Proceedings of the 2011 IEEE International Symposium on
Safety,Security and Rescue Robotics
Kyoto,Japan,November 1-5 2011

On the Performance and Scalability of Multi-Robot Patrolling Algorithms

David Portugal
Institute of Systems and Robotics
University of Coimbra
3030-290 Coimbra, Portugal
davidbsp@isr.uc.pt

Rui P. Rocha
Institute of Systems and Robotics
University of Coimbra
3030-290 Coimbra, Portugal
rprocha@isr.uc.pt

Abstract — Several distinct multi-robot patrolling strategies have been presented for the last decade in the context of security applications. However, there is a deficit of studies comparing these strategies, namely in terms of their performance and the scalability in the number of robots. For that reason, in this paper, an evaluation of five representative patrolling approaches is presented. This analysis is based on realistic simulation results using ROS and a performance metric represented by the average idleness of the topological environment (i.e., graph) that represents the area to patrol.

The results presented help to identify which strategies enable enhanced team scalability and which are the most suitable approaches given any environment.

Keywords: *Multi-Robot Systems, Patrolling, Security, Scalability and Performance.*

I. Introduction

This work focus on surveillance tasks using multiple mobile robots, which involve frequent visits to every point of the environment. Therefore, the words "Patrol" and "Patrolling" are implicitly used in this sense.

The major motivation for studying this issue relates to its spectrum of applicability in the context of security systems and the potential to replace or assist human operators in dangerous real-life scenarios, like mine clearing, rescue operations or surveillance, easing arduous and time-consuming tasks and offering the possibility to relieve human beings, enabling them to be occupied in nobler tasks like monitoring the system from a safe location.

Cooperation among robots is one of the most decisive issues in this context; since robots must efficiently work together in order to improve the performance of the system as a whole.

In addition, multi-robot patrol is a challenging problem, because agents must navigate autonomously, coordinate their actions, be distributed in space and must be independent of the number of robots and the environment's dimension.

This work presents a comparative study between five different state of the art patrolling strategies using distinct topological environments and different teamsizes, in order to analyze the performance and scalability of each approach. Conclusions drawn in this field of research may support the development of future approaches not only in this domain but also in other multi-robot applications.

II. Related Work

The existing algorithms in the literature for patrolling an environment with multiple mobile agents present many differences in terms of strategy, communication paradigm, cooperation scheme, performance evaluation and other features. They can be divided into Pioneer methods [1], [2]; Graph Theory methods, [3], [4], [5]; and Alternative Coordination methods [6], [7], [8], [9].

Pioneer strategies include simple architectures with agents with different capabilities that move in the environment mostly looking for locations that have not been visited for some time, aiming to maintain a high frequency of visits in every place of the area.

Graph Theory strategies look for solutions of classical problems like finding Hamilton cycles, Graph partitioning and others to assign efficient routes for the robot's patrolling missions. These strategies typically rely on a centralized coordinator to calculate those routes.

Recently, many alternative coordination methods have also been presented, aiming to solve the problem through the usage of approaches that have presented good results in multi-robot systems in general, like task allocation, reinforcement learning, negotiation mechanisms and swarm-based strategies.

A pioneer work was presented in [1], where several architectures for multi-agent patrol were proposed. These architectures have distinct agents' behavior, perception, communication paradigms and decision-making. Additionally, they have contributed with criteria to evaluate the performance of the approaches based on the average and maximum idleness of the vertices of the graph that represents the topological environment.

In [2], the architectures proposed by Machado were enhanced with advanced decision-making, based on both the instantaneous idleness of vertices and the distance to them, as well as with advanced path finding, which considers distance (or cost) of the edges and the idleness of vertices in the path towards a goal. Also, the tests were run on more and distinct environment topologies, which was a strong limitation of Machado's work. Nonetheless, Almeida's work contains several simplifications that are overcome in this article by using realistic simulations that consider both the dynamic of robots and the actual time to measure performance instead of using iterative simulation cycles.

Approaches based on graph theory and operational research (OR) commonly address the patrolling problem by computing minimal-cost cycles that visit all points in the target area. The agents are employed uniformly along the path and follow the same patrol route over and over again. For example, [3] presented an area patrol algorithm based on the computation of Hamilton cycles that guarantees that each point in the target area is covered at the same optimal frequency. On the other hand, [4] described a cyclic algorithm based on heuristics that approximates a Travelling Salesman Problem (TSP) cycle on top of the topological representation of the environment.

These strategies are robust, being independent of the number of robots and are recognized for their results in terms of visit frequency. However, they have a deterministic nature, which means that an intelligent intruder that apprehends the patrolling scheme may take advantage of the idle time between passages of robots in some points of the area.

In [5], the Multilevel Subgraph Patrolling (MSP) Algorithm is described. In this approach, balanced graph partitioning is projected in order to assign different patrolling regions (subgraphs) of the environment for each mobile agent. The algorithm subsequently computes effective paths for every robot using classical graph theory approaches. Results confirmed the flexible and high performance nature of the approach, which benefits from being non-redundant and not needing inter-agent communication.

Some alternative methods have been presented throughout the years. In [6], patrolling is addressed in a task allocation perspective, where each robot is assigned a different region to visit. Robots send their current state to a centralized system running on a remote computer, through a wireless communication network, to compute the task strength and drive the robot through propagated data.

Moreover, reinforcement learning was used in [7] to solve the patrolling problem by automatically adapting the agents' strategies to the topology of the environment. Additionally, approaches based on negotiation mechanisms have also been proposed in [8] as well as in [10]. In these works, agents exchange vertices of the environment graph to patrol, using an auction-based system. The agents will naturally aim to obtain a set of vertices in the same region of the graph. Despite the results obtained, reinforcement learning and negotiation mechanisms prove to be extremely complex when compared to pioneer strategies, with nearly no communication ability, which achieve similar results [11].

Recently, swarm intelligence has also been used to tackle the multi-robot patrolling problem [9], [12]. In such works, the environment is represented by a grid and agents only have local perception, deciding their next move according to the artificial state of grid cells, which depends on information previously dropped by agents on those cells (similar to pheromones in ant colonies).

For a more thorough survey of multi-robot patrolling architectures, one should refer to [13].

III. Problem Formulation

As mentioned before, it is common to represent the area to patrol by an undirected connected graph G = (V, E) with v_i $\in$ V vertices and $e_{i,j} \in$ E edges. Therefore, G corresponds to the topological map for the patrolling mission and is assumed to be known a priori. Also, in these maps, vertices correspond to important places or landmarks, connected by edges that represent the paths between them.

In order to address and compare the performance of different patrolling algorithms, it is important to establish an evaluation metric.

The instantaneous idleness (Idl_{t_k}) of a vertex $v_i \in$ V in time t_k, with t = {0, ..., t_k}, is defined as:

$$Idl_{t_k}(v_i) = t_k - t_{last\_visit}, \tag{1}$$

where t_{last_visit} corresponds to the last time instant when the vertex v_i was visited by any robot of the team. Consequently, the average idleness (Idl_m) of a vertex $v_i \in$ V in a total time T can be defined as:

$$Idl_m(v_i) = \frac{\sum_{k=0}^{T} Idl_{t_k}(v_i)}{T} \tag{2}$$

Finally, in order to obtain a generalized measure, the average idleness of the graph G (Idl_G) is defined as:

$$Idl_G = \frac{\sum_{i=0}^{|V|} Idl_m(v_i)}{|V|}, \tag{3}$$

where $|V|$ represents the cardinality of the set V.

Considering a patrolling path as an ordered array of vertices of G, the multi-robot patrolling problem can thus be described as the problem of finding a set of paths **x** which visit all vertices $v_i \in$ V of the graph G, using an arbitrary team of R robots, with the overall team goal of minimizing Idl_G:

$$f = \underset{\mathbf{x}}{\operatorname{argmin}}\, Idl_G \tag{4}$$

By finding:

$$\mathbf{x} = [x_1, ..., x_r, ..., x_R] \tag{5}$$

Such that:

$$x_r = \{v_{r,1}, v_{r,2}, ..., v_{r,N}\} \tag{6}$$

$$v_{r,n} \in V$$

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

$$1 \leq n \leq N, N \in \mathbb{N}$$

Subject to:

$$\forall v_i \in V, \exists x_r \in \mathbf{x} : v_i \in x_r \tag{7}$$

Note that x_r represents the patrolling path of robot *r* which can either be calculated *a priori*, which is typically done by centralized algorithms, or online to consider and incorporate the dynamics of the system in a given time step, which is the usual approach of distributed approaches.

TABLE I
COMPARATIVE TABLE OF THE ANALYZED ARCHITECTURES

Algorithm	Complexity	Agent Perception	Decision-Making	Planning
CR	*	Reactive	Local Idleness-based	Online
HCR	**	Reactive	Local Heuristic-based	Online
HPCC	***	Cognitive	Global Heuristic-based	Online
CGG	****	Cognitive	Cycle Computation (OR)	Offline
MSP	****	Cognitive	OR inside each region	Offline

IV. EVALUATED PATROLLING ALGORITHMS

Having analyzed the literature, five representative algorithms were implemented. These algorithms were chosen from among all previous research works based on the good performance results that they have obtained and the different properties assumed like agent perception, decision-making and planning, as it is shown in Table 1. In this section, those algorithms are examined and compared in detail. Due to space limitations, the pseudo-code of the approaches is not presented. No alternative approaches were implemented in this work mainly due to the fact that the complexity of their implementation does not lead to better results when compared to simpler approaches, as previously concluded by [11].

Besides the analysis of the performance of the diverse algorithms, this work also addresses the scalability of the studied approaches. In the context of multi-robot systems, scalability is related to how well a given strategy performs as the dimension of the team grows and how the individual productivity of each robot is influenced by the increase of several number of agents in the team. Having this in mind, the interference between robots is measured in every experiment as the number of times that different agents share nearby areas, having to avoid each other.

A. Conscientious Reactive (CR)

Ranked one of the top algorithms in the study of Machado *et al.* [1], Conscientious Reactive is a simple pioneer approach, in which agents decide locally which vertex they should move to in the next step, taking only into consideration the instantaneous idleness of the neighbors of the current vertex, where the robot is located at the moment.

B. Heuristic Conscientious Reactive (HCR)

Heuristic Conscientious Reactive is an algorithm presented by Almeida in [2]. It is similar to CR with an important modification on the decision-making process, where the authors calculate a decision value that considers not only the instantaneous idleness of the neighbors of the current vertex as well as the distance to them.

C. Heuristic Pathfinder Conscientious Cognitive (HPCC)

Unlike the two previous approaches, which use reactive agents that move only to close by vertices, Heuristic Pathfinder Conscientious Cognitive plans on the global graph to decide which vertex to move to subsequently. HPCC was also presented by Almeida [2] as a modified version of an approach called "Conscientious Cognitive" previously described in [1].

Agents use a similar decision-making process as in HCR. However, instead of only moving to vertices in their neighborhood, they can move to any vertex of the graph. In addition, the algorithm takes into account the vertices on the way from the current one to the calculated destination. The chosen path depends on the instantaneous idleness and the distance of the vertices along the way. This is possible by computing new edge costs and running a Dijkstra shortest path algorithm considering the new costs.

D. Cyclic Algorithm for Generic Graphs (CGG)

Inspired on the work of Elmaliach et al. [3], a Cyclic Algorithm was implemented and used in [5]. It is essentially an offline graph theory based method which looks for Hamilton cycles or paths in the graph in order to visit all vertices. When no such cyles or paths exist, the method looks for long paths and non hamiltonian cycles as an alternative and computes detours to unvisited vertices. In this work, each robot is endowed with the ability of computing the final cycle. Hence the algorithm is run in a totally distributed fashion like the three previous ones.

E. Generalized MSP Algorithm (MSP)

The MSP Algorithm [5] is an offline graph theory based method, which partitions the graph into regions, where agents perform the patrol task. In the first phase of the algorithm, the graph is partitioned by a centralized entity, which then assigns regions to different robots. In a second phase, robots patrol their independent areas in a cyclic way, using a similar approach to the CGG method in their own subgraphs. The word "Generalized" was added since the algorithm can partition the graph into arbitrarily high k regions (it could only be partitioned into a maximum of 8 regions in the original work) being limited up to the point that the graph can no longer be partitioned. The performance of the algorithm strongly depends on how balanced the partitioning of the graph is.

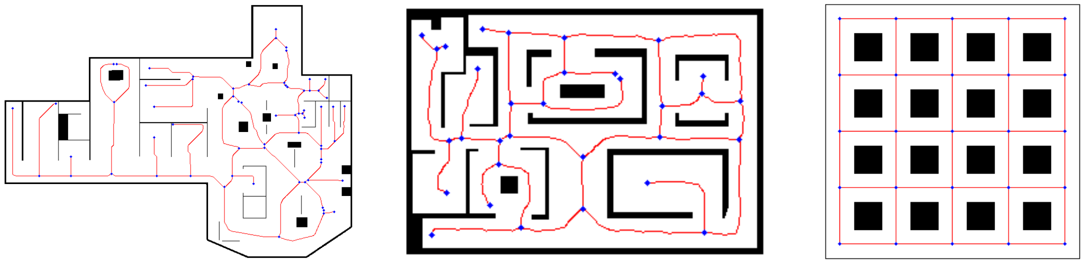
V. SETTING UP THE EXPERIMENTS

The performance and scalability of the five algorithms were compared using three topological maps chosen due to their different connectivity and complexity. To address the connectivity of the graph, a well-known metric of the graph was analyzed: the Fiedler value or algebraic connectivity [14]. In order to remove its dependency on the number of vertices in the spectrum of the Laplacian matrix, the Normalized Laplacian $\mathscr{L}$ [15] was adopted to obtain the Fiedler value of each graph.

All eigenvalues of $\mathscr{L}$ are non-negative and $\lambda_0 = 0$. For non-complete connected graphs (as is our case), the Fiedler Value λ_1 is the smallest non-zero eigenvalue of $\mathscr{L}$ and:

$$0 < \lambda_1 \leq 1 \tag{8}$$

Table II presents the connectivity properties of the graphs chosen for the experiments. All three graphs and the respective



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

Fig. 1. Environments used in the experiments with respective topological map.

TABLE II

CONNECTIVITY PROPERTIES OF THE GRAPHS USED IN THE EXPERIMENTS

Topological Map	Environment Area	$\|V\|$	Fiedler Value (λ_1)
A	1357.17 m^2	66	0.0080
B	1542.30 m^2	32	0.0317
C	665.64 m^2	25	0.1313

environments are presented in Figure 1. Since it is intended that the simulations are as realistic as possible, the Stage 2D multi-robot simulator was chosen together with ROS [16], a framework for robot software development. The graph information of a given environment is loaded by every robot in the beginning of each simulation, which then runs one of the five algorithms described. Robots 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 [17] and a probabilistic localization system, more specifically the adaptive Monte Carlo localization approach [18], which uses a particle filter to track the pose of a robot against a known map. In addition, robots are non-holonomic and have a maximum velocity of 0.2 m/s.

VI. RESULTS AND DISCUSSION

The simulation process involved running the five described patrolling strategies with six different teamsizes (1, 2, 4, 6, 8 and 12 robots) in all three environments. Robots had the same starting positions for all algorithms when using the same teamsize and environment. Every trial was repeated three times, in a total of 264 simulations[1], which lasted around 345 hours with a cluster of four processors that were used due to the powerful computation demands of simulations, mainly those with higher teamsizes. Simulations were stopped when the value of the average graph idleness (Idl_G) after each patrolling cycle, i.e. every vertex visited, converges with 2.5% of tolerance. This resulted into an average simulation time of 1h18m, which led to accurate and similar results between different trials; hence, there was no need to repeat the trials several times as testified by the overall average standard deviation of the results: $\overline{\sigma} = 4.42\%$.

[1]The ROS simulation code is available at: http://www.ros.org/wiki/isr-uc-ros-pkg#patrol

The chart in Figure 2 represents environment connectivity vs. teamsize and depicts some general insights about the most suited solutions in different regions of the design space, providing a graphical overview of the results obtained. It is possible to verify that offline planning strategies (MSP and CGG) perform better in weakly connected environments than in strongly connected ones. This occurs because one can take better advantage of offline planning in such environments, while there are more path alternatives in strongly connected environments, where online planning performs adequately.

Generally, MSP is the algorithm with the best Idl_G values for larger teams, up to the point where the algorithm can no longer partition the graph. The method is not able to partition the topologies B and C in the 12 regions case, which happens due to limitations of the partition stage of the algorithm. Nevertheless, these good results can be explained by low interference between agents when compared to other strategies, because each robot operates in a specific section of the environment. For smaller teams, the approach is not usually worth to employ, because it is more complex than simple reactive approaches and it does not lead to enhanced performance, mostly when the partitioning in regions is not as balanced as it would be desirable.

Moreover, CGG is the most regular algorithm, achieving fairly good results for all cases, especially in weakly connected environments or using larger teams, similarly to the MSP. However, it does not scale as well as the MSP as seen in Figure 3a; e.g., note the 12 robots case.

On the other hand, HPCC proves to be an algorithm with good performance mostly for smaller teams, independently of the graph connectivity, given that, although it plans its decisions online, the entire graph is considered (unlike HCR and CR). Also, for the same reason, its performance drops for larger teams, because all robots wander and plan in the whole environment, which raises the probability of encounters between them. Results also show that this approach is the one that converges sooner to an Idl_G value, as the number of robots is increased, which indicates reduced scalability.

Moving on to reactive algorithms, it is interesting to observe that HCR does not present evident improvements when compared to CR. According to its authors, HCR can eventually be tuned to give different weights to the vertices' distance and the instantaneous idleness of neighbors during decision-making.

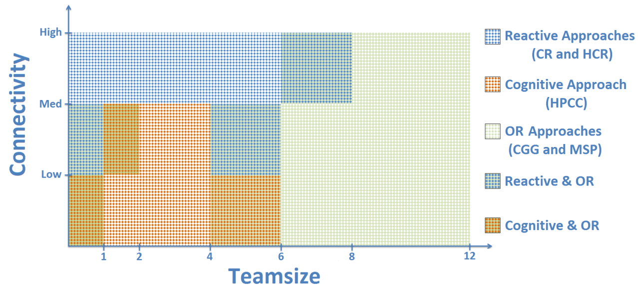


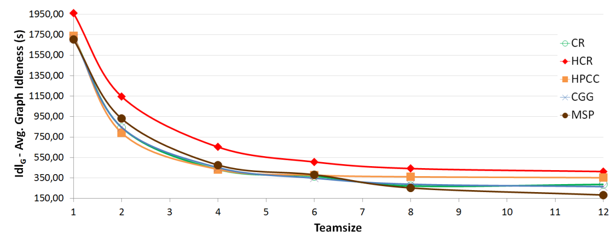
Fig. 2. General Simulation Results. In this chart, the best strategies for a given map connectivity and team size are shown. Note that the figure presents some intersections of design solutions.

In this work, the same weight for both parameters of the decision process was used and it was verified that for weakly connected environments, HCR was the algorithm with the worst performance (Figure 3a). This happens because robots tend to stay longer in regions with close vertices, causing high interference between robots, which compete to reach those vertices, reducing overall performance dramatically. As for the CR algorithm, it scales better than HCR and HPCC, only staying behind the MSP and CGG for large teams. Reactive algorithms have good performance especially in strongly connected environments, as seen in Figure 3c, where agents have alternatives to decide at the very moment, which vertex to move next to, taking into consideration the state of the system. Nevertheless, even in less connected environments, at some point when increasing the teamsize, the CR algorithm obtains better performance than the HPCC, since it scales better than the latter one.

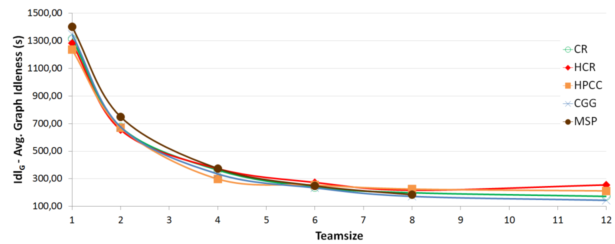
As expected, all algorithms display increasing performance only until reaching a certain group size, around which the group productivity stagnates and even drops with the addition of robots; e.g., HCR in environment B as illustrated by Figure 3b. In theory, productivity should grow during size scale-up; however spatial limitations increase the interference between robots causing the decrease of performance. For example, calculating Balch's speedup measure [19] for increasing teamsizes:

$$S[i] = \frac{\frac{P[1]}{i}}{P[i]} \tag{9}$$

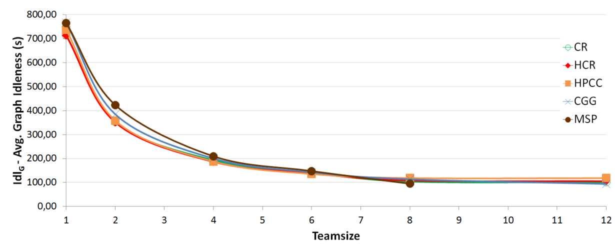
where $P[i]$ is the performance for i robots, it is straightforward to conclude that such systems rapidly enter in sublinear performance ($S[i] < 1$), as shown on top of Figure 4 for environment A. On the other hand, in the bottom of Figure 4 it is represented the interference measured for the same environment, which was calculated as the number of times that robots had to avoid each other in order not to collide. Online planning strategies were the ones which presented more interference. It can be seen that speedup and interference are negatively correlated. For larger teamsizes, instead of cooperating, robots tend to compete to firstly reach a given vertex than their teammates. Designing strategies which account for the teammates' goal can be beneficial for multi-robot patrolling, since they can take advantage of cooperation over competition between agents.



(a) Results for Environment A.



(b) Results for Environment B.



(c) Results for Environment C.

Fig. 3. Simulation Results: Idl_G performance curves.

It is also interesting to see that Figure 3 shows that, even though map B has a larger area to patrol when compared to map A, all algorithms obtain lower Idl_G values for the same number of robots in environment B, due to its greater connectivity. These results prove that graph connectivity is a very important parameter to consider when employing a patrolling algorithm in a given environment. Expectedly, the performance of the team is also greatly affected by graph dimension. However, when independent of the connectivity, graph size is seen as a scale factor when considering fixed teamsizes.

Furthermore, the median graph idleness value corresponds typically to around 85% of the average graph idleness, meaning that the frequency distribution is usually positively skewed (this is true in 96% of the trials). CR is the algorithm which has closest values between Idl_G and the median, which shows that the algorithm normally does not let points in the environment stay idle for too long, balancing more its visits to the graph's vertices when compared to other approaches. However, this property does not lead to better performance in terms of Idl_G values.

The maximum idleness (most unvisited vertex of each graph) was also calculated. It is typically around 2.7 times

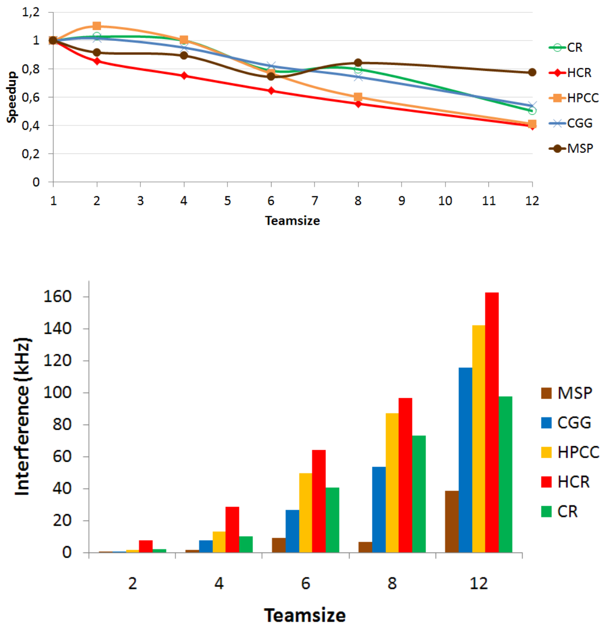


Fig. 4. Speedup and Interference in environment A.

larger than the average graph idleness. This ratio grows consistently with teamsize for all algorithms, being lower (around 2 times in average) for small teamsizes and increasing for higher teamsizes. As expected, CR due to its balanced property is the approach with a lower overall ratio of around 2.25 and surprisingly, if we consider the little difference between the two approaches, HCR is the algorithm with a higher ratio of around 3.25. The other three approaches have a ratio of around 2.6-2.7.

VII. Conclusion and Future Work

In this work, a study of the scalability and performance of five different multi-robot patrolling strategies was presented. This study is unprecedented in this field because it overcomes many limitations and simplifications of previous works by using generic environments with different topological connectivity properties and weighted edges; realistic simulations that consider the robots' dynamics; and is based on the actual time in its performance metric instead of atomic iterations or simulation cycles. It was shown that different types of algorithms perform differently according to the environment and the number of robots running the patrol task. Consequently, the choice of a patrolling strategy for teams of multiple robots should take into consideration these two important parameters. Moreover, to improve the team's performance, scalable methods should be developed to minimize interference between robots.

In the future, we intend to deepen the study on the scalability properties of multi-robot patrolling algorithms by analyzing the behavior of the variables of the problem in order to present an estimation method to dimension a team of robots in such missions according to the environment to patrol. Additionally, we intend to develop new scalable approaches for multi-robot patrol and test it in mobile robots and real scenarios.

Acknowledgment

The authors gratefully acknowledge Gonçalo Cabrita (ISR, University of Coimbra), Daniel Spielman (Department of Computer Science, Yale University), Eitan Marder-Eppstein and Brian Gerkey (Willow Garage and ROS Developers) for their contribution and feedback.

This work was financially supported by a PhD grant (SFRH/BD/64426/2009) from the Portuguese Foundation for Science and Technology (FCT) and the Institute of Systems and Robotics (ISR-Coimbra).

References

[1] A. Machado, G. Ramalho, J. Zucker and A. Drogoul. "Multi-Agent Patrolling: an Empirical Analysis of Alternative Architectures", *Multi-Agent-Based Simulation*, 3rd Int. Workshop, 155-170, Italy, July, 2002.

[2] A. Almeida. "Patrulhamento Multiagente em Grafos com Pesos". M.Sc. Thesis, Centro de Informática, Univ. Federal de Pernambuco, Recife, Brasil, 2003 (In Portuguese).

[3] Y. Elmaliach, N. Agmon and G. Kaminka. "Multi-Robot Area Patrol under Frequency Constraints", *In Proc. of the IEEE International Conf. on Robotics and Automation (ICRA 2007)*, 385-390, Italy, April, 2007.

[4] Y. Chevaleyre. "Theoretical Analysis of the Multi-agent Patrolling Problem", *In Proc. of the Intelligent Agent Technology: IAT '04, IEEE/WIC/ACM Int. Conf.*, 302-308, Beijing, China, Sep. 20-24, 2004.

[5] D. Portugal and R. Rocha. "MSP Algorithm: Multi-Robot Patrolling based on Territory Allocation using Balanced Graph Partitioning", *In Proceedings of 25th ACM Symposium on Applied Computing (SAC 2010)*, 1271-1276, Sierre, Switzerland, March 22-26, 2010.

[6] F. Sempé and A. Drogoul. "Adaptive Patrol for a Group of Robots". *Int. Conf. on Intelligent Robots and Systems*, Las Vegas, October, 2003.

[7] H. Santana, G. Ramalho, V. Corruble and B. Ratitch. "Multi-Agent Patrolling with Reinforcement Learning", *Int. Conf. on Autonomous Agents and Multiagent Systems*, 1122-1129, Vol.3, New York, 2004.

[8] T. Menezes, P. Tedesco and G. Ramalho. "Negotiator Agents for the Patrolling Task". *In Proceedings of the International Conf. IBERAMIA/SBIA*, LNAI. 4140, 48-57, Ribeirão Preto, São Paulo, 2006.

[9] H. Chu, A. Glad, O. Simonin, F. Sempé, A. Drogoul and F. Charpillet. "Swarm Approaches for the Patrolling Problem, Information Propagation vs. Pheromone Evaporation", *In Proceedings of the International Conference on Tools with Art. Intelligence*, 1, 442-449, France, 2007.

[10] K. Hwang, J. Lin and H. Huang. "Cooperative Patrol Planning of Multi-Robot Systems by a Competitive Auction System". *ICROS-SICE Int. Joint Conf.*, Japan, August, 2009.

[11] A. Almeida, G. Ramalho, H. Sanana, P. Tedesco, T.Menezes, V. Corruble and Y. Chaveleyre."Recent Advances on Multi-Agent Patrolling", *In Brazilian Symposium on Artificial Intelligence (SBIA 2004)*, 474-483, Vol. 3171, São Luís, Brazil, September, 2004.

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

[13] D. Portugal and R. Rocha. "A Survey on Multi-Robot Patrolling Algorithms". *In Proc. of the Doctoral Conf. on Computing, Electrical and Industrial Systems*, Costa da Caparica, Portugal, February, 2011.

[14] M. Fiedler. "Algebraic connectivity of Graphs", Czechoslovak Mathematical Journal: 23 (98), 1973.

[15] F. Chung. "Spectral Graph Theory", CBMS Lecture Notes, AMS, Providence, RI, 1997.

[16] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler and A. Ng. "ROS: an open-source Robot Operating System", *In Proc. of the 2009 Int. Conf. on Robotics and Automation (ICRA 2009)*, Workshop On Open Source Software, Kobe, Japan, May, 2009.

[17] E. Marder-Eppstein, E. Berger, T. Foote, B. Gerkey and K. Konolige. "The Office Marathon: Robust Navigation in an Indoor Office Environment". *In Proc. of the 2010 IEEE Int. Conf. on Robotics and Automation (ICRA 2010)*, 300-307, Anchorage, Alaska, USA, May 3-8, 2010.

[18] S. Thrun, D. Fox, W. Burgard and F. Dellaert. "Robust Monte Carlo Localization for Mobile Robots". *Artificial Intelligence (AI)*, 99-141, Vol. 128, No. 1-2, 2001.

[19] T. Balch and R. Arkin. "Communication in reactive multiagent robotic systems".*Autonomous Robots*, 1(1):27-52, 1994.