed decisions quite often. It can be seen that the system is able to perform well assuming such restrictions, especially for smaller teamsizes. The performance obtained using only local communication closely resembles to that obtained when dropping 50% of the messages for all teamsizes.

In short, these results show that the approach is robust to communication failures and only slightly degrades its performance when communication errors rate is moderate (e.g., 25%). Evidently, the higher the rate of failures, the more affected performance is. Additionally, communication failures have more impact in the performance of systems with a larger number of robots.

8. Conclusion

In this work, two methods based on Bayesian interpretation, inspired on conditional probability distributions, were proposed to solve the MRPP. It was shown that both are able to tackle the problem, resulting in adaptive, effective and distributed cooperative patrolling. Breaking away from conventional techniques, this work goes beyond classical centralized approaches that rely on precomputed cyclic routes or partition schemes for multi-robot patrolling, giving the robots the autonomy to deal with uncertainty and select actions according to the state of the system at the time.

The State Exchange Bayesian Strategy (SEBS) is an extension of the Greedy Bayesian Strategy (GBS), which attests the flexibility of employing Bayesian-based formalism to solve this problem. Also, as expected, SEBS generally performs better due to accounting for the future immediate state of the system, preventing robots from competing to reach the same goals, consequently reducing interference and enhancing scalability, as verified by simulations and experiments with multi-robot systems. Additionally, when placed

in comparison with other distributed state-of-the-art approaches, SEBS outperforms them, not only in terms of performance, but also in terms of scalability.

It is the authors' belief that research in this field should be more oriented towards effective solutions with applicability in the real world. In this work, the results obtained have demonstrated that the approach is able to scale to a high number of robots, being robust to robot failures and communication failures, and having the ability to adapt to constraints, e.g., different agent velocities, since the decision-making is done online with the information that each agent has collected about the system. Experiments were conducted using real robots and mixed teams of both virtual and real agents, in a large indoor infra-structure, proving the effectiveness of the approach and the potential to use it in the real world.

In the future, due to its flexibility and simplicity, the model can be extended with more variables in order to employ it in different applications and/or use others sensors in the robots; e.g., readings from a temperature sensor may be included in the model, guiding robots towards heat sources in the environment. In addition, we are currently extending our model formulation into a generalizable framework with the capability to make autonomous decisions based on robot's collective experience, i.e., past decisions will increment the previous knowledge database and will influence future decisions. Moreover, the system will have memory, which means that at each vertex, decisions made previously by agents will be taken into consideration. Finally, we intend to devise an analytical method to compute the most adequate teamsize for a patrolling mission according to the environment topology and temporal constraints.

Acknowledgments

This work was financially supported by a Ph.D. grant (SFRH/BD/64426/2009) from the Portuguese Foundation for Science and Technology (FCT), the Institute of Systems and Robotics (ISR-Coimbra) and by the research project CHOPIN (PTDC/EEA-CRO/119000/2010) also under regular funding by FCT.

The authors gratefully acknowledge João B. Campos, Luís Santos, Amadeu Fernandes, André Araújo, Micael Couceiro and Gonçalo Cabrita (ISR, University of Coimbra) for their contribution and feedback.

Appendix. Terminology

See Tables A.1 and A.2.

Table A.1
Notation table.

Notation	Description
$\mathcal{G}$	Undirected and connected navigation graph.
$\mathcal{V}$	Set of vertices of $\mathcal{G}$.
$\mathcal{E}$	Set of edges of $\mathcal{G}$.
v_i	Vertex i.
e_{ij}	Edge that connects v_i to v_j.
t	Current time step.
t_l	Time step when a vertex was last visited.
C_i	Number of visits to v_i.
τ	Stopping time of the patrol mission.
$\mathcal{I}_{v_i}(t)$	Instantaneous idleness of vertex v_i at time t.
$\overline{\mathcal{I}_{v_i}}$	Average idleness of a vertex v_i over time.
$\max(\overline{\mathcal{I}_{\mathcal{V}}})$	Maximum average idleness of all vertices.
$\overline{\mathcal{I}_{\mathcal{G}}}$	Average idleness of the graph $\mathcal{G}$ over time.
$\widetilde{\mathcal{I}_{\mathcal{G}}}$	Median idleness of the graph $\mathcal{G}$ over time.
σ	Standard deviation.
R	Number of robots.
x_r	Patrolling route of robot r.
N	Number of vertices in x_r.
$\mathbf{x}$	Set of all R patrolling routes.
G_i	Gain of moving to v_i, a continuous random variable.
c	Constant robot speed.
$\lvert e_{\min}\rvert$	Edge weight threshold.
$N_{\mathcal{G}}(v_i)$	Neighborhood of v_i: set of adjacent vertices of v_i.
$deg(v_i)$	Degree of v_i: number of adjacent vertices of v_i.
$P(\cdot)$	Probability.
$P(\cdot\lvert\cdot)$	Conditional probability.
$f(g)$	Probability density function of gain.
$F(g)$	Cumulative distribution function of gain.
L	Minimum probability value of $F(g)$.
M	Gain saturation: maximum value of G_i.
S_i	State of vertex v_i, a discrete random variable.
$f(s)$	Probability mass function of S_i.
$\lambda_{\mathcal{G}}$	Fiedler value or algebraic connectivity of $\mathcal{G}$.
$\upsilon(R)$	Speedup of R robots.
$\Psi(R)$	Performance of R robots.
ξ	Rate of communication failures.

Table A.2
Table of acronyms.

Acronym	Description
AMCL	Adaptive Monte Carlo Localization
CGG	Cyclic Algorithm for Generic Graphs
CR	Conscientious Reactive Algorithm
GBS	Greedy Bayesian Strategy
HCR	Heuristic Conscientious Reactive Algorithm
HPCC	Heuristic Pathfinder Conscientious Cognitive
ISR	Institute of Systems and Robotics
MANET	Mobile *Ad Hoc* Network
MRPP	Multi-Robot Patrolling Problem
MRS	Multi-Robot Systems
MSP	Multilevel Subgraph Patrolling Algorithm
TSP	Traveling Salesman Problem
SEBS	State Exchange Bayesian Strategy
ROS	Robot Operating System

References

[1] L.E. Parker, Distributed intelligence: overview of the field and its application in multi-robot systems, Journal of Physical Agents 2 (2) (2008) 5–14. Special issue on Multi-robot systems.
[2] Webster's Online dictionary, December, 2012. Available at: http://www.websters-onlinedictionary.org.
[3] R. Murphy, J. Kravitz, S. Stover, R. Shoureshi, Mobile robots in mine rescue and recovery, IEEE Robotics & Automation Magazine 16 (2) (2009) 91–103.
[4] A. Kleiner, J. Prediger, B. Nebel, RFID technology-based exploration and SLAM for search and rescue, in: Proceedings of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'2006, Beijing, China, October 9–15, 2006, pp. 4054–4059.
[5] A. Renzaglia, L. Doitsidis, A. Martinelli, E. Kosmatopoulos, Multi-Robot three-dimensional coverage of unknown areas, International Journal of Robotics Research (IJRR) 31 (6) (2012) 738–752.
[6] R. Murphy, Human–robot interaction in rescue robotics, IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 34 (2) (2004) 138–153.
[7] N. Agmon, G. Kaminka, S. Kraus, Multi-robot adversarial patrolling: facing a full-knowledge opponent, Journal of Artificial Intelligence Research (JAIR) 42 (2011) 887–916.
[8] N. Basilico, N. Gatti, T. Rossi, S. Ceppi, F. Amigoni, 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, Milan, Italy, 2009, pp. 557–564.
[9] A. Marino, L.E. Parker, G. Antonelli, F. Caccavale, Behavioral control for multi-robot perimeter patrol: a finite state automata approach, in: Proceedings of the 2009 IEEE international conference on Robotics and Automation, ICRA'2009, Kobe, Japan, May, 2009, pp. 831–836.
[10] E. Jensen, M. Franklin, S. Lahr, M. Gini, Sustainable multi-robot patrol of an open polyline, in: Proceedings of the 2011 IEEE International Conference on Robotics and Automation, ICRA'2011, Shanghai, China, May 9–13, 2011.
[11] Y. Chevaleyre, Theoretical analysis of the multi-agent patrolling problem, in: Proceedings of the 2004 International Conference on Agent Intelligent Technologies, IAT'04, Beijing, China, September 20–24, 2004, pp. 30–308.
[12] D. Portugal, R.P. Rocha, Multi-robot patrolling algorithms: examining performance and scalability, Advanced Robotics Journal 27 (5) (2013) 325–336. Taylor & Francis.

[13] Y. Elmaliach, N. Agmon, G. Kaminka, Multi-robot area patrol under frequency constraints, in: Proceedings of the 2007 IEEE International Conference on Robotics and Automation, ICRA'2007, Rome, Italy, April 10–14, 2007, pp. 385–390.
[14] S. Ruan, C. Meirina, F. Yu, K.R. Pattipati, R.L. Popp, Patrolling in a stochastic environment, in: Proceedings of the 10th International Command and Control Research and Technology Symposium, 10th ICCRTS, McLean, Virginia, USA, June 13–16, 2005.
[15] S. Smith, D. Rus, Multi-robot monitoring in dynamic environments with guaranteed currency of observations, in: Proceedings of the IEEE Conference on Decision and Control, CDC'2010, Atlanta, GA, December, 2010, pp. 514–521.
[16] F. Pasqualetti, A. Franchi, F. Bullo, On cooperative patrolling: optimal trajectories, complexity analysis, and approximation algorithms, IEEE Transactions on Robotics 28 (3) (2012) 592–606.
[17] F. Pasqualetti, J. Durham, F. Bullo, Cooperative patrolling via weighted tours: performance analysis and distributed algorithms, IEEE Transactions on Robotics 28 (5) (2012) 1181–1188.
[18] A. Machado, G. Ramalho, J. Zucker, A. Drogoul, Multi-agent patrolling: an empirical analysis of alternative architectures, in: Multi-Agent-Based Simulation II, in: Lecture Notes in Computer Science, vol. 2581, Springer-Verlag, 2003, pp. 155–170.
[19] A. Almeida, G. Ramalho, H. Santana, P. Tedesco, T. Menezes, V. Corruble, Y. Chaveleyre, Recent advances on multi-agent patrolling, in: Advances in Artificial Intelligence—SBIA 2004, in: Lecture Notes in Computer Science, vol. 3171, Springer-Verlag, 2004, pp. 474–483.
[20] P. Fazli, A. Davoodi, P. Pasquier, A.K. Mackworth, Complete and robust cooperative robot area coverage with limited range, in: Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS'2010, Taipei, Taiwan, October 18–22, 2010, pp. 5577–5582.
[21] Y. Gabriely, E. Rimon, Spanning-tree based coverage of continuous areas by a mobile robot, in: Annals of Mathematics and Artificial Intelligence, Vol. 31, Kluwer Academic Publishers, 2001, pp. 77–98.
[22] T. Sak, J. Wainer, S. Goldenstein, Probabilistic multiagent patrolling, in: Advances in Artificial Intelligence—SBIA 2008, in: Lecture Notes in Computer Science, vol. 5249, Springer, 2008, pp. 124–133.
[23] R. Stranders, E.M. de Coteb, A. Rogers, N.R. Jennings, Near-optimal continuous patrolling with teams of mobile information gathering agents, in: Artificial Intelligence, Elsevier, 2012.
[24] F. Sempé, A. Drogoul, Adaptive patrol for a group of robots, in: Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS'2003, Las Vegas, USA, 2003.
[25] R. Zlot, A. Stentz, Complex task allocation for multiple robots, in: Proceedings of the IEEE International Conference on Robotics and Automation, ICRA'2005, Barcelona, Spain, 18–22 April, 2005, pp. 1515–1522.
[26] K. Hwang, J. Lin, H. Huang, Cooperative patrol planning of multi-robot systems by a competitive auction system, in: Proceedings of the ICROS-SICE International Joint Conference, Japan, August, 2009.
[27] C. Poulet, V. Corruble, A. Seghrouchini, Working as a team: using social criteria in the timed patrolling problem, in: Proceedings of the 24th IEEE International Conference on Tools with Artificial Intelligence, ICTAI'2012, Athens, Greece, November 7–9, 2012.
[28] H. Santana, G. Ramalho, V. Corruble, B. Ratitch, Multi-agent patrolling with reinforcement learning, in: Proceedings of the International Conference on Autonomous Agents and Multiagent Systems, Vol. 3, AAMAS'2004, New York, 2004, pp. 1122–1129.
[29] Y. Ishiwaka, T. Sato, Y. Kakazu, An approach to the pursuit problem on a heterogeneous multiagent system using reinforcement learning, in: Robotics and Autonomous Systems (RAS), Vol. 43, Elsevier, 2003, pp. 245–256. (4).
[30] J. Pita, M. Tambe, C. Kiekintveld, S. Cullen, E. Steigerwald, GUARDS—innovative application of game theory for national airport security, in: Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Vol. 3, IJCAI'11, Barcelona, Spain, July 16–22, 2011, pp. 2710–2715.
[31] J. Marier, C. Besse, B. Chaib-draa, Solving the continuous time multiagent patrol problem, in: Proceedings of the 2010 IEEE International Conference on Robotics and Automation, ICRA'2010, Anchorage, Alaska, USA, May, 2010.
[32] X. Chen, T.S. Yum, Patrol districting and routing with security level functions, in: Proceedings of the 2010 IEEE International Conference on Systems, Man and Cybernetics, SMC'2010, Istanbul, Turkey, October 10–13, 2010, pp. 3555–3562.
[33] V. Yanovski, I.A. Wagner, A.M. Bruckstein, A distributed ant algorithm for efficiently patrolling a network, Algorithmica 37 (2003) 165–186.
[34] M. Baglietto, G. Cannata, F. Capezio, A. Sgorbissa, Multi-robot uniform frequency coverage of significant locations in the environment, in: Distributed Autonomous Robotic Systems 8, Springer-Verlag, Berlin, Heidelberg, 2009, pp. 3–14.
[35] M. Jansen, N. Sturtevant, Direction maps for cooperative pathfinding, in: Proceedings of the 4th Artificial Intelligence and Interactive Digital Entertainment Conference, AAAIDE'08, Stanford, California, October 22–24, 2008.
[36] P. Sampaio, G. Ramalho, P. Tedesco, The gravitational strategy for the timed patrolling, in: Proceedings of the 22nd IEEE International Conference on Tools with Artificial Intelligence, ICTAI'10, Arras, France, October 27–29, 2010, pp. 113–120.
[37] D. Portugal, R.P. Rocha, A survey on multi-robot patrolling algorithms, in: Luis M. Camarinha-Matos (Ed.), Technological Innovation for Sustainability, Vol. 349, Springer, Berlin, Heidelberg, 2011, pp. 139–146.
[38] G. Cabrita, P. Sousa, L. Marques, A. de Almeida, Infrastructure monitoring with multi-robot teams, in: Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS'2010, Workshop on Robotics for Environmental Monitoring, Taipei, Taiwan, October 18–22, 2010.
[39] D. Portugal, R.P. Rocha, MSP algorithm: multi-robot patrolling based on territory allocation using balanced graph partitionin, in: Proceedings of the 25th ACM Symposium on Applied Computing, SAC 2010, Special Track on Intelligent Robotic Systems, Sierre, Switzerland, March 22–26, 2010, pp. 1271–1276.
[40] L. Iocchi, L. Marchetti, D. Nardi, Multi-robot patrolling with coordinated behaviours in realistic environments, in: Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS'2011, San Francisco, CA, USA, September 25–30, 2011, pp. 2796–2801.
[41] D. Portugal, R.P. Rocha, Decision methods for distributed multi-robot patrol, in: Proceedings of the 10th IEEE International Symposium on Safety, Security and Rescue Robotics, SSRR'2012, College Station, Texas, USA, November 8–11, 2012.
[42] T. Furukawa, F. Bourgault, B. Lavis, H.F. Durrant-Whyte, Recursive Bayesian search-and-tracking using coordinated UAVs for lost targets, in: Proceedings of the IEEE International Conference on Robotics and Automation, ICRA'2006, Orlando, Florida, May 2006.
[43] B. Julian, M. Angermann, M. Schwager, D. Rus, Distributed robotic sensor networks: an information theoretic approach, International Journal of Robotics Research (IJRR) (2012).
[44] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, A. Ng, 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, Japan, May 12–17, 2009.
[45] D. Portugal, R.P. Rocha, On the performance and scalability of multi-robot patrolling algorithms, in: Proceedings of the IEEE International Symposium on Safety, Security, and Rescue Robotics, SSRR'2011, Kyoto, Japan, 50–55, November 1–5, 2011.
[46] D. Portugal, R.P. Rocha, Extracting topological information from grid maps for robot navigation, in: Proceedings of the 4th International Conference on Agents and Artificial Intelligence, ICAART'2012, Vilamoura, Algarve, Portugal, February 6–8, 2012, pp. 137–143.
[47] F. Jansen, T. Nielsen, Bayesian Networks and Decision Graphs, in: Information Science and Statistics Series, vol. XVI, second ed., Springer Verlag, 2007.
[48] M.R. Garey, D.S. Johnson, Computers and Intractability, Springer, 1979.
[49] B. Korte, J. Vygen, Combinatorial Optimization: Theory and Algorithms, fourth ed., in: Series Algorithmics and Combinatorics, vol. 21, Springer Verlag, 2007.
[50] N. Christofides, J.E. Beasley, The period routing problem, Networks 14 (2) (1984) 237–256.
[51] D. Applegate, W. Cook, A. Rohe, Chained Lin–Kernighan for large traveling salesman problems, INFORMS Journal on Computing 15 (2003) 82–92.
[52] R. Vaughan, B. Gerkey, Reusable robot code and the player/stage project, in: D. Brugali (Ed.), Software Engineering for Experimental Robotics, in: Springer Tracts on Advanced Robotics, Springer, 2007, pp. 267–289.
[53] S. Thrun, D. Fox, W. Burgard, F. Dellaert, Robust Monte Carlo localization for mobile robots, Artificial Intelligence (AI) 128 (1–2) (2001) 99–141.
[54] M. Fiedler, Algebraic connectivity of graphs, Czechoslovak Mathematical Journal 23 (98) (1973).
[55] T. Balch, R. Arkin, Communication in reactive multiagent robotic systems, Autonomous Robots 1 (1) (1994) 27–52.
[56] A. Araújo, D. Portugal, M. Couceiro, C. Figueiredo, R.P. Rocha, Traxbot: assembling and programming of a mobile robotic platform, in: Proceedings of the 4th International Conference on Agents and Artificial Intelligence, ICAART'2012, Vilamoura, Algarve, Portugal, February 6–8, 2012, pp. 301–304.
[57] M. Couceiro, C. Figueiredo, D. Portugal, R.P. Rocha, N.M. Ferreira, Initial deployment of a robotic team—a hierarchical approach under communication constraints verified on low-cost platforms, in: Proceedings of the 2012 IEEE International Conference on Intelligent Robots and Systems, IROS'2012, Vilamoura, Algarve, October 7–12, 2012.
[58] M. Banzi, Getting Started with Arduino, second ed., O'Reilly Media Inc., 2011.
[59] ActivMedia Robotics Pioneer 3 Operations Manual, Version 3, Mobile Robots Inc., January 2006.

David Portugal is a Doctoral Candidate on Electrical and Computer Engineering at the Faculty of Sciences and Technology of University of Coimbra and a researcher at the Institute of Systems and Robotics (ISR), University of Coimbra, Portugal. He is sponsored by a Ph.D. scholarship from the Portuguese Foundation for Technology and Sciences. He holds an M.Sc. degree in Electrical Engineering and Computers obtained in September, 2009. Currently, David Portugal is also collaborating as a researcher in the nationally funded Project CHOPIN, which deals with cooperation between human and robotics teams in catastrophic incidents. His research interests include cooperative robotics, multi-agent systems and meta-heuristics and optimization.

Rui P. Rocha completed his Ph.D. degree on May 2006, from the Faculty of Engineering of the University of Porto. Between February 2000 and May 2006, he has been 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 on 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.