David Bina Siassipour Portugal

EFFECTIVE COOPERATION AND SCALABILITY IN MULTI-ROBOT TEAMS FOR AUTOMATIC PATROLLING OF INFRASTRUCTURES

Tese de Doutoramento em Engenharia Electrotécnica e Computadores orientada pelo Professor Doutor Rui Paulo Rocha e apresentada à Faculdade de Ciências e Tecnologia da Universidade de Coimbra

Setembro de 2013

• U C •

Universidade de Coimbra

· U C ·

Department of Electrical Engineering and Computers
Faculty of Science and Technology
University of Coimbra

Effective Cooperation and Scalability in Multi-Robot Teams for Automatic Patrolling of Infrastructures

Ph.D. Thesis

David Bina Siassipour Portugal

Coimbra, September 2013

Department of Electrical Engineering and Computers
Faculty of Science and Technology
University of Coimbra

Effective Cooperation and Scalability in Multi-Robot Teams for Automatic Patrolling of Infrastructures

Thesis submitted:

to the Department of Electrical Engineering and Computers of the Faculty of Science and Technology of the University of Coimbra in partial fulfilment of the requirements for the Degree of Doctor of Philosophy.

David Bina Siassipour Portugal

Coimbra, 2013

This Ph.D. thesis was developed under the Supervision of

Doctor Rui P. Rocha
Professor of the Faculty of Science and
Technology, University of Coimbra

This research work was supported by a Ph.D. scholarship (SFRH/BD/64426/2009), the CHOPIN R&D research project (PTDC/EEA-CRO/119000/2010) and by the Institute of Systems and Robotics (project PEst-C/EEI/UI0048/2011), all of them funded by the Portuguese science agency FCT - Fundação para a Ciência e a Tecnologia (Foundation for Science and Technology).

Although this may seem a paradox,
all exact science is dominated by
the idea of approximation.

Bertrand Russell, 1931.

Abstract

In the digital era that we live in, advances in technology have proliferated throughout our society, quickening the completion of tasks that were painful in the old days, improving solutions to the everyday problems that we face, and generally assisting human beings both in their professional and personal life. Robotics is a clear example of a broad technological field that evolves every day. In fact, scientists predict that in the upcoming few decades, robots will naturally interact and coexist alongside human beings.

While it is true that robots already have a strong presence in industrial environments, *e.g.*, robotic arms for manufacturing, the average person still looks upon robots with suspicion, since they are not acquainted by such type of technology. In this thesis, the author deploys teams of mobile robots in indoor scenarios to cooperatively perform patrolling missions, which represents an effort to bring robots closer to humans and assist them in monotonous or repetitive tasks, such as supervising and monitoring indoor infrastructures or simply cooperatively cleaning floors.

In this context, the team of robots should be able to sense the environment, localize and navigate autonomously between way points while avoiding obstacles, incorporate any number of robots, communicate actions in a distributed way and being robust not only to agent failures but also communication failures, so as to effectively coordinate to achieve optimal collective performance. The referred capabilities are an evidence that such systems can only prove their reliability in real-world environments if robots are endowed with intelligence and autonomy. Thus, the author follows a line of research where patrolling units have the necessary tools for intelligent decision-making, according to the information of the mission, the environment and teammates' actions, using distributed coordination architectures.

An incremental approach is followed. Firstly, the problem is presented and the literature is deeply studied in order to identify potential weaknesses and research opportunities, backing up the objectives and contributions proposed in this thesis. Then, problem fundamentals are described and benchmarking of multi-robot patrolling algorithms in realistic conditions is conducted. In these earlier stages, the role of different parameters of the problem, like environment connectivity, team size and strategy philosophy, will become evident through extensive empirical results and statistical analysis. In addition, scalability is deeply analyzed and tied with inter-robot interference and coordination, imposed by each patrolling strategy.

After gaining sensibility to the problem, preliminary models for multi-robot patrol with special focus on real-world application are presented, using a Bayesian inspired formalism. Based on these, distributed strategies that lead to superior team performance are described. Interference between autonomous agents is explicitly dealt with, and the approaches are shown to scale to large teams of robots. Additionally, the robustness to agent and communication failures is demonstrated, as well as the flexibility of the model proposed. In fact, by later generalizing the model with learning agents and maintaining memory of past events, it is then shown that these capabilities can be inherited, while at the same time increasing team performance even further and fostering adaptability. This is verified in simulation experiments and real-world results in a large indoor scenario.

Furthermore, since the issue of team scalability is highly in focus in this thesis, a method for estimating the optimal team size in a patrolling mission, according to the environment topology is proposed. Upper bounds for team performance prior to the mission start are provided, supporting the choice of the number of robots to be used so that temporal constraints can be satisfied.

All methods developed in this thesis are tested and corroborated by experimental results, showing the usefulness of employing cooperative teams of robots in real-world environments and the potential for similar systems to emerge in our society.

Keywords: Distributed Systems, Multi-Robot Patrol, Multi-Agent Learning, Security, Graph Theory, Topological Maps, Scalability, Performance and Robustness.

Resumen

En la era digital en la que vivimos, los avances tecnológicos han proliferado entre la sociedad, acelerando la realización de tareas que antes resultaban difíciles, mejorando las soluciones a los problemas que nos enfrentamos día a día, y en general, proporcionando asistencia a los seres humanos tanto en la vida profesional como la personal. La robótica es un claro ejemplo de un ancho campo tecnológico que evoluciona día a día. De hecho, los científicos predicen que en las décadas venideras, los robots interaccionarán y coexistirán de forma natural con los seres humanos.

Si bien es cierto que los robots tienen ya una fuerte presencia en entornos industriales, p. ej. brazos robóticos para la fabricación, la población en general mira todavía a los robots con cierta desconfianza, ya que no está familiarizada con este tipo de tecnología. En esta tesis, el autor desarrolla equipos de robots móviles en escenarios de interiores cuyo objetivo es llevar a cabo misiones de patrullaje de manera cooperativa. Esto supone el esfuerzo de acercar los robots a los humanos, para que les asistan en tareas repetitivas y monótonas, tales como la supervisión y la monitorización de infraestructuras de interior o simplemente la limpieza de suelos de manera cooperativa.

En este contexto, el equipo de robots debería ser capaz de percibir el entorno, localizarse y navegar de manera autónoma entre puntos predefinidos y esquivando posibles obstáculos; incorporar un número indefinido de robots; comunicar acciones de manera distribuida; y ser robustos no sólo ante fallos en los propios agentes, sino también ante fallos en las comunicaciones, para así coordinarse de manera efectiva y conseguir un comportamiento colectivo óptimo. Las mencionadas capacidades son una evidencia de que estos sistemas sólo pueden demostrar su fiabilidad en entornos reales si los robots son dotados de inteligencia y autonomía. Así, el autor sigue una línea de investigación en la que las unidades de patrullaje deben ser dotadas de las herramientas necesarias para poder llevar a cabo una toma de decisiones inteligente, de acuerdo a la información de la misión, el entorno y las acciones de los compañeros de equipo, utilizando arquitecturas de coordinación distribuida.

Para ello se sigue una aproximación incremental. En primer lugar, se presenta el problema y se estudia a fondo la literatura, con el objetivo de identificar potenciales debilidades y oportunidades de investigación, respaldando los objetivos y contribuciones propuestos en esta tesis. A continuación, se presentan los fundamentos del problema en sí, y se llevan a cabo *benchmarks* de algoritmos de patrullaje de sistemas multi-robot en condiciones realísticas. En estas etapas iniciales, el papel de los diferentes parámetros del problema, como la conectividad del entorno, el tamaño del equipo y la filosofía de la estrategia, se harán evidentes, a través de extensos resultados empíricos y análisis estadístico. Además, se analiza en profundidad la escalabilidad y se enlaza con la coordinación e interferencias entre robots, impuesta por cada estrategia de patrullaje.

Después de incrementar la sensibilidad hacia el problema, se presentan modelos preliminares para patrullaje multi-robot con especial énfasis en aplicaciones del mundo real, utilizando un formalismo inspirado en la decisión Bayesiana. Basadas en esto, se describen estrategias distribuidas que llevan a un mayor desempeño de los equipos. Se abordan de manera explícita las interferencias entre agentes autónomos, y las aproximaciones se muestran escalables a grandes equipos de robots. Además, se demuestra la robustez ante fallos de los agentes e de comunicación, así como la flexibilidad del modelo propuesto. De hecho, se muestra a continuación que estas capacidades pueden ser heredadas haciendo una posterior generalización del modelo con agentes capaces de aprender y de mantener en memoria los eventos pasados, incrementando a su vez aun más el desempeño y adaptabilidad del equipo. Esto último es verificado tanto en experimentos en simulación como con resultados en un gran entorno real de interiores.

Además, puesto que el problema de la escalabilidad de los equipos es un tema importante en esta tesis, se ha propuesto un método para estimar el tamaño óptimo del equipo en una misión de patrullaje, de acuerdo a la topología del entorno. Se proporcionan límites superiores de desempeño del equipo previos al inicio de la misión, ayudando a la elección del número de robots a ser utilizados, de manera que puedan satisfacerse las restricciones temporales.

Todos los métodos desarrollados en esta tesis han sido testeados y corroborados con resultados experimentales, mostrando la utilidad de utilizar equipos de robots cooperativos en entornos del mundo real, y la potencialidad de que sistemas similares puedan emerger en nuestra sociedad.

Palabras clave: Sistemas Distribuidos, Patrullaje multi-robot, Aprendizaje multi-agente, Seguridad, Teoría de Grafos, Mapas Topológicos, Escalabilidad, Desempeño y robustez.

Resumo

Na era digital em que vivemos, os avanços na tecnologia proliferam em diversos sectores da sociedade, acelerando a conclusão de tarefas que outrora eram dolorosas, melhorando soluções para os problemas que enfrentamos no quotidiano, e ajudando, em geral, os seres humanos, tanto na vida profissional como pessoal. A Robótica é um exemplo claro de uma ampla área tecnológica que evolui a cada dia. De facto, cientistas prevêem que nas próximas décadas, robôs irão interagir e conviver naturalmente com os seres humanos.

A Robótica possui hoje em dia uma presença forte em ambientes industriais, um exemplo disso mesmo são os braços robóticos para manufactura. No entanto, a pessoa comum ainda olha para os robôs com desconfiança, uma vez que não está familiarizada com este tipo de tecnologia. Nesta tese, o autor foca-se em equipas de robôs móveis em ambientes fechados, que executam missões de patrulhamento cooperativas. Este trabalho representa um esforço para aproximar os robôs dos seres humanos, podendo ajudá-los em tarefas monótonas ou repetitivas, tais como a supervisão e monitorização de infra-estruturas, ou simplesmente a limpeza cooperativa de pisos.

Neste contexto, a equipa de robôs deve ser capaz de interagir correctamente com o ambiente, localizar-se e navegar de forma autónoma entre pontos enquanto evita obstáculos, incorporar um número arbitrário de robôs, comunicar acções de forma distribuída e ser robusta não só a falhas de agentes, mas também a falhas de comunicação, de forma a coordenar-se eficazmente e alcançar um desempenho colectivo ideal. As capacidades que se referem atrás são uma evidência de que esses sistemas só podem provar a sua fiabilidade em ambientes do mundo real se os robôs forem dotados de inteligência e autonomia. Assim sendo, o autor segue uma linha de investigação onde os agentes de patrulhamento possuem as ferramentas necessárias para tomada de decisão inteligente, de acordo com informação decorrente da missão, do meio ambiente e das acções dos companheiros de equipa, utilizando arquitecturas de coordenação distribuída.

A abordagem seguida é incremental. Em primeiro lugar, o problema é apresentado e a literatura é profundamente estudada a fim de identificar potenciais pontos fracos e oportunidades de pesquisa, justificando os objectivos e contribuições propostas nesta tese. Em seguida, as bases necessárias para o problema são descritas e procede-se a uma análise comparativa de algoritmos de patrulhamento multi-robô em condições realistas. Nesta fase preliminar, o papel dos diferentes parâmetros do problema, tais como a conectividade do ambiente, tamanho da equipa e filosofia da estratégia tornar-se-á evidente, através de extensos resultados empíricos e análise estatística. Para além disso, a escalabilidade é profundamente analisada e relacionada com interferência inter-robô e a coordenação imposta por cada estratégia de patrulhamento.

Depois de ganhar sensibilidade para o problema, são apresentados modelos preliminares para patrulha multi-robô com foco especial na aplicação do mundo real, utilizando um formalismo inspirado em decisão Bayesiana. Baseado nestes modelos, são descritas estratégias distribuídas de patrulhamento que levam a um superior desempenho da equipa. As abordagens tratam explicitamente da interferência entre agentes autónomos, e demonstra-se a sua habilidade para integrar grandes equipas de robôs. Além disso, a robustez a falhas de agentes e a falhas de comunicação é demonstrada, bem como a flexibilidade do modelo proposto. De facto, mais tarde ao generalizar o modelo com aprendizagem por parte dos agentes e memorização de eventos passados, é mostrado que estas capacidades podem ser herdadas, enquanto que, ao mesmo tempo, se melhora ainda mais o desempenho da equipa e se promove adaptabilidade. Isto é verificado em experiências em simulação e resultados no mundo real num cenário interior de largas dimensões.

Adicionalmente, uma vez que a questão da escalabilidade da equipa é extremamente importante nesta tese, um método para estimar o tamanho ideal da equipa numa missão de patrulhamento, de acordo com a topologia do ambiente é proposto. São fornecidos majorantes de desempenho da equipa ainda antes do início da missão, assistindo a escolha do número de robôs a ser utilizado, de modo a satisfazer restrições temporais da tarefa.

Todos os métodos desenvolvidos nesta tese são testados e comprovados por resultados experimentais, demonstrando a utilidade de equipas cooperativas de robôs em ambientes no mundo real e do potencial destes sistemas na nossa sociedade.

Palavras-Chave: Sistemas Distribuídos, Patrulha Multi-Robô, Aprendizagem Multi-Agente, Segurança, Teoria de Grafos, Mapas Topológicos, Escalabilidade, Desempenho e Robustez.

Acknowledgment

During the last four years, several people have been fundamental for the conclusion of this Thesis. Firstly, I wish to express my deepest gratitude to my supervisor Prof. Rui Rocha, for his accompaniment and input, which increased the quality of the work, as well as his wisdom and kindness, in every step of the way. I would also like to thank the support of the Institute of Systems and Robotics (ISR) at the University of Coimbra (UC), granting me the conditions to develop the work in this thesis.

I would like to make a much deserved special mention to Micael Couceiro, who was my travel companion in this long Ph.D. journey. We worked together in several occasions such as in the Doctoral program, when building and developing the Traxbot and Stingbot robot platforms, as well as in every task in the scope of the CHOPIN project. His terrific work capacity is inspirational and I feel lucky to have had such a talented colleague.

Important parts of this thesis also arose from fruitful discussions and exchange of views with some of my colleagues at the Mobile Robotics Lab (MRL), from which I would like to especially highlight Luís Santos, Pedro Trindade, Ricardo Martins, Diego Faria, Hadi Aliakbarpour, Kamrad Khoshhal, José Prado and Amílcar Ferreira, whom have been with me, most of the time that I spent at the MRL. I would also like to mention other MRL team members, which I had the pleasure to meet: Bruno Patrão, Christiana Tsiourti, Jafar Hosseini, João Quintas, João Filipe Ferreira, Luís Almeida, Nuno Dias, Paulo Drews Jr, Pedro Machado and Rita Catarino.

My appreciation goes, as well, to the MSc. students that worked under my co-supervision, namely André Araújo, Amadeu Fernandes, Walter Miani, João M. Santos and João Martins. I was able to learn and become a better researcher with every single one of them. Also, they formed my team of “shepherds” along with some other people, by helping me controlling the sometimes painful experiments at the ISR corridors when several robots were used. Within the team of “shepherds”, I also thank deeply the kind help of Gonçalo Augusto, Nuno L. Ferreira and José S. Pereira, for taking the time in helping me with my experiments.

I am greatly thankful to Prof. Hélder Araújo from ISR/UC, for conceding three Pioneer-3DX robots used in the experiments in chapter 5, as well as to Gonçalo Cabrita, who showed me that ROS was the next big thing in Robotics back in 2009, when no-one was using it at the ISR but him. In addition, I acknowledge the contribution of Charles E. Pippin, from the Georgia Tech Research Institute, in Atlanta (US), whom I had the pleasure to meet in SAC 2013, in Coimbra, and with whom I jointly wrote a paper on near-optimal routes using deterministic strategies, which served as the basis for the estimation method proposed in chapter 6 of this thesis. I also thank Jorge Sales for his kindness in translating the abstract of this thesis to Spanish.

Most of all, I would like to thank my parents, family and girlfriend Vanessa, for the neverending support and patience to endure my constant unavailability in the course of this work. Without your comprehension and motivation, it just would not be possible. This thesis is dedicated to you.

I would also like to give a big thanks to all my friends from the “Quartel”, especially to João B. Campos and José Martins, for our weekly discussions and for putting up with my often late arrivals for lunch.

I thank also the members of the jury, for the time and commitment on the evaluation of this thesis, as well as the anonymous reviewers of my scientific papers for the constructive comments and corrections.

All research work leading to this thesis was supported by a FCT scholarship (SFRH/BD/64426/2009) granted since January 2010 until December 2013.

Declaration

The work in this Ph.D. Dissertation is based on research carried out at the Mobile Robots Laboratory (MRL) of the Institute of Systems and Robotics (ISR) in Coimbra, Portugal. No part of this thesis has been submitted elsewhere for any other degree or qualification and it is all my own work unless referenced to the contrary in the text.

Contents

List of Figures

List of Tables

List of Algorithms

Acronyms

2D	Two-Dimensions/Two-Dimensional
AGV	Automated Guided Vehicle
AI	Artificial Intelligence
AMCL	Adaptive Monte Carlo Localization
ANOVA	Analysis of Variance
CBLS	Concurrent Bayesian Learning Strategy
CCO	Command Center of Operations
CDT	Constrained Delaunay Triangulation
CHOPIN	Cooperation between Human and rObotic teams in catastroPhic INcidents
CGG	Cyclic Algorithm for Generic Graphs
CR	Conscientious Reactive Algorithm
DEEC	Department of Electrical Engineering and Computers
DFS	Depth-First Search
EHP	Evolutionary Heuristic for the min-max cost closed walk Problem
EVG	Extended Voronoi Graph
GBS	Greedy Bayesian Strategy
GUI	Graphical User Interface
GPU	Graphics Processing Unit
HCR	Heuristic Conscientious Reactive Algorithm
HPCC	Heuristic Pathfinder Conscientious Cognitive
HTSP	Heuristic to approximate the TSP tour

ISR	Institute of Systems and Robotics - University of Coimbra
LIDAR	Light Detection And Ranging
LRF	Laser Range Finder
MANET	Mobile *Ad Hoc* Network
MRL	Mobile Robotics Laboratory (of the ISR)
MRPP	Multi-Robot Patrolling Problem
MRS	Multi-Robot Systems
MSP	Multilevel Subgraph Patrolling Algorithm
MST	Minimum Spanning Tree
MSTt	MST Tour approximation algorithm
OLSR	Optimized Link State Routing Protocol
OpenCV	Open Source Computer Vision
OpenGL	Open Graphics Library
OpenRAVE	Open Robotics Automation Virtual Environment
OSRF	Open Source Robotics Foundation
TMR	Tracked Mobile Robot
SaR	Search and Rescue
TSP	Traveling Salesman Problem
SEBS	State Exchange Bayesian Strategy
SIGA	Santander Interactive Guest Assistants
ROS	Robot Operating System

Notation

Graph Theory

$\mathcal{G}$	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.
$e_{i,j}$	Edge that connects v_i to v_j.
$\lvert e_{i,j}\rvert$	Weight of $e_{i,j}$.
$\lvert\mathcal{V}\rvert$	Cardinality of $\mathcal{V}$: Total number of vertices.
$\lvert\mathcal{E}\rvert$	Cardinality of $\mathcal{E}$: Total number of edges.
x	Generic path in the graph, composed by an array of vertices.
$N_{\mathcal{G}}(v_i)$	Neighborhood of v_i: set of adjacent vertices of v_i.
$deg(v_i)$	Degree of v_i: number β of adjacent vertices of v_i.
D	Graph Density.
$\mathscr{L}$	Normalized Laplacian matrix of $\mathcal{G}$.
λ_1	Fiedler value or algebraic connectivity of $\mathcal{G}$.
η	Ratio between the maximal and minimal edge weight on $\mathcal{G}$.
Γ	Chain Graph.
X	Closed walk on $\mathcal{G}$.
$\mathbf{P}$	Partition set on $\mathcal{G}$.
P_r	Partition assigned to robot r.

Graph Theory (continued)

$\mathcal{L}(X)$	Length (or cost) of a walk X.
$\mathcal{T}$	Minimum Spanning Tree of $\mathcal{G}$.
$\mathcal{G}_C$	Complete Graph obtained from $\mathcal{G}$.

General MRPP Notation

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.
$\mathcal{WI}$	Worst idleness of any vertex of the graph.
$\max(\overline{\mathcal{I}_{\mathcal{V}}})$	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.
R	Number of robots.
r	Robot ID.
c	Constant robot speed.
π_r	Patrolling route of robot r.
Π	Set of all R patrolling routes.
$v(R)$	Speedup of R robots.
$\Psi(R)$	Performance of R robots.
ξ	Rate of communication failures.
ζ_i	Normalized number of visits to v_i.
ς	Normalized Distance per Patrol.
$\mathcal{C}$	Average Number of Vertices Visited on the Patrol.
Ω	Idleness Timing Constraint (in seconds).

Probabilities and Statistical Notation

σ	Standard Deviation.
α	Significance level of the ANOVA F-test.
$\mathbf{H_0}$	Null hypothesis.
$\mathbf{H_1}$	Alternative hypothesis.
SS	Sum of Squares.
dof	Degrees of Freedom.
MS	Mean Squares.
F	F-ratio.
$P(\cdot)$	Probability.
$P(\cdot\|\cdot)$	Conditional Probability.
H	Entropy.
$\mathcal{H}$	Normalized Entropy.

Proposed MRPP Strategies

G_i	Gain of moving to v_i, a continuous random variable.
$\|e_{min}\|$	Edge weight threshold.
$f(g)$	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.
$\theta_{0,i}$	Arc Strength from v_0 to v_i.
$\boldsymbol{\theta}$	Arc Strength set.
$\gamma_{0,i}$	Reward of the arc $e_{0,i}$.
$\mathcal{S}_{0,i}$	Reward Sign of the arc $e_{0,i}$.
κ	Initial Arc Strength value.
ω	Look-ahead weight.
Φ	Exploration factor in EHP.

Chapter 1

Introduction

One of the fundamental areas in Robotics is multi-robot systems. More particularly, this thesis addresses the cooperation of a team of mobile robots in patrolling missions. The main aspects studied herein are strategies for effective patrolling performance, agents' coordination, scalability and applicability in real-life situations.

This introductory chapter presents the context of the research in order to clarify the motivation and significance of the problem. In addition, some guidelines about multi-robot systems in general and, more specifically, agents in patrolling missions are herein introduced to lay the groundwork to approach the problem in hands. Finally, an overview of the document is given.

1.1 Context and Motivation

In recent years, robotics has been one of the scientific fields with the most substantial advances. Within the diverse areas that it embraces, mobile robotics has had great focus in the last decades from roboticists (*i.e.*, researchers on robotics) around the world. In particular, issues like autonomous navigation, path planning, self-localization, coordination of robots, cooperative dynamics, mapping, exploration and coverage have become popular and have benefited from the progress of artificial intelligence, control theory, real-time systems, sensors' development, electronics, communication systems and systems integration [Parker, 2008].

Nowadays, we expect to see robots with many different shapes operating in different environments as on land, underwater, in the air, suspended on wires, climbing and so on. This evident growth is extremely motivating for the development and contribution of new developments by the community.

Security applications are a fundamental task with unquestionable impact on society. Combining this fact with the technological evolution observed in the last decades, it becomes clear that robot assistance can be a valuable resource by taking advantage of robots' expendability. In particular, multi-robot patrolling has high utility and is considered as a contemporary area with some relevant work presented in the last decade, especially in terms of strategies for coordinating teams of robots. However, many of the studies in the literature present unrealistic simplifications, strong limitations or questionable applicability as illustrated later on. Therefore, there is an eminent potential to explore in this context.

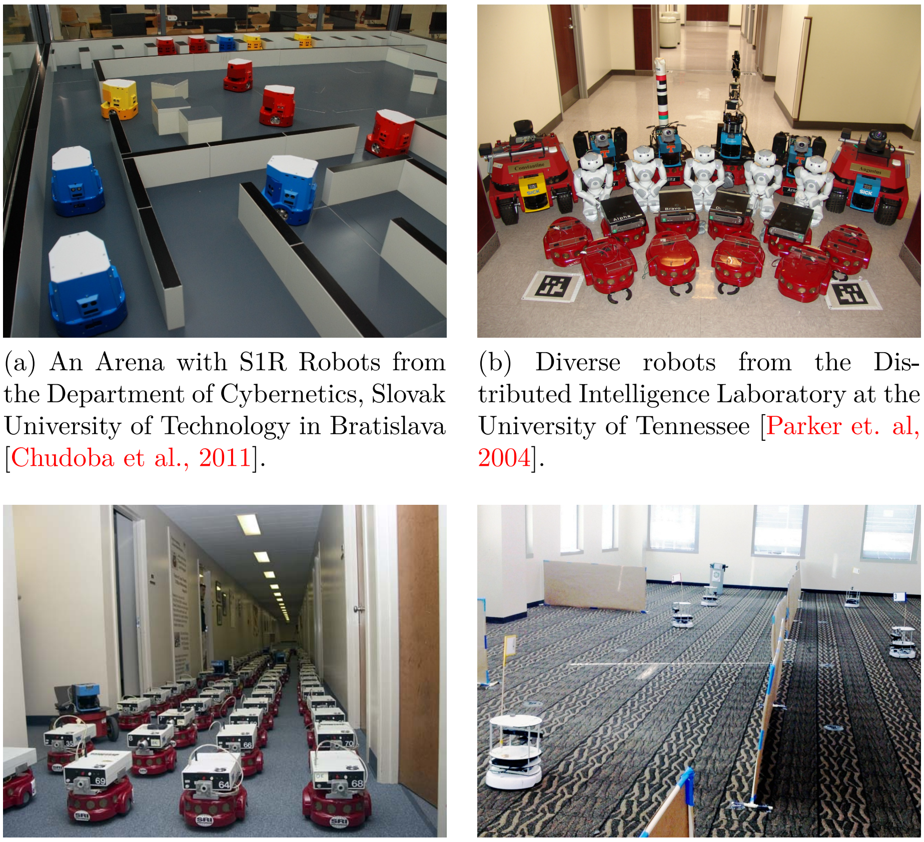
A crucial research question is: "Why is it important to study the Patrolling Problem?". Similarly to other studies on robotics like [Marjovi et al., 2009], multi-robot patrolling is a task with the potential to replace or assist human operators in dangerous real-life scenarios like mine clearing [Murphy et. al, 2009] or rescue operations in catastrophic scenarios [Couceiro et al., 2013]. The spectrum of applicability is vast. Note however that replacing human operators should not generate social controversy, as for instants, by increasing future unemployment rate. The key conception is to safeguard human lives, reducing the risk to human operators in the face of dangerous scenarios or assist them in monotonous and repetitive tasks, in a similar way as an industrial manipulator would do, by easing arduous, tiring and time-consuming tasks. Replacing people with autonomous robots in these scenarios provides inestimable benefits. Furthermore, it can be difficult to coordinate human teams to guarantee that the area is free of intruders. This suggests using robots especially equipped for assistance, offering the possibility to relieve human beings, enabling them to be occupied in nobler tasks like, for example, monitoring the system from a safe location [Murphy, 2004].

Moreover, the patrolling problem is very challenging in the context of multi-robot systems, because agents must navigate autonomously, coordinate their actions in a distributed way and acquire information about the surrounding space, possibly with communication constraints and independently of the number of robots in the team and the environment's dimension. Clearly, 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. All of these features lead to an excellent case study in mobile robotics and conclusions drawn from such studies may support the development of future approaches not only in the patrolling domain but also in multi-robot systems, in general.

1.2 Multi-Robot Systems

During the last two decades, researchers in the field of mobile robotics have begun to investigate problems that involve multiple robots rather than using single robots (*cf.* Figure 1.1), and research in multi-robot systems (MRS) has witnessed notorious progress as never before.

In many applications, an autonomous mobile robot equipped with different sensors may adequately complete a given assignment. However, in several situations, it proves to be more expensive, less efficient and less robust than using a multi-robot system. In some cases, due to the need of combining different tasks and the dynamics of the environment, it is only viable to achieve the mission with



(a) An Arena with S1R Robots from the Department of Cybernetics, Slovak University of Technology in Bratislava [Chudoba et al., 2011].

(b) Diverse robots from the Distributed Intelligence Laboratory at the University of Tennessee [Parker et. al, 2004].

(c) The DARPA Centibots Project (SRI International, Stanford University, University of Washington and ActivMedia) [Fox et al., 2006] .

(d) A team of Turtlebots from the center for Robotics and Intelligent Machines at Georgia Tech Research Institute [Pippin et al., 2013].

Figure 1.1: Examples of multi-robot systems.

a multiple distributed autonomous robotic system. According to [Rocha, 2005], “For some robotic tasks, especially those that are intrinsically distributed and complex (...), a team of several cooperative mobile robots - a cooperative multi-robot system - is required to either make viable the mission accomplishment or, at least, accomplish the mission with better performance than a single robot”.

Some characteristics of multi-robot systems include distributed control, autonomy, communicative agents and greater fault-tolerance. A single robot may be vulnerable to hostile environments or attackers, for example, in military actions. In such scenarios, agents would greatly benefit from the assistance of nearby agents during emergencies, failures or malfunctions.

Another important advantage is the possibility of having many robots in numerous places, carrying out diverse tasks at the same time, *i.e.*, space distribution. Most missions are solved much quicker if robots operate in parallel. Increasing robustness and reliability of the solution is also feasible in MRS by introducing redundancies in the capabilities across robot team member and graceful performance degradation, remaining functional if some of the agents fail.

From the standpoint of cost-effectiveness, it can actually be cheaper and more practical to build a set of less capable and simpler robots that cooperate, instead of one single robot to perform the entire mission, depending on the intended application.

In more complex problems, which require solving distinct tasks in different locations of the environment, it can be useful to divide the problem in simpler subtasks and assign them to different robots of the team. Task decomposition together with effective cooperation can be a major advantage of a multi-robot system, if correctly designed. This feature can be used, for example, in exploration of unknown environments. To increase reliability and robustness, in comparison to a single autonomous mobile robot incorporated with all kind of sensors and abilities, a team of multiple robots may be heterogeneous by having spread resources.

The complexity of the presented multi-robot systems throughout the years has been growing. Mobile robot teams are expected to respond robustly, reliably and adaptively to unknown and dynamic environments, mechanical or communication failures, learning of new skills or addition of new robots.

One of the main difficulties when approaching these systems is to coordinate many robots to perform a complex, global task in an efficient manner, maximizing group performance under a wide range of conditions, with the flexibility to take advantage of the resources available, embrace the requirements and constraints imposed and resolve issues like action selection, coherence, conflict resolution and communication. This cannot be done by just increasing the number of robots assigned to a task. A coordination mechanism must exist to establish relationships

Table 1.1: Examples of applications of multi-robot patrol.

Area of Application	Example
Rescue Operations	Monitoring trapped or unconscious victims in the aftermath of a catastrophic scenario
Military Operations	Mine clearing
Surveillance and Security	Clearing a building
Supervision of Hazardous Environments	Patrolling toxic environments
Safety	Preventive patrol for gas leak detection
Environmental Monitoring	Sensing humidity and temperature levels inside a facility
Planetary Exploration	Collecting samples
Cooperative Cleaning	Household vacuum and pool cleaning
Areas with restricted access	Sewerage inspection
Vehicle Routing	Transportation of elderly people
Industrial Plants	Stock Storage
Computer Systems	War-game simulations

between agents so that they can accomplish the mission effectively.

1.3 The Multi-Robot Patrolling Problem

Patrolling an infrastructure with multiple robots is no different than other multi-robot assignments, in the sense that it incorporates all the previously mentioned characteristics of MRS. To understand this problem, it is important to firstly introduce the definition of patrol.

Definition 1 (Patrol)**.** *According to the Webster's online dictionary [Webster's, 2013], to patrol is literally "the activity of going around or through an area at regular intervals for security purposes".*

In the context of this thesis, it is assumed that instead of having a human guard or a group of men, the patrolling task should be performed by multiple autonomous and cooperating mobile robots in a real-life environment.

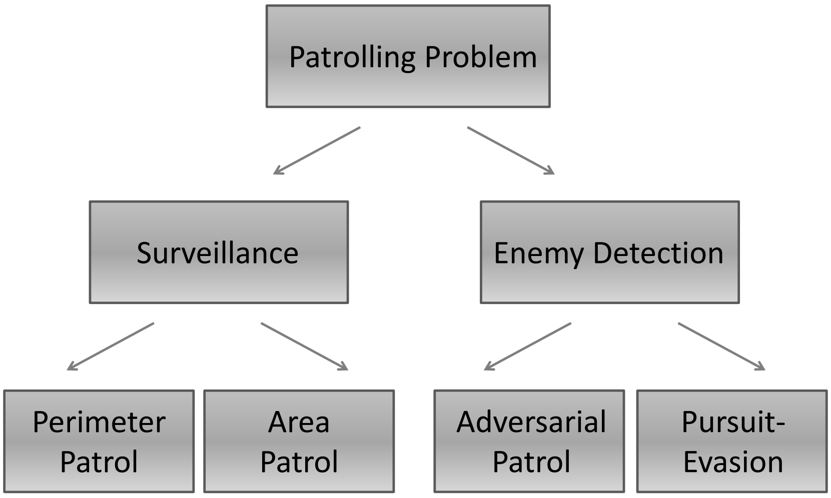


Figure 1.2: Research areas inside the Patrolling problem.

Patrolling is as a somehow complex multi-robot mission, requiring an arbitrary number of agents to coordinate their decision-making with the ultimate goal of achieving optimal group performance. It also aims at monitoring, protecting and supervising environments, obtaining information, searching for objects and detecting anomalies in order to guard the grounds from intrusion. Hence, a wide range of applications are possible, as exemplified in Table 1.1.

It is the author's belief that employing teams of robots for active surveillance tasks has several advantages over, for instance, a camera-based passive surveillance system. Robots are mobile and have the ability to travel in the field, collect environmental samples, act or trigger remote alarm systems and inspect places that can be hard for static cameras to capture. These capabilities are greatly beneficial to safeguard human lives and in terms of the flexibility of the deployed system..

Essentially, investigation on the Multi-Robot Patrolling Problem (MRPP) is divided in two classes, as seen in Figure 1.2, surveillance and enemy detection. Both have different goals and metrics. While a good strategy for detecting enemies does not necessarily require constant visits to all locations of the environment, a good surveillance strategy (also known as supervision or monitoring) intrinsically involves such frequent visits to all places in the environment[1].

On one hand, enemy detection techniques are usually divided in two types of strategies in the literature: adversarial patrol [Basilico et al., 2009, Agmon et al., 2011, Aguirre and Taboada, 2012] and pursuit-evasion [Ishiwaka et al., 2003, Vieira et al., 2009, Strom et al., 2010]. Both of these assume a target or an intruder inside the environment. However, while the target is mobile in the latter one, in the former one the target may be motionless and an explicit search for the opponent may not exist. For example, agents in an adversarial patrol task may strategically position themselves in the environment to attain maximal environmental coverage. Thus, the problem finds inspiration from the classical art gallery problem [Sherman, 1992], which is a visibility problem that aims to find the minimum number of static guards who together can observe an entire polygonal region. The goal in enemy detection is to recognize the target as quickly as possible. Usually behavior models of the adversary are considered, and performance is evaluated according to exploration time and other similar metrics.

On the other hand, surveillance techniques can be divided in perimeter patrol and area patrol. As their name suggests, perimeter patrol addresses supervision around a closed area [Agmon et al., 2008, Marino et al., 2009, Jensen et al., 2011], while area patrol is concerned with supervision inside a region of space [Machado et al., 2003, Iocchi et al., 2011, Pasqualetti et al., 2012a]. Oppositely to enemy detection, in surveillance methods, agents are continuously traveling inside an area for an indefinite period of time, seeing that every position in the environment, or at least the ones that require surveillance, must be regularly visited. Therefore, metrics based on the time spent without visiting important places in the environment or the frequency of visits have been proposed to gauge the performance of several strategies. Note that an intruder may not necessarily exist in these scenarios, *e.g.*, cooperative cleaning.

The focus in this thesis is on monitoring and supervision of environments, more specifically on area patrolling missions. Therefore, the words “Patrol” and “Patrolling”, are implicitly used in this sense throughout this document.

[1] Alternatively, it may involve visiting all important designated locations instead of visiting *every single* location.

Some distinct area patrolling strategies for teams of multiple robots have been presented in the last decade and it is consensual that a good strategy should minimize the time lag between visits in strategic places of the environment. More detail on the literature is given in the next chapter.

Also, it is important to address the characteristics of the patrolled environment. For example, the environment may be static or dynamic, *i.e.*, there may be changes in the environment during execution of the mission due, for example, to mobile obstacles. In these cases, the agents have to keep track of all changes and update their local representation of the environment. There may be also areas of the environment with different patrolling priorities if, for example, a region is more critical or more susceptible to attacks than others. In this case, such regions will need to be visited more often.

Robotic agents are normally endowed with a representation of the environment, which is typically an occupancy grid model, which in turn, is normally abstracted by a simpler, yet precise representation: a topological map (*i.e.*, a graph).

As it is shown in the next chapter, most of the previous works in this field are based on topological representations of the environment. By having a graph representation, one can use vertices to represent specific locations and edges to represent the connectivity between these locations. The multi-robot patrolling problem can, thus, be reduced to coordinate robots in order to visit all vertices of the graph ensuring the absence of intruders or other abnormal situations, with respect to a predefined optimization criterion.

Beyond the representation of the environment, it is important to consider other properties of the robotic team like their perception, which is not necessarily global. Agents may only have local awareness of the environment around them, which, in general, makes the problem harder to tackle and more dependent on communication mechanisms, in order to update knowledge about the state of the environment and teammates.

Moreover, it is important to define such communication mechanism. Agents may need to share state information, communicate their intentions, negotiate patrolling regions with other agents or exchange other information that might be important, considering the strategy proposed to better achieve their goals or the team's goals. With this aim, they must respect a communication protocol, which can be accomplished, for example, through explicit peer-to-peer messages between them or using a blackboard scheme. Nevertheless, some strategies proposed in the literature do not use inter-robot communication at all. For example, a centralized coordinator unit may compute a set of patrolling routes *a priori* and simply assign each local trajectory to each robot so as to cover the whole environment. Regarding coordination, these strategies are called centralized. Those that do not rely on a central unit are called decentralized or distributed.

In distributed strategies, agents may have a reactive behavior, simply interacting with the immediate surroundings or may have enhanced capabilities, like autonomous decision-making. In such architectures, agents should continuously decide where to move next after clearing each location. As a consequence, distributed strategies generally benefit from greater **robustness** to agents' failures, due to distribution of intelligence among the components of the system, as shown in this thesis later on.

In addition, on any given team of robots, agents capabilities is an important issue. For example, in a centralized strategy, it may be appropriate to have a heterogeneous team of robots, where the central coordinator would assign different specific tasks according to each agent's distinct capabilities. Using a homogeneous or heterogeneous architecture is a decision that relies on the actual cost of the multi-robot system, the application domain and the intended performance of the team. It is yet left to be proven which organization is advantageous in this context.

In this work, it is foremost studied **distributed** patrolling architectures with robots endowed with local perception capabilities, in environments with fixed topology (though not necessarily static), their design, effectiveness, potential to scale to larger team sizes, as well as their application in **realistic** scenarios. Secondary questions, like dealing with intruders or monitoring topological changes in the environment were not addressed in this thesis and are left as future work.

According to [Sempé and Drougoul, 2003], "a good collective architecture should adapt itself to the robots group size and environment size and topology because some robots may break down and some areas may be temporarily restricted". The authors refer to the **scalability** of a multi-robot system, which is one of the crucial points addressed in this thesis.

In the context of multi-robot patrolling, the definition of scalability is as follows:

Definition 2 (Scalabilty)**.** *Scalability consists on how well a given collective strategy performs as the dimension of the team grows and how the individual productivity of each robot is influenced by having additional agents in the team.*

In this work, scalability is always related with the dimension of a team of cooperative patrolling agents and should not be mistaken for scalability of the environment. The latter refers to how easily an approach scales with the increase of the dimensions of the environment.

Furthermore, in terms of **performance** of the group as a whole, both efficient and effective cooperation are intended to be achieved. While "efficient" refers to the resources used in the patrolling strategy, both computational as well as physical, "effective" is related to the actual results obtained using any strategy, considering a predetermined optimization criterion. An approach which results in great performance but is computationally expensive is deemed as an effective approach but not an efficient one.

All these concepts and issues are essential to lay the groundwork to approach the patrolling problem and are consistently addressed along this work.

1.4 Overview of the Thesis

Having answered to the questions "What problem is addressed in this thesis?" and "Why are we studying this problem?", it is fundamental to answer an additional question, which is: "How is the problem going to be solved?". This question requires a more in-depth answer, that is detailed throughout the rest of thesis.

Initially, an analysis of relevant literature concerning related work to the MRPP is conducted in chapter 2. This allows to formulate the problem and extract some weaknesses inherent to previous works, so as to present the goals and contributions proposed in this thesis.

In Chapter 3, several state-of-the-art patrolling strategies are compared in order to draw important conclusions on the performance and scalability using different approaches on distinct environments and variable team sizes.

Later on, in Chapter 4, a preliminary framework to solve the MRPP is presented by making use of distributed coordination and yielding a scalable and robust team behavior.

Afterwards, a more complex solution is presented in chapter 5, where each robot consists of a learning agent that adapts its behavior in order to maximize team effectiveness. It is demonstrated that exceeding performance can be obtained, when compared to several other strategies.

In Chapter 6, a method for estimating the optimal size of a team of mobile robots in a patrolling mission is proposed. Theoretical results on the verge of optimality are discussed and used as an upper bound for team performance in practical experimentation with any number of robots in the patrol team.

Finally, the last chapter sums up the work and provides final conclusions and future directions of research.

Chapter 2

Background and Fundamentals

In this chapter a survey of multi-agent patrolling[2] strategies is presented. As stated before, this is a recently growing field, which picked the interest of the robotics community, during the last decade. This interest stems from the variety of possible approaches, the potential applications of such algorithms in several distinct areas and the important social function of these systems. In this survey, it should be noticed that the words “agent” and “robot” are usually interchangeable. However, “agent” accounts for any intelligent individual, having a general scope referring also to software or even human patrolling units, beyond robotic ones. The works herein referred present many differences in terms of strategy, communication paradigm, cooperation scheme, performance evaluation and other features, as discussed in this chapter.

After presenting related work on the MRPP, the main gaps in the literature in this area are identified and following this analysis, the contributions proposed in this work are presented. This chapter ends with the formulation of the problem to be solved in this thesis in order to lay the foundation for the remaining chapters of this thesis.

[2] Also known as *repetitive sweeping*, *multi-robot monitoring* or *graph coverage* in the literature.

2.1 Related Work

Despite the high potential utility of Multi-Robot Patrol, it has only been rigorously addressed in the last decade. In the first chapter of this thesis, a taxonomy of the research areas related to the patrolling problem has been proposed (*cf.* Figure 1.2, page 7). In this section, a detailed look into existing works on *Area Patrol* and variants of the surveillance problems with multiple robots is given.

Important theoretical contributions on the MRPP have been presented by [Chevaleyre, 2004, Smith et al., 2010, Pasqualetti et al., 2012a]. Considering the *idleness* criteria (*cf.* subsection 2.4.2), also known as *minimum refresh time*, and assuming global and centralized information, Pasqualetti et al. showed that the problem is NP-Hard, *i.e.*, no polynomial time algorithm is known to compute an optimal solution to the problem. This was demonstrated by reduction from the Traveling Salesman Problem (TSP). In addition, Chevaleyre proved that it can be optimally solved with a single robot by finding a TSP tour in the graph that describes the environment to patrol.

As for the multi-robot case, solving the problem with a TSP approach by spacing each of the robots evenly along the path [Chevaleyre, 2004, Smith et al., 2010] has proven to be an effective approach, in general. However, it was shown that partitioning-based strategies may perform better than a TSP cyclic strategy "for graphs containing long edges", *i.e.*, when there are long corridors or edges separating clusters of vertices [Chevaleyre, 2004]. Furthermore, in [Pasqualetti et al., 2012a], approximation algorithms on specific graph instances are addressed in order to obtain known bounds related to the optimal solution.

Beyond these important theoretical contributions, several strategies have been presented to tackle the problem in a variety of different ways. A survey of such strategies, grouped according to their features, is presented below.

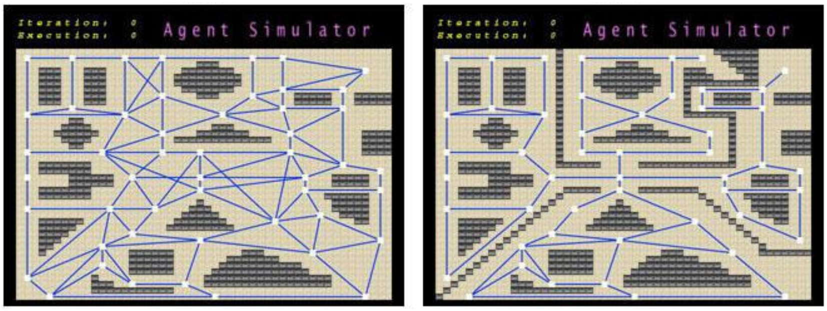


Figure 2.1: Maps used in the simulation trials of [Machado et al., 2003].

2.1.1 Pioneering Approaches (2002-2004)

Pioneer strategies include simple architectures with agents endowed 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.

One of the pioneer works in this field is described in [Machado et al., 2003] and in more detail in [Machado, 2002], where a discussion of multi-agent patrolling issues is presented, as well as several architectures and evaluation criteria. Moreover, a simplistic patrolling simulator was developed to compare different architectures proposed by the authors.

Different agent behaviors are employed in the approaches described therein, namely in the agent's perception, which can be *reactive* (with local information) or *cognitive* (with access to global information). Also, these architectures differ in the *communication* mechanism and in *decision-making* of the next vertex to be visited in the topological map.

To analyze the performance of each technique, criteria based on the average and maximum idleness of the vertices were proposed. Random decision algorithms scored the worst results and simple techniques conducted by the vertices' idleness scored close results to the same technique using a centralized coordinator. In general, the best strategy was a local strategy with no communication, based on individual idleness and without a centralized coordinator, called "Conscientious Reactive". Other good results were obtained by "Conscientious Cognitive", which is a similar method. However, agents are no longer reactive, choosing the next vertex to visit on the global graph (instead of their neighborhood). Additionally, raising the agents population up to 25 agents increased the performance of the team in every case, as expected.

There are a few weaknesses in this work. Conclusions were drawn based only on two scenarios (map A and B on Figure 2.1). Also, unweighted edges were used, meaning that agents travel from one vertex to another in a single iteration, independently of the distance between them, which is a rather hard assumption. Moreover, the solutions presented are more directed to virtual agents in simulation environments as no real robots were used during the experiments.

In [Almeida et al., 2004], the architectures proposed by Machado were enhanced with advanced path finding decision-making towards a goal, based on both the idle time of vertices of the graph and the distance to them. Also, the tests were run on more and *distinct environment topologies*. Benefits and disadvantages of each approach tested were specified and it was shown that the best strategy depends on the topology of the environment and the agents' population size.

Generally, a cyclic approach based on the TSP, as described by [Chevaleyre, 2004], has the best performance for most cases. This can be explained by its disciplined coordination scheme, which is very effective. However, this architecture will have problems in dynamic environments, graphs with thousands of nodes (due to the complexity of computing a cycle in these cases), graphs containing long edges, and patrolling regions with different assigned priorities, due to its predefined and fixed nature. On the other hand, agents moving randomly achieved very bad results. Additionally, agents with no communication ability, whose strategies consisted of moving towards the vertex with the highest idleness, performed nearly as well as the most complex pioneer algorithm implemented. In general, heuristic agents and reinforcement learning techniques considered have the second best performance, followed by the considered negotiation mechanisms techniques. Nonetheless, Almeida's work contains several simplifications, like using *unrealistic simulations* that do not consider the dynamic of the robots as well using iterative simulation cycles instead of the actual time to measure performance.

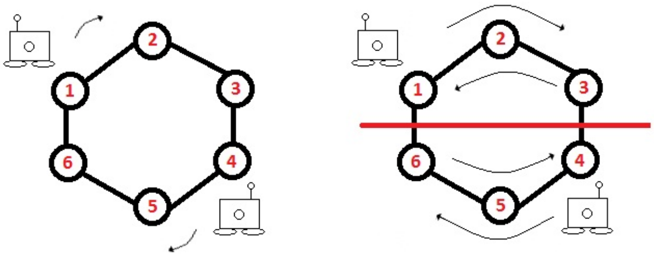


Figure 2.2: On the left, an example of a cyclic strategy. On the right, an example of a partition-based strategy.

Beyond the theoretical contribution in [Chevaleyre, 2004], diverse patrolling strategy classes are also described and compared, focusing mostly on two graph-theory centralized planning strategies: cyclic and partitioning strategies. A simple illustration of these strategies is presented in Figure 2.2.

A good strategy is considered to be the one that minimizes the time lag between two passages to the same place and for all places. The author makes reference to the fact that very simple strategies, with nearly no communication ability, can achieve impressive results. The paper aims to answer some key questions, such as whether the existing algorithms generate optimal strategies, whether there are effective near-optimal approximations, and how good partitioning and cyclic algorithms are, having agents following the same fixed path with uniform distribution.

The main conclusion presented is that cyclic and partitioning strategies have generally good performance when compared to Almeida's architectures [Almeida et al., 2004]. The first one is better suited for graphs that are *highly connected* or have large closed paths and the second one is better for graphs having long corridors separating regions.

The first known patrolling approach, which was focused and implemented on robotic agents as opposed to software agents, was presented in [Sempé and Drougoul, 2003] and [Sempé, 2004]. This work described a reactive and *adaptive* approach in which robots are distributed in regions of the environment to solve the area patrolling problem, through task data propagation.

Patrolling is seen in a task allocation perspective, where each robot is assigned a different region to visit. Robots have localization and local navigation abilities and can estimate their remaining autonomy. They 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. In the experimental setup, battery recharges are taken into account, unlike the software agents' case, and *physical interference* can occur.

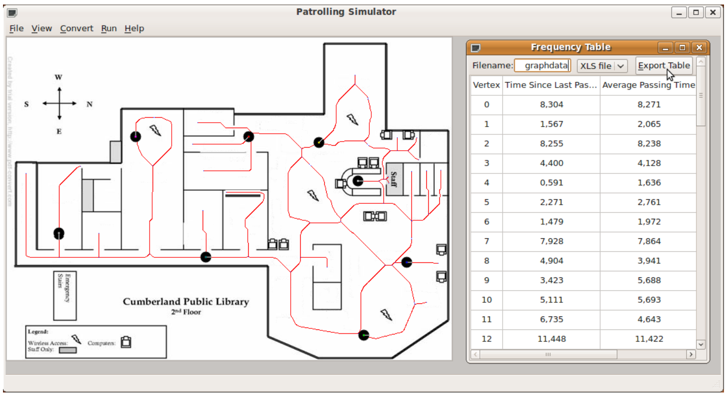
The authors claim that efficient patrol is achieved, considering an evaluation criterion based on idleness of the vertices of the topological environment, and interesting properties of adaptability concerning group size and the environment are shown.

2.1.2 Graph Theory Approaches

Graph theory strategies look for solutions of classical problems like finding Hamilton cycles, graph clustering, spanning trees and others to assign efficient routes for the robot's patrolling missions. These strategies typically rely on a centralized coordinator to calculate minimal-cost routes that visit all points in the target area, therefore agents follow the same patrol routes over and over again.

The problem of generating patrolling paths for a team of mobile robots within a certain environment and following a given frequency optimization criterion is considered in [Elmaliach et al., 2007]. The area patrol algorithm developed guarantees that each point in the target area, represented by an occupancy grid, is covered at the same frequency. This is possible by computing minimal-costs cyclic patrol paths that visit all points in the target area, *i.e.*, Hamilton cycles[3]. Agents are *uniformly distributed* along this path and they follow the same patrol route over and over again. Movement direction and velocity constraints may change in different parts of the environment. One of the key aspects of this strategy is the fact that it is independent of the number of robots. Uniform frequency of the patrolling task is achieved as long as there is, at least, one robot working prop-

[3] Hamilton cycles consist of closed paths that contain every vertex of a graph, according to [Bondy and Monty, 1976].



Filename: graphdata

Vertex	Time Since Last Pas...	Average Passing Time
0	8,304	8,271
1	1,567	2,065
2	8,255	8,238
3	4,400	4,128
4	0,591	1,636
5	2,271	2,761
6	1,479	1,972
7	7,928	7,864
8	4,904	3,941
9	3,423	5,688
10	5,111	5,693
11	6,735	4,643
12	11,448	11,422

Figure 2.3: MSP running on the Patrolling Simulator window [Portugal, 2009].

erly. Logically, the visit frequency grows linearly with the number of robots. A possible disadvantage of this approach is its deterministic nature. 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.

The first step taken by our investigation group to deal with the MRPP was presented in the author's Master dissertation [Portugal, 2009], which focused on patrolling algorithms and related issues like graph extraction [Portugal and Rocha, 2012a]. An original, scalable, centralized and efficient algorithm was presented, called Multilevel Subgraph Patrolling (MSP) Algorithm [Portugal and Rocha, 2010].

The MSP Algorithm assumes that robots are endowed with the environment map and the ability for self-localization and navigation. The algorithm generates a topological representation of the environment and partitions the environment in generic K subgraphs, as described in [Portugal and Rocha, 2011a]. Each balanced subgraph is then assigned to a mobile agent. The algorithm deals, then, with effectively patrolling each region by computing paths for every robot in the assigned subgraph. To accomplish this, it searches each subgraph, using a classical algorithm for Euler tour[4] and various heuristics for Hamiltonian cycles, non-Hamiltonian Cycles and Longest paths.

The algorithm was compared to a cyclic algorithm. In order to carry out this

[4] An Euler tour is a closed walk that traverses each edge of a graph exactly once [Euler, 1736].

comparison, the patrolling simulator shown in Figure 2.3 was developed, which incorporates both approaches. Six different maps were used. The MSP Algorithm scored slightly better results in three cases and obtained slightly worse results in the other three cases. Given that cyclic algorithms are well-known for their performance, in terms of visiting frequency these results were very optimistic and confirmed the flexible, *scalable* and high performance nature of the approach, which also benefits from being non-redundant, does not need communication between agents, and collision avoidance between them is only needed in the frontiers of their regions. Following such work, the authors also proceeded to seek effective methods for finding long paths in graphs [Portugal et al., 2010].

A successful implementation of the previous algorithm is reported in [Cabrita et al., 2010], where a team of Roomba robots cooperatively patrol an indoor facility. The work addresses environmental monitoring, therefore robots collected samples of alcohol concentration and temperature during the mission. Similarly, in [Iocchi et al., 2011], both cyclic and partitioning strategies are analyzed and compared and the authors also show that a mixed strategy may perform better in particular graph instances. The focus of the work is put on *coordinated robot behavior*, and experiments with realistic simulations and through Erratic platforms in an indoor environment were conducted. Additionally, Pasqualetti et al. [Pasqualetti et al., 2012b] focused on constructing tours using graph-theoretic techniques, instructing the robots to travel according to an Equal-Time-Spacing trajectory, while Standers et al. [Stranders et al., 2013] have focused on partitioning strategies for continuous patrolling with teams of mobile information gathering agents.

Finding a minimum spanning tree (MST) on the graph is also a popular approach for MRPP, not only for partitioning the environment on disjoint sets that are patrolled by each robot [Fazli, 2013], but also to find cycles that circumnavigate the MST and visit every place in the environment [Gabriely and Rimon, 2001].

Graph-theory strategies are robust, being independent of the number of robots and are recognized for their results in terms of visit frequency and idleness. 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.

2.1.3 Swarm-based and Reactive Approaches

Recently, swarm intelligence has also been used to tackle the multi-robot patrolling problem. In such works, agents only have *local perception*, deciding their next move according to the artificial state of the surrounding environment, which depends on information previously "dropped" by members of the team, similarly to pheromones in ant colonies.

In [Chu et al., 2007], the environment is unknown and markers are used to implement indirect communication among agents. Being unknown, an empty grid is used initially and it is updated continuously, where each cell may be free, occupied or unreachable.

A new algorithm is proposed, which relies on the evaporation process of pheromones dropped by agents, *i.e.*, an indicator of time passed since the last visit. The agents' behavior is naturally defined by moving towards cells containing less pheromone quantity. Agents have limited local perception, and follow paths according to the pheromone quantity in their neighboring cells.

Simulation experiments using different map configurations and team sizes revealed that an approach with *global perception* is more effective in more complex infrastructures, in terms of the average idleness and the worst idleness. However, it proves to be twice as costly in terms of *computational complexity*. Due to the marking of the environment, the system self-organizes and an effective patrolling behavior emerges. As expected, the average idleness decreases with the number of agents, guaranteeing scalability.

Similarly, [Elor and Bruckstein, 2010] introduces an approach using ant-like swarm agents, where agents only have local perception and knowledge, cannot communicate directly, and use the environment to drop markers for later sensing. Although being apparently distributed, a single agent, called "leader", runs a plan-

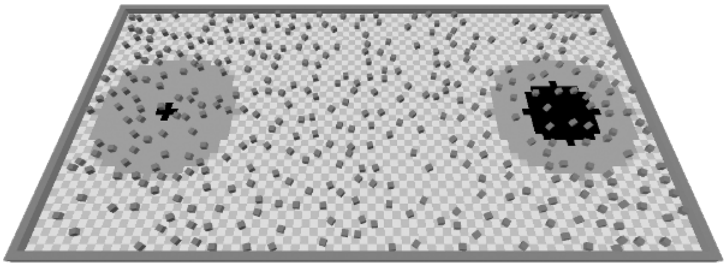


Figure 2.4: Simulation of a swarm of robots [Schmickl et al., 2007].

ning algorithm to find a path that covers the whole graph in a cyclic way, relying on Hamiltonian cycle computations. All other agents will follow this computed path, distributed in time, maintaining even gaps, using time stamps to estimate distances between agents.

The great innovation in this strategy is its robustness and *fault-tolerance* nature. Agents autonomously reorganize uniformly in case of breakdown in any agent and the leader will find a new patrolling route autonomously when there are changes in the graph. If the leader breaks down, an identifying stamp on the markers will get old, the team will realize it and one of the agents can replace the leader, for example, using an election algorithm.

Additionally, in [Yanovski et al., 2003], a simple multi-agent graph exploration algorithm is presented and analyzed. The authors focus on a very special case of graphs: Eulerian graphs[5], and show lower bounds on the number of steps that each memoryless agent would need to cover the whole graph.

2.1.4 Negotiation Approaches

Market-based coordination is becoming popular in MRS. In *patrolling auctions*, agents are assigned with a set of vertices and trade their vertices with other agents in order to maximize own performance or the team's global performance. For example, in [Menezes et al., 2006], agents reveal a scalable and reactive behavior, being able to patrol infrastructures of all sizes and diverse topology types. Besides

[5] An Eulerian graph is a graph containing an Euler tour [Bondy and Monty, 1976].

criteria based on patrolling frequency and idleness, in this work some other important measures like scalability (in terms of environment size), stability (uniformity or variation of the nodes visited) and *reactiveness* (concerning performance on environments with different topologies) are introduced.

Each agent acts as a negotiator and receives a set of random graph vertices to patrol. Agents negotiate those vertices using auctions in order to change one or even two of them with other agents. Aiming to minimize visits to the same node, these agents will naturally bid to obtain a set of vertices in the same region of the graph. The agents' are directed towards nodes with high idleness, considering their distance. Interestingly, simulation experiments in different graph topologies showed that, in terms of average graph nodes idleness, patrolling strategies using negotiation mechanisms with self-interested agents generally performed better than with cooperative agents.

When comparing this negotiation strategy with previous pioneering patrolling strategies in subsection 2.1.1, the single-cycle approach proposed in [Chevaleyre, 2004] was the only one that outperformed the negotiation strategy in terms of nodes idleness. However, the single-cycle approach is less *robust* to failures. As for stability, the proposed approach presents better results than all other approaches analyzed. Also, the negotiator agent is able to run in worlds of any size. Good indicators were also obtained regarding reactiveness, since the agents need no learning time or path pre-computation.

The results presented in the paper were obtained through simulations. A cooperative negotiation mechanism implies global knowledge about the utility function of all agents, which can only be done in a centralized manner or continuously synchronizing the communication between all robots during runs. Like many previous strategies presented in this chapter, it would be interesting to test a similar approach with real robots to confirm its performance in real scenarios.

In [Hwang et al., 2009], a cooperative auction system is also proposed to solve the problem of patrol planning. Robots have no reasoning abilities, so they always bid in every auction, independently of their interest. A centralized system is responsible for assigning points to the most suitable robot. If the winner of the auction becomes the robot with the maximal path length to patrol, a re-auction process will take place in order to choose one of the winner's points, except the new one to be auctioned again.

After performing the cooperative auction system, the proposed approach suggests a patrol path for each robot among the points they are responsible for. The authors evaluate their approach in a simulated environment using three criteria, which should be minimal: the *total energy* that robots consume, the *length of the patrolling routes* among robots and the *average waiting time* for visiting the points of the environment.

The authors claim that their approach leads to a decrease of time complexity, lower routing path cost and better workload balancing among robots. However, they do not compare it with alternative approaches. Also, they consider a somehow unrealistic simplification: the environment is represented as an open space, with no obstacles or barriers. Collision problems while patrolling are also not considered. Nevertheless, despite its weaknesses, the proposed approach to achieve cooperation among robots is innovative and has the potential to be used in some real-world situations.

Similarly, in [Poulet et al., 2012] two decentralized, cooperative, auction-based patrolling strategies are presented. Inspired from the computational social choice theory, agents trade the nodes they have to visit while reasoning on the performance of the group rather than on their own. The authors show that the performance of the approaches is at least as good as centralized pioneering strategies by comparing their approaches through simulations, with the work of [Almeida et al., 2004], in the same topologies. The simulator used, which was also used in the pioneering works in subsection 2.1.1, is described in [Moreira et al., 2007].

2.1.5 Adaptive Approaches

Within all the strategies pursued so far, the creation of adaptive behaviors that allow agents to learn how to effectively patrol a given scenario are the more promising in the context of security missions, because such adaptability fosters the unpredictability principle in a way that eventual intruders may be unable to anticipate patrolling trajectories.

Certain works in this field have adopted machine learning methods aiming to adapt agents' behavior. For instance, in [Santana et al., 2004], the patrolling task is modeled as a Reinforcement Learning problem in an attempt to allow automatic *adaptation of the agents' strategies* to the topology of the environment. The authors justify the choice of using such approach based on findings from previous studies in this domain, in which proposed strategies perform badly in particular environment topologies, due to the topology-dependent coordination between the agents' actions inherent to the patrolling task.

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. However, to make sure that agents do not interfere with each other, and to ensure a satisfactory global behavior, penalties are given when agents compete for idleness on the same node.

Two Reinforcement Learning techniques using different communication schemes were implemented and compared to non-adaptive architectures in simulations. Although not always scoring the best results, the adaptive solutions are superior in most of the experiments using different number of agents in the team. The main attractive characteristics in this work is *distribution* (no centralized communication is assumed) and the adaptive behavior of agents, which can be desirable in this domain.

Additionally, in [Marier et al., 2009, Marier et al., 2010] the patrolling problem is cast as a multi-agent Markov decision process, where reactive and planning-based techniques are compared. The authors focus on the information retrieval aspect of patrolling. The value of the information decreases with its age, forcing the agent to update its information on vertices as frequently as possible. Another important aspect in these papers is the continuous-time formulation to allow real-valued duration.

Being adaptive, the first approach is robust in case of failures and can deal with environment changes. It is based on states and actions and the decision process relies on local and current information. The second approach tries to maximize information retrieved from the nodes of the graph in the long-term. It uses a heuristic search in the state space and a branch-and-bound scheme to approximate the long-term expected value of any state. The authors concluded that both perform similarly, with the latter being slightly superior in general, since it *looks further ahead* than the former, which is purely local. However, the reactive technique runs much faster, suggesting that a simple and computationally cheaper approach can be used in many applications, instead of more complex strategies which only perform slightly better.

Furthermore, in [Ruan et al., 2005], effective patrolling in a dynamic and stochastic environment is considered. The patrol locations are modeled with different priorities and patrol units respond to call-for-service. An infinite-horizon Markov Decision Process methodology and a learning algorithm are used to obtain a deterministic patrol route to respond to incidents. In addition, an action selection method is applied to devise several preventive patrol routes, and agents randomly choose a single route from those generated by this method.

General results show that complex solutions are usually not elegant, since they require higher cost, only to slightly improve the outcome when compared to simple architectures, with nearly no communication ability [Almeida et al., 2004].

2.1.6 Variants of the Problem

Despite being focused on *adversarial patrol*, a comprehensive study presented in [Sak et al., 2008] explores important concepts such as *unpredictability* in the multi-agent patrol task. By applying unpredictable actions in the patrolling method, intruders will not have access to the patrolling trajectory information to avoid being detected by agents. Also, graph partitioning techniques to assign different sites for each agent are analyzed. The evaluation criteria considers three models of intruder behavior with distinct order of intelligence, assuming that intruders will choose a vertex and will stay some time at that location to achieve the attack.

Metrics are presented in terms of probability of catching intruders with different models of behavior. The authors also propose to evaluate different sequencing algorithms, which set the rules for the sequence of vertex visits for each agent:

- Random algorithm: the agent will choose randomly the next vertex to visit;
- Original TSP: sequence generated by the TSP cycle solution on the graph;
- TSP rank of solutions: besides computing one TSP original solution, sub-optimal solutions are also computed and kept in a queue. After each cycle, the agents will choose randomly one of the computed solutions.
- Multilevel graph partitioning [Karypis et al., 1998]: partitioning scheme which creates regions with the same number of vertices;
- K-Means Partititioning [Jain et al., 1999]: Graph clustering algorithm based on the Euclidean distance of each vertex to the prototype of each cluster;
- Agglomerative Hierarchical Clustering [Jain et al., 1999]: Clustering algorithm that uses a different metric, in which close by vertices should be joined in the same cluster.

Intensive simulation results made evident some distinct facts. Traditional partitioning schemes are more effective against random attackers. However, non-partitioning schemes perform better when the attacker has some level of intelligence. With random attackers or attackers with restricted information, the deterministic Original TSP cycle was the best solution found, because it covers all the nodes with the minimal time needed. The algorithm that performed better against highly intelligent intruders was the TSP rank, which is *non-deterministic*, thus confirming the importance of unpredictability in the patrolling task. The random-based strategy, although being very unpredictable, leads to very high idle time in some vertices, which makes it generally useless for the patrolling problem. The authors state that if the terrain which is being patrolled is static and the patrolling task will last for a long time, it is probably better to compute centralized solutions with global information and calculations in the graph, as presented in the paper, due to the potential to achieve better global solutions than with a distributed strategy.

Another variant of the patrolling problem is *Uniform Coverage*, where the goal of the work is to repeatedly visit a set of locations in the environment while maintaining uniform frequency in all locations. This is done according to a specified frequency distribution, instead of considering average visiting rate to each location. In [Baglietto et al., 2008] and then in [Cannata et al., 2011], the authors address this problem using robots with minimal computational memory and communication capabilities and also aim for unpredictability of the paths followed by robots. Thus, a minimalist algorithm is described, where agents communicate implicitly through the environment using *smart nodes*, similar to the *stigmergy*[6] concept of ant colonies.

An offline phase that is executed only once in the beginning of the mission is necessary to define the frequency distribution, and an online phase consists of having each agent executing the algorithm in parallel in a fully distributed way. Using this approach, agents can be added or removed during the mission without requiring a configuration phase. Unfortunately the work is only verified through simulations.

Furthermore, Pippin et al. [Pippin et al., 2013] deal with the problem of having agents in a cooperative patrolling team that do not perform as expected. Therefore, in order to avoid less effective patrol, a central entity monitors robot performance and dynamically reassigns tasks from members that patrol inefficiently. An auction system is triggered when assignment is necessary and nearby agents bid to exchange vertices with the poor performer. Therefore, members of the team cooperate by balancing the load according to their individual capabilities, so as to increase overall team performance.

On the upside, the work presents realistic simulations and real-world experiments with teams of robots, and the concern with maximizing performance by online monitoring of the agents is innovative and interesting. However, the as-

[6] *Stigmergy* is a self-organizing mechanism of indirect coordination between agents by modifying their local environment [Bonabeau et al., 1999].

sumption that an external monitor can fully observe the visits to each node may not be applicable in large environments. Also, the monitor centralizes information and is not allowed to fail during run time.

Table 2.1 presents an overview of the main multi-robot patrolling approaches previously referred.

2.2 Discussion and weaknesses of the literature

Beyond the previous contributions, some authors have proposed distinct strategies for multi-robot coordination in patrolling missions based on a variety of alternative concepts like task allocation [Zlot et al., 2005], neural networks [Guo et al., 2007], game theory [Pita et al., 2011], artificial forces [Sampaio et al., 2010], linear programming modeling [Keskin et al., 2012] or Gaussian theory processes [Marino et al., 2012]. However, most of these works propose patrolling methods and overlook other relevant problems that should be addressed in multi-robot patrolling missions. In this section, studies focused on *robustness to failures* and *scalability* are analyzed and the weaknesses found on the literature related to the patrolling problem are summarized.

2.2.1 Robustness to Failures

Most multi-robot systems are expected to be robust in the sense that the system should maintain its functionality, possibly with degraded performance, in case a subset of the robotic agents fails, *e.g.*, due to battery discharge. In this context, totally decentralized approaches benefit from enhanced fault-tolerance when compared to centralized approaches, which suffer from the single point of failure phenomenon.

Among the works surveyed, only few deal explicitly with failures during the patrol mission. One of them is [Fazli et al., 2010]. This work focused on fault-tolerance in a novel approach for multi-robot graph coverage in a known and static

Table 2.1: Overview of selected architectures analyzed [Portugal and Rocha, 2011b].

Proposed Strategy	**Type/Perception**	**Communication**	**Coordination**	**Decision-Making**
Conscientious Reactive [Machado et al., 2003]	Reactive/ Local	None	Emergent	Local Idleness-based
Conscientious Cognitive [Machado et al., 2003]	Cognitive/ Global	None	Emergent	Global Idleness-based
Idleness Coordinator Monitored [Machado et al., 2003]	Cognitive/ Global	Coordinator Messages	Centralized	Idleness-based with Monitoring
Heuristic Pathfinder Cognitive Coordinated [Almeida et al., 2004]	Cognitive/ Global	Coordinator Messages	Centralized	Heuristic (Idleness + Distance)
STP [Elmaliach et al., 2007]	Cognitive/ Global	Coordinated Cyclic Trajectory	Centralized	Hamilton Cycle Computation
Cyclic Approach [Chevaleyre, 2004]	Cognitive/ Global	Coordinated Cyclic Trajectory	Centralized	TSP Heuristic Calculation
Partitioning Approach [Chevaleyre, 2004]	Cognitive/ Global	Coordinated Cyclic Trajectory	Centralized	TSP Heuristic inside each region
Gray-Box Learner Agent [Santana et al., 2004]	Reactive/ Local	Flags	Emergent	Reinforcement Learning (Adaptive)
Bidder Agent [Menezes et al., 2006]	Cognitive/ Global	Bidding Messages	Auctions	Self-Interested Idleness-based
Sequential Single-Item Auctions [Hwang et al., 2009]	Cognitive/ Global	Bidding Messages	Auctions	Minimize the max patrol path
Heuristic Pathfinder Two-Shot Bidder [Almeida et al., 2004]	Cognitive/ Global	Bidding Messages	Two-Shot Auctions	Heuristic (Idleness + Distance)
Task Propagation approach [Sempé and Drougoul, 2003]	Cognitive/ Global	Task Propagation Messages	Centralized	Task strength + Idleness (Adaptive)
EVAP [Chu et al., 2007]	Reactive/ Local	Flags	Emergent	Local Idleness-based
CLInG [Chu et al., 2007]	Reactive/ Local	Flags	Emergent	Idleness-based information diffusion
TSP rank of solutions [Sak et al., 2008]	Cognitive/ Global	Coordinator Messages	Centralized	Random TSP sub-optimal solution
MSP [Portugal and Rocha, 2010]	Cognitive/ Global	Coordinated Cyclic Trajectory	Centralized	Graph-based inside each region
Left-Induced Partition [Pasqualetti et al., 2012a]	Cognitive/ Global	Coordinated Cyclic Trajectory	Centralized	Graph Partitioning with known bounds
Cyclic Approximation [Pasqualetti et al., 2012b]	Cognitive/ Global	Coordinated Cyclic Trajectory	Centralized	TSP approximation with known bounds
Mixed Strategy [Iocchi et al., 2011]	Cognitive/ Global	Coordinated Cyclic Trajectory	Centralized	Mixed Cyclic + Partitioning
PatrolGRAPH* [Baglietto et al., 2008, Cannata et al., 2011]	Reactive/ Local	Flags	Emergent	Uniform frequency guided visits
Auction reassignment approach[Pippin et al., 2013]	Cognitive/ Global	Coordinated Cyclic Trajectory	Centralized	Partitioning with performance-based reassignment

2D environment represented by a graph. A series of decompositions of the graph are employed to obtain cycles, which are then assigned to covering robots. Firstly, a technique called Constrained Delaunay Triangulation (CDT) is used to extract the graph of the environment. Then, the graph is reduced to decrease the distance traveled by each robot and improve efficiency. Afterwards, the authors make use of an extended version of the Prim's algorithm [Prim, 1957] that builds MSTs on weighted graphs, in order to partition the graph and form as many partial spanning trees as the number of covering robots. Finally, cycles are built on each resultant tree to form navigating areas for each agent.

Upon failure of a robot, all of the vertices of its assigned tree are released and robots in adjacent areas will expand their trees, using the same extended version of the Prim's algorithm, to repossess the released vertices and cover the environment again. However, no hints on how failures are detected by other robots in the team are given.

As referred before, a procedure is described in [Elor and Bruckstein, 2010] for the reorganization of agents in simulations, when a member of the team fails. However, the issue of robustness to leader breakdowns was left as future work and the authors suggest the use of an election algorithm. Also, in [Pippin et al., 2013] performance of agents is monitored during the patrol mission to handle the situation when robots are not performing as expected. However, no hints are given on what happens if some of the agents fail. In fact, several authors refer how the failure problem could eventually be handled in their own framework, but there is a manifest lack of solutions implemented, especially in the real-world.

2.2.2 Scalability

Scalability of the robot team is also a very important characteristic for patrolling strategies. A good strategy should work independently of the number of agents in the team and with any environment topology. In the context of the MRPP, most works do not focus on this issue and the ones that test strategies with variable number of agents do not address the scalability issue explicitly. There have been, however, a few scalability studies presented for multi-robot systems in general.

According to economists and conclusions from previous works on mobile robotics [Sweeney et al., 2003], groups display increasing *marginal performance* only until a certain group size. Beyond this point, the groups' productivity drops with the addition of robots. In theory, productivity should grow during size scale-up. However, spatial limitations increase the *interference* between agents causing the decrease of performance. Effective coordination algorithms maintain increasing productivity. To improve the group's performance, methods can be developed to minimize interference between robots as in [Rosenfeld et al., 2006], where the impact of scalability of homogeneous robots in the productivity of a multi-robot system applied in groups with different coordination algorithms is studied.

In order to measure the interference level with different coordination algorithms and promote comparisons, an interference variable was defined by the total time robots deal with resolving team conflicts, *e.g.*, like re-planning due to collision avoidance. This study was first applied in the foraging domain, where robots have to locate target items from a search region and deliver them to a goal region. This domain was chosen due to the wide existing research of coordination algorithms within this environment and the likeliness of arising spatial conflicts between members of the same team in a limited field of operation. The study was limited to groups of homogeneous robots without the ability to communicate, and without previous knowledge of the environment to minimize the factors involved in the experiment.

In order to check if the same results were obtained in different domains with restricted spatial resources, the authors also conducted similar experiences in a limited search domain where the main objective is to find the exit out of the room as quickly as possible.

By evaluating exhaustive simulation trials with eight different coordination algorithms in both domains, important conclusions were drawn:

- The coordination method strongly impacts the scalability of the group;
- A high negative correlation between group interference and productivity was found.

Besides the critical point in scalability where productivity starts to decrease, there is a second point where the addition of more robots ceases to negatively

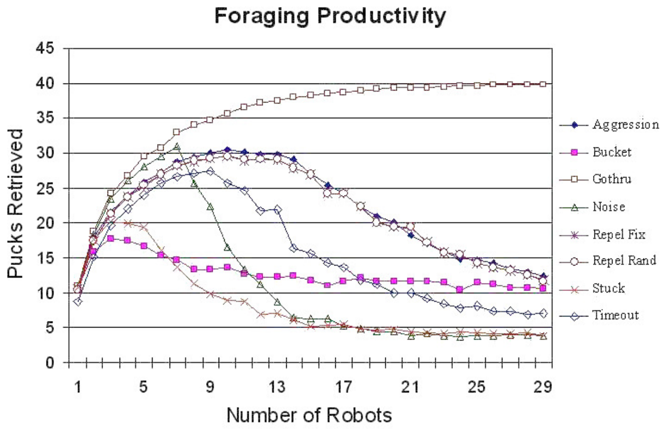


Figure 2.5: Results obtained in [Rosenfeld et al., 2006] using different coordination schemes in the foraging domain.

affect the groups' performance. This is illustrated in Figure 2.5.

Authors conclude that the interference measure should be considered in future works to produce effective coordination methods with the ability to scale to larger team sizes and aid in the study of scalability qualities of robots.

Furthermore, in [Balch and Arkin, 1994] the *speedup measure* is proposed. This metric reveals how much more efficient several robots are than just one in completing a task. If $\Psi(R)$ is the performance function for R robots[7], the *speedup measure* is given by:

$$\upsilon(R) = \frac{\Psi(R)}{\Psi(1) \cdot R}. \tag{2.1}$$

If the speedup measure is equal to 1.0, R robots complete the task with a performance that is exactly R times better than one robot. This is called *linear improvement*, *i.e.*, performance proportional to the number of robots. Speedup values inferior to 1.0 reflect *sub-linear* performance and values greater than 1.0 reflect *super-linear* performance.

The works of [Balch and Arkin, 1994] and [Rosenfeld et al., 2006] served as

[7] Assuming that a higher value of Ψ means better performance.

a great inspiration in this thesis to address the scalability issue in the context of multi-robot patrol, as shown later on.

2.2.3 Weaknesses of preceding work

Studies on the last decade revealed some crucial facts about the patrolling task. First of all, it is clearly an assignment where the utilization of teams of robots can be advantageous. Secondly, strategies to approach this problem using multiple robotic agents may vary in terms of the *agent's type*, *communication* model, the *coordination* scheme and *decision-making*, as evidenced in Table 2.1, on page 30. Thirdly, despite the diversity of strategies, comparative studies between approaches invariably reveal the absence of a superior approach for every case with respect to different criteria, given that distinct approaches perform differently, depending especially on the topology of the environment.

Beyond these conclusions, the literature provides other important tools for studying the multi-robot patrolling problem, such as performance *evaluation criteria*, *environment abstraction*, *classification* of strategies and hints for the application of the correct strategy, taking into account the *connectivity* and layout of the patrol area. However, as seen before and underlined during the analysis made on previous approaches, there are still several drawbacks common to most architectures. It is the author's belief that some of these aspects need to be carefully analyzed and potentially overcome. A list of weaknesses identified, which have been addressed in this thesis, is provided below:

- No comparison between the actual time spent by different strategies to finish their patrolling cycles. These are typically compared through simulation runs, where each run may last differently for each case;
- Lack of a standard testbed for researchers all over the world to implement different strategies and compare precisely with previous ones already implemented.
- Lack of experimental work using a team of robots in real scenarios, especially

distributed ones;

- Absence of studies especially focused on scalability, complexity, flexibility, resource utilization, interference, redundancy, communication load or workload among robots when performing the patrolling task;
- Several simplifications assumed;
- Lack of variability or classification of the environments tested;
- Deficiency of works guaranteeing fault-tolerance;
- Impracticability of several simulation approaches towards experiments with real robots;
- Deterministic nature of several centralized approaches.

In the next section, starting from the weaknesses pointed out in past works, the main objectives and proposed contributions are presented.

2.3 Objectives and Contributions

As seen before, there are many gaps in previous works that need to be bridged.

Numerous strategies previously referred were compared through simulations, and their performance was analyzed based solely on algorithm runs instead of the actual time different strategies took to complete their patrolling cycles. This is seen as a limitation, because all strategies have different computational complexity and the run time for each simulation trial is variable. Concerning simulations in this thesis, it is intended to analyze the actual time of simulation trials, using a time-based performance metric, and it is very important to minimize the simplifications assumed, which is an identified drawback of preceding works as well. Any simulator uses simplifying approximations to some extent. When too many simplifications are assumed, simulations become *unrealistic*, jeopardizing the validity of the outcome. Also, migration to real mobile robots is a much more onerous step, and real-world results will not be in agreement with simulation results, because many conditions change, *e.g.*, noise in sensor data, errors in localization estimation, etc. In the upcoming studies, unrealistic simplifications like unweighted edges on the graph, absence of obstacles or only considering certain environment topologies will not be present.

In fact, the lack of diversity of environment topologies tested in previous works is also a major weakness identified. The literature shows us that a strategy that always performs better independently of the topology of the infrastructure to patrol does not exist. The fact is: different approaches perform differently with the environment topology. In this sense, it is absolutely crucial to evaluate the performance of new approaches using *different topologies*, in order to conclude in which cases one given approach is expected to patrol more effectively a given environment and which cases it is not. The ultimate goal should be to devise a strategy that can adapt and perform effectively in any topology.

In addition, many of the previous strategies proposed, especially the ones which rely on a centralized coordinator and are based on graph theory concepts, are *deterministic*. Most of these approaches present good overall performance in terms of minimizing the idleness of every vertex of the graph. However, they lead to a predictable team behavior. On one hand, this can be ideal in cooperative tasks like cleaning. Yet, in surveillance missions, determinism is not a desirable feature, since it becomes easier for an intruder to calculate and predict good areas of intrusion in the environment, because all the robotic agents will follow the same patrolling cycle over and over again. This patrolling cycle may be apprehended by an intelligent attacker and compromise the objective of the patrolling task. It is essential to explore then, strategies which enclose any kind of *unpredictability* to avoid similar situations. Nevertheless, conclusions drawn from previous studies showed that random approaches are almost useless in this context, thus a tradeoff between determinism and unpredictability should be made.

Other aspect which is also related to desired properties of the patrolling strategy is *feasibility*. For example, strategies which use ant-like swarm agents rely on markers (*e.g.*, RFID tags) on the environment to achieve indirect communication between agents. Such strategies are usually limited to simulations, because they are practically unfeasible in experiments with real robots, mostly because of the amount of equipment required to perform experiments in a real scenario and the assumption that the environment can be changed before the mission starts. It is the author's opinion that feasibility is one of the key properties for new solutions to the multi-robot problem in the real-world, representing an important contribution to this field.

Regarding previous studies on this field, it is clear that numerous questions are still left for analysis. Many authors focused their attention purely on patrolling strategies and their effective performance. A few of them addressed different agent behaviors like self-interested or cooperative. Others went further ahead, also addressing unpredictability in the patrolling task, robustness, environment scalability, stability and even reactiveness. However, in this context, there are still open issues dealing with *scalability* in terms of team size, *complexity*, *flexibility*, *resource utilization*, *interference*, *redundancy*, *communication* or *workload* among robots using different strategies. Further understanding of such issues is fundamental to enable comparisons between strategies, in order to assist upcoming implementations and advance the knowledge towards having multi-robot systems patrolling real-world environments somewhere soon in the future.

In this work's particular case, the focus is on *team scalability* as defined in section 1.3. This is seen as vital question, since any patrolling strategy is expected to perform better with the increase of the number of patrolling agents, but the increase of effectiveness of the patrolling strategy does not necessarily means an increase of efficiency. Therefore, one should evaluate the individual contribution of each robot with different team sizes and verify how, and especially when, the interference between agents leads to the degradation of the individual productivity of each agent. Logically, this is expected to vary according to the patrolling architecture employed and it represents a crucial inherent property of each strategy that will have impact on the ability to scale to larger team sizes.

Perhaps the most noticeable limitation in previous works, and probably one of the main challenges in this context, is the surprising lack of *experimental work* using *physical teams of robots* patrolling real scenarios. Despite the variety of approaches presented, only a few were verified using teams of mobile robots. Being mainly a practical problem, it is essential to validate convincing real-world solutions for the MRPP. The majority of methods implemented in physical multi-robot systems have been especially focused on centralized policies to coordinate the team of agents, such as [Cabrita et al., 2010, Iocchi et al., 2011, Pasqualetti et al., 2012b, Pippin et al., 2013]. The work herein presented will address new distributed methods for multi-robot patrol in the real-world, taking advantage of the precedent research about the problem. It is important to notice that the number of distributed approaches presented in the literature is low. Also, those that exist proved to have competitive results in previous works. Moreover, although presenting good results, most of the conclusions were drawn through simulations and there are very few experiments with robots in real scenarios. Finally, by presenting an innovative, cooperative and distributed method for patrolling infrastructures, dealing with unpredictability in the multi-robot patrolling task will be considered, as well as proving its feasibility in the real-world.

The main objectives of this work are related to the successful completion of a number of novel contributions to the multi-robot patrol field of research. This list of contributions is the result of an extensive analysis of the weaknesses of previous works in an attempt to fill the identified gaps. Briefly, the key scientific contributions of this thesis include:

- Benchmarking of several approaches in the literature to deeply analyze the performance of existing multi-robot patrolling algorithms using realistic simulations and conducting statistical analysis [Portugal and Rocha, 2011c, Portugal and Rocha, 2012b, Portugal and Rocha, 2013d];

- Evaluation of the ability of the team of robots to increase its size with additional robots, using diverse patrolling strategies and different topologies in order to analyze team scalability [Portugal and Rocha, 2011c, Portugal and Rocha, 2012b, Portugal and Rocha, 2013d];

- Description of preliminary distributed, scalable and effective patrolling strategies, making use of a Bayesian-based mathematical formalism to coordinate the team of robots [Portugal and Rocha, 2012c, Couceiro et al., 2013, Portugal and Rocha, 2013e];

- Implementation of a framework for multi-robot patrol in real-world scenarios that demonstrates the method's feasibility, without any centralized coordinator unit and using simple communication methods between agents to cooperatively solve the problem [Portugal and Rocha, 2012c, Portugal and Rocha, 2013e].
- Demonstration of the robustness against robot failures and communication errors in the experimental testbed considered [Portugal and Rocha, 2013e];
- Proposal of an innovative probabilistic strategy for multi-robot patrolling using a team of autonomous mobile robots, extending the antecedent preliminary approach with the ability to deal with uncertainty and adapt to the system's state by means of concurrent Bayesian learning [Portugal et al., 2013, Portugal and Rocha, 2013b, Portugal and Rocha, 2014a];
- Demonstration of the adaptability of the system to constraints such as heterogeneous robots with different speed profile [Portugal and Rocha, 2014a].
- Evaluation of the performance and scalability of the techniques proposed and comparison to previous work in the literature [Portugal and Rocha, 2012c, Portugal and Rocha, 2013e, Portugal and Rocha, 2013b, Portugal and Rocha, 2014a].
- Devisal of an analytical method for estimation of the team size for a patrolling mission according to the environment topology and temporal constraints [Portugal et al., 2014b].

2.4 Preliminaries

In this thesis, the problem of efficiently patrolling a given environment with an arbitrary number of robots is studied. In this section, details are given on how

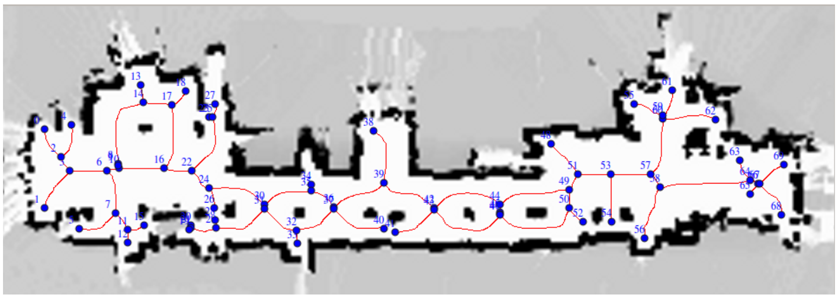


Figure 2.6: Illustration of a topological map obtained using the technique in [Portugal and Rocha, 2013c].

agents obtain the representation of the environment and about the performance metric used throughout the work.

2.4.1 Obtaining a Topological Representation

In coverage problems (*cf.* [Ge and Fua, 2005]), the environment is usually modeled as a grid-like map requiring the team of robots to sweep all cells of the environment. Instead, 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, *i.e.*, vertices of the graph, without necessarily visiting every location in the environment.

In order to obtain the topological navigation graph from an existing metric map, the procedure described in [Portugal and Rocha, 2013c] is followed. In brief, a skeleton representation of the initial grid is obtained by computing the Extended Voronoi Graph (EVG) [Beeson et al., 2005]. The EVG is a pixel-based thinning algorithm which is a fast approximation of the Voronoi Diagram [Voronoi, 1908]. Afterwards, a filtering technique is applied to remove clusters in the skeleton that arise due to aliasing. This technique does not affect the integrity and connectivity of the final graph. Then, with the existing filtered skeleton, vertices and edges are identified by image processing in the resulting undirected graph. The final step is to extract relevant information concerning the graph connectivity, such as vertice coordinates, vertice degree[8], IDs of neighbor vertices and edge costs. Results have attested the applicability of the method proposed for graph extraction. An illustration of an acquired topology is shown in Figure 2.6.

The topological map extracted is used to represent the area to patrol by a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with vertices $v_i \in \mathcal{V}$ and edges $e_{i,j} \in \mathcal{E}$, enabling robots to assess the topology of its surroundings. In this representation, vertices correspond to important locations that must be visited regularly and edges represent the connectivity between those locations. The cost of an edge $|e_{i,j}|$ is defined by the metric distance between vertex v_i and v_j. $|\mathcal{V}|$ and $|\mathcal{E}|$ represent the cardinality of the set $\mathcal{V}$ and $\mathcal{E}$, respectively. Seeing as undirected graphs are assumed, then: $|\mathcal{E}| \leq \frac{|\mathcal{V}| \cdot (|\mathcal{V}|-1)}{2}$. A path x is composed of an array of vertices in $\mathcal{V}$.

Since the topological maps considered in this context represent real-world 2D environments, it is assumed that $\mathcal{G}$ has the following properties:

- *Undirected*; where $|e_{i,j}| = |e_{j,i}|$, and the edge weights satisfy the triangle inequality[9];
- *Connected*; where $\forall v_h, v_i \in \mathcal{V}, \exists x = \{v_h, ..., v_i\}$;
- *Simple*; where two neighbor vertices v_i and v_j are connected by a unique edge $e_{i,j}$ and no graph loops exist;
- *Planar*; where a pair of edges $e_{g,h}, e_{i,j} \in \mathcal{E}$ never crosses each other.

As a consequence of these properties, $\mathcal{G}$ is usually *non-complete*[10]; *i.e.*, for every pair v_h, v_i of $\mathcal{V}$ there may not exist an edge $|e_{h,i}|$ connecting each pair of vertices.

[8] The *degree* (or *valency*) of a vertex of a graph is the number of edges incident to the vertex [Bondy and Monty, 1976].

[9] The *triangle inequality* states that for any triangle, the sum of the lengths of any two sides must be greater than the length of the remaining side, *i.e*, for any $v_h, v_i, v_j \in \mathcal{V}, e_{h,j} \leq e_{h,i} + e_{i,j}$ [Blumenthal, 1953].

[10] In a complete graph, every pair of distinct vertices is connected by a unique edge [Gries and Schneider, 1993].

In this work, it noteworthy that any generic planar graph may be addressed, as opposed to specific instances such as chain graphs, cyclic or acyclic graphs, tree graphs, Hamiltonian graphs, *etc.*

2.4.2 Formulation of the Multi-Robot Patrolling Problem

Having previously described how agents extract the topological map, in this subsection the Multi-Robot Patrolling Problem (MRPP) is formulated, assuming the existence of a navigation graph of the environment.

As referred in [Chevaleyre, 2004], 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 find R coordinated trajectories $\Pi = \{\pi_1, ..., \pi_R\}$ for each robot in order to visit frequently all $v_i \in \mathcal{G}$, ensuring the absence of atypical situations with regard to a predefined optimization criterion.

In order to address and compare the performance of different patrolling algorithms, it is important to establish an evaluation metric. Diverse criteria have been previously proposed to 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 [Iocchi et al., 2011]. In this work, the first one has been considered, given that it measures the elapsed time since the last visit from any agent in the team to a specific location. The idleness metric uses time units, which is particularly intuitive, *e.g.*, in the analysis of possible attacks to the system or enabling comparisons between different approaches.

Definition 3 (Idleness)**.** *The idleness of a vertex corresponds to the period of time that passes without the vertex being visited by any robot.*

In the upcoming equations, important variables used in the remaining chapters are defined. Firstly, the instantaneous idleness of a vertex $v_i \in \mathcal{V}$ in time step t is defined as:

$$\mathcal{I}_{v_i}(t) = t - t_l, \tag{2.2}$$

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

$$\overline{\mathcal{I}_{v_i}}(t) = \frac{\overline{\mathcal{I}_{v_i}}(t_l) \cdot C_i + \mathcal{I}_{v_i}(t)}{C_i + 1}, \tag{2.3}$$

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:

$$\overline{\mathcal{I}_{\mathcal{V}}} = \left\{ \overline{\mathcal{I}_{v_1}}, ..., \overline{\mathcal{I}_{v_i}}, ..., \overline{\mathcal{I}_{v_{|\mathcal{V}|}}} \right\}, \tag{2.4}$$

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

$$\max(\overline{\mathcal{I}_{\mathcal{V}}})(t) = \max\left(\overline{\mathcal{I}_{v_1}}(t), ..., \overline{\mathcal{I}_{v_i}}(t), ..., \overline{\mathcal{I}_{v_{|\mathcal{V}|}}}(t) \right). \tag{2.5}$$

For simplicity of notation, the argument (t) is omitted 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:

$$\overline{\mathcal{I}_{\mathcal{G}}} = \frac{1}{|\mathcal{V}|} \sum_{i=1}^{|\mathcal{V}|} \overline{\mathcal{I}_{v_i}}. \tag{2.6}$$

A similar assumption to other works in the literature (*cf.* [Chevaleyre, 2004], [Almeida et al., 2004]) is taken in the beginning of the experiments, where for all $v_i \in \mathcal{V}$, $\mathcal{I}_{v_i}(0) = 0$, as if every vertex had just been visited at the beginning of the mission. Consequently, there is a transitory phase in which the $\overline{\mathcal{I}_{\mathcal{G}}}$ values tend to be low, not corresponding to the reality in steady-state, as will be seen in experimental tests conducted in this thesis. 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

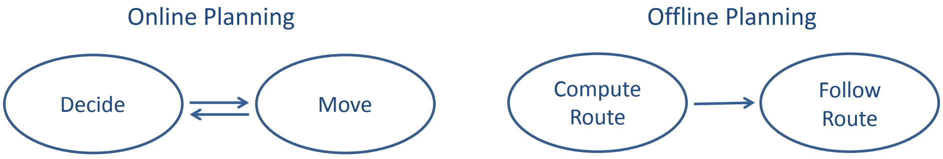


Figure 2.7: Properties of the two planning approaches to solve the patrolling problem with teams of multiple mobile robots.

problem with an arbitrary team of R robots can be described as the problem of finding a set of R robot trajectories that visit all vertices $v_i \in \mathcal{V}$ of $\mathcal{G}$, with the overall team goal of minimizing $\overline{\mathcal{I}_{\mathcal{G}}}$:

$$f = \underset{\mathbf{x}}{\operatorname{argmin}}(\overline{\mathcal{I}_{\mathcal{G}}}), \tag{2.7}$$

by finding:

$$\Pi = \{\pi_1, ..., \pi_r, ..., \pi_R\}, \tag{2.8}$$

such that:

$$\pi_r = \{v_a, v_b, ...\}, \tag{2.9}$$

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

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

subject to:

$$\forall v_i \in \mathcal{V}, \exists \pi_r \in \Pi : v_i \in \pi_r. \tag{2.10}$$

Note that π_r represents the patrolling trajectory of robot r, and $v_a, v_b, ...$ are vertices successively connected in $\mathcal{V}$. The route π_r of each robot can either be calculated *a priori*, which is typically done by classical approaches, *e.g.*, by assigning predefined tours to robots; or online to consider and incorporate the dynamics of the system in a given time step, which is usual in distributed approaches, where each agent incrementally decides on its moves autonomously. This is illustrated in Fig 2.7.

2.5 Summary

Related work on the patrolling problem has been presented in this chapter and a wide variety of existing approaches have been categorized and analyzed. Several concepts and issues involved in patrolling missions have been introduced such as *agent's perception*, *decision-making*, *communication model*, *environment topology*, *scalability*, *robustness*, *unpredictability*, *feasibility* and *team coordination*. These concepts are continuously mentioned throughout this work.

In order to contribute to this field, the author listed in this chapter a number of scientific objectives and proposed novelties that are described, in more detail, in the remaining chapters of this thesis, based on the weaknesses of the literature previously pointed out.

Moreover, the fundamental multi-robot patrolling problem to address in this thesis was formulated. Starting from an *a priori* metric representation, the environment is abstracted by a topological map, which is described by an undirected graph $\mathcal{G}$. Using the navigation graph concept, the idleness performance criterion has also been defined.

In the next chapter, the first two objectives proposed (*cf.* section 2.3) are addressed. More specifically, a study on performance and scalability of several existing strategies for Multi-Robot Patrol is presented, by resorting to realistic simulations, distinct graph topologies and conducting statistical analysis on the results to further understand the significance of the different factors involved in the problem.

Chapter 3

Performance and Scalability: Benchmarking Patrol Strategies

Lately, the interest of the research community in this field has been focused in the development of patrol strategies. However there is a deficit of studies comparing such strategies, namely in terms of their performance and team scalability in different environments. On the other hand, the ones that exist (*cf.*, [Machado et al., 2003], [Almeida et al., 2004]) rely on several unrealistic assumptions, as described previously.

For this reason, an evaluation of five representative patrol approaches is presented in this chapter. Aiming to analyze the performance, ability to scale and the behavior resulting from interactions between teammates, extensive realistic simulation using the Robot Operating System (ROS) [Quigley et al., 2009] together with Stage [Gerkey et al., 2003] was conducted. The metric used to compare the performance is the average idleness of the topological environment (*i.e.*, graph), that represents the area to patrol. This is represented by $\overline{\mathcal{I}_{\mathcal{G}}}$, being computed as described in section 2.4.2, equation (2.6).

The comparative study conducted evaluates five different state-of-the-art patrolling strategies using distinct topological environments and different team sizes, in order to analyze the performance and scalability of each approach. Additionally, these results help to identify which strategies enable enhanced team scalability, and which are the most suitable approaches given any environment. Also, statistical analysis on the results, enables to identify the significance of the different factors involved in the problem. Thus, this study may support future research directions in the field and possibly the development of new approaches in this domain, as well as in other multi-robot applications, in general. The work in this chapter is partially covered in [Portugal and Rocha, 2011c], [Portugal and Rocha, 2012b] and [Portugal and Rocha, 2013d].

This chapter is organized as follows. An overview of the ROS robotic framework and the Stage simulation environment is presented in the next section. Section 3.2 describes the patrolling strategies evaluated and compared in this work. The subsequent section presents the motivation behind the topological maps used and important aspects about the simulations performed. In section 3.4, experimental results are presented and discussed. Finally, the chapters ends with conclusions that summarizes the benchmarking work presented herein.

3.1 Robotic Framework and Simulation Environment

Several possible robotic frameworks and realistic simulation environments could be considered so as to evaluate different multi-robot patrolling strategies. In this section, an overview of *ROS* and *Stage*, the frameworks chosen in this work, is conducted in order to clarify the reasons for such choice.

3.1.1 Robot Operating System (ROS)

Despite the existence of many different robotic frameworks that were developed in the last decade, ROS[11] has already become the most trending and popular robotic framework, being used worldwide due to a series of features that it encompasses

[11] http://www.ros.org

(a) ROS Logotype. (b) ROS Hydro Medusa distribution.

Figure 3.1: Robot Operating System (ROS). The current distribution of ROS, named Hydro Medusa, was released in September 4th, 2013.

and being the closest one to become the standard that the robotics community urgently needed.

The required effort to develop any robotic application can be daunting, as it must contain a deep layered structure, starting from driver-level software and continuing up through perception, abstract reasoning and beyond. Robotic software architectures must also support large-scale software integration efforts. Therefore, usually roboticists end up spending excessive time with engineering solutions for their particular hardware setup [Araújo et al., 2013]. In the past, many robotic researchers solved some of those issues by presenting a wide variety of frameworks to manage complexity and facilitate rapid prototyping of software for experiments, thus resulting in the many robotic software systems currently used in academia and industry, like *Player* [Gerkey et al., 2003], *YARP* [Metta et al., 2006], *Orocos* [Bruyninckx, 2001], *CARMEN* [Montemerlo et al., 2003] or *Microsoft Robotics Studio* [Jackson, 2007]. Those frameworks were designed in response to perceived weaknesses of other available frameworks or to place emphasis on aspects which were seen as most important in the design process. ROS [Quigley et al., 2009] is the product of trade-offs and prioritizations made during this process.

ROS was originally developed in 2007 by the Stanford Artificial Intelligence Laboratory, and currently the development of ROS is shared between Willow

Garage[12] and the Open Source Robotics Foundation (OSRF)[13]. The major goals of ROS are hardware abstraction, low-level device control, implementation of commonly-used functionalities, message-passing between processes and package management. ROS promotes code reuse with different hardware by providing a large amount of libraries available for the community, like visual SLAM [Grisetti et al., 2006], 3D point cloud-based object recognition [Rusu and Cousins, 2011], among others; as well as tools for 3D visualization (*rviz*), recording experiments and playing back data offline (*rosbag*) and much more.

Regular updates enable the users to obtain, build, write, test and run ROS code, even across multiple computers, given its support for many processors running distributedly. Additionally, since it is highly flexible, with a simple and intuitive architecture, ROS allows reusing code from numerous other open-source projects such as several Player robot drivers, the *Stage* 2D [Vaughan, 2008] and *Gazebo* 3D [Koenig and Howard, 2004] simulation environments, the *Orocos* stack, which wraps this modular framework mostly used in industrial robots and machine control, vision algorithms from the Open Source Computer Vision (OpenCV) library [Bradski and Kaehler, 2008] and planning algorithms from the Open Robotics Automation Virtual Environment (OpenRAVE) [Diankov and Kuffner, 2008]. Therefore, integrating robots and sensors in ROS is highly beneficial, since it strongly reduces the development time.

Due to its peer-to-peer, modular, tools-based, free and open-source nature, ROS helps software developers in creating robotic applications in a quick and easy way. These applications can be programmed in C++, Phyton, LISP or Java, making ROS a language-independent framework.

At the file system level, ROS resources files are distributed in groups of *packages*, which are the main unit for organizing software in ROS. A *package* may contain ROS runtime processes (*i.e.*, *nodes*), a ROS-dependent library, datasets and/or configuration files. *Stacks* are collections of *packages* that provide aggregate functionality. At the computation level, a peer-to-peer network of ROS processes is responsible to process all the data together. ROS places virtually all

[12] http://www.willowgarage.com

[13] http://www.osrfoundation.org

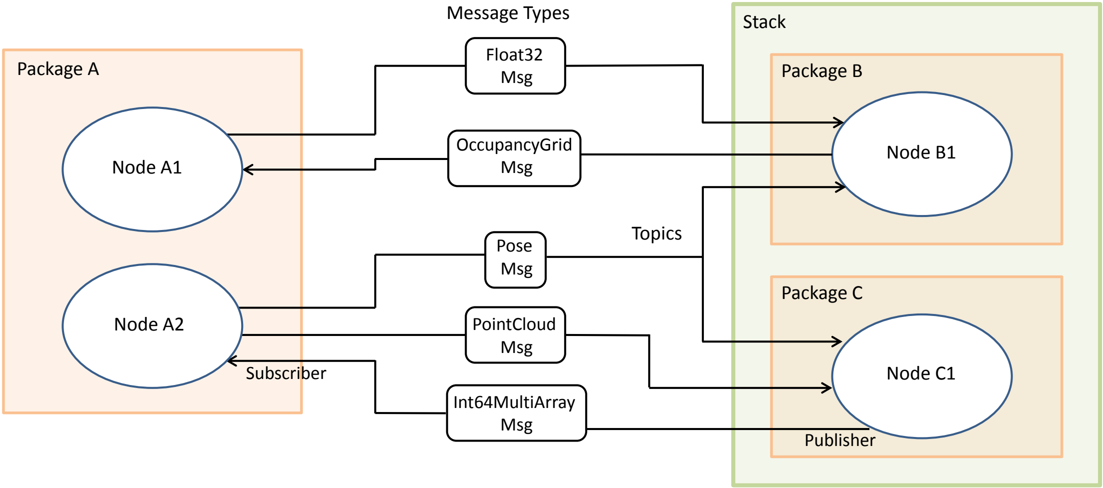
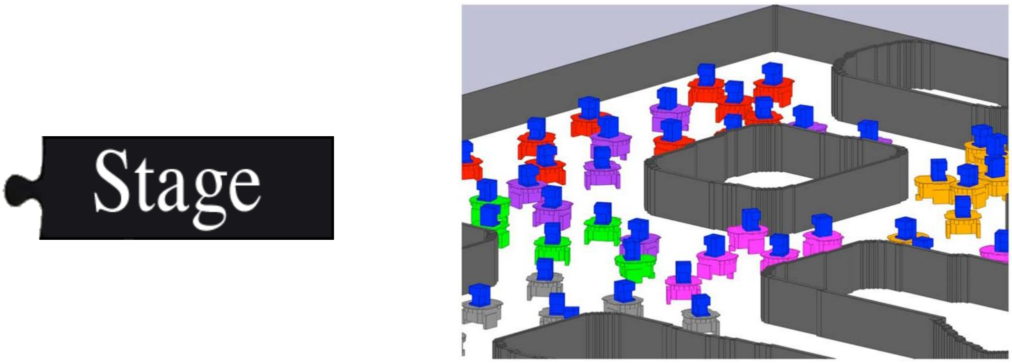


Figure 3.2: ROS architecture example diagram.

complexity in libraries, only creating small executables, *i.e.*, *nodes*, which expose library functionalities to ROS. The basic computation graph concepts of ROS are *nodes*, *Master*, *messages*, *services* and *topics*, all of which provide data to the graph in different ways. This is shown in Fig. 3.2.

Nodes communicate by publishing or subscribing to messages at a given *topic*. The *topic* is a name that is used to identify the content of the message. Hence, a node that requires a certain kind of data, subscribes to the appropriate topic. There may be multiple concurrent publishers and subscribers for a single topic, and a single node may publish and/or subscribe to multiple topics. The idea is to decouple the production of information from its consumption. In other words, one can think of a *topic* as a strongly typed message bus. Each bus has a name, and any entity can connect to the bus to send or receive messages as long as these are of the right type.

Beyond the easiness of using the available tools, ROS also provides effortlessly integration of new sensors without needing hardware expertise. As a result, the overall time spent in developing software is greatly reduced due to code reuse and hardware abstraction, when using available ROS drivers to interface with the hardware.



(a) Stage Logotype.

(b) Example of a simulation in Stage.

Figure 3.3: Stage Simulator.

3.1.2 Stage

Stage[14] is a C++ software library designed to support research into multi-agent autonomous systems. Stage simulates not only a population of mobile robots, but also sensors and objects in a two-dimensional bitmapped environment. It is a 2D dynamic physics simulator with some 3D extensions, therefore adopting terminology from computer graphics and video games, it is described as 2.5D (two-and-a-half dimensional) simulator. Its graphical interface is designed using *OpenGL*, which takes advantage of graphics processor (GPU) hardware, being fast, ease of use, and having wide availability. An example of the stage graphical user interface (GUI) is shown in Figure 3.3.

Stage was originally developed as the simulation backend for the Player/Stage system [Gerkey et al., 2003]. It was verified that Player *clients* developed using Stage would work with little or no modification with real robots and vice-versa. Thus, Stage allows rapid prototyping of controllers destined for real robots. This a powerful argument to support the real-world validity of Stage-only experiments and a major advantage of using a well-known simulator instead of home-grown, project-specific code. Stage also allows experiments with realistic robot devices that one may not happen to have. Various sensors and actuator models are pro-

[14] http://playerstage.sourceforge.net/index.php?src=stage

vided, including range-finders (sonar, laser scanners, infrared), vision (color blob detection), 3D depth-map camera, odometry (with drift error model), and differential steer robot base.

According to [Vaughan, 2008], Stage has several other important technical features, beyond the seamless interface with Player clients and providing models of many of the common robot sensors. Stage is relatively easy to use, it is realistic for many purposes, striking a useful balance between fidelity and abstraction that is different from many alternative simulators. It runs on Linux and other Unix-like platforms including Mac OS X, which is convenient for most roboticists, and it supports multiple robots sharing a world. Moreover, Stage is also free and open-source, has an active community of users and developers worldwide and has reached a well-known status of being a robust simulaton platform.

Stage enables experiment sharing, which is a crucial aspect. Thus, standard test scenarios can emerge, in which users can compare their algorithms in the same conditions. For this reason, it is ideal for benchmark of several different algorithms, like the one described later in this chapter.

Stage is made available for ROS, through the *stageros* node from the *stage*[15] package, which wraps the Stage multi-robot simulator. Using standard ROS topics, *stageros* provides odometry data from each virtual robot and scans from the corresponding laser model. Additionally, a ROS node may interact with Stage by sending velocity commands to differentially drive the virtual robot.

3.2 Evaluated Patrolling Algorithms

Having analyzed the literature, five representative algorithms were implemented for comparison in this benchmark evaluation. These algorithms were chosen from among all previous research works based on the good performance reported in previous works and the different properties assumed like agent perception, decision-making and planning, as it is shown in Table 3.1. In this section, those algorithms

[15] http://www.ros.org/wiki/stage

Table 3.1: Comparative Table of the Analyzed Architectures.

Algorithm	Complexity	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	Offline
MSP	****	Cognitive	Graph-based inside each region	Offline

Algorithm 3.1: Conscientious Reactive.

```
r ← load($robot\_id$);
$\mathcal{G}$ ← load($topological\_map$);
$v_n$ ← load($initial\_vertex$);
add($v_n$ $to$ $\pi_r$);
init($\mathcal{I}_{\mathcal{V}}(t)$);
while $true$ do
    $v_{n+1}$ ← argmax($\mathcal{I}_{N_{\mathcal{G}}(v_n)}$);        /* Next vertex is the neighbor of the current vertex with highest idleness. */
    move_robot($v_{n+1}$);
    $v_n$ ← $v_{n+1}$;
    add($v_n$ $to$ $\pi_r$);
    update($\mathcal{I}_{\mathcal{V}}(t)$);
```

are examined in detail.

Besides the analysis of the performance of the diverse algorithms, the scalability of the approaches studied is also addressed in this work. 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.

3.2.1 Conscientious Reactive (CR)

Ranked one of the top algorithms in the study of Machado *et al.* [Machado et al., 2003], Conscientious Reactive is a very 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 open neighborhood of the current

Algorithm 3.2: Heuristic Conscientious Reactive.

```
r ← load($robot\_id$);
$\mathcal{G}$ ← load($topological\_map$);
$v_n$ ← load($initial\_vertex$);
add($v_n$ to $\pi_r$);
init($\mathcal{I}_{\mathcal{V}}(t)$);
while true do
    MaxIdl ← max($\mathcal{I}_{N_{\mathcal{G}}(v_n)}$);
    MaxDist ← max $|e_{n,j}|, \forall e_{n,j} \in N_{\mathcal{G}}(v_n)$;
    forall the $v_i \in N_{\mathcal{G}}(v_n)$ do
        NormIdl $[v_i]$ ← $\frac{\mathcal{I}_{v_i}}{\text{MaxIdl}}$;
        NormDist $[v_i]$ ← $\frac{\text{MaxDist} - |e_{n,i}|}{\text{MaxDist}}$;
        Decision $[v_i]$ ← NormIdl $[v_i]$ + NormDist $[v_i]$;
    $v_{n+1}$ ← argmax(Decision $[N_{\mathcal{G}}(v_n)]$); /* Next vertex is the neighbor of the current vertex with highest heuristic decision value. */
    move_robot($v_{n+1}$);
    $v_n$ ← $v_{n+1}$;
    add($v_n$ to $\pi_r$);
    update($\mathcal{I}_{\mathcal{V}}(t)$);
```

vertex, $N_G(v_i)$[16], where the robot is located at the moment.

Basically, every time the robot arrives to a given vertex, it will move then to the adjacent vertex, which has not been visited for the longest time. This is shown in Algorithm 3.1, where the pseudo-code of the approach is presented.

3.2.2 Heuristic Conscientious Reactive (HCR)

Heuristic Conscientious Reactive is an algorithm presented by Almeida in [Almeida, 2003]. It is a more advanced version of CR, which has an important modification on the decision-making process. The authors calculate a decision value that con-

[16] The open neighborhood $N_{\mathcal{G}}(v_i)$ of a vertex v_i in a graph $\mathcal{G}$ is the induced subgraph of $\mathcal{G}$ consisting of v_i and all vertices adjacent to it [Diestel, 2005].

Algorithm 3.3: Heuristic Pathfinder Conscientious Cognitive.

```
r ← load($robot\_id$);
$\mathcal{G}$ ← load($topological\_map$);
$v_n$ ← load($initial\_vertex$);
MaxCost ← max $|e_{i,j}|, \forall e_{i,j} \in \mathcal{E}$;
MinCost ← min $|e_{i,j}|, \forall e_{i,j} \in \mathcal{E}$;
add($v_n$ to $\pi_r$);
init($\mathcal{I}_{\mathcal{V}}(t)$);
while true do
    MaxIdl ← max($\mathcal{I}_{\mathcal{V}}$);
    MaxDist ← max(dijkstra(from $v_n$ to all $v_i \in \mathcal{V}$));
    // Heuristic Decision:
    forall the $v_i \neq v_n \in \mathcal{V}$ do
        dist ← dijkstra(from $v_n$ to $v_i$);
        NormIdl $[v_i]$ ← $\frac{\mathcal{I}_{v_i}}{\text{MaxIdl}}$;
        NormDist $[v_i]$ ← $\frac{\text{MaxDist}-\text{dist}}{\text{MaxDist}}$;
        Decision $[v_i]$ ← NormIdl $[v_i]$ + NormDist $[v_i]$;
    v$_{n+1}$ ← argmax(Decision $[\mathcal{V}]$);
    // Path-Finding (Compute new edge costs):
    forall the $e_{i,j} \in \mathcal{E}$ do
        IdleCost ← $\frac{\text{MaxIdl}-\mathcal{I}_{v_j}}{\text{MaxIdl}}$;
        DistCost ← $\frac{|e_{i,j}|-\text{MinCost}}{\text{MaxCost}-\text{MinCost}}$;
        NewEdgeCost $[i,j]$ ← IdleCost + DistCost;
    $x_r$ ← dijkstra_path(from $v_n$ to v$_{n+1}$); // Using NewEdgeCosts.
    move_robot($x_r$);
    add($x_r$ to $\pi_r$); // Add path $x_r$ without $1^{st}$ vertex ($v_n$).
    $v_n$ ← v$_{n+1}$;
    update($\mathcal{I}_{\mathcal{V}}(t)$);
```

siders not only the instantaneous idleness of the neighbors of the current vertex as well as the distance to them. In order to compute this heuristic decision, the idleness and distance to a vertex are normalized according to the maximal idleness and distance respectively, in the neighborhood of the vertex where the agent is located. Algorithm 3.2 presents the pseudo-code of the approach.

3.2.3 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 (HPCC) plans on the global graph to decide which vertex to move to subsequently. HPCC was also presented by Almeida [Almeida, 2003] as a modified version of an approach called "Conscientious Cognitive" previously described in [Machado et al., 2003].

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 [Dijkstra, 1959], as seen in line 21 of Algorithm 3.3, which presents the pseudo-code of the approach.

3.2.4 Cyclic Algorithm for Generic Graphs (CGG)

Previous studies like [Machado et al., 2003] and [Chevaleyre, 2004] identify cyclic strategies based on the TSP as a method of guaranteeing low average idleness values. Solving TSP is NP-hard and is typically calculated using complete graphs and known-bound approximations. The problem becomes more complex when using generic non-complete graphs, like topological maps.

Consequently, inspired on these previous works, a Cyclic Algorithm was implemented in [Portugal and Rocha, 2010]. 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. Due to the NP-Completeness of finding an Hamilton cycle, a fast heuristic algorithm proposed in [Angluin and Valiant, 1979] was used. When no such cycles or paths exist, the method looks for long paths and non-Hamiltonian

Algorithm 3.4: Cyclic Algorithm for Generic Graphs.

```
build_$\pi_r$($\mathcal{G}$) {
    main_path ← hamilton($\mathcal{G}$);
    if main_path = $\emptyset$ then
        cycle ← non_hamilton_cycle($\mathcal{G}$) ;
        if size(cycle) > $|\frac{\mathcal{V}}{2}|$ then
            main_path ← cycle;
        else
            main_path ← longest_path($\mathcal{G}$) ;
        end
    end
    final_path ← main_path + detours($all\ e_{i,j} \in \mathcal{E}$) ;
    if main_path = hamilton_path ∨ longest_path then
        final_path ← add_inverse_path() ;
    end
    return $\pi_r$ ← final_path;
}
r ← load($robot\_id$);
$\mathcal{G}$ ← load($topological\_map$);
$\pi_r$ ← build_$\pi_r$($\mathcal{G}$);
$v_n$ ← load($initial\_vertex$);
k ← 0 ;
while $true$ do
    $v_{n+1}$ ← $\pi_r$ [k+1] ;
    move_robot($v_{n+1}$);
    k ++ ;
    if k = size($\pi_r$)−1 then k ← 0;
end
```

cycles as an alternative and computes detours to unvisited vertices, as seen in the pseudo-code of Algorithm 3.4. 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.

Algorithm 3.5: Mulilevel Subgraph Patrolling (MSP) Algorithm.

```
r ← load($robot\_id$);
$P$ ← $R$-way_Partitioning($\mathcal{G}$); // Creates partitions $P = \{P_1, ..., P_R\}$.
$\pi_r$ ← build_x$_r$($P_r$); // Build path in the assigned partition subgraph.
$v_n$ ← load($initial\_vertex$);
k ← 0 ;
while $true$ do
    v$_{n+1}$ ← $\pi_r$ [k + 1] ;
    move_robot(v$_{n+1}$);
    k ++ ;
    if k = size($\pi_r$) − 1 then k ← 0;
```

3.2.5 MSP Algorithm (MSP)

The MSP Algorithm [Portugal and Rocha, 2010] is an offline graph theory partitioning-based method, which splits the graph into regions, where agents perform the patrolling task. In the first phase of the algorithm, the graph is partitioned into R subgraphs, corresponding to the number of robots in the team. Graph partitioning is conducted using a fast multilevel approach for partitioning irregular graphs inspired on the work of Karypis and Kumar [Karypis et al., 1998]. Afterwards, each resulting subgraph is assigned to a robot, according to the robot's ID. In the following phase, robots patrol their independent areas in a cyclic way, using the same approach described in the CGG method.

The performance of the algorithm strongly depends on how balanced the partitioning of the graph is, and the size of the team used is limited to the point where the graph can no longer be split with the hierarchical multi-level partitioning approach, described in [Portugal and Rocha, 2011a]. Algorithm 3.5 presents the pseudo-code of the approach.

Table 3.2: Connectivity Properties of the Graphs used in the Experiments.

Topological Map	Environment Area	Graph Density (D)	Fiedler Value (λ_1)
A	1357.17 m^2	0.0308	0.0080
B	1542.30 m^2	0.0746	0.0317
C	665.64 m^2	0.1333	0.1313

3.3 Setting up Comparative Experiments

In order to compare the performance and scalability of the five algorithms described previously, three topological maps were adopted, being 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 [Fiedler, 1973]. In order to remove its dependency on the number of vertices in the spectrum of the Laplacian matrix, the Normalized Laplacian $\mathscr{L}$ [Chung, 1997] was adopted to obtain the Fiedler value of each graph.

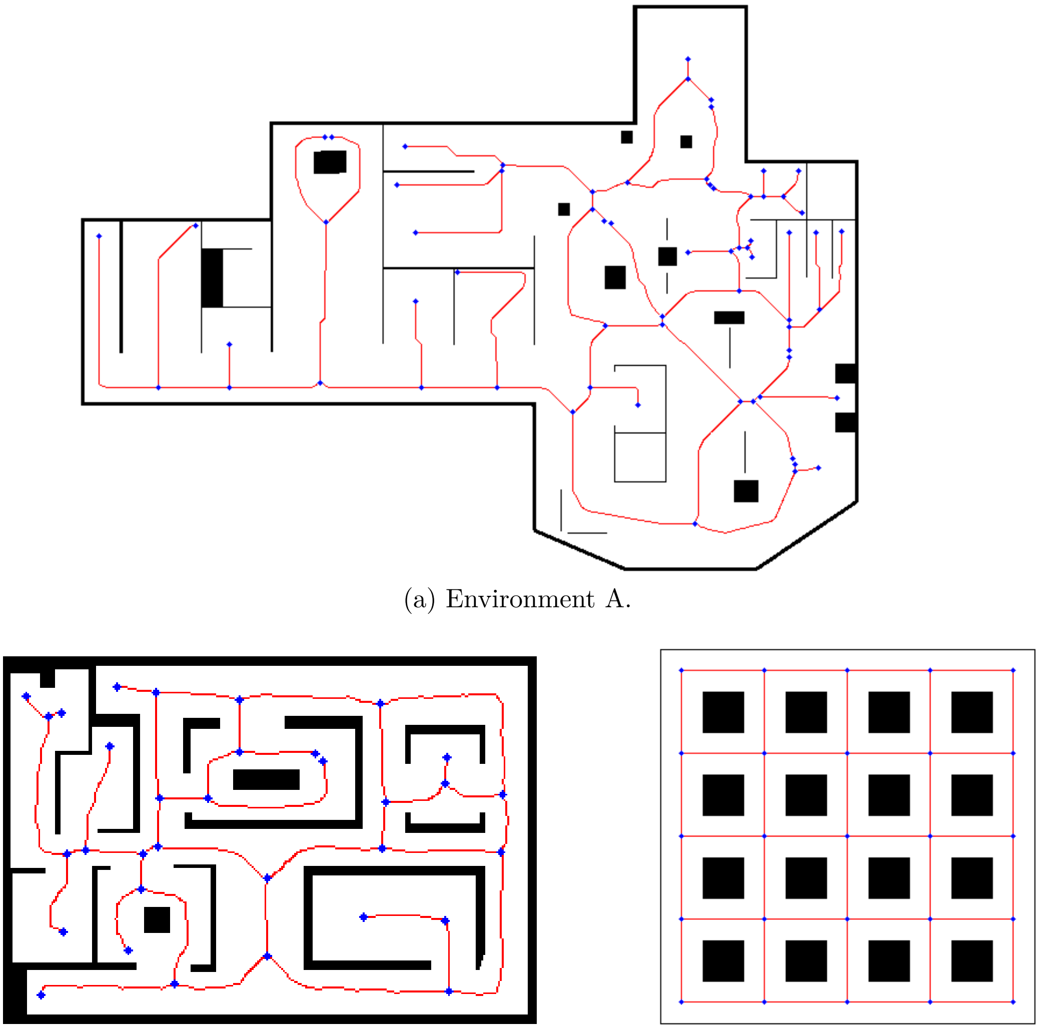
All eigenvalues of $\mathscr{L}$ are non-negative and $\lambda_0 = 0$. For non-complete connected graphs (as is those considered herein), the Fiedler Value λ_1 is the smallest non-zero eigenvalue of $\mathscr{L}$ and:

$$0 < \lambda_1 \leq 1. \tag{3.1}$$

Table 3.2 presents the connectivity properties of the graphs chosen for the experiments. Beyond the Fiedler Value, the Graph Density (D) was also calculated. This value represents a ratio between the actual number of edges and all possible edges if it were a complete graph:

$$D = \frac{2|E|}{|V|(|V|-1)}. \tag{3.2}$$

All three graphs and the respective environments are presented in Figure 3.4. In addition, it was necessary to resort to a simulator since it would not be possible to obtain the extent of results presented in the next section, within reasonable time limits, if teams of real robots had been used. Therefore, a recognized simulator with realistic modeling was chosen: the Stage 2D multi-robot simulator [Gerkey



(a) Environment A.

(b) Environment B.

(c) Environment C.

Figure 3.4: Environments used in the experiments with respective topological map.

et al., 2003, Vaughan, 2008], described in section 3.1.2.

Stage brings up the virtual world and the robots' models and ROS is used to program all five algorithms from the standpoint of each robot in the team and test their collective performance. 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.

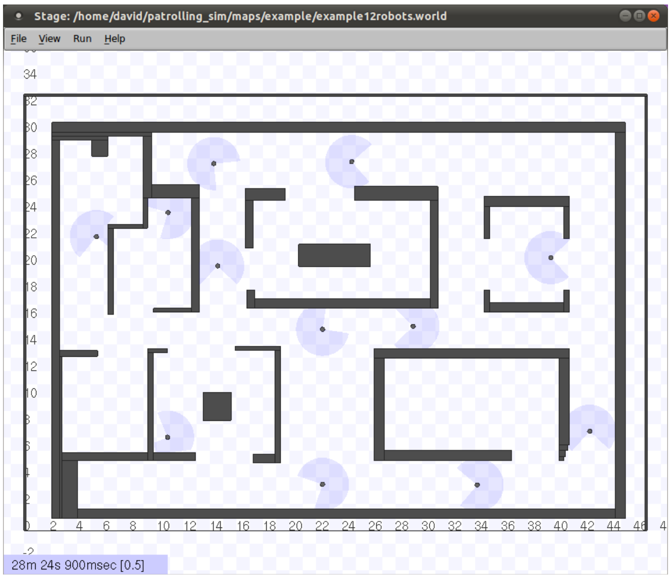


Figure 3.5: Snapshot of a Stage Simulation with 12 robots in Map B.

Robots navigate safely in the environment by heading towards their goals and avoiding collisions with walls and other robots through the use of ROS navigation stack[17] [Marder-Eppstein et al., 2010], which includes a probabilistic localization system, more specifically the adaptive Monte Carlo localization (AMCL) approach[18] [Thrun et al., 2001]. AMCL uses a particle filter to track the pose of a robot against a known map. These dynamics are implicit in all algorithms, when robots move. In addition, in these experiments robots are non-holonomic and have a maximum velocity of 0.2 m/s. Figure 3.5 illustrates a simulation run with 12 robots.

3.4 Evaluation and Discussion

The simulation process involved running the five described patrolling strategies with six different team sizes (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

[17] http://www.ros.org/wiki/navigation
[18] http://www.ros.org/wiki/amcl

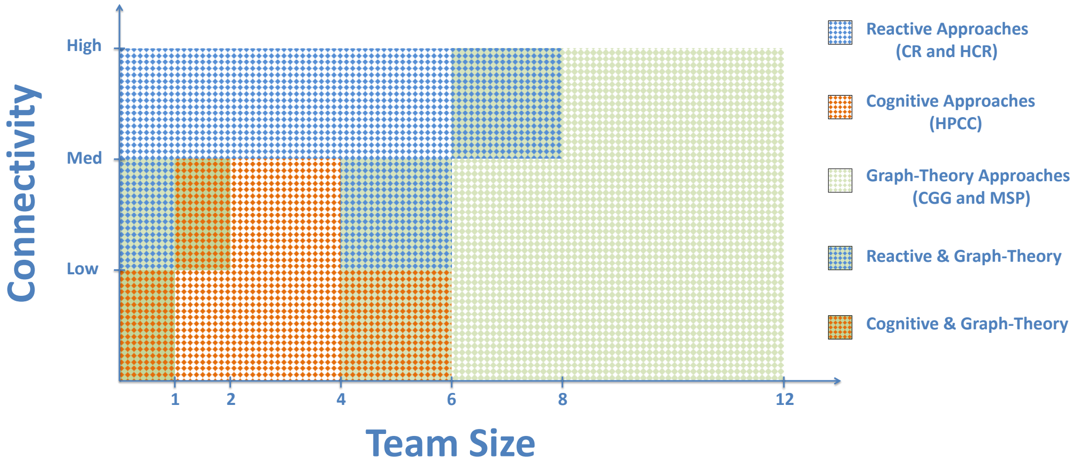


Figure 3.6: General Simulation Results. In this chart, the best strategies for a given map connectivity and team size are shown. Note that the figure presents some intersections of design solutions.

team size and environment. Every trial was repeated three times, in a total of 264 simulations[19], 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 team sizes. Simulations were stopped when the value of the average graph idleness ($\overline{\mathcal{I}_{\mathcal{G}}}$) after each patrolling cycle, *i.e.* every vertex visited, converged with 2.5% of tolerance. This resulted into an average simulation time of 1h18m, which led to accurate and similar results between different trials. There was no need to repeat the trials several times as testified by the overall average standard deviation of the results: $\overline{\sigma} = 4.42\%$.

The chart in Figure 3.6 represents environment *connectivity vs. team size* 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.

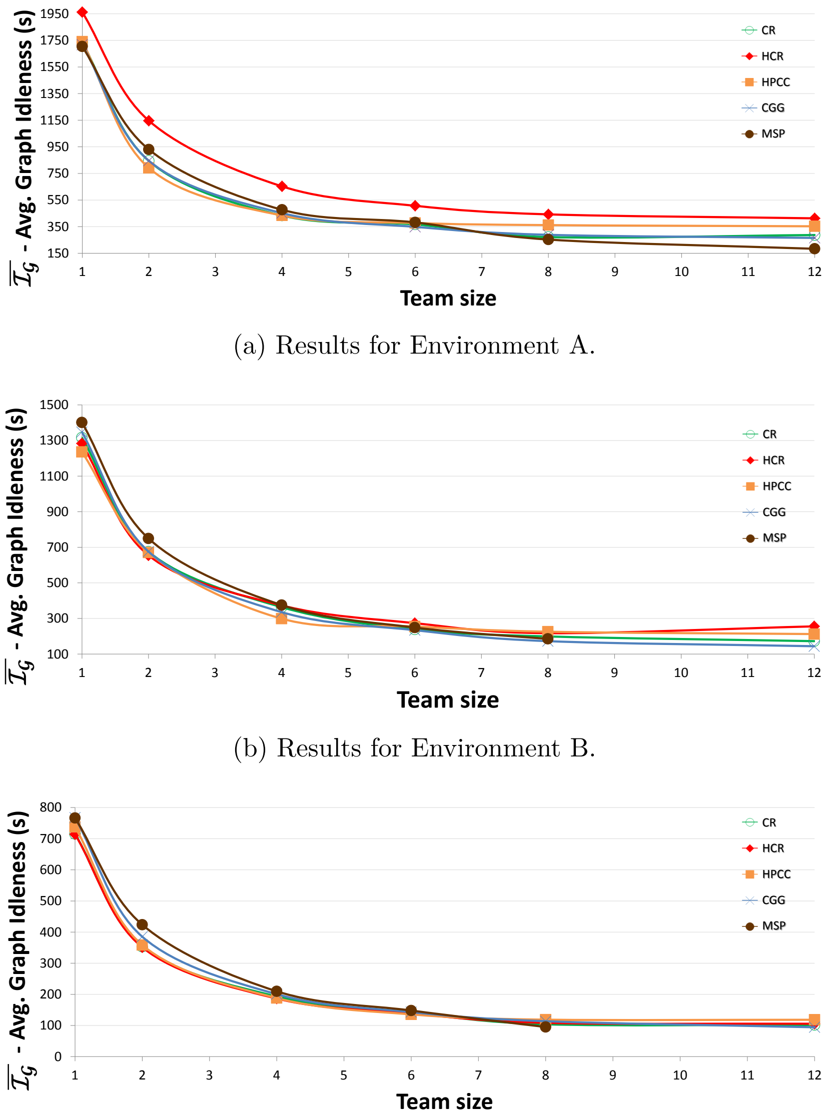
[19] The ROS simulation code is available at: http://www.ros.org/wiki/patrolling_sim

Generally, MSP is the algorithm with the best $\overline{\mathcal{I}_{\mathcal{G}}}$ 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. This confirms the suitability of employing cyclic strategies for multi-robor patrol in generic graphs. However, it does not scale as well as the MSP as seen in Figure 3.7a; *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 $\overline{\mathcal{I}_{\mathcal{G}}}$ 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. 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 3.7a and Table 3.3). This happens because agents tend to stay longer in regions with close vertices, causing high interference between



(a) Results for Environment A.

(b) Results for Environment B.

(c) Results for Environment C.

Figure 3.7: Simulation Results: $\overline{\mathcal{I}_\mathcal{G}}$ performance curves.

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 3.7c and Table 3.5, where agents have alternatives to decide at the very moment, which vertex to move next to, taking into consideration the state of the system. Never-

Table 3.3: Numerical Results for Map A.

Team	Average Graph Idleness $(\overline{\mathcal{I}_{\mathcal{G}}})$				
size	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 3.4: Numerical Results for Map B.

Team	Average Graph Idleness $(\overline{\mathcal{I}_{\mathcal{G}}})$				
size	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	172.4	255.62	212.3	143.94	-

Table 3.5: Numerical Results for Map C.

Team	Average Graph Idleness $(\overline{\mathcal{I}_{\mathcal{G}}})$				
size	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	-

theless, even in less connected environments, at some point when increasing the team size, the CR algorithm obtains better performance than the HPCC, since it scales better than the latter one.

Tables 3.3-3.5 show in detail a summary of all numerical results obtained in the simulation experiments. These were used to build the curves in Figure 3.7

completely clarifying and assisting the comparison between approaches, which is not always evident when the curves are too close. Each $\overline{\mathcal{I}_{\mathcal{G}}}$ value in the table is an average of three trials with the given algorithm, team size and map.

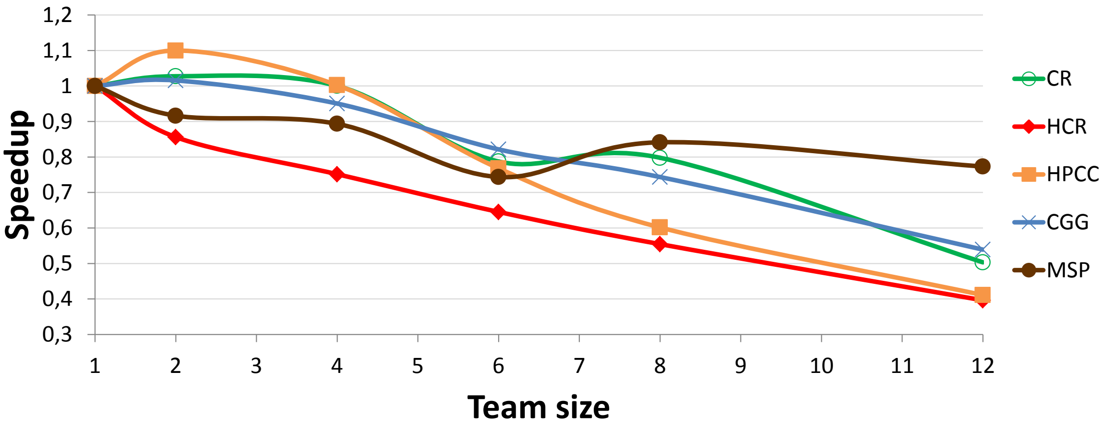
Additionally, as expected, all algorithms display increasing performance only until reaching a certain group size, around which the group productivity stagnates and even drops with the addition of robots; *e.g.*, HCR in environment B as illustrated by Figure 3.7b. In theory, productivity should grow during size scale-up; however spatial limitations increase the interference between robots causing the decrease of performance. For example, calculating Balch's speedup measure [Balch and Arkin, 1994] for increasing team sizes[20]:

$$\upsilon(R) = \frac{\Psi(1)/R}{\Psi(R)}, \tag{3.3}$$

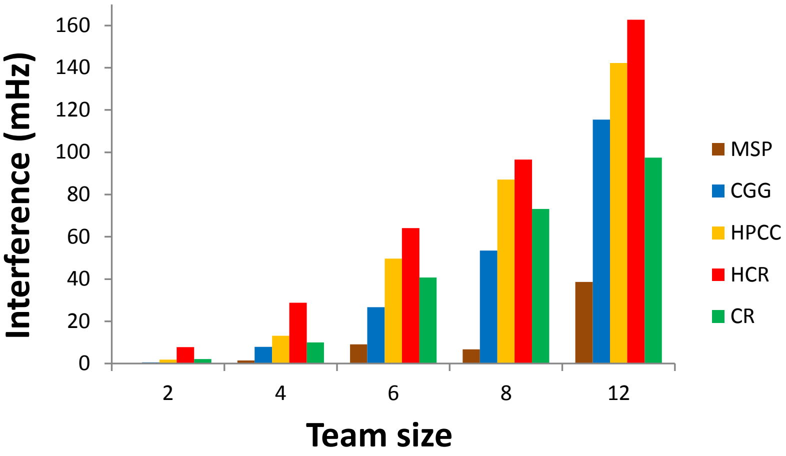
where $\Psi(R)$ is the performance for R robots, it is straightforward to conclude that such systems rapidly enter in sublinear performance ($\upsilon(R) < 1$), as shown in Figures 3.8a, 3.9a and 3.10a for all three environments tested. On the other hand, in Figures 3.8b, 3.9b and 3.10b, a measure of interference in the three environments is presented. Interference is 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 team sizes, instead of cooperating, robots tend to compete to firstly reach a given vertex than their teammates. In such cluttered situations, robots spend inestimable time avoiding teammates, which highly affects performance. 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.

It is also interesting to see that Figure 3.7 and Tables 3.3-3.4 show that, even though map B has a larger area to patrol when compared to map A, all algorithms obtain lower $\overline{\mathcal{I}_{\mathcal{G}}}$ values for the same number of robots in environment B, due to

[20] In this case Balch's speedup equation should be adapted since lower values mean higher performance.



(a) Speedup Chart.

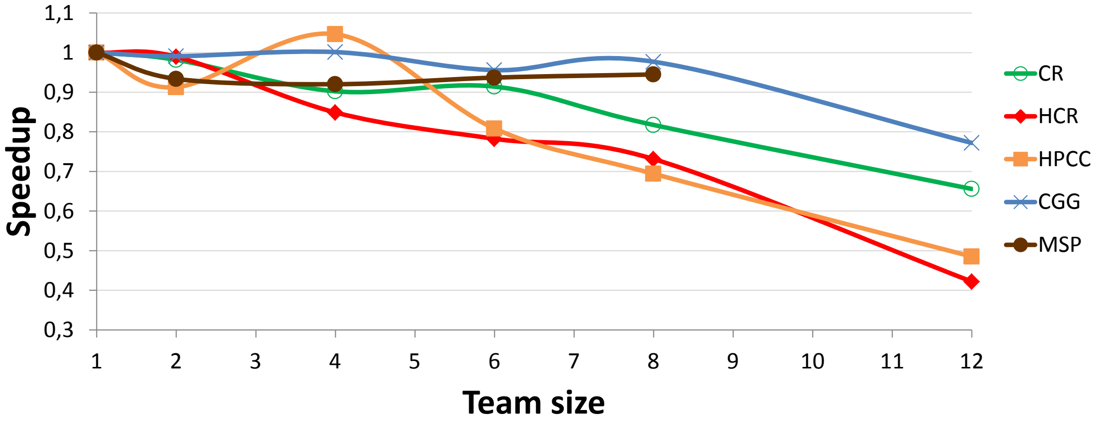


(b) Interference Chart.

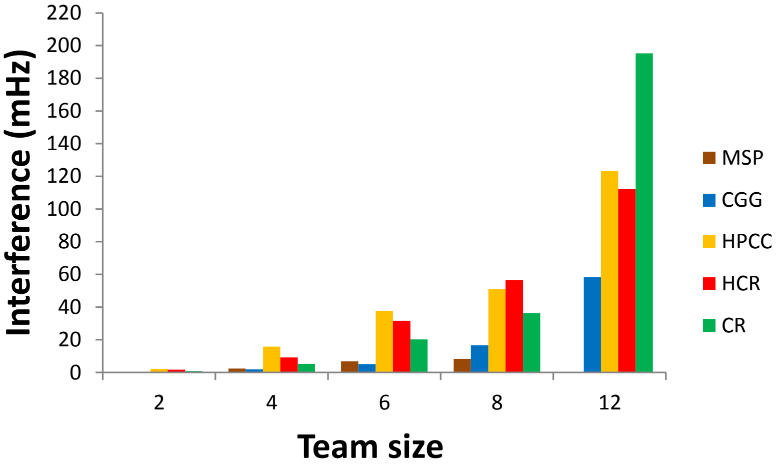
Figure 3.8: Speedup and Interference in environment A.

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. As expected, the performance of the team is also greatly affected by graph dimension.

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 $\overline{\mathcal{I}_{\mathcal{G}}}$ and the median $\widetilde{\mathcal{I}_{\mathcal{G}}}$, which shows that the algorithm



(a) Speedup Chart.

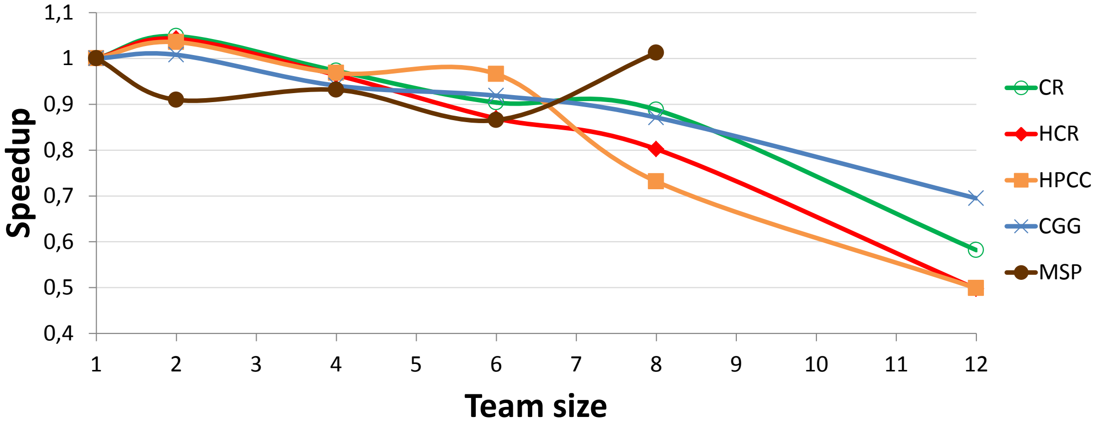


(b) Interference Chart.

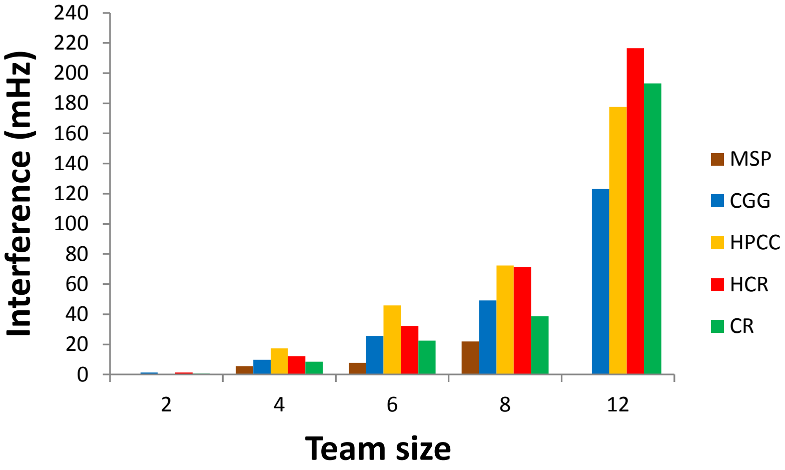
Figure 3.9: Speedup and Interference in environment B.

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.

The maximum average idleness of all vertices $max(\overline{\mathcal{I}_{\mathcal{V}}})$ was also analyzed. It is typically around 2.7 times larger than the average graph idleness. This ratio grows consistently with team size for all algorithms, being lower (around 2 times in average) for small team sizes and increasing for higher team sizes. As expected, CR due to its balanced property is the approach with a lower overall ratio of around 2.25 and surprisingly, if one considers the little difference between the two



(a) Speedup Chart.

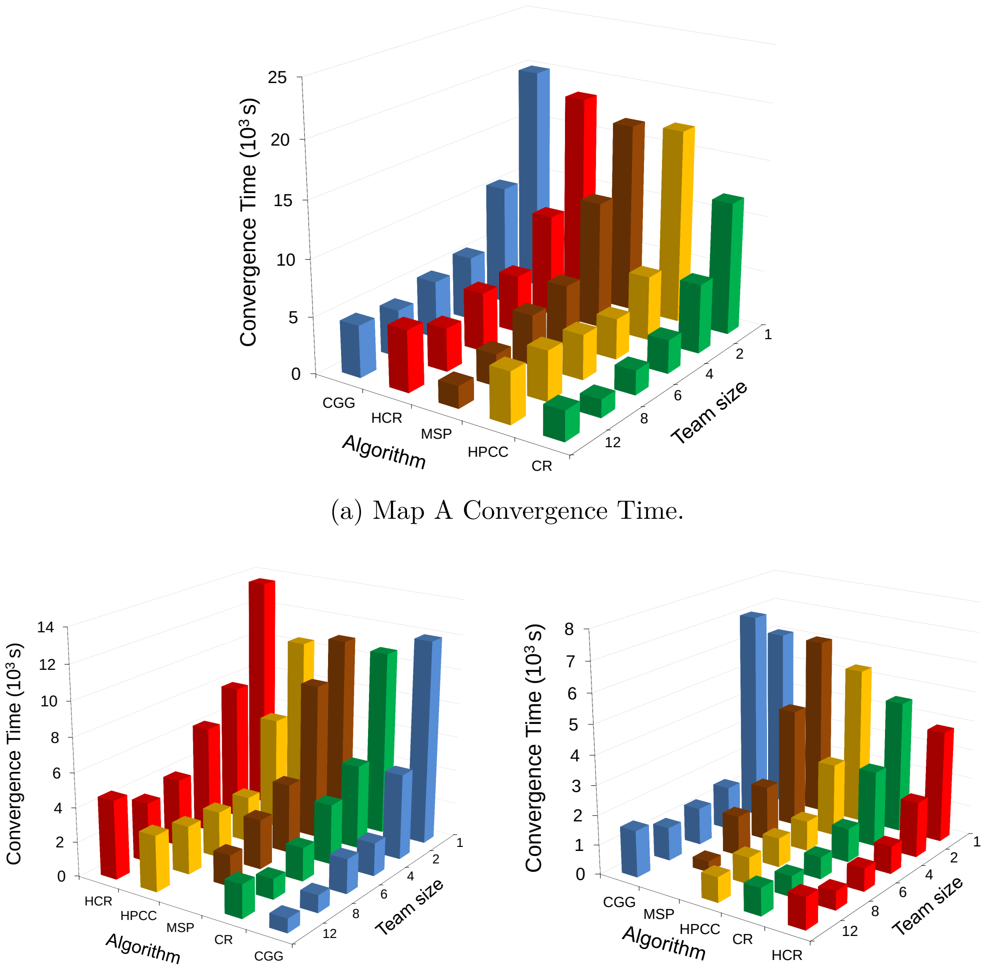


(b) Interference Chart.

Figure 3.10: Speedup and Interference in environment C.

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.

In terms of time taken to conclude the patrol task, it can be seen that CR usually needs less time to converge than the remaining approaches, as shown in Figure 3.11. This happens due to its property of constantly visiting places that have been idle for a long time, regardless of the distance to them. Consequently, it maintains a similar visit rate to all places. Despite this interesting aspect, it does not lead to better performance, when compared to other approaches. In



(a) Map A Convergence Time.

(b) Map B Convergence Time.

(c) Map C Convergence Time.

Figure 3.11: Average Convergence Time for each Algorithm with different team size.

fact, there is no apparent relation between performance and convergence time. Differences between the approaches are more marked with low number of robots as well as with different environment connectivity. Nevertheless, global trends can be observed. For instance, convergence time generally drops when team size increases from 8 to 12 robots.

In order to verify the significance of the problem's parameters tested in the experiments, Analysis of Variance (ANOVA) [Scheffé, 1959] was applied to measure quantitatively the group's variable effect. ANOVA is a powerful statistical technique which enables the comparison between parameters of more than two populations. From the analysis of the total dispersion present in a data set, it allows us to identify the source of the variations that led to that dispersion and evaluate the contribution of each factor, determining whether a significant relation exists between variables. The experiments presented in this work consider three factors (*algorithm*, *team size* and *connectivity*), thus the analysis of variance used to study their effects is called a three-way ANOVA.

Linear models were considered, assuming that the probability distribution of the response is normal, mutually independent and homoscedastic, *i.e.*, the variance of the data inside the groups is equivalent. The ANOVA fundamental test, which makes use of the F-statistics distribution usually with 95% of confidence bounds, verifies the significance of the factors by checking the group's variable effect (α_i):

$$\mathbf{H_0} \quad : \quad \alpha_1 = \alpha_2 = ... = \alpha_I = 0, \tag{3.4}$$

$$\mathbf{H_1} \quad : \quad \exists \alpha_i \neq 0, \tag{3.5}$$

where H_0 is the null hypothesis and H_1 is the alternative hypothesis. If the probability of the null hypothesis is near 0, a main effect is present due to the associated factor, meaning that the result is statistically significant. Arbitrarily high n-way ANOVA divides the total variation, given by the deviation of all observations from the global mean, into variations given by different factors and residual variation (or error). Also, if the model has a non-additive effect, variations given by the interaction of factors are also considered. These variations are calculated through sums of squares.

In this work, a test using the F-statistics distribution with 95% of confidence bounds and a model with first-order interaction effects between pairs of factors was adopted. This model explains 99.75% of the variation of the results. The ANOVA table, presented in Table 3.6, illustrates the model used.

Table 3.6: ANOVA Table.

Source	*SS*	*dof*	*MS*	*F*	*Prob > F*
Algorithm	64953.125	4	16238.281	14.603	0
Team size	12763618.172	5	2552723.634	2295.671	0
Connectivity	2574860.148	2	1287430.074	1157.789	0
Algorithm*Team size	36814.014	20	1840.701	1.655	0.0891
Algorithm*Connectivity	120574.630	8	15071.829	13.554	0
Team size*Connectivity	1279680.351	10	127968.035	115.082	0
Error	42254.968	38	1111.973	-	-
Total	17234127.010	87	-	-	-

The only factor that presents no relevant significance is the *algorithm*team size* interaction, seeing as the null hypothesis was accepted:

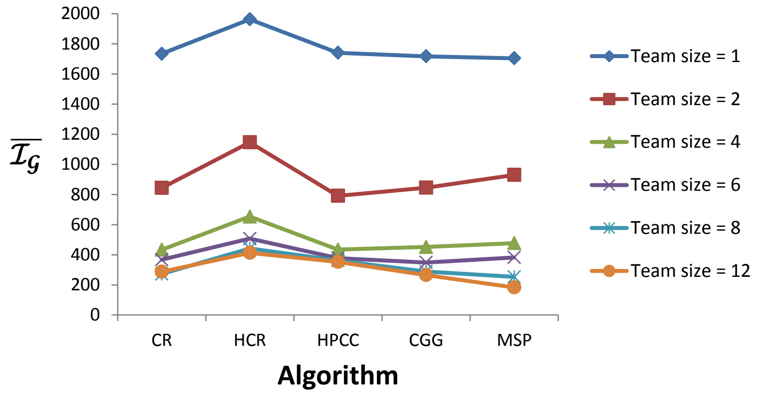
$$F\text{–}test_{Alg*TS} = \frac{MS_{Alg*TS}}{MSE} = \frac{1840.701}{1111.973} = 1.655, \qquad (3.6)$$

$$1.655 < F_{20,38}(\alpha = 0.05) \simeq 1.85. \qquad (3.7)$$

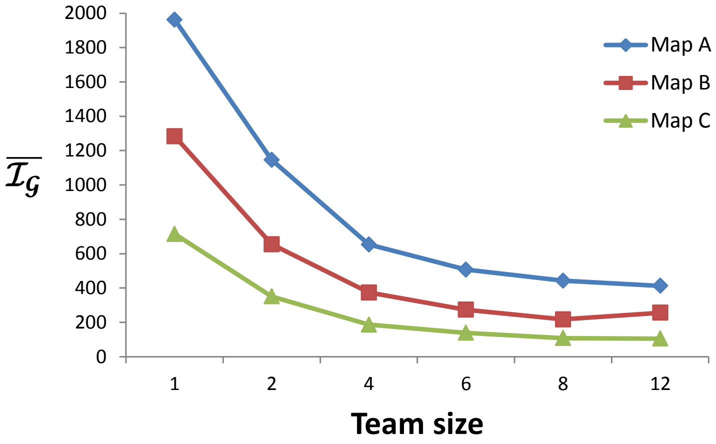
In fact, a clear indication of the low significance of this interaction is given by the $\overline{\mathcal{I}_\mathcal{G}}$ values of Tables 3.3-3.5, which do not differ much when each of the associated columns are compared as a whole. In addition, analyzing the individual factors, it can be seen that the influence of team size and connectivity in the results is greater than in the algorithm case. As a consequence, the interaction factor between team size and connectivity is the most significant interaction. Figure 3.12a depict the approximately additive effect of interaction *team size-algorithm* and the non-additive effect of the interaction *team size-connectivity* is presented in Figure 3.12b.

A more complete model could eventually be obtained by considering second-order interaction of the three factors. However, this interaction is only responsible for the remaining 0.25% variation of the results.

These results are the natural evidence that performance relies heavily on the number of members in the team and the environment to patrol.



(a) Approximate additive effect of the *algorithm*team size* factor.



(b) Non-additive effect of the *team size*connectivity* factor.

Figure 3.12: In a) the factors algorithm and team size present an approximate additive effect, which is clear by the near parallel curves. In this chart, the environment used was map A. In b) the factors team size and connectivity present a clear non-additive effect. In this chart, the algorithm used was HCR.

3.5 Summary

In this chapter, 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.

Quantitative analysis through three-way ANOVA was used to understand the significance of the different factors involved in the problem. The results presented were somehow unforeseen, given that the patrolling algorithms proved to be much less significant as a factor than team size and connectivity of the environment, which represents an important conclusion to this field.

Therefore, upcoming work should be guided towards approaches that ensure scalability and are appropriate, or perhaps can adapt, to all kinds of environment in order to improve the team's performance and minimize interference between robots.

Beyond covering the benchmarking objective with realistic assumptions and the objective of evaluation of the scalability of diverse strategies in different topologies, conducting simulation experiments in well-known platforms like ROS and Stage and making the code used in these simulations publicly available, represents a step forward, towards the objective of presenting a standard environment for researchers all over the world to implement different strategies and precisely compare with previously existing ones.

In the next chapter, the focus is shifted so as to validate multi-robot experimentation in real-world scenarios, using a distributed coordination architecture and aiming to minimize the interference that arises in teams with large numbers of robots, as well as studying robustness against robot failures and communication errors.

Chapter 4

A Distributed, Scalable and Robust Framework

In the Multi-Robot Patrolling Problem (MRPP), agents must coordinate their actions while continuously deciding which place to move next after clearing their locations. Classical approaches commonly address this problem using centralized planners with global knowledge, and/or calculating *a priori* routes for all robots before the beginning of the mission. According to [Pasqualetti et al., 2012b], "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".

Contrarily, in this work, two totally distributed techniques to solve the problem are proposed. The patrolling route π_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 computed by a centralized entity.

Despite the diversity of techniques proposed in the literature, there is an evident lack of implementation using physical MRS in non-centralized architectures. This serves as a motivation for the need to continuously coordinate teams of robots in patrolling missions in a distributed way, to validate these systems in the real-world and the possibility to add or remove patrolling agents (*e.g.*, due to faults).

The two proposed solutions to address this problem employ a Bayesian-based mathematical formalism for decision-making and coordination of the team of mobile robots. As will be seen, the main advantages of this framework are: providing the robots with suitable autonomy[21]; the straightforwardness of implementation; and the quality of the technique when placed in comparison against other techniques, leading to a minimization of inter-robot interference and, therefore, providing a scalable and fault-tolerant solution.

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 solutions is preliminarily conducted through realistic simulations. Subsequently, the work is validated in experiments with robot platforms in a small lab scenario. Then, further experiments are done in a large indoor real-world environment using a team of autonomous mobile robots to assess performance, scalability and fault-tolerance. The work in this chapter is partially covered in [Portugal and Rocha, 2012c], [Portugal and Rocha, 2013a] and [Portugal and Rocha, 2013e].

In the last chapter, 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 [Furukawa et al., 2006], [Julian et al., 2012]. 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

[21] In Robotics literature, "autonomy" is a concept that can easily be misinterpreted. Autonomy in this context is related to the capacity of an individual robot to make an informed and uncoerced decision. It should not be mistaken for robot "energetic" autonomy, which relates to the operation time allowed by robot batteries.

herein can attain exceeding results in the field. To summarize, the contributions to the state-of the-art of the work presented in this chapter 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 autonomous decision-making 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 [Quigley et al., 2009].
- 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, the system is robust to robot failures and communication errors, and it is shown that simulations conducted are realistic and present similar results to real-world tests.

The next section describes the two proposed distributed multi-robot patrolling strategies. Sections 4.2, 4.3 and 4.4 present the results obtained both through simulations and hardware experiments, as well as a discussion of the facets of the problem and a description of each of the robotic platforms used in this work. Finally, the chapter ends with conclusions and open issues for further research.

4.1 Distributed Patrolling Strategies

In this section, two novel 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[22] 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 $(\overline{\mathcal{I}_{\mathcal{G}}})$. In this section, special focus is given to the selection of proper statistical distributions to model the data, in order to ensure the quality of the results [Jansen and Nielsen, 2007].

Additionally, it is worth noting that distributed strategies are addressed herein, where agents only decide to move progressively and amongst their local neighborhood. Also, the proposed approaches are based on heuristics, since the problem is known to be NP-hard (*cf.* section 2.1). Nevertheless, the effectiveness of the approaches is shown later in section 4.2, when compared with other approaches in the literature.

4.1.1 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

[22] see footnote 8.

decision stage, where it must decide the direction it should travel next, among all β adjacent vertices. Having that in mind, two fundamental random variables are defined. The first one is boolean and simply represents the act of moving (or not) to a neighbor vertex v_A:

$$move(v_A) = \{true, false\}, \tag{4.1}$$

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:

$$G_A(t) = c \cdot \left(\frac{\mathcal{I}v_A(t) - \mathcal{I}v_A(t + \Delta t)}{|e_{val}|} \right), \tag{4.2}$$

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 (4.2) is equivalent to:

$$G_A(t) = c \cdot \frac{\mathcal{I}v_A(t)}{|e_{val}|}. \tag{4.3}$$

For simplicity of notation, hereafter G_A is used 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 (4.4) 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):

$$|e_{val}| = \begin{cases} |e_{min}|, & \text{if } \mathtt{max}\{e_{0A}, \ldots, e_{0\beta}\} > 2\mathtt{min}\{e_{0A}, \ldots, e_{0\beta}\} \wedge |e_{0A}| < |e_{min}| \\ |e_{0A}|, & \text{otherwise.} \end{cases} \tag{4.4}$$

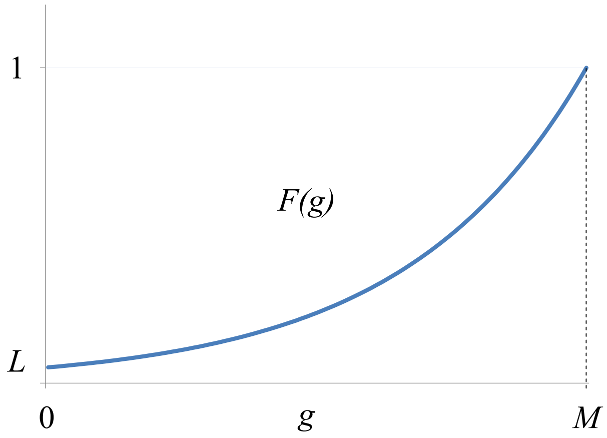


Figure 4.1: Distribution function of gain, given that *move* is true.

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:

$$P(move(v_i)|G_i) = \frac{P(move(v_i))P(G_i|move(v_i))}{P(G_i)}. \tag{4.5}$$

$P(move(v_i))$ 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 [Jansen and Nielsen, 2007], 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:

$$P(G_i \leq g) = \int_{-\infty}^{g} f(g)\,dg = \int_{0}^{g} f(g)\,dg = F(g), \tag{4.6}$$

$$\text{with:} \quad G_i \in [0, \infty]. \tag{4.7}$$

Note that $F(g)$ is the distribution function of G_i, and it is defined 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. 4.1:

$$F(g) = ae^{bg}; \quad a > 0, \tag{4.8}$$

$$\text{where:} \quad F(0) = L \Leftrightarrow a = L, \tag{4.9}$$

$$\text{and:} \quad 1 = Le^{bM} \Leftrightarrow b = \frac{\ln(1/L)}{M}. \tag{4.10}$$

This results in:

$$F(g) = L \cdot \exp\left(\frac{\ln(1/L)}{M} g\right), \tag{4.11}$$

$$\text{with:} \quad L, M > 0 \quad \text{and} \quad g < M. \tag{4.12}$$

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 (4.3) using an upper bound of $\mathcal{I}_{v_A}$. Finally, the probability density function $f(g)$ is obtained by

Algorithm 4.1: Greedy Bayesian Strategy (GBS).

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

differentiating $F(g)$:

$$f(g) = F'(g) = \frac{1}{M} \cdot \ln(1/L) \cdot \exp\left(\frac{\ln(1/L)}{M} g\right). \tag{4.13}$$

Now that the distribution model is defined, $P(move(v_i)|G_i)$ can be estimated via (4.5), for each vertex involved in the decision process. In algorithm 12 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, the previous formalism is used so as to easily add a new variable to the model in the next section and denote its flexibility.

4.1.2 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.

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), a discrete variable Vertex state S_i is defined, representing 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:

$$S_i \in \mathbb{N}_0 \cap [0, R-1], \quad R > 1. \tag{4.14}$$

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 robots' 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

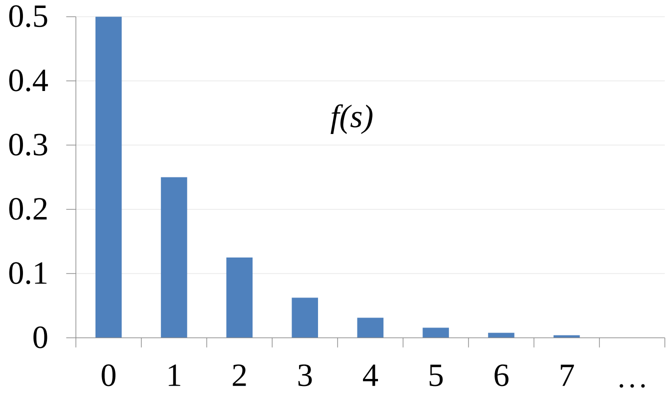


Figure 4.2: Distribution function of the vertex state, given that *move* is true.

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:

$$f_{S_i}(s)_{R\to\infty} = P(S_i = s)_{R\to\infty} = \frac{1}{2^{s+1}}, \tag{4.15}$$

as shown in Fig. 4.2. This geometric sequence is used to guarantee that the total probability for all S_i equals 1:

$$\sum_{s=0}^{R-1} f_{S_i}(s) = 1. \tag{4.16}$$

Eq. (4.15) 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 (4.15) is assumed:

$$f_{S_i}(s) = P(S_i = s) = \frac{2^{R-(s+1)}}{2^R - 1}; \quad R > 1, \tag{4.17}$$

Algorithm 4.2: State Exchange Bayesian Strategy (SEBS).

```
while $true$ do
    add($v_n$ to $\pi_r$);    // 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($\mathcal{I}_{\mathcal{V}}(t)$);
    $v_n \leftarrow v_{n+1}$;
```

which still holds condition (4.16). With the discrete probability distribution model characterized, robots can now decide moving to a specific vertex given its gain and state:

$$P(move(v_i)|G_i, S_i) \propto P(move(v_i))P(G_i|move(v_i))P(S_i|move(v_i)). \tag{4.18}$$

Algorithm 15 presents a high-level pseudo-code of the SEBS approach, which runs locally on each robot.

4.2 Experimental Validation

4.2.1 Simulation Experiments

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

Similarly to the simulations presented in section 3.4, the graph information of a given environment is loaded by every robot in the beginning, which then runs one of the algorithms. The robots navigate safely in the environment by heading towards their goals and avoiding collisions with walls and other robots through the use of ROS navigation stack and AMCL localization. This dynamic is implicit in both patrolling strategies, when robots move, and has a non-negligible effect on performance results. For this reason, the presented simulation experiments to study the MRPP are deemed as realistic. This is later confirmed on section 4.4. 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. 3.4 on page 61 presents three environment topologies with different algebraic connectivity λ_1. These topologies were used in the previous chapter, where they were classified as: lowly (A), mildly (B) and highly (C) connected, having a Fiedler value of $\lambda_{1A} = 0.0080$, $\lambda_{1B} = 0.0317$ and $\lambda_{1C} = 0.1313$, respectively. In this section, 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 team sizes (1, 2, 4, 6, 8 and 12)[23]. 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

[23] The simulation code including the two new strategies proposed in this chapter, is available at http://www.ros.org/wiki/patrolling_sim

Table 4.1: Overview of the Simulation Results with GBS and SEBS ($\overline{\mathcal{I}_\mathcal{G}}$ values in seconds).

	Map A		Map B		Map C	
team size	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

determined experimentally, being 3.75m and 3m 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 (4.4) does not apply. Table 4.1 presents the results obtained, wherein $\overline{\mathcal{I}_\mathcal{G}}$ was measured in seconds and was used as the performance metric.

Analyzing Table 4.1, 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 team size starts to rise, especially in teams of 6 or more robots. One explanation for this phenomenon is the growing interference between robots as team size R increases, which is shown in Fig. 4.3.

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, robots have to compete in cluttered areas to reach the same goal when adopting GBS.

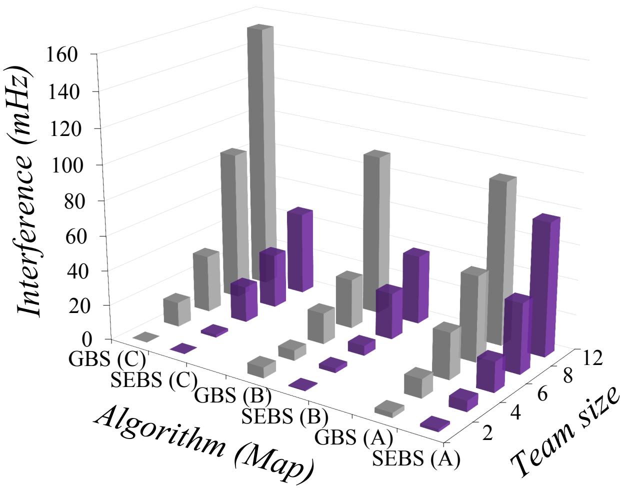


Figure 4.3: Interference levels for all experiments.

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 leads to 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 section 3.2, benchmarking tests were conducted with several state-of-the-art patrolling approaches using the same three environments and the same performance metric. In brief, Conscientious Reactive (CR) [Machado et al., 2003], Heuristic Conscientious Reactive (HCR) and Heuristic Pathfinder Conscientious Cognitive (HPCC) [Almeida et al., 2004] 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 inspired on the work of Chevaleyre [Chevaleyre, 2004] and [Portugal and Rocha, 2010] respectively, 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 section 3.2 in the performance comparison, tables 3.3-3.5 (*cf.* chapter 3) and table 4.1 show that GBS is the

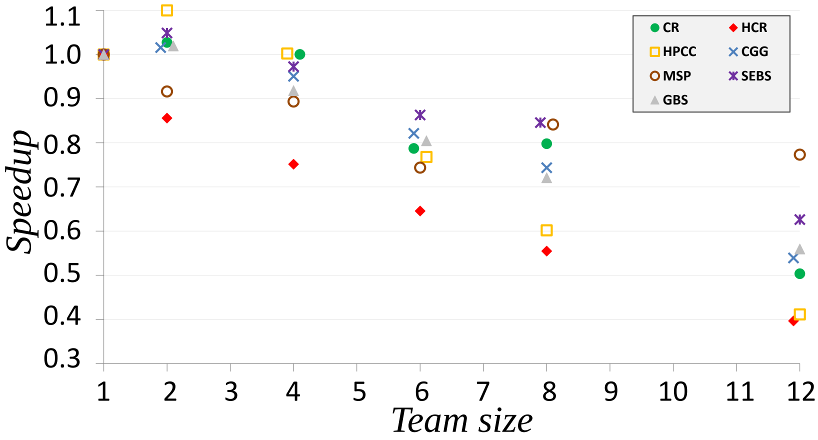


Figure 4.4: Speedup comparison with other approaches in map A.

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 (*team size, map*) pair, SEBS performance is always in the top 3, considering the total of 7 approaches, which demonstrates the suitability of SEBS independently of team size and graph connectivity, as well as the potential of employing Bayesian inspiration in the MRPP.

In order to analyze how well different strategies scale, Balch's speedup measure (eq. 3.3, page 67) was calculated for each strategy. Fig. 4.4 presents a chart comparing the speedup for each strategy in map A (including those in chapter 3). It can be seen that most systems enter progressively in sublinear performance ($v(R) < 1$) with team size, 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 team size, when compared to other approaches. Both perform effectively regardless of team size, 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 team sizes 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 team sizes.

These results show that the strategies proposed in this work are highly scalable, when compared to other distributed strategies. Furthermore, the proposed methods are able to adapt to non-standard situations like robot failures, while offline strategies (*e.g.*, CGG or MSP) are not able to account for these situations unless some adaptive online behavior is provided. In the next section, the robots used to validate the work in a preliminary experimental setup are presented.

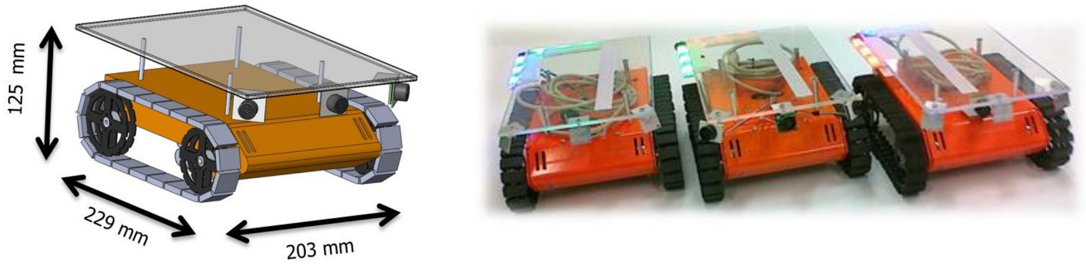
4.2.2 TraxBot

Earlier, mobile robotics research was especially focused on large and medium robotic platforms. However, with recent advances in sensor miniaturization and the increasing computational power and capability of microcontrollers in the past years, the emphasis has been put on the development of smaller and lower cost robots. Such low-cost platforms make affordable the experimentation with a larger number of robots (*e.g.*, in cooperative robotics and swarm robotics) and are also ideal for educational purposes. With such assumptions in mind, our research group has been doing engineering and research work with a custom mobile platform: the TraxBot [Araújo et al., 2012], [Couceiro et al., 2012].

Traxbot is a compact educational mobile robotic platform built around an Arduino controller board. The choice fell upon Arduino solutions, since it presents an easy-to-learn programming language (derived from C++) that incorporates various complex programming functions into simple commands that are much easier for students to learn. Moreover, the simplicity of the Arduino to create, modify and improve projects, as well as its open-source nature and reduced cost makes it among the most used microcontroller solutions in the educational context.

The TraxBot design has been chosen essentially due to the following reasons:

- Robustness: All hardware is either aluminum or stainless steel;



(a) Traxbot dimensions.

(b) A team of three Traxbots.

Figure 4.5: The Traxbot mobile robot.

- Low Cost: The platform costs around 485€, which makes affordable experiments with multiple robots;

- Operability: It has the ability to maneuver in many different types of terrain and surface topographies. Since it is a tracked mobile robot (TMR), it provides adequate mobility in unstructured environments;

- Autonomy: It can operate continuously around 2-3 hours. Robots should have a long battery life since they may have to operate for a long time during a mission;

- Sensor System: Equipped with ultrasonic range sensors to allow interaction with the environment;

- Dimension: It is adequate for both indoor and outdoor experiments. Its detailed dimensions are illustrated in Fig. 4.5a;

- Flexibility: It can incorporate many new extensions and components (*e.g.*, LEDs, cameras, LIDARs, grippers, etc.);

- Hybrid design: It is able to work with and without a small netbook on top of the platform according to the user's computational power;

- Communication: Supports wireless ZigBee communication or Wireless 802.11 b/g communication when a notebook is used on top;

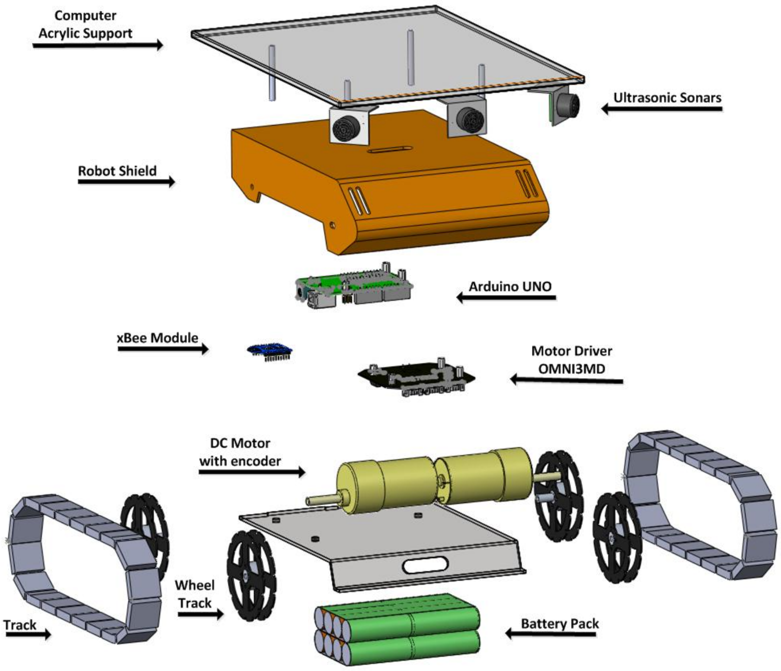


Figure 4.6: Mechanical structure of the Traxbot.

The embedded Arduino control board inside the platform accesses the motor encoders and other information from the power motor driver, like temperature and battery state, being also able to send commands to the motors, read sonar information and exchange messages natively through Zigbee.

The light and robust aluminum chassis is equipped with 2 DC gearhead motors on the front wheels, with quadrature wheel encoders of 624 pulses resolution. The Arduino board includes a microcontroller ATmega 328p [Banzi, 2011], that controls the platform's motion and exchanges data with the Bot'n Roll OMNI-3MD motor driver (*cf.* Fig. 4.6). This driver has the ability to control three motors in omnidirectional platforms by sending linear velocity, direction and speed commands, performing both velocity and position control.

The microcontroller is able to deal with data received from sensors, to send control signals to actuators and send information to computers or others devices. Furthermore, it offers advantages over higher processing systems, such as low-cost and lower power consumption. In addition, shields can be plugged on top of the

Arduino board, thus extending its capabilities. In this case a ZigBee shield module was added. ZigBee is used for exchanging short messages between robots when operating without a notebook and running simple coordination algorithms.

Also, a flat acrylic structure was added on the top of the platform to support a 10" notebook and the three ultrasonic range sonars, as presented in Fig. 4.6. The netbook may extend the processing power and provide Wi-Fi wireless communication, supporting larger bandwidth with heavy data exchange than Zigbee technology. The three Maxbotix Sonars MB1300 are used for distance sensing. In order to avoid the crosstalk phenomenon between sonars, a chain loop was created to provide synchronized cadence values from individual sonars. The robot is also equipped with two packs of 12V 2300mAh Ni-MH batteries and it has the flexibility to incorporate even more custom sensors.

TraxBot is a differential non-holonomic drive robot, with two motors independently controlled from each other. Its tracked configuration provides a better stability and traction, thus allowing performing a wider range of tasks in different surfaces when compared to wheeled robots.

Having developed this robot, it is seen as an ideal platform for education and research, since it can provide basics required for autonomous robot development, both at the hardware level (mechanics, energy, locomotion, embedded electronics, sensors) and software level (control theory, microcontroller programming, robot navigation trajectory planning, localization, etc.). The setting up, development and programming of the robot was motivated by experimentation and research in cooperative multi-robot systems.

4.2.3 Preliminary Experiments in a Lab Scenario

Simulation experiments allow attesting and comparing empirically the performance of distinct patrolling strategies in 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.

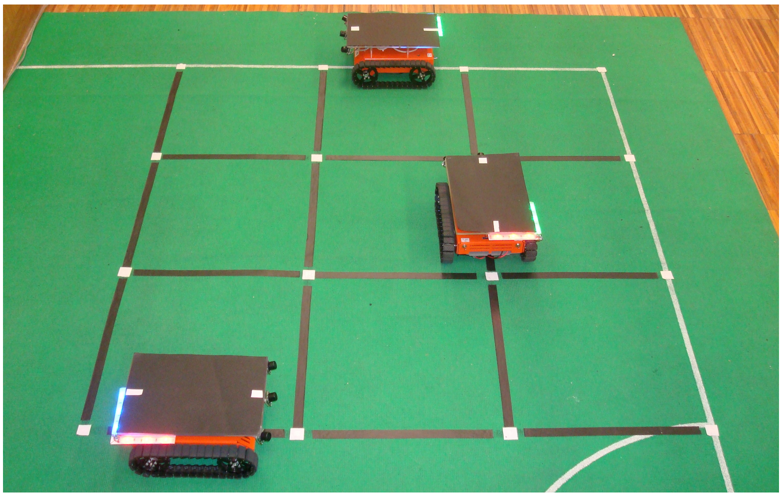
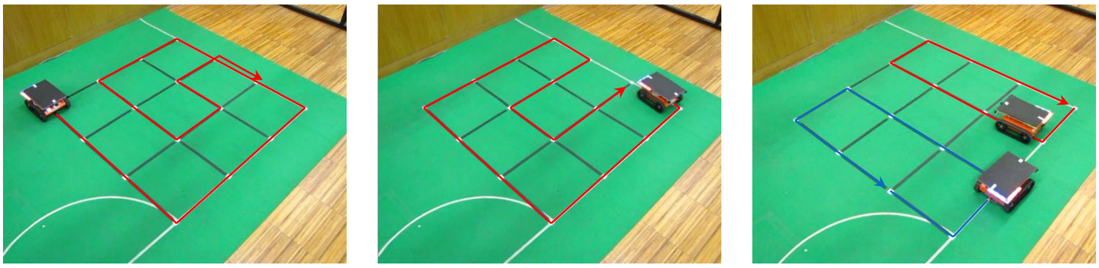


Figure 4.7: Lab Scenario with 3 TraxBots deployed in a 4×4 Grid map.

Therefore, preliminary experiments with a team of TraxBot platforms were conducted. In the presented experiments, these low-cost custom-made robots move at the same nominal speed, have low processing power since no netbook is used, and each robot represents a node of a self-configuring infrastructureless network, *i.e.*, a mobile *ad hoc* network (MANET), being able to communicate with its teammates through the use of ZigBee modules.

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 Fig. 4.7. 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 Imaging Source Firewire CCD Color Camara [Imaging Source, 2013] facing the ground was mounted on top of the scenario at a height of around 4 meters, tracking the robots' pose. This was done by identifying the colored ribbon LED strips on top of the platforms, which uniquely identifies each robot. The process includes background subtraction, detection of the robots' colored LEDs, position and heading calculation, and image to real-world coordinate transformation. The



(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.

Figure 4.8: Preliminary experiments with 1 and 2 robots. Arrows represent trajectories followed by robots.

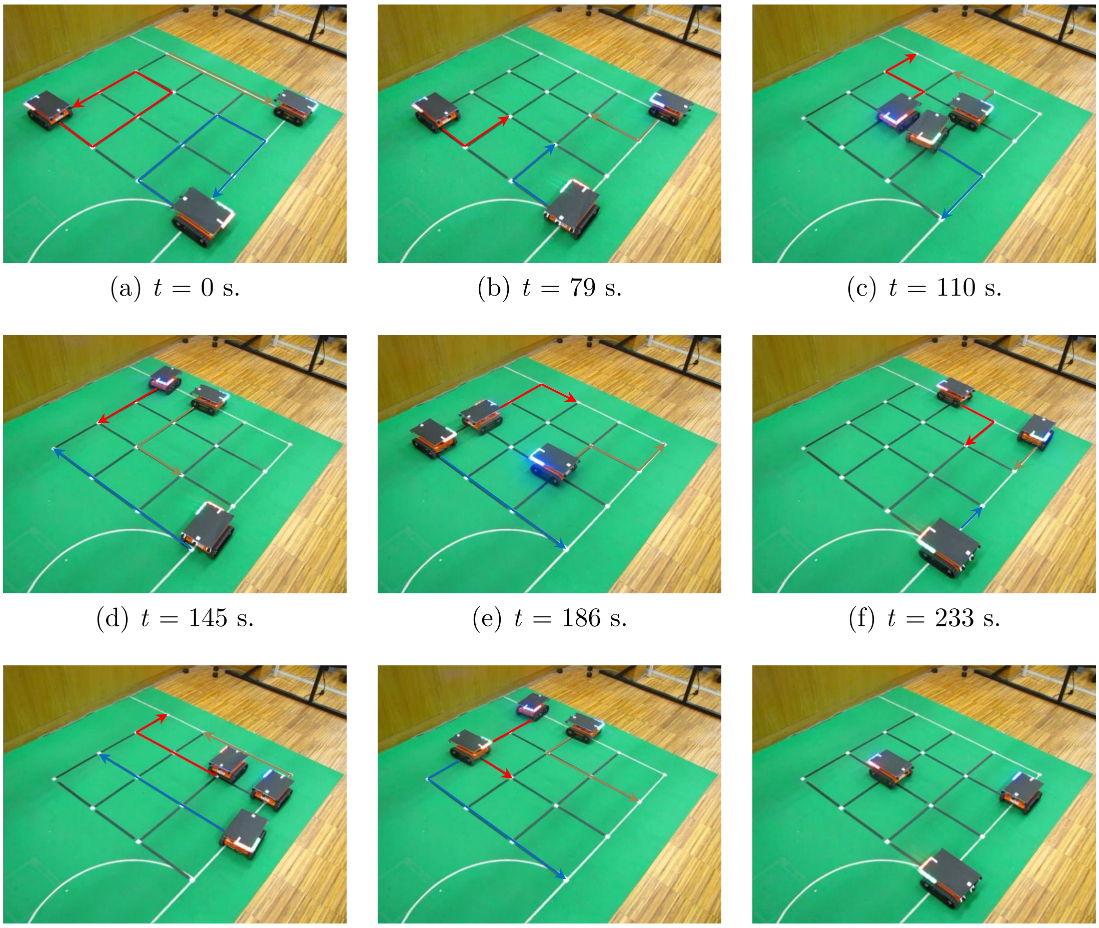
result of the system is the robot's position and heading, which is communicated through the Zigbee network. Up to three robots were deployed in the confined space, where they run the multi-robot patrolling algorithm, programmed in the microcontroller of each robot.

One important result, which can be seen in the video of these experiments[24] and in Fig. 4.8, 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. 4.9, 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[25].

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

[25] The theoretical lower bound for the number of moves would be 5.33.



(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.

Figure 4.9: Preliminary experiments with 3 robots. Arrows represent trajectories followed by robots.

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 team size grows, which is tantamount to saying that performance increases with team size.

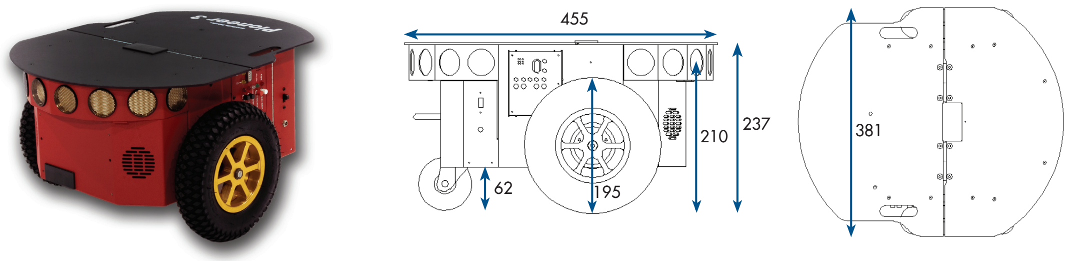
4.3 Experiments in a real-world Environment

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 the next section, the robotic platform used in the experiments is described.

4.3.1 Pioneer-3DX

Pioneer robots are the world's most popular mobile robots for research [ActivMedia, 2006]. More specifically, the Pioneer 3-DX is a lightweight two-wheel differential drive robot for indoor use, with a sturdy aluminum body and balanced drive system, which is equipped with an array of eight ultrasonic sonars in the front, as shown in Figure 4.10a. The sonar position in the array is fixed, with one on each side, and six facing outward at 20-degree intervals.

Pioneer 3-DX drive system uses two high-speed, high-torque, reversible-DC motors, each equipped with a high resolution optical quadrature shaft encoder for precise position, speed sensing and advanced dead-reckoning. The Pioneer 3-DX with onboard PC is a fully autonomous intelligent mobile robot, leveraging the microcontroller with ARCOS firmware based on the new-generation 32-bit Renesas SH2-7144 RISC microprocessor.



(a) A Pioneer 3-DX.

(b) Pioneer 3-DX dimensions (mm).

Figure 4.10: The Pioneer 3-DX mobile robot.

These robots are highly popular due to their versatility, reliability and durability. The robot uses 3 lead/acid batteries and can operate continuously for 8-10 hours, with a maximum load of 23 kg on top of the platform. Additionally, it can reach speeds of up to 1.2 m/s.

In terms of dimensions, the robot has a diameter of 45.5 cms and 23.7cms of height, as shown in Figure 4.10b. Also, the robot comes with foam-filled solid tires with knobby treads and contains a programmable piezo buzzer.

Pioneer robots are pre-assembled, highly customizable, upgradeable, and rugged enough to last through years of usage. Even though they are geared towards indoor use, the robot easily handles small gaps and minor bumping. For all these reasons, it is easy to find such robots in the majority of robotics research institute, typically with a wide variety of extensions like laser range finders (LRFs), grippers, stereo vision systems, electronic compasses, bumper rings and many more.

Unlike other commercially available robots, Pioneer's middle-size lends itself very well to navigation in tight quarters and cluttered spaces, such as classrooms, laboratories, and small offices. To make the robot fully capable of mapping and localization, a LRF is typically used to find its way home and performing other sophisticated path-planning tasks [Zaman et al., 2011]. Additionally, one can easily plug a laptop on top of the robot to extend its processing capability. This is shown in the upcoming sections.

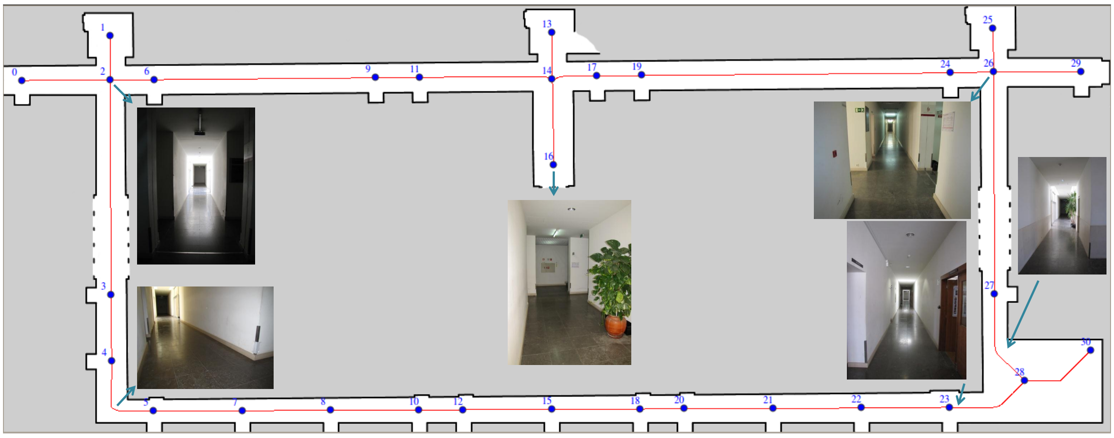


Figure 4.11: Topological map of the "ISR-Floor0" Environment.

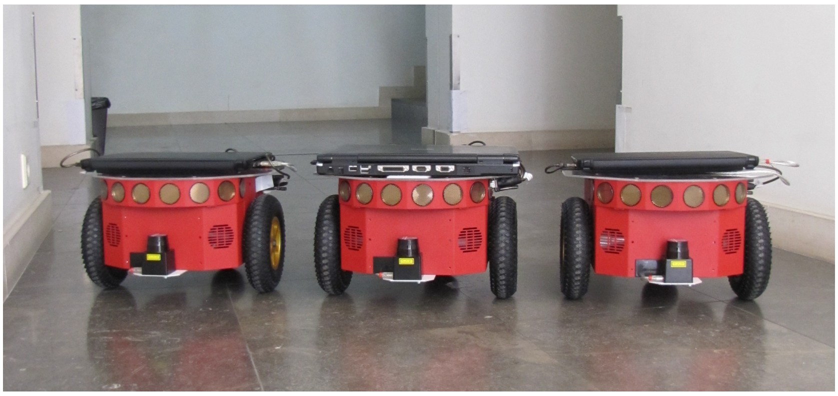


Figure 4.12: Robots used in the experiments.

4.3.2 Initial Experiments

All 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. 4.11 shows a few snapshots of the corridors of the ISR and the extracted topological map on top of the 67.85 × 26.15 meters environment, which was obtained using the algorithm in [Portugal and Rocha, 2012a]. 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 4.2.3.

A team of three Pioneer-3DX robots (*cf.* section 4.3.1), equipped with an

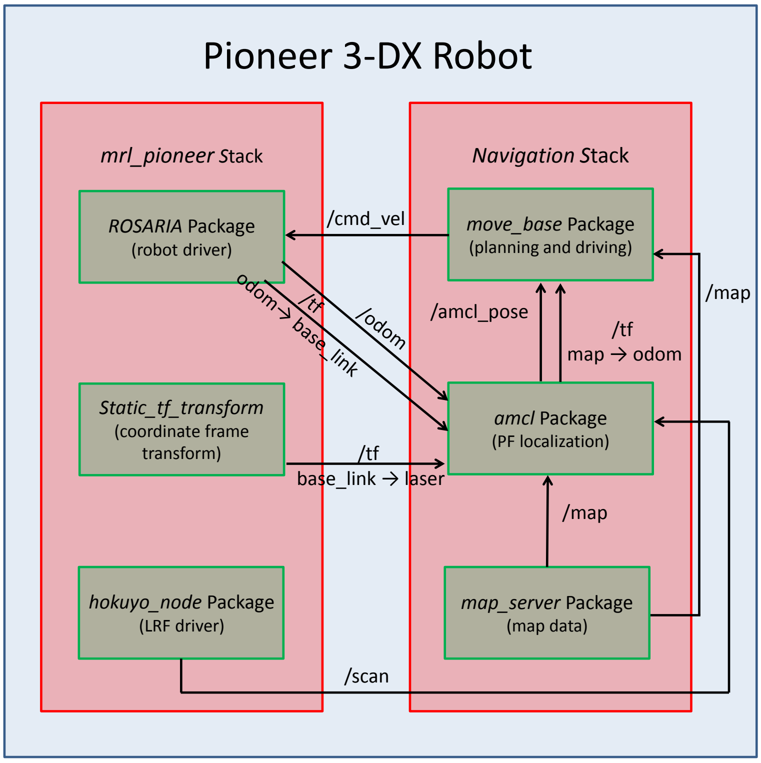

Pioneer 3-DX Robot
mrl_pioneer Stack
Navigation Stack
ROSARIA Package (robot driver)
move_base Package (planning and driving)
/cmd_vel
/map
/amcl_pose
/tf map → odom
odom → base_link
/tf
/odom
Static_tf_transform (coordinate frame transform)
amcl Package (PF localization)
/tf base_link → laser
/map
hokuyo_node Package (LRF driver)
map_server Package (map data)
/scan

Figure 4.13: Overview of the ROS system running on each robot.

Hokuyo URG-04LX-UG01 laser in the front and a laptop on top was used, as seen in Fig. 4.12. 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. The ROS architecture running inside each robot is depicted in Fig. 4.13. All robots are limited to a 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*[26] 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

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

Table 4.2: Experiments with 1 to 3 Robots (all values in seconds).

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

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.

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, $\texttt{max}(\overline{\mathcal{I}_\mathcal{V}})$.

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

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 Kms.

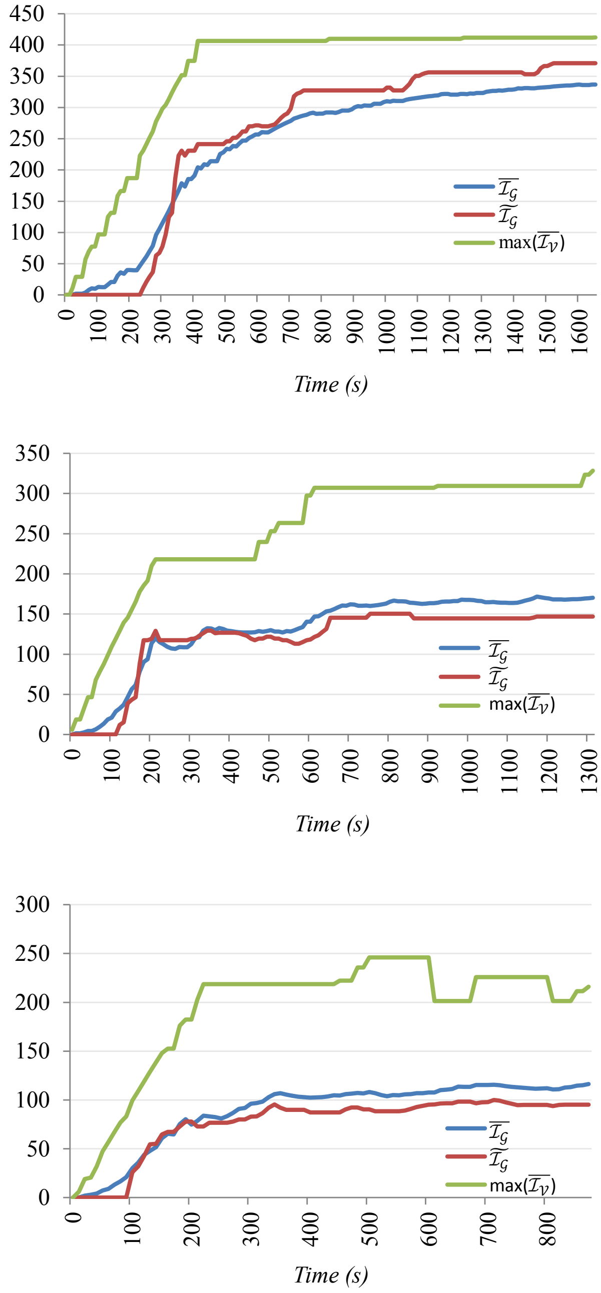


Figure 4.14: Evolution of the idleness along time a) with 1 robot, b) with 2 robots and c) with 3 robots.

Table 4.2 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 team size, 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 average idleness, $\mathtt{max}(\overline{\mathcal{I}_{\mathcal{V}}})$, 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 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 team size of 2 and 3 robots, respectively[27].

Figure 4.14 shows the evolution of the idleness in three different experiments with 1 to 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.

4.3.3 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 team size. 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.

In order to test the approach with greater team size 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.

Three trials were conducted with a total of 6 agents comprising 3 physical robots and 3 simulated ones. Three additional trials were performed with a team

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

Table 4.3: Experiments with 6 and 9 Robots (all values in seconds).

team size	$\overline{\mathcal{I}_{\mathcal{G}}}$	$\texttt{max}(\overline{\mathcal{I}_{\mathcal{V}}})$	$\widetilde{\mathcal{I}_{\mathcal{G}}}$	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

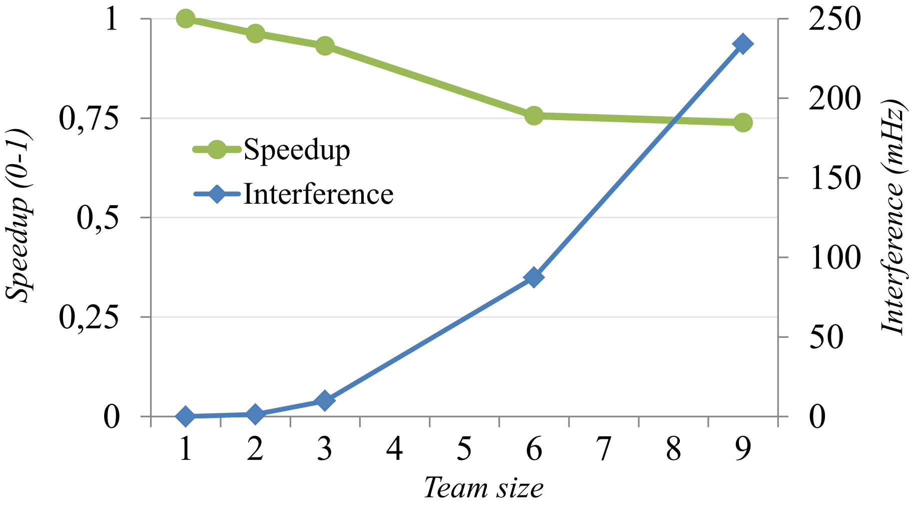


Figure 4.15: Interference and Speedup against team size.

size of 9, comprising 3 physical robots and 6 simulated ones. Similarly to [Iocchi et al., 2011], the software layer is used unchanged both on real robots and in simulation.

Results in Table 4.3 show that the overall values of $\overline{\mathcal{I}_{\mathcal{G}}}$, $\texttt{max}(\overline{\mathcal{I}_{\mathcal{V}}})$, $\widetilde{\mathcal{I}_{\mathcal{G}}}$, σ and τ are within the expected, following the trend shown in the cases of two and three robots.

Fig. 4.15 presents the speedup chart using different team sizes. It can be seen that speedup and interference are negatively correlated, since the system enters progressively in sublinear performance with team size, 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 team size. In addition, it is also illustrated that the individual contribution of each robot, as team size 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 4.2.1.

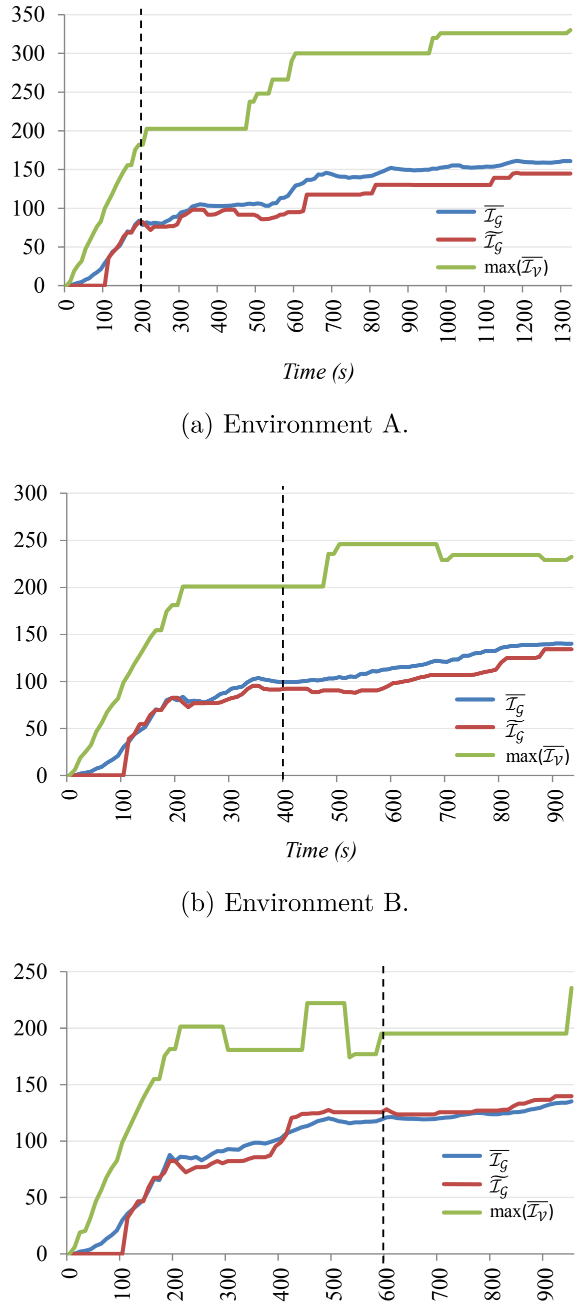
4.3.4 Fault-Tolerance

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

Generally, it can be seen in Table 4.4 that the results obtained in the first experiment resembles 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.



(a) Environment A.

(b) Environment B.

(c) Environment C.

Figure 4.16: Evolution of the idleness along time in experiments with robot failures. a) Failure at 200s. b) Failure at 400s. c) Failure at 600s.

Table 4.4: Experiments with 3 robots with failure of a robot in different instants of time (all values in secs).

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

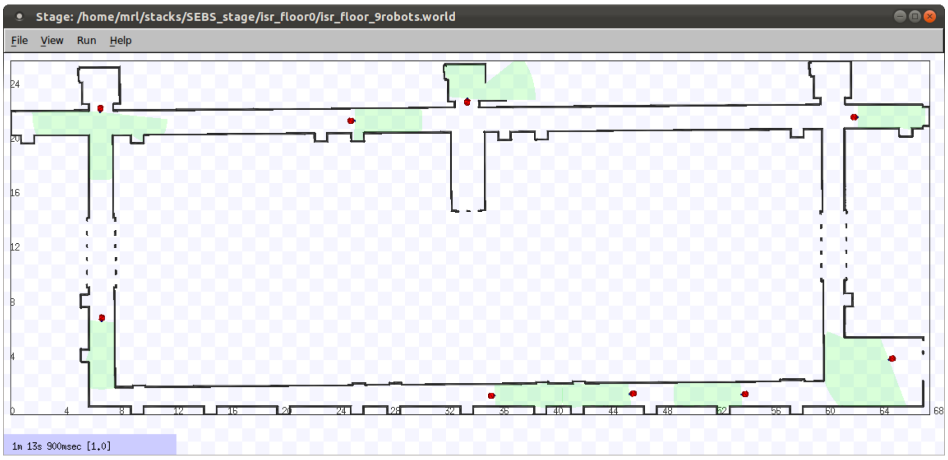
Analyzing now the influence of the failures in the evolution of the results, one can verify that in all three cases, when the failure occurs, the values of $\overline{\mathcal{I}_{\mathcal{G}}}$ and $\widetilde{\mathcal{I}_{\mathcal{G}}}$ increase after a while, which is particularly visible in Fig. 4.16a and Fig. 4.16b. These results prove the robustness of the system, enabling graceful degradation, as long as one robot remains operational.

4.4 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 4.4.1. Additionally, these simulations are seized by introducing different error rates in multi-robot communication in order to analyze how team performance is affected. This is discussed in 4.4.2.

4.4.1 Simulation Realism

In the experimental validation of the techniques (*cf.* section 4.2), it was shown that both strategies presented in this chapter 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



Stage: /home/mrl/stacks/SEBS_stage/isr_floor0/isr_floor_9robots.world

Figure 4.17: Snapshot of a simulation in the “ISR-Floor0” environment with a team of 9 robots.

potential of employing these systems in the real-world. In this section, the tests conducted in section 4.3 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 results drawn from simulation tests are from those obtained with the real robots, thus demonstrating how realistic 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. Figure 4.17 illustrates a snapshot of a simulation with 9 virtual robots in the environment and in Table 4.5 the new simulation results are presented.

Generally, the results in Table 4.5 show close resemblances to those in Tables 4.2 and 4.3. In fact, the difference in performance (in terms of $\overline{\mathcal{I}_{\mathcal{G}}}$) between simulated and real results is $\simeq 3.6\%$, which is remarkably low. Nevertheless, the difference in performance is more noticeable with larger team sizes, especially with 9 robots, which suggests that agents in simulations are less affected by multi-robot interference.

Additionally, the values of $\overline{\mathcal{I}_{\mathcal{G}}}$, $\mathtt{max}(\overline{\mathcal{I}_{\mathcal{V}}})$, $\widetilde{\mathcal{I}_{\mathcal{G}}}$, σ and τ tend to be marginally

Table 4.5: Simulation experiments in the "ISR-Floor0" environment (all values in seconds).

team size	$\overline{\mathcal{I}_\mathcal{G}}$	$\texttt{max}(\overline{\mathcal{I}_\mathcal{V}})$	$\widetilde{\mathcal{I}_\mathcal{G}}$	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

lower in simulations and the variance between each different trial with the same configuration is inferior. Therefore, 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 10Hz, which explains why the τ values only have one decimal place.

4.4.2 Influence of Communication Errors

The models proposed to solve the MRPP in section 4.1 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 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 keeps 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 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 Figure 4.18 presents an overview of the simulation results with communication failures, using team sizes 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 team sizes, is evident in the bars for $\xi = 100\%$:

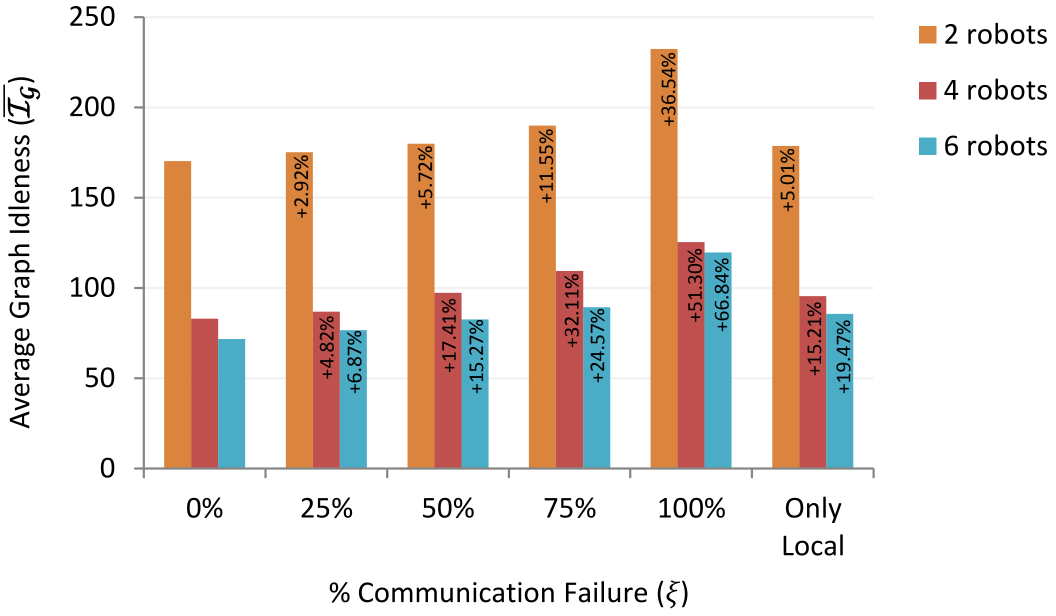


Figure 4.18: Influence of Communication Failures in Team Performance.

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 team sizes. The performance obtained using only local communication closely resembles to that obtained when dropping 50% of the messages for all team sizes.

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%). Obviously, 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.

On a final note, being a research member of the R&D CHOPIN[28] project, the author has also had the possibility to apply the SEBS algorithm in one of the project's work described in [Couceiro et al., 2013], which addresses distributed architectures of cooperation for multi-robot teams in search and rescue (SaR) missions. In the reconnaissance phase of the SaR mission, a fleet of cooperative mobile robots searches thoroughly the catastrophic scenario, while performing cooperative exploration and mapping of the environment, thus signalizing the presence of victims and possible evolution of the disaster. Afterwards, in the rescuing phase, the mobile robotic team covers the scenario, identifying the location of any remaining victims and the possible evolution of the disaster, *e.g.*, monitor the fire evolution in a firefighting operation and transmit this information to the command center of operations (CCO).

In the latter phase, the SEBS algorithm has been employed to inspect previously defined locations in the urban SaR scenario and results with different team configurations have confirmed that performance increases with the number of robots in the team and the communication range of each robot.

4.5 Summary

In this chapter, 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 coordinated, effective and distributed cooperative patrolling. Breaking away from conventional techniques, this work goes beyond classical approaches that rely on pre-computed cyclic routes or partition schemes for multi-robot patrolling, giving the robots the autonomy to decide locally and sequentially their actions without requiring a central planner. This way, it was shown that agents can coordinate effectively, using distributed communication, independently of the number of robots in the team.

The State Exchange Bayesian Strategy (SEBS) is an extension of the Greedy

[28] CHOPIN stands for **C**ooperation between **H**uman and r**O**botic teams in catastro**P**hic **IN**cidents. http//chopin.isr.uc.pt

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 author's belief that research in this field should be more oriented towards effective solutions with applicability in the real-world. The results obtained herein have demonstrated that the approach is able to scale to a high number of robots, being robust to robot failures. Experiments were conducted using real robots and mixed teams of both virtual and real agents, in a a large indoor infrastructure, proving the effectiveness of the approach and the potential to use it in the real-world. Moreover, an important contribution was the assessment of the realism of Stage/ROS simulations and the analysis of how communication errors affect the system's performance.

In the next chapter, the formulated model is extended 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. This is done by updating the prior term at each step and use a reward-based learning technique to adapt the likelihood function according to the evolution of visits at each vertex of the graph.

Chapter 5

Adopting Bayesian Learning to Promote Adaptive Patrol

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

Considerable scientific work presented recently span across the boundaries of Robotics and AI, and in the particular case of MRS, these are commonly limited to verification through simulations or controlled test scenarios. Some exceptions include works such as Iocchi *et al.* [Iocchi et al., 2011] and Pippin *et al.* [Pippin et al., 2013], among others, which have employed teams of robots in real-world scenarios. Successful examples of solutions that have proliferated in public places include automated guided vehicles (AGVs)[29] and the Santander Interactive Guest Assistants (SIGA)[30] and teams of multiple robots have been increasingly used in

[29] http://www.cybercars.org

[30] http://www.ydreamsrobotics.com/projects

military and security applications, taking advantage of space distribution, parallelism, task decomposition and redundancy [Parker, 2008].

Security applications are a fundamental task with unquestionable impact on society. Combining progress witnessed in AI with the technological evolution observed in the last decades, it becomes clear that intelligent and adaptable robot assistance can be a valuable resource in surveillance missions.

In this chapter, a distributed and adaptive multi-robot solution for indoor patrol is proposed, which extends the work presented in chapter 4. This new strategy rectifies weaknesses previously identified, by describing a probabilistic multi-robot patrolling strategy, where a team of concurrent learning agents adapt their moves to the state of the system at the time, using Bayesian decision based on the robot's accumulated experience and distributed intelligence. When patrolling a given site, each agent evaluates the context and adopts a reward-based learning technique that influences future moves.

Firstly, extensive simulation experiments are conducted, and afterwards the approach is validated in a large real-world environment. It is proven that: the approach can be applied in any generic environment, independent of its topology; withstands failures in robotic patrol units; can be performed with heterogeneous robots; and accomplishes exceeding performance. Therefore, its potential is shown as a solution for real-world MRS. Additionally, it is shown that the approach presents superior results when compared to existing state-of-the-art methods and outperforms previous strategies described in this thesis.

In the next section, a Bayesian model is formulated in order to solve the MRPP using a probabilistic framework. Afterwards, the model is tested in simulations to enable the evaluation and comparison with existing patrolling techniques and study the effect of *look-ahead* in the mission. Then, in section 5.3 hardware experiments are conducted and the proposed approach is placed in comparison with a near-optimal TSP cyclic strategy in a real-world scenario. In the end, section 5.4 summarizes relevant concluding thoughts. The work in this chapter is partially covered in [Portugal et al., 2013], [Portugal and Rocha, 2013b] and [Portugal and Rocha, 2014a].

The contribution of the work described in this chapter is listed below:

- A new probabilistic, distributed and scalable approach to solve the MRPP is described, whose effectiveness is attested in the experiments conducted;
- Bayesian decision is employed in the context of the MRPP for the first time, as far as the author's knowledge goes, providing enhanced adaptability to the system;
- Advantages of using the proposed algorithm become evident by evaluating its performance and effectiveness against several previous approaches using realistic simulations in Stage/ROS;
- The approach is implemented in a large indoor scenario with up to six physical robots and compared with the classical multi-agent TSP approach, which is theoretically optimal using one robot and nearly-optimal for generic teams of robots;
- Fault-tolerance and scalable behaviors are verified in real experiments, as well as the possibility to use the approach in heterogeneous teams of robots.

5.1 Bayesian Model for Multi-Robot Patrolling

In the previous chapter of this thesis, simple preliminary Bayesian-based techniques to tackle the MRPP were studied. Even though the results obtained were satisfactory, two main drawbacks were identified: a uniform prior distribution was adopted, assuming that all decisions were equiprobable; and the likelihood distributions were immutable, representing a fixed function of random variables. In this work, robots are endowed with increased intelligence, since the previous Bayesian models are extended with likelihood reward-based learning and continued prior update.

Once again, the model represents the decision of moving from one vertex of the graph to another. For β neighbors of the current vertex v_0, where $\beta = deg(v_0)$, the model is applied β times. Each decision is considered independent and the agents have the ability to choose the action which has the greatest expectation of utility, weighted by the effects of all possible actions. Consequently, each robot's patrol route is built progressively, at each decision step, adapting to the system's needs, *i.e.*, aiming at minimizing $\overline{\mathcal{I}_{\mathcal{G}}}$. In the following subsections, more details on the Concurrent Bayesian Learning Strategy (CBLS) to solve the MRPP are presented.

5.1.1 Distribution Modeling

As stated before, when reaching a vertex v_0 of the navigation graph $\mathcal{G}$, each robot is faced with a decision stage, where it must decide the direction it should travel next (*cf.* Fig. 5.1). For that reason, two fundamental random variable are defined. The first one is the same as in section 4.1, eq. 4.1, being boolean and simply representing the act of moving (or not) to a neighbor vertex:

$$move_i = \{true, false\}, \tag{5.1}$$

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

$$\theta_{0,i} \in \boldsymbol{\theta}; \quad 0, i \in \mathbb{N}_0; \quad \text{and} \quad |\boldsymbol{\theta}| = 2|\mathcal{E}|. \tag{5.2}$$

Note that graph $\mathcal{G}$, which describes the environment to patrol, is an undirected graph, as outlined in section 2.4.1, where an edge $e_{j,k}$ represents a connection from v_j to v_k and vice versa. $e_{j,k}$ has an edge cost or weight $|e_{j,k}| = |e_{k,j}|$, given by the distance between the two vertices. Nevertheless, the term "arc" instead of "edge" is used intentionally, since it implies a direction of traveling. In a situation where an agent is at v_j, it will look for the suitability of traveling to v_k, given by $\theta_{j,k}$. Under those circumstances, the suitability of traveling in the opposite direction is not relevant, thus $\theta_{j,k} \neq \theta_{k,j}$. As a consequence, the set $\boldsymbol{\theta}$ has a population of

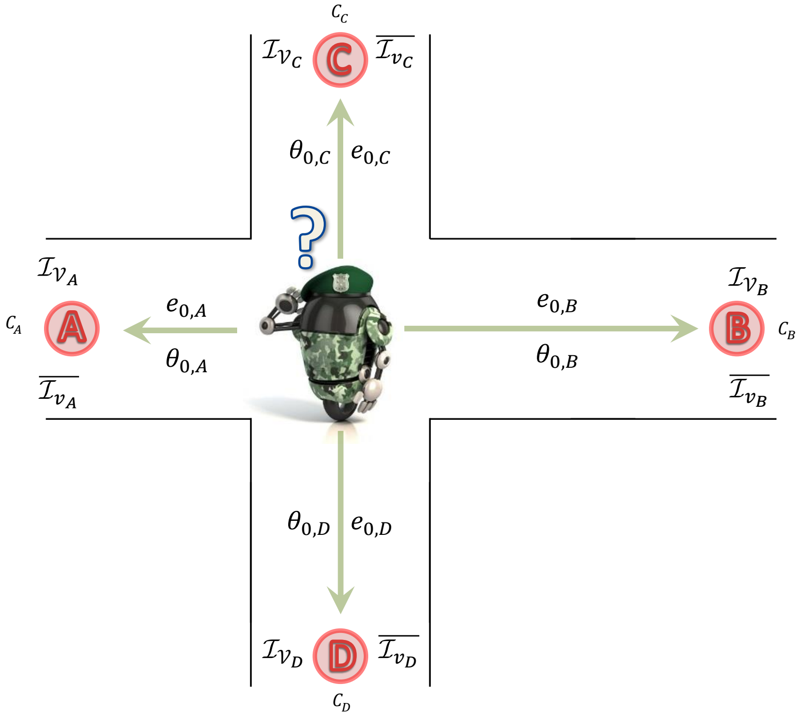


Figure 5.1: Illustration of a patrolling decision instance. In this example, the robot may choose to visit one out of four neighbor vertices. Considering a generic neighbor v_x, $\mathcal{I}_{v_x}$ is the instantaneous idleness of v_x; $\overline{\mathcal{I}_{v_x}}$ is the average idleness of v_x; $e_{0,x}$ gives the edge weight between the current vertex and v_x; C_x gives the number of visits to v_x and $\theta_{0,x}$ represents the arc strength of traveling from v_0 to v_x.

$2|\mathcal{E}|$, where $|\mathcal{E}|$ is the cardinality of the set of edges $\mathcal{E}$ of $\mathcal{G}$, and informally, higher values of *arc strength* lead to the edge being traversed more often in the specified direction.

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

$$P(move_i|\theta_{0,i}) = \frac{P(move_i)P(\theta_{0,i}|move_i)}{P(\theta_{0,i})}. \tag{5.3}$$

The posterior probability $P(move_i|\theta_{0,i})$ is estimated via Bayesian inference from the prior $P(move_i)$ and likelihood $P(\theta_{0,i}|move_i)$ distributions and the denominator term is regarded as a normalization factor. The prior represents the belief obtained from analyzing past data, and unlike the preliminary model formulated in chapter 4, this new model considers constant prior term update. In the MRPP, prior information about each vertex is encoded in the average idleness $\overline{\mathcal{I}_{v_i}}$ of a vertex v_i given by (2.3). Therefore, $P(move_i)$ is defined as:

$$P(move_i) = \frac{\overline{\mathcal{I}_{v_i}}}{\sum\limits_{k=1}^{|\mathcal{V}|} \overline{\mathcal{I}_{v_k}}}, \tag{5.4}$$

thus decisions of moving to vertices with higher values of average idleness have intuitively higher probability. During the patrol mission, robots are continuously visiting new places and the $\overline{\mathcal{I}_V}$ values change over time. Each agent computes these values internally by tracking its own visits to $\mathcal{V}$ and communicating to other teammates when they arrive to a new vertex. In order to make an informed decision, at each decision step, the agent updates the prior information through (5.4), just before adopting (5.3) to obtain a degree of belief of moving to a neighbor vertex v_i.

In addition to the prior distribution, it is also necessary to define the likelihood through a statistical distribution to model the *arc strength* $\theta_{0,i}$. In the patrolling problem, agents must visit all $v_i \in \mathcal{G}$, thus, theoretically, assigning a uniform value for every arc would not be unreasonable. However, in such a dynamic system, where the number of visits to different locations in the environment is permanently evolving, it is usually advantageous to avoid traversing certain edges at a given time and favoring the use of others, in order to improve performance. Furthermore, task effectiveness is strongly related to the environment topology.

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

5.1.2 Multi-Agent Reward-Based Learning

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

The concept of delayed reward with a 1-step horizon model is explored. Each agent chooses an action of moving from v_0 to a neighbor v_i, based on (5.3). After reaching v_i, the information on its neighborhood has changed, namely the instantaneous idlenesses have been updated, *i.e.*, $\mathcal{I}_{v_i}(t) = 0$ and $\mathcal{I}_{v_0}(t) > 0$. Through information observed after making the move, a reward-based mechanism is used to punish or benefit the arcs involved in the decision to move from v_0 to v_i. This influences future moves starting in v_0, by introducing a bias towards arcs which ought to be visited ahead in time.

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

$$H(move_i|\boldsymbol{\theta}) = -\sum_{i=1}^{\beta} P(move_i|\theta_{0,i}) \log_2(P(move_i|\theta_{0,i})), \tag{5.5}$$

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

$$\mathcal{H}(move_i|\boldsymbol{\theta}) = \frac{H(move_i|\boldsymbol{\theta})}{\log_2(\beta)}. \tag{5.6}$$

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

$$\gamma_{0,i} = \mathcal{S}_{0,i}(C_i, \mathcal{I}_{v_i}(t)) \cdot (1 - \mathcal{H}(move|\theta)), \tag{5.7}$$

$$\text{with:} \quad \mathcal{S}_{0,i} \in \{-1, 0, 1\}. \tag{5.8}$$

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

$$\zeta_i = \frac{C_i}{\deg(v_i)}. \tag{5.9}$$

This is used in the punish/reward procedure given that vertices with higher degree are naturally more visited than vertices with lower degree, being often traversed to reach isolated vertices that tend to have a lower number of visits. The rules for assigning the sign $\mathcal{S}_{0,i}$ of the rewards are given by:

$$\mathcal{S}_{0,i} = \begin{cases} -1, & \textbf{if} \quad (\beta > 1) \wedge (\underset{j \in N_G(v_0)}{\texttt{argmax}} \; \zeta_j = i) \wedge (|\underset{j \in N_G(v_0)}{\texttt{argmax}} \; \zeta_j| = 1); \\ -1, & \textbf{if} \quad (\beta > 1) \wedge (\underset{j \in N_G(v_0)}{\texttt{argmax}} \; \zeta_j = i) \wedge (|\underset{j \in N_G(v_0)}{\texttt{argmax}} \; \zeta_j| > 1) \wedge \\ & \quad (\underset{j \in N_G(v_0)}{\texttt{argmin}} \; \mathcal{I}_{v_j}(t)) = i); \\ 1, & \textbf{if} \quad (\beta > 1) \wedge (\underset{j \in N_G(v_0)}{\texttt{argmin}} \; \zeta_j = i) \wedge (|\underset{j \in N_G(v_0)}{\texttt{argmin}} \; \zeta_j| = 1); \\ 1, & \textbf{if} \quad (\beta > 1) \wedge (\underset{j \in N_G(v_0)}{\texttt{argmin}} \; \zeta_j = i) \wedge (|\underset{j \in N_G(v_0)}{\texttt{argmin}} \; \zeta_j| > 1) \wedge \\ & \quad (\underset{j \in N_G(v_0)}{\texttt{argmax}} \; \mathcal{I}_{v_j}(t)) = i); \\ 0, & \textbf{otherwise}. \end{cases} \tag{5.10}$$

$$\text{with:} \quad \beta = \deg(v_0) = |N_G(v_0)|, \tag{5.11}$$

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

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

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

- $\mathcal{S}_{0,i} = 0$, in every other situation that differs from the above.

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

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

$$\forall \theta_{0,i} \in \theta,\, \theta_{0,i}(t_0) = \kappa. \tag{5.12}$$

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

$$\theta_{0,i}(t) = \theta_{0,i}(t-1) + \gamma_{0,i}(t). \tag{5.13}$$

Note that the larger the value of κ is set in (5.12), the less immediate influence the rewards received will have on $\theta_{0,i}$. In all experimental tests conducted in this work, $\kappa = 1.0$ was used. This reward-based procedure is expected to make the values of $\theta_{0,i}$ fluctuate as time goes by, informing robots of moves which are potentially more effective, but keeping in mind that robots must visit all vertices v_i in the patrolling mission.

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

$$P(\theta_{0,i}|move_i) = \frac{\theta_{0,i}}{\sum_{j}^{|\mathcal{E}|}\sum_{k}^{|\mathcal{E}|}\theta_{j,k}}, \tag{5.14}$$

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

5.1.3 Decision-Making and Multi-Agent Coordination

Having described how agents learn their likelihood distribution, it is necessary to address agent coordination to completely characterize the CBLS solution for patrolling tasks presented in this section. Being a concurrent learning approach, each agent is adapting its behavior via its own learning process and has no control or knowledge of how other agents behave nor their internal state, *i.e.*, they do not know their teammates' likelihood distribution $P(\theta_{0,i}|move_i)$ and cannot predict their moves. This allows the reduction of complexity of the problem, however it is necessary to guarantee the coordination of robots.

In collective missions with a common goal, multi-agent coordination plays a fundamental role in the success of the mission. Particularly in this context, it is highly undesirable that agents move to the same positions. The asynchronous and distributed communication system that is used to inform teammates of the current vertex v_0 is therefore augmented with the information of the vertex v_i chosen for the next move.

This way, simply by sending and receiving messages from its teammates, each robot can update the information about the state of the system, namely the idleness values, and decide its moves taking that information into account, as well as its progressively acquired experience. When agents are close by, they can coordinate by inspecting if a teammate has already expressed intention to move to a given vertex v_i in its local neighborhood and if so, remove it from its decision.

Finally, the decision-making process of the agent consists of choosing the move to the neighbor vertex with the *maximum a posteriori* (MAP) probability:

$$\begin{aligned} move_{MAP} =& \underset{move}{\texttt{argmax}} \; P(move_i|\theta_{0,i}) \\ =& \underset{move}{\texttt{argmax}} \; P(move_i) \cdot P(\theta_{0,i}|move_i). \end{aligned} \tag{5.15}$$

5.1.4 Incorporating Vertex Look-ahead

Up to now, only agents that decide their moves upon knowledge on the local 1-step neighborhood have been considered. While it is true that expanding the search horizon increases the complexity of the problem, it is also true that it may lead

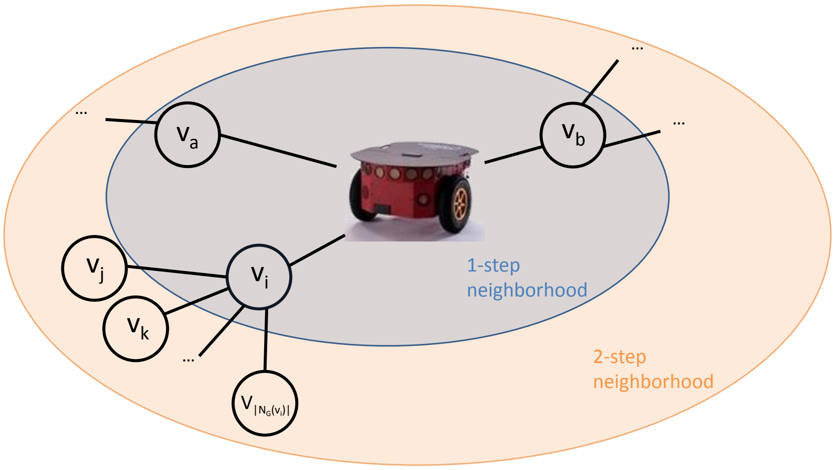


Figure 5.2: Illustration of the 2-step look-ahead in the navigation graph.

to a higher number of correct decisions, since agents deliberate with additional information.

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

Prior to making a decision, agents will now look not only for the average idleness of the vertices v_i in its immediate neighborhood, but also for the neighbors of these $(v_j, v_k, ...)$, as shown in Fig. 5.2, to find the maximum average idleness among them.

Accordingly, vertex look-ahead is incorporated in the prior distribution, by modifying (5.4):

$$P(move_i) = \frac{\omega \cdot \overline{\mathcal{I}_{v_i}} + (1-\omega) \cdot \mathtt{max}(\overline{\mathcal{I}_{N_G(v_i)}})}{\sum_{k=1}^{|\mathcal{V}|} \omega \cdot \overline{\mathcal{I}_{v_k}} + (1-\omega) \cdot \mathtt{max}(\overline{\mathcal{I}_{N_G(v_k)}})}, \tag{5.16}$$

$$\text{where:} \qquad \omega \in [0,1], \tag{5.17}$$

$$\mathtt{max}(\overline{\mathcal{I}_{N_G(v_i)}}) = \mathtt{max}\{\overline{\mathcal{I}_{v_j}}, \overline{\mathcal{I}_{v_k}}, ...\}, \tag{5.18}$$

$$v_j, v_k, ... \in N_G(v_i). \tag{5.19}$$

The factor ω assigns weights to the observations on the immediate neighborhood and on the 2-step neighborhood. When $\omega = 1.0$, no look-ahead is considered and (5.16) is equivalent to (5.4). The lower the ω value is, the higher the weight of observations beyond the 1-step neighborhood are. However, intuitively one should set ω for values above 0.5 (and below 1.0), because the information on the immediate 1-step neighborhood is always reliable and up-to-date when the decision is made, while the information on the 2-step neighborhood may change when the robot reaches v_i. Note also, that when reaching v_i, the robot has another decision instance and in rare cases may deliberate contrarily to the initial plan, in case the settings in its surroundings have changed in the meantime. The influence of the ω factor is analyzed in the preliminary simulation results presented in the next section, so as to establish an appropriate value to use in experiments with physical robots.

Having described the model for multi-robot patrol in this section, the pseudocode of the CBLS approach, running on each individual robot of the team, is presented in Algorithm 5.1.

Algorithm 5.1: Concurrent Bayesian Learning Strategy (CBLS).

```
while $true$ do
    add($v_k$ to $\pi_r$);                                        // current vertex $v_k$
    forall the $v_i \in N_G(v_k)$ and $v_i \notin$ intended by teammate do
        $P(move_i) \propto \omega \cdot \overline{\mathcal{I}_{v_i}} + (1-\omega) \cdot \mathtt{max}(\overline{\mathcal{I}_{N_G(v_i)}})$;    // prior
        $P(\theta_{k,i}|move_i) \propto \theta_{k,i}$;                      // likelihood
        $P(move_i|\theta_{k,i}) \propto P(move_i) \cdot P(\theta_{k,i}|move_i)$;     // posterior
    $H(move|\theta) \leftarrow - \sum_{i=1}^{|N_G(v_k)|} P(move_i|\theta_{k,i}) \log_2(P(move_i|\theta_{k,i}))$;
    $\mathcal{H}(move|\theta) \leftarrow H(move|\theta)/\log_2(|N_G(v_k)|)$;        // normalized entropy
    // choose neighbor of $v_k$ with highest posterior probability
    $v_{k+1} \leftarrow$ argmax($P(move_i|\theta_{k,i})$);
    send_msg(current: $v_k$, next: $v_{k+1}$);
    while move_robot to $v_{k+1}$ do
        read_msg(arrivals, intentions);
        update($\mathcal{I}_{\mathcal{V}}(t)$);
    forall the $v_i \in N_G(v_k)$ do
        compute $\mathcal{S}_{k,i}$;                              // eq. 5.10
        $\gamma_{k,i} \leftarrow \mathcal{S}_{k,i} \cdot (1-\mathcal{H}(move|\theta))$;        // reward
        $\theta_{k,i} \leftarrow \theta_{k,i} + \gamma_{k,i}$;                    // arc strength
    $v_k \leftarrow v_{k+1}$;
```

5.2 Simulation Results

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

In the preliminary simulation experiments, the main goal is to study the effect of the ω parameter in the vertex look-ahead method and to enable comparisons

Table 5.1: Final $\overline{\mathcal{I}_{\mathcal{G}}}$ values (in seconds) using CBLS in Environment A with different ω.

	CBLS (map A)		
R	$\omega = 1.0$	$\omega = 3/4$	$\omega = 2/3$
1	1445.45	1432.21	**1323.55**
2	707.92	**686.23**	701.73
4	402.28	**357.94**	371.55
6	261.36	**231.32**	251.07
8	188.11	**167.45**	181.53
12	168.85	**120.91**	133.92

Table 5.2: Final $\overline{\mathcal{I}_{\mathcal{G}}}$ values (in seconds) using CBLS in Environment B with different ω.

	CBLS (map B)		
R	$\omega = 1.0$	$\omega = 3/4$	$\omega = 2/3$
1	1249.45	**1110.66**	1199.97
2	575.06	557.00	**554.00**
4	284.88	**275.96**	283.35
6	197.33	**192.77**	194.25
8	143.36	142.74	**142.50**
12	108.22	96.48	**94.16**

Table 5.3: Final $\overline{\mathcal{I}_{\mathcal{G}}}$ values (in seconds) using CBLS in Environment C with different ω.

	CBLS (map C)		
R	$\omega = 1.0$	$\omega = 3/4$	$\omega = 2/3$
1	701.98	**684.42**	688.79
2	359.09	**355.65**	359.38
4	183.91	**175.40**	182.55
6	126.62	**121.36**	125.68
8	96.81	**90.25**	91.84
12	75.43	**63.00**	64.19

with other strategies in the literature. Hence, the three environments illustrated in Fig. 3.4, page 61, have been used once again to test the approach with different team sizes of $R = \{1,2,4,6,8,12\}$ robots. The three illustrated topologies present different algebraic connectivity being classified as: lowly (A), mildly (B) and highly (C) connected. In this work, these are again adopted to enable comparative analysis against previously described MRPP strategies, and the Stage 2D multi-robot simulator together with ROS, was adopted to implement simulations and guaranteeing the same exact conditions as in chapters 3 and 4.

Tables 5.1, 5.2 and 5.3 present the performance results of the distributed patrolling strategy, CBLS, described in this chapter, given by $\overline{\mathcal{I}_{\mathcal{G}}}$ in seconds, and using the environments of Fig. 3.4, with $\omega = 1.0$, $\omega = 3/4$ and $\omega = 2/3$. Results

Table 5.4: Final $\overline{\mathcal{I}_{\mathcal{G}}}$ values (in seconds) using different state-of-the-art strategies on Environment A.

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

Table 5.5: Final $\overline{\mathcal{I}_{\mathcal{G}}}$ values (in seconds) using different state-of-the-art strategies on Environment B.

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

prove the intuition that superior performance can be obtained with $0.5 < \omega < 1.0$, seeing as, in general, the best results were obtained for $\omega = 3/4$. It is also interesting to verify that the results without look-ahead ($\omega = 1.0$) are always inferior to those obtained with vertex look-ahead. This confirms that looking further beyond the local neighborhood has the potential to increase the number of correct decisions of each agent and improve team performance. In view of this, $\omega = 3/4$ is specified in the experiments with physical robots.

Using findings from the previous two chapters, Tables 5.4, 5.5 and 5.6 were built. In these tables, performance of 7 state-of-the-art approaches, including the preliminary models presented in section 4.1, with the same team size is compared using the $\overline{\mathcal{I}_{\mathcal{G}}}$ metric in the same three environments. For more details on the

Table 5.6: Final $\overline{\mathcal{I}_{\mathcal{G}}}$ values (in seconds) using different state-of-the-art strategies on Environment C.

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

various strategies and tests previously conducted, the interested reader should refer to sections 3.2 and 4.1, in this thesis.

As can be seen by inspecting all six tables, in general, CBLS clearly outperforms the other 7 algorithms. This is the case even when no look-ahead is considered. The difference is apparent especially for lowly (environment A) and mildly (environment B) connected environments. In the highly connected grid environment (map C), vertices have, in general, greater degree and there are usually many alternative routes to reach a given goal. This fact makes other state-of-the-art strategies perform better in this case than in less connected (more typical) environments. Nevertheless, CBLS presents a strong performance for smaller team sizes of 1, 2 and 4 robots; and outperforms the other strategies for larger team sizes of 6, 8 and 12 robots, which suggests that CBLS scales better than the rest of the strategies. Additionally, these results prove that CBLS is able to adapt to all kinds of environment topologies independently of team size.

Looking now at the results in more detail, the evolution of the likelihood function of a patrolling mission with two robots in environment A and $\omega = 3/4$ is illustrated in Fig. 5.3. Note that each robot apprehends a different distribution and has no control or knowledge on the internal state of its teammates. As expected, peaks in the histograms emerge with the increasing number of decisions. Despite that, it is also clear that values fluctuate around the initial uniform value (represented by the red line in each chart), which comes as a consequence of robots

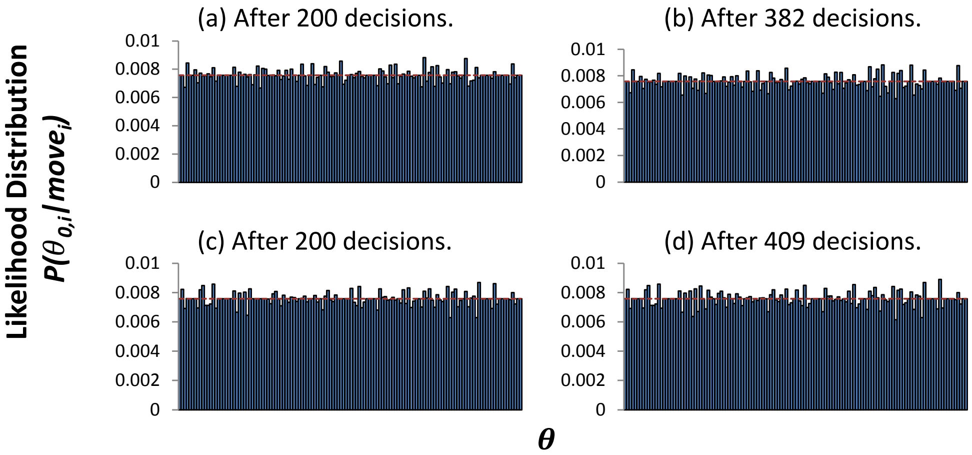


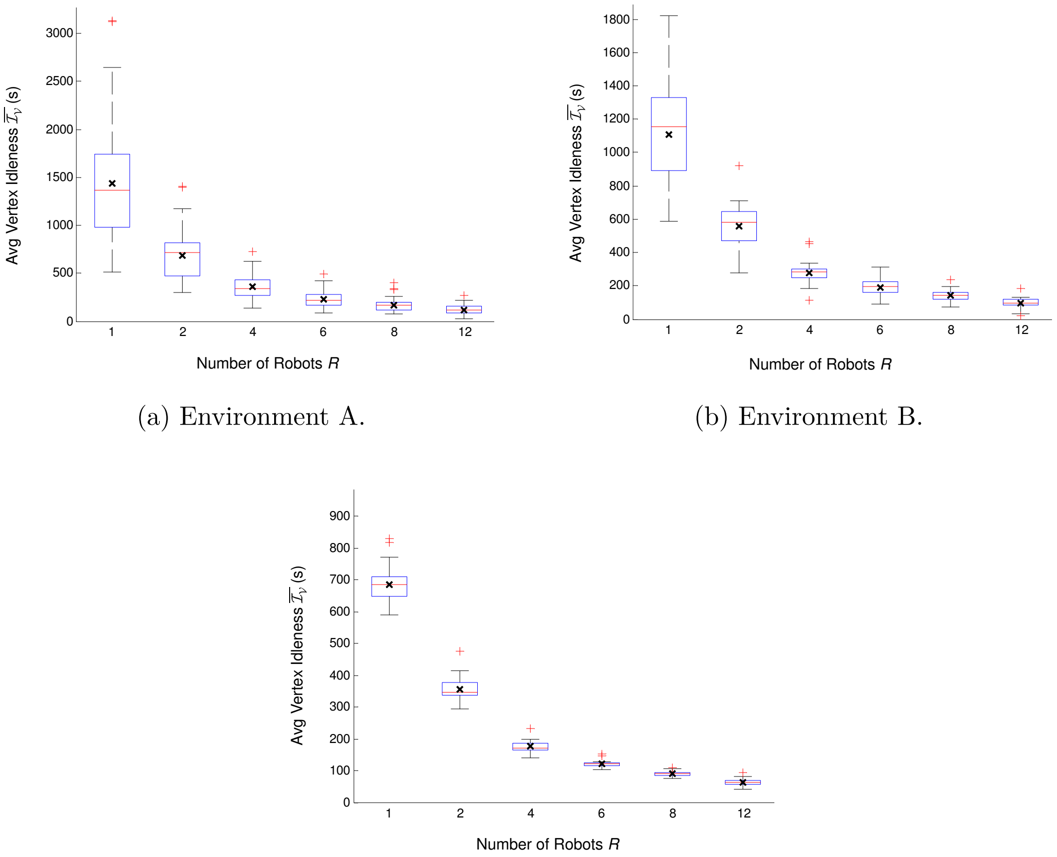
Figure 5.3: Evolution of the likelihood distribution in a mission with 2 robots in Environment A, with $\omega = 0.75$. a) robot 1 after 200 decisions; b) robot 1 at the end of the mission; c) robot 2 after 200 decisions; d) robot 2 at the end of the mission. The red line represents the initial distribution when the mission started.

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

Moving on to the performance of the algorithm, the boxplot charts in Fig. 5.4 represent the $\overline{\mathcal{I}_{\mathcal{V}}}$ values, in seconds, for each tested team size on all three maps. The average value is represented by a black cross, providing a generalized measure: the average graph idleness, $\overline{\mathcal{I}_{\mathcal{G}}}$ (*cf.* Eq. 2.6). The ends of the blue boxes and the horizontal red line in between correspond to the first and third quartiles and the median values of $\overline{\mathcal{I}_{\mathcal{V}}}$, respectively.

As expected, the idleness values decrease when the number of robots grow. Despite the increasing performance displayed by the CBLS approach, the individual contribution of adding more robots gradually reduces with team size. Group productivity will eventually converge with a large R. In theory, productivity should grow during size scale-up. However, spatial limitations increase the number of times the robots meet and beyond a given R, it is argued that they will spend more time avoiding each other than effectively patrolling on their own.

Another interesting aspect illustrated in the boxplot of Fig. 5.4 is the fact that the median $\widetilde{\mathcal{I}_{\mathcal{G}}}$ is close to the mean $\overline{\mathcal{I}_{\mathcal{G}}}$ in all configurations, being usually lower. This means that the $\overline{\mathcal{I}_{\mathcal{V}}}$ values are positively skewed, *i.e.*, most of the values are



(a) Environment A.

(b) Environment B.

(c) Environment C.

Figure 5.4: Overall results running CBLS with $\omega = 3/4$ and different team sizes.

below the average, $\overline{\mathcal{I}_\mathcal{G}}$, and, as a consequence, most outliers are above the third quartile.

Finally, on a more general note, visual inspection of the trajectories of robots using CBLS showed that prediction of patrolling routes is far from being straightforward, as opposed to most strategies presented in tables 5.4, 5.5 and 5.6. This stochastic behavior, together with the promising results obtained, proves the effectiveness of the approach and the potential to be applied in actual security systems with physical teams of robots, yielding unpredictable patrol trajectories.

5.3 Experiments with Physical Robots

In order for distributed intelligence systems to be useful in the real-world, it is essential to go beyond simulation experiments and validate convincing solutions that prove the reliability of the proposed strategy in more demanding scenarios.

In this section, the implementation of a system with teams of physical robots and corresponding experimental results in a large indoor scenario are discussed. Fully autonomous robots decide locally and sequentially their patrol routes according to the state of the system, as previously described, validating the CBLS distributed approach. Beyond the coordination which arises from the distributed communication of agents, it is also shown that the approach is scalable, robust to robot failures, *i.e.*, fault-tolerant, and supports heterogeneous agents with different speed profiles. CBLS is firstly compared to an algorithm called TSP Cyclic strategy, which finds the shortest tour on the graph similarly to solving the TSP [Fazli, 2013], but in the case of a non-complete graph as the one presented in the experiments. This classical algorithm is optimal for the single robot case and near optimal for multi-robot scenarios, being perfectly suited for a comparative analysis.

Once again, experiments were conducted in the floor 0 of the Institute of System and Robotics (ISR), in the University of Coimbra, as illustrated in Fig. 4.11, page 101. The resulting topology is a non-complete, connected and sparse graph, like most real-world environments.

When conducting experiments in the real-world, one must overcome noisy sensor readings, localization issues and even robot failures, which are usually ignored or not precisely modeled in simulation experiments. Therefore, a team of six Pioneer-3DX robots equipped with an Hokuyo laser in the front and a laptop on top was used, as seen in Fig. 5.5. Similarly as before, each laptop runs the ROS navigation stack using the AMCL algorithm for Localization, being responsible for controlling the robot's motion, which reaches speeds of up to 1 m/s. Inter-robot multimaster communication is provided using a distributed publish/ subscribe mechanism, as described in section 4.3.

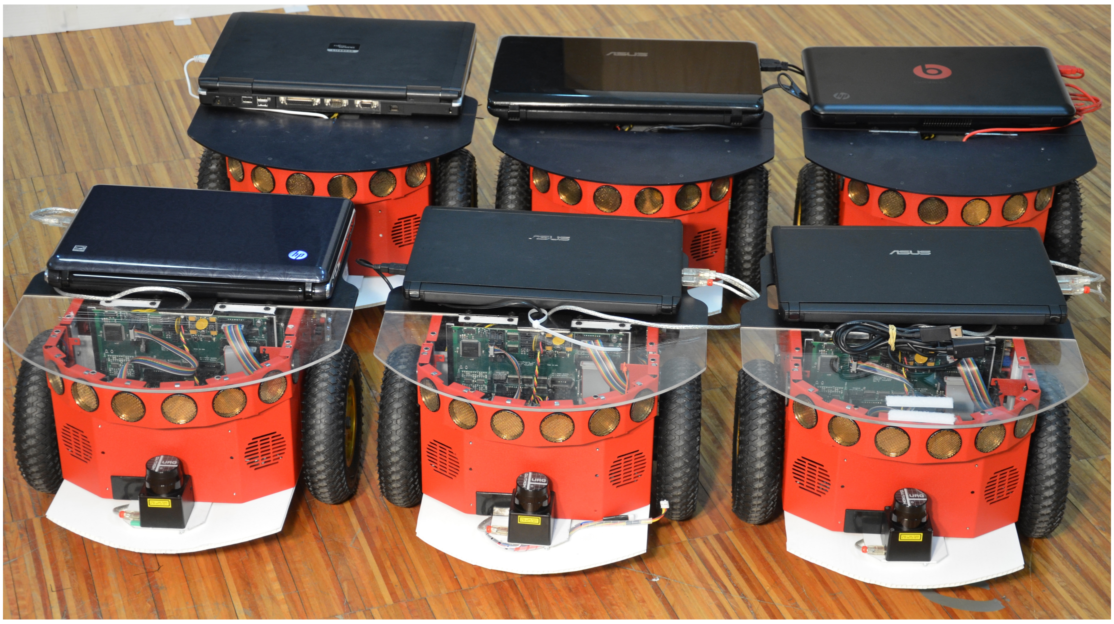


Figure 5.5: Robots used in the experiments.

The average graph idleness along time, $\overline{\mathcal{I}_{\mathcal{G}}}$, is used as the performance metric, and the median $\widetilde{\mathcal{I}_{\mathcal{G}}}$, standard deviation σ, and the maximum average idleness of a vertex along time, $\mathtt{max}(\overline{\mathcal{I}_{\mathcal{V}}})$ are also analyzed.

Firstly, experiments with teams from one to six robots were done, using the proposed strategy and comparing it with the TSP Cyclic Strategy. After this comparative analysis, CBLS was tested by adding one, and then, two agents to the patrolling mission with different speed profile, in order to prove that the approach adapts well even when heterogeneous teams of robots are used. Finally, to prove its robustness, experiments which included failures in the robots at different time instants were analyzed. Whenever CBLS is adopted in these experiments, $\omega = 3/4$ is considered. The total estimated distance traveled by the robots during the course of all experiments was 50 kms ($\simeq$ 31 miles). In addition, Each experiment finished after 4 complete patrolling cycles *i.e.*, after every $v_i \in \mathcal{G}$ has been visited at least 4 times. This stopping condition is adequate, as the $\overline{\mathcal{I}_{\mathcal{G}}}$ converges in all experiments.

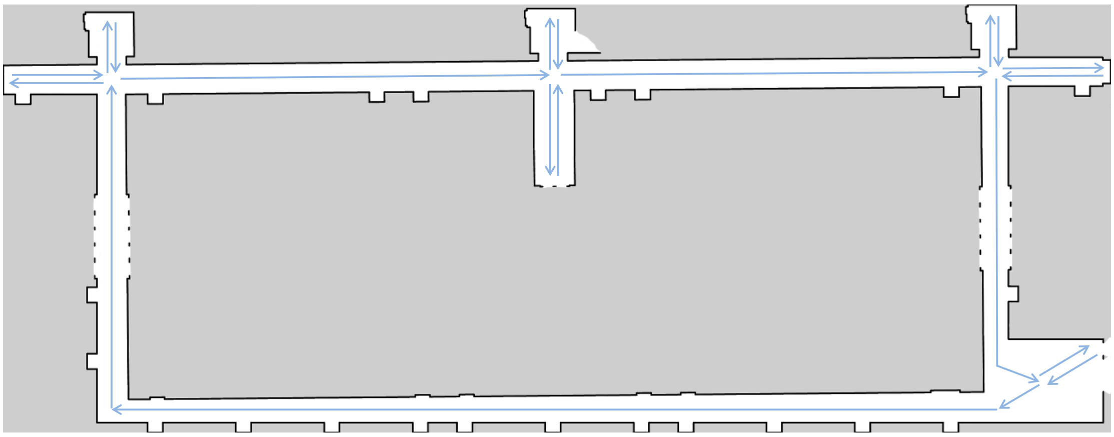


Figure 5.6: TSP cyclic trajectory in the “ISR-Floor0” environment.

5.3.1 Scalability and Performance

In this subsection, the scalability of the approach is addressed and its performance is analyzed when compared to the TSP Cyclic Strategy. As mentioned before, the latter is a recognized classical approach for multi-robot patrolling which finds the minimal tour that visits every vertex of the graph. This can be obtained using heuristic methods like the Chained Lin-Kernighan algorithm [Applegate et al., 2003]. The approach is theoretically optimal for the single robot case, while in the multi-robot case, robots are equally spaced along the tour, and the approach provides at least a near optimal solution. Note also that the deterministic route is computed *a priori* and offline, in contrast to the presented approach where agents have the autonomy to decide online their own moves. It also assumes strictly homogeneous robots without supporting communication between agents and robot failures, as opposed to CBLS.

In the particular case of the environment used in the experiments, the TSP cyclic tour is composed of the clear rectangular cycle pattern that exists in the map, with short detours to vertices with $deg(v_i) = 1$, as illustrated in Fig. 5.6.

Using the team of Pioneer-3DX robots, experiments with different team size from 1 to 6 robots were performed in the “ISR-Floor0” scenario. Each experiment was repeated 3 times and the results are presented in tables 5.7 and 5.8.

Table 5.7: Experiments with 1 to 6 robots using CBLS (all values in seconds).

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

Table 5.8: Experiments with 1 to 6 robots using TSP cycle (all values in seconds).

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

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

Perhaps the most interesting aspect of these results is the performance attained by the proposed strategy with mobile robot teams. In the experiments conducted, CBLS is generally slightly superior to TSP cycle, achieving differences in performance of up to $\simeq 3.5\%$. However, one cannot say that CBLS is superior to TSP

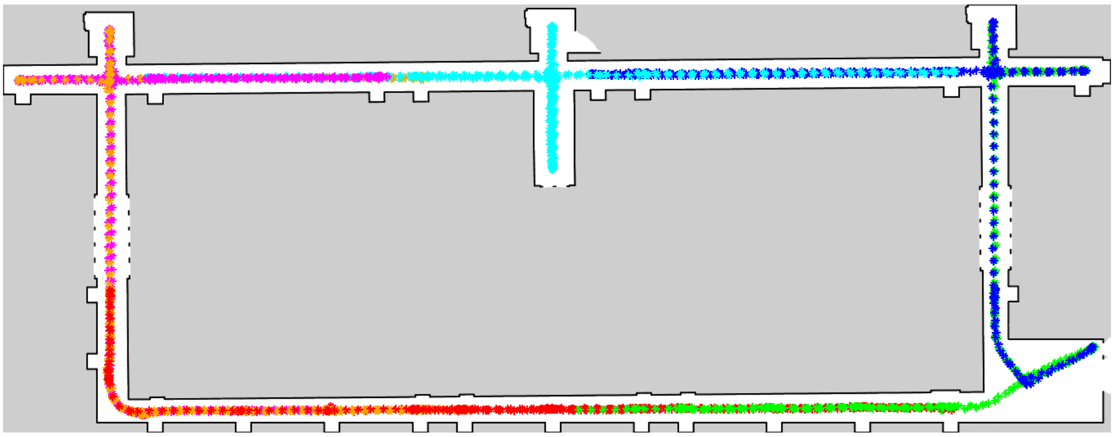


Figure 5.7: Robots positions during an experiments with 6 robots using CBLS. Implicit formation of dynamic regions.

for MRS without conducting more tests. In general, both strategies perform similarly in the collected results, with TSP presenting even superior results of around $\simeq 2.5\%$ in the 4 robots situation.

The tests in teams of growing number of robots show that the approach is able to scale well, performing in a near optimal way, similarly to TSP cycle. This is remarkable, considering the distributed and non-deterministic nature of the approach, as opposed to TSP cycle. A careful look at the behavior of the robots shows us that they tend to create dynamic regions where each agent patrols more often, as shown in Fig 5.7. As a result, there is little interference between agents[31]. In addition, since robots only share their current and future immediate goals, the bandwidth requirements are negligible even with larger teams.

It is clear in Table 5.8 that by making the robots follow the same global route, the TSP cycle always presents lower values of standard deviation, σ, when compared to CBLS, promoting more uniform visits to vertices. This is also suggested by the maximum average idleness of the vertices, $\texttt{max}(\overline{\mathcal{I}_{\mathcal{V}}})$, which are smaller than in CBLS.

Another interesting aspect observed in the experiments is the median value,

[31] A video of an experiment with 6 robots running CBLS is available at: http://isr.uc.pt/~davidbsportugal/videos/AIJournal/

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

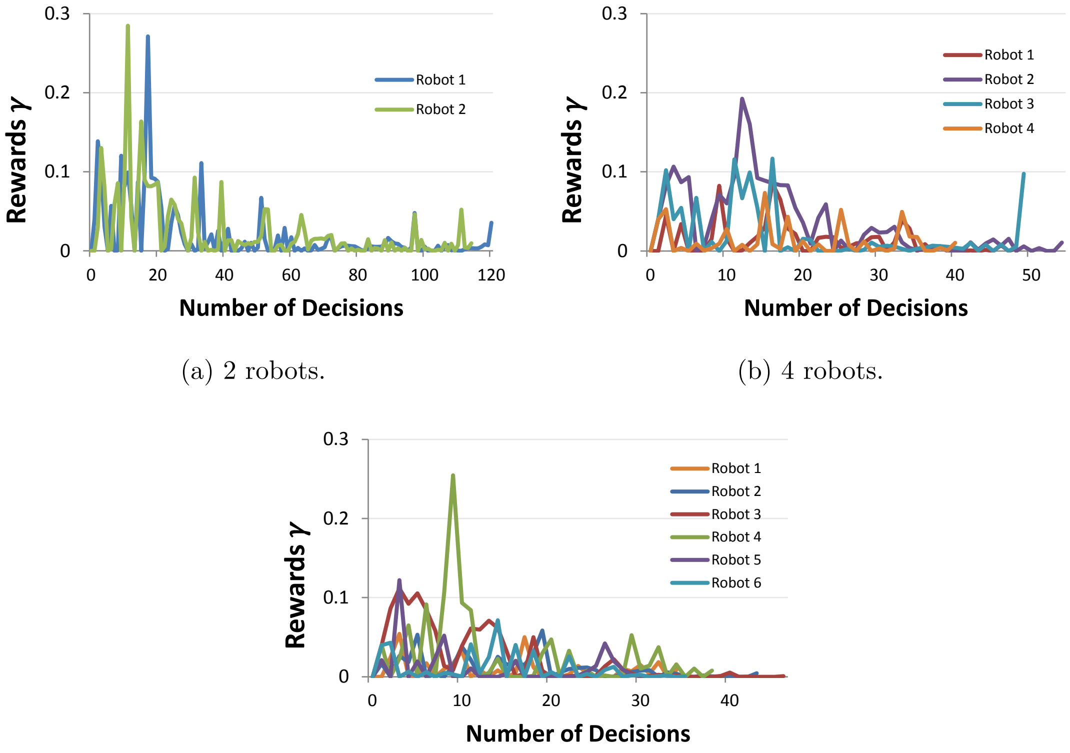
Looking more closely at CBLS in the experiments with physical robots, a descending trend is shown by the absolute reward values, given by the $(1-\mathcal{H})$ factor in (5.7), along the experiments with different configurations. Fig. 5.8 illustrates how these values evolve in missions with three different team sizes. Despite the occasional peaks, such values tend to decrease with the number of decisions. This is because, in general, as the system progresses, the $\overline{\mathcal{I}_{\mathcal{V}}}$ values of different vertices become more balanced and, as a consequence, the degree of belief in moving to distinct neighbors comes closer. In such situations, the closer the posterior probabilities are, the higher the entropy becomes, therefore the reward values descend gradually. The peaks observed are justified by situations where agents share nearby areas, temporarily perturbing the $\overline{\mathcal{I}_{\mathcal{V}}}$ values in the neighborhood of other agents. For that reason, peaks are more observable in larger teams.

5.3.2 Heterogeneous Teams

In this section, the aim is to further explore the scalability of the approach and test it with teams of heterogeneous robots. Being a distributed patrolling strategy, CBLS should support an arbitrary high team size.

Furthermore, being composed of concurrent learning agents, the team should not only adapt to the system's state but also to different robot profiles. Hence, virtual agents, running in the stage simulator, were added to the physical team, resulting in a mixed and interacting team of real and simulated robots, which communicate seamlessly.

Virtual agents have the same properties of the ones used in the simulation experiments. Thus, they travel slower than the physical robots. Three trials were conducted with a total of 7 agents composed by 6 physical robots and 1 simulated



(a) 2 robots.

(b) 4 robots.

(c) 6 robots.

Figure 5.8: Evolution of the absolute reward values along three experiments with different team size.

robot; and three more trials were performed with a team size of 8, composed by 6 physical robots and 2 simulated ones. Similarly as before (*cf.*, section 4.3.3), the software layer is used unchanged both on real robots and in simulation.

Results in Table 5.9 show that the overall values of $\overline{\mathcal{I}_{\mathcal{G}}}$, $\texttt{max}(\overline{\mathcal{I}_{\mathcal{V}}})$, $\widetilde{\mathcal{I}_{\mathcal{G}}}$ and σ are within the expected, following the trend shown in Table 5.7. The additional virtual agents are integrated into the remaining team and are able to interact with teammates. In addition, their contribution to the global performance is minor, as expected. This happens not only because of the progressive decrease of individual contribution of each robot as team size grows, but also due to lower speed at which these robots travel. The boxplot of Fig. 5.9 illustrates this trend.

It is noteworthy that incorporating robots that travel at different speeds with

Table 5.9: Experiments with a team of 6 physical robots using CBLS extended with one and two virtual agents (all values in seconds).

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

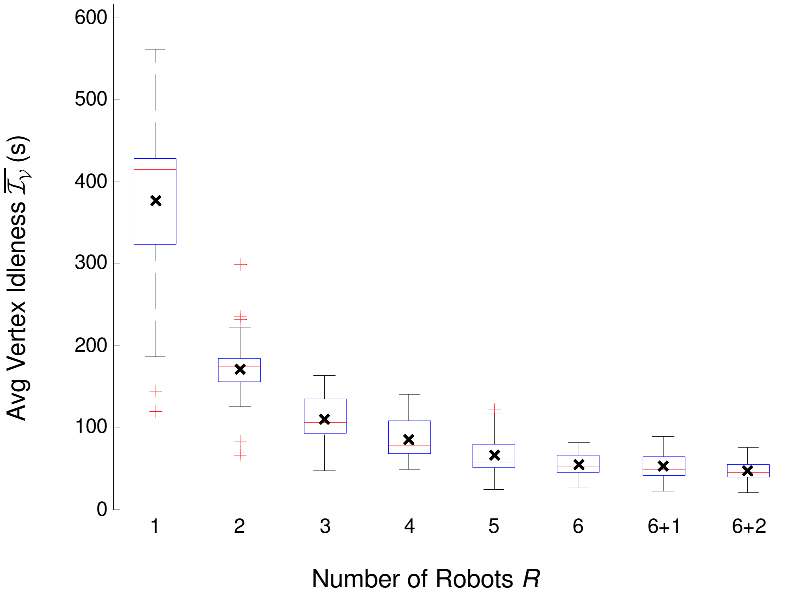


Figure 5.9: Overview of the results with teams of physical robots and mixed teams of real and virtual robots, using CBLS.

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

Table 5.10: Experiments with a team of 6 physical robots with one and two failures during the mission (all values in seconds).

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

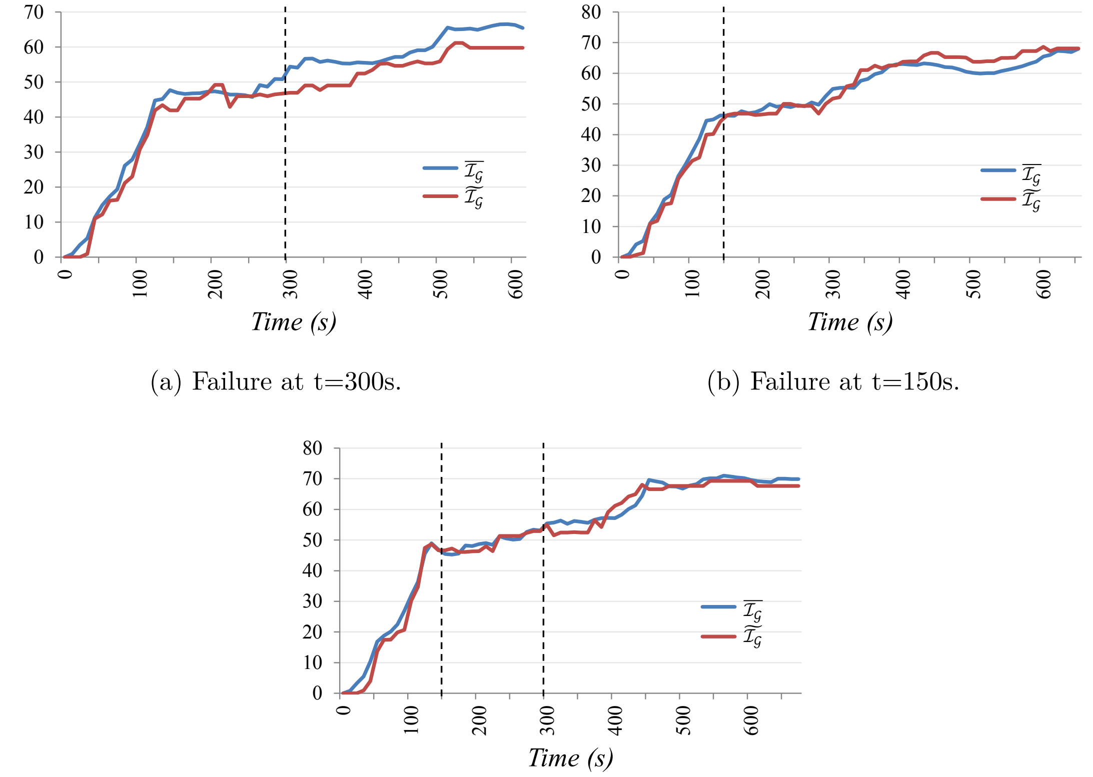
5.3.3 Robustness and Fault-Tolerance

Distributed systems usually benefit from being robust to failures, since each autonomous agent can perceive the state of the environment in its surroundings and adapt to unforeseen constraints. Therefore, these systems enable redundancy and should remain functional if some of the robots fail.

Similarly as in chapter 4, the robustness of the approach is demonstrated in three experiments using the Pioneer 3-DX robots, where 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 all three tests, the team starts with six robots. In the first test a robot is shutdown 300 seconds into the experiment. Similarly, in the second experiment, a robot is shutdown after 150 seconds. Finally, in the third experiment, one robot is shutdown after 150 seconds and a second robot is shutdown after 250 seconds. The other robots assume that a teammate has failed when no message has been received from it within 2 minutes.

Table 5.10 indicates that the results obtained in the first two experiments resemble those obtained with five robots. In these experiments, the average idleness values converged after the fault occurs. Figure 5.10 shows the evolution of the average vertex idleness in the three experiments. In the first experiment, since approximately half of the mission was spent with six robots, the final performance was slightly better and the overall mission time τ was shorter than in the second experience, *i.e.*, it took longer in the second test to fulfill the mission because the fault occurred earlier.



(a) Failure at t=300s.

(b) Failure at t=150s.

(c) Failure at t=150s and t=250s.

Figure 5.10: Evolution of the idleness along time using CBLS with teams of six physical robots with failures.

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

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

5.4 Summary

In this chapter, cooperative multi-agent learning has been addressed in order to solve the patrolling problem in a distributed way. Each robot decides its local patrolling moves online, without requiring any central planner. Decision-making is based upon Bayesian reasoning on the state of the system, considering the history of visits and teammates actions, so as to promote effective coordination in the behavior of patrolling agents. Concurrent reward-based learning has been adopted given that, in this domain, the decomposition of the problem reduces the complexity of the general mission by distributing computational load among each independent learner. The model is complete and generalizable, including vertex look-ahead to enhance decision-making.

Experimental results have shown that the method is able to effectively tackle the problem, since it can deal with uncertainty and the actions are selected according not only to prior knowledge about the problem, but also the state of the system at the time. Moreover, the learning robots adapt to constraints and the dynamics of the system, *e.g.*, different agent velocities, since the decision-making is done online with the information that each agent has collected about the system.

Evaluating the performance of the approach against several state-of-the-art approaches through simulations, CBLS is generally superior independently of team size and, in the real-world experiments conducted, the approach was able to obtain near optimal results, which is particularly remarkable given the limited search space considered and distributed nature of the approach. Thus, proving the potential of the proposed multi-robot patrolling strategy for real-world applications, through assessment on the adaptability, scalability and fault-tolerance nature of the approach.

It would also be interesting to relax the assumption of perfect communication, as done in chapter 4, testing the performance of CBLS under communication failures and using only local interactions between robots in the same range. It is the author's belief that similar results would be obtained, due to its distributed nature and the decision-making autonomy of each patrol unit, eventually proving the robustness to occasional communication failures.

In the next chapter, methods for estimating performance of teams of robots with variable size are addressed. With that goal in mind, algorithms that extract the theoretical performance on classical near-optimal approaches in patrolling missions are studied, in order to serve as a predictable upper bound for the performance of the team in a generic environment using any patrolling approach, given some temporal constraints. The analytical methods presented are able to extract performance without the need to run simulated or real-world experiments as done up to now.

Chapter 6

Optimal Team Dimensions for Patrolling Missions

As seen before, multi-robot patrolling is a problem that has important applications in many different fields. In this chapter, a final step is taken to conclude the study on the scalability and performance with teams of any number of agents in patrolling missions. In this case, one should answer the following research question:

"Given a generic environment, represented by graph $\mathcal{G}$, how many robots are necessary in the patrolling task, such that each point is <u>*at least*</u> *visited every Ω seconds?"*

The answer to this question requires an estimate of the performance of teams of arbitrary high R robots in generic environments, prior to the mission. Note however, that a slightly different approach should be taken, when compared to previous chapters. In this case, all locations must be visited *at least* every Ω seconds. This means that the average idleness of vertices should not be optimized, but rather the worst vertex idleness, *i.e.*, the critical point of intrusion in the system. For that reason, a variation of the problem is proposed in section 6.1.

Estimation of performance to dimension teams of robots in patrolling tasks is a challenging issue. As seen throughout this thesis, probabilistic strategies benefit from difficult intrusion for evaders and each agent is an autonomous decision-making entity in the collective patrol task. These have been in focus up to now. However, foreseeing team performance is extremely complex for such situations, due to the unpredictability involved. On the other hand, deterministic policies follow very strict rules and, under some assumptions, it is possible to estimate their theoretical performance. In addition, such performance may allegedly be optimal or at least sub-optimal if the right policy is selected according to the environment to patrol. Thereunto, four algorithms are presented to estimate the outcome of deterministic and theoretically optimal approaches to approximate and serve as an upper bound for the performance of any generic multi-robot patrol strategy. These algorithms make use of graph-theory procedures and do not require running an entire patrol mission, being able to predict the result beforehand.

However, the optimal task assignment in multi-robot patrol is known to be NP-hard. Classical centralized approaches consider evenly spacing the robots in a cyclic TSP based tour or partitioning the graph of the environment. The trade-off in computing time, overall team travel cost and coordination in both classes of strategies is firstly analyzed. Both theoretical analysis and experimental results across multiple environments are provided. The proposed algorithms will provide the intended upper bound for estimation and will confirm that graph topology and team size determine the best choice for an optimal patrol strategy.

The key challenge in this work is to design effective deterministic patrol routes in order to optimize the worst idleness criterion. Like most existing work in the literature [Chevaleyre, 2004, Pasqualetti et al., 2012a, Portugal and Rocha, 2013d, Iocchi et al., 2011], it is assumed that robots are homogeneous, travel with the same average and maximal speed, and are expected to visit every strategic position of the environment. Therefore, owning adequate sensing range, complete coverage of the environment is achieved by visiting all the important locations in the area.

In this chapter, theoretical results known so far are studied and certain problems that are still open in this area are posed. In addition, practical implementation issues on real-world robots are discussed. In Section 6.1, the worst idleness performance criteria is defined and the variant of the problem is formulated.

Section 6.2 describes four techniques to estimate the performance of cyclic and partitioning-based tours. Afterwards, the estimation outcome of these techniques using teams from 1 to 20 robots are discussed. In the end, conclusions are drawn and directions for future research are outlined.

6.1 Problem Definition

As stated before, the general multi-robot patrolling problem is known to be NP-hard. In fact, Chevaleyre [Chevaleyre, 2004] provided the first theoretical analysis of the problem and was able to prove that it can be optimally solved with one robot using a TSP Tour, which in turn, is NP-Hard. Nevertheless, for metric problem instances[32] of the TSP, there are several adequate approximation algorithms. For instance, Christofides Algorithm [Christofides, 1975] 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 [Applegate et al., 2003], typically finds tours within 5% of the optimal in $O(|\mathcal{V}|^{2.2})$ computation time. For the multi-robot case, Chevaleyre showed that partitioning-based strategies may perform better than evenly spacing robots along a TSP cycle for specific graph instances, *e.g.*, when there are long corridors or edges separating clusters of vertices.

Following this, Pasqualetti *et al.*, which used the concept of “refresh time” as a synonym for “worst idleness”, proved that the team refresh time problem is NP-hard, *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. To address the general problem with R robots, Pasqualetti *et al.* studied approximation algorithms to obtain known bounds related to the optimal result on specific graph instances. Unlike [Pasqualetti et al., 2012a], the focus in this work is on generic graphs describing any real-world environment and effective heuristics are also considered to obtain near-optimal results. It is shown that these heuristics consistently result in superior trajectories

[32] In the metric TSP, the edge weights satisfy the triangle inequality (*cf.*, footnote 9) as in the case of metric graphs.

(or at least equally good) to constant factor approximation algorithms.

As referred before, it is common to represent the area to patrol by a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $v_i \in \mathcal{V}$ vertices and $e_{i,j} \in \mathcal{E}$ edges. Wherefore, $\mathcal{G}$ corresponds to the topological map for the patrolling mission, and vertices correspond to important places or landmarks and edges represent the connectivity between those locations. As stated in section 2.4, the topological maps considered represent real-world 2D environments, being generic, undirected, connected, simple, planar and generally non-complete.

The multi-robot patrolling problem is then reduced to find R trajectories $\Pi = \{\pi_1, ..., \pi_R\}$ for each robot in order to visit frequently all vertices $v_i \in \mathcal{V}$ with respect to a predefined optimization criterion. An effective strategy should minimize the time lag between two passages to the same place and for all places. Thus, considering the instantaneous idleness $\mathcal{I}_{v_i}$ of a vertex v_i as defined in eq. 2.2, the worst idleness $\mathcal{WI}$ corresponds to the largest idleness value for all $v_i \in \mathcal{V}$ that occurred during the patrolling task that lasted τ time units:

$$\mathcal{WI} = \max_{v_i \in \mathcal{V}} \max_{t \in [0, \tau]} \mathcal{I}_{v_i}. \tag{6.1}$$

Using now $\mathcal{WI}$ as the optimization criterion, a varant of the multi-robot patrolling problem is formulated as follows:

Problem 1 (Multi-Robot Patrol). *Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and a team of R robots, find a policy Π^* such that:*

$$\Pi^* = \underset{\Pi}{\mathrm{argmin}} \, (\mathcal{WI}). \tag{6.2}$$

Remark 1 (Computational Complexity). *It has been shown that Problem 1 is NP-Hard by reduction from the Traveling Salesman Problem (cf. Theorem II.1 in [Pasqualetti et al., 2012a]).*

6.2 Effective Deterministic Trajectories for Multi-Robot Patrol

Two classical types of strategies have been used previously in the literature to obtain optimal and/or near-optimal results: cyclic-based and partitioning-based strategies [Chevaleyre, 2004, Pasqualetti et al., 2012a]; and considering the worst idleness criterion, superior approaches to those are still not known.

Using a variety of graph theory concepts, it is possible to devise patrolling trajectories for the team of robots based on the two classes of approaches, defined below.

Definition 4 (Cyclic-based Strategy)**.** *Given a closed walk* $X = \{v_a, v_b, ..., v_a\}$ *in* $\mathcal{G}$*, such that* $\forall v_i \in \mathcal{V} : v_i \in X$ *and possibly visiting vertices more than once, a Cyclic-based strategy* Π_{Cyc} *places* r *agents, that move at the same speed, equally spaced along* X*, while keeping a constant gap between them.*

Definition 5 (Partitioning-based Strategy)**.** *Given a set of disjoint partitions* $\mathbf{P} = \{P_1, ..., P_R\}$ *in* $\mathcal{G}$*, such that* $\cup_{r=1}^{R} P_r = \mathcal{V}$ *and* $P_i \cap P_j = \emptyset$ *with* $i \neq j$*, in a Partitioning-based strategy* Π_P*, each agent* r *visits the vertices of a single partition* P_r*, by following a strategy* π_r*.*

In the next two subsections, two cyclic-based and two partitioning-based strategies are presented. In the cyclic approach, it is necessary to compute a patrolling-effective close walk X. For the partitioning approach, it is necessary not only to compute an effective set of partitions $\mathbf{P}$, but also to define each agent's strategy π_r on each partition P_r.

For each class of approach, one method with a known constant factor approximation and one heuristic method are proposed. These heuristics are employed to obtain closer results to the theoretical optimum and to further understand the potential of each class of strategy in graphs with different connectivity and using different team sizes.

6.2.1 Cyclic-based Strategies

In a cyclic-based strategy, the time taken for an agent to visit a vertex for the second time is at most $v \cdot \mathcal{L}(X)$, where v is the average agent's speed and $\mathcal{L}(X)$ is the length of the walk X. Without lack of generality, let us assume that agents move at unitary speed. The worst idleness of any cyclic based-strategy Π_{Cyc}, using a single agent, is given by:

$$\mathcal{WI}_{\Pi_{Cyc}} = \mathcal{L}(X), \quad R = 1, \tag{6.3}$$

since agents are equally spaced along X, one can extend (6.3) for a multi-robot situation with R robots:

$$\mathcal{WI}_{\Pi_{Cyc}} = \frac{\mathcal{L}(X)}{R}. \tag{6.4}$$

Clearly, by minimizing $\mathcal{L}(X)$, *i.e.*, finding the smallest X walk that visits every vertex of $\mathcal{G}$, $\mathcal{WI}$ becomes minimal in (6.4). Consequently, it becomes evident that the Traveling Salesman Problem (TSP) solution for $\mathcal{G}$ is the best possible solution among all cyclic-based strategies:

$$\Pi^*_{Cyc} = \Pi_{TSP}. \tag{6.5}$$

TSP is a classical NP-complete problem and no polynomial time algorithm is known to compute an optimal solution to it. In this section, two different methods for approximating a metric TSP tour in a generic graph $\mathcal{G}$ are discussed.

The first method presented is a well-known approximation for the metric TSP tour, based on the Minimum Spanning Tree (MST) concept, as shown in Fig 6.1. Consider algorithm 6.1, named MST Tour.

Theorem 1 (Constant factor approximation)**.** *MST Tour is a 2-approximation for the metric TSP.*

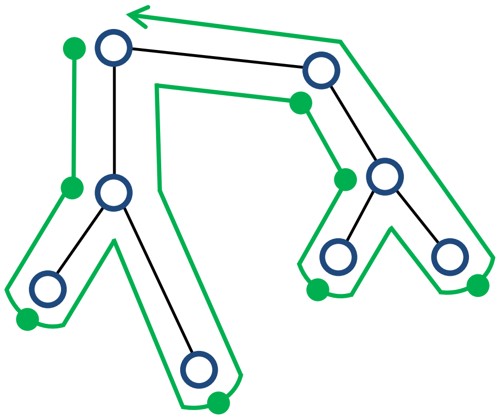


Figure 6.1: Obtaining a closed walk using a MST (Algorithm 6.1).

Algorithm 6.1: MST Tour Approximation.

i) *Find a Minimum Spanning Tree $\mathcal{T}$ in $\mathcal{G}$.*

ii) *Conduct a depth-first search (DFS) to visit all $v_i \in \mathcal{T}$ in a depth-first order.*

iii) *Build a closed walk π_{MSTt} that visits all vertices, following the order of DFS discovery.*

iv) *Equally space R moving robots along π_{MSTt}.*

Proof. Let $\mathcal{L}(\pi_{TSP})$ be the cost of an optimal TSP tour. Recall that by removing an edge from π_{TSP}, one obtains a spanning tree. Therefore, the Minimum Spanning Tree provides a lower bound for the optimal tour: $\mathcal{L}(\mathcal{T}) \leq \mathcal{L}(\pi_{TSP})$. Notice that the length of a depth-first tour of the connected tree $\mathcal{T}$ equals twice the sum of the length of the edges of $\mathcal{T}$: $\mathcal{L}(\pi_{MSTt}) = 2 \cdot \mathcal{L}(\mathcal{T})$. Hence, $\mathcal{L}(\pi_{MSTt}) \leq 2 \cdot \mathcal{L}(\pi_{TSP})$. ∎

Corollary 1. *It immediately follows that the worst idleness of the MST tour $\mathcal{WI}_{\Pi_{MSTt}}$ with R agents is at most 2 times the worst idleness of the optimal TSP tour $\mathcal{WI}_{\Pi_{TSP}}$ with R agents.*

Algorithm 6.1 can quickly obtain a 2-approximate solution for the TSP in feasible time. In fact, the implementation considered (*cf.* section 6.3), uses Kruskal's

Algorithm 6.2: Heuristic to approximate the TSP Tour.

i) *Create a complete graph $\mathcal{G}_C = (\mathcal{V}_C, \mathcal{E}_C)$, by copying all vertices and edges of $\mathcal{G}$, and for all non-existing edges between pairs of vertices v_h, v_i in $\mathcal{G}$, create a unique edge $e_{h,i}$ in $\mathcal{G}_C$, where $|e_{h,i}| = \texttt{dijkstra}(v_h, v_i)$.*

ii) *Compute an efficient heuristic for the TSP in $\mathcal{G}_C$.*

iii) *Convert the TSP tour obtained in $\mathcal{G}_C$ into a minimum cost closed walk π_{HTSP} in $\mathcal{G}$ by translating each edge $|e_{h,i}| \in [\mathcal{E}_C \cap \overline{\mathcal{E}}]$ of the TSP tour into a shortest path of $\mathcal{G}$ in π_{HTSP}.*

iv) *Equally space R moving robots along π_{HTSP}.*

algorithm in step i to compute the minimum spanning tree [Kruskal, 1956], which runs in $O(|\mathcal{E}| \log |\mathcal{V}|)$ time.

Despite the performance guarantees given by Π_{MSTt}, in general Algorithm 6.1 does not lead to an optimal TSP tour. This is clear in the estimates reported in the evaluation section. For a generic graph $\mathcal{G}$, an additional cyclic-based method has been tested, being described by Algorithm 6.2.

Before proceeding, it is important to demonstrate that solving the TSP in a complete graph $\mathcal{G}_C$ is equivalent to solve the minimum cost closed walk problem in a non-complete graph $\mathcal{G}$, where ocasionally repeating visits to vertices is allowed.

Theorem 2 (Minimum Cost Closed Walk)**.** *The TSP tour of the complete graph $\mathcal{G}_C$ is equivalent to the minimum cost closed walk in $\mathcal{G}$.*

Proof. When decoding a TSP tour of $\mathcal{G}_C$ into a closed walk π_{HTSP} in $\mathcal{G}$ (step *iii* of Alg. 6.2), clearly all edge costs are preserved. Assume now that there is a shorter closed walk π_s such that $\mathcal{L}(\pi_s) < \mathcal{L}(\pi_{HTSP})$ in $\mathcal{G}$. Translate π_s into a tour in $\mathcal{G}_C$ by selecting the vertices in order they appear first. This would imply a tour in $\mathcal{G}_C$ shorter than the optimal TSP tour. Hence, one has a contradiction. ∎

Algorithm 6.2 presents an heuristic solution Π_{HTSP} for the minimum cost closed walk with R agents. It should be noted that Dijkstra's algorithm [Dijkstra, 1959] is used to obtain the shortest path in $\mathcal{G}$ and, in this implementation (see section 6.3), the heuristic chosen to approximate the TSP in a complete graph (step *ii*) is a genetic algorithm named TSP_GA[33]. Despite not having performance guarantees, the results have shown that the approach is able to quickly compute the optimal solution, known *a priori* for every graph tested. The graphs considered have several tens of vertices, similarly to those that commonly represent real-world areas.

6.2.2 Partitioning-based Strategies

In the past, it has been shown [Chevaleyre, 2004, Iocchi et al., 2011, Pasqualetti et al., 2012a, Portugal and Rocha, 2013d] that partitioning strategies may have superior performance than cyclic ones. In fact, two important contributions by Chevaleyre [Chevaleyre, 2004] are recalled:

i) The optimal partition-based strategy Π_P^* is a disjoint partition, where each agent behaves optimally inside each subgraph, by running a TSP tour on it (*i.e.*, a minimum cost closed walk).

ii) Cyclic strategies are not suited for graphs containing long edges, as shown by the following result: $\mathcal{WI}_{\Pi_{Cyc}^*} \leq \mathcal{WI}_{\Pi_P^*} + 3 \cdot \mathtt{max}|e_{i,j}|$.

These two contributions are of high importance in the multi-robot patrolling literature. Nevertheless, they lead to two important follow-up questions:

In i), each agent's strategy π_n becomes evident, however how should one optimally compute a set of R partitions P in the first place?

In ii), the inequality can be verified with $\mathcal{WI}_{\Pi_{Cyc}^*} > \mathcal{WI}_{\Pi_P^*}$ or $\mathcal{WI}_{\Pi_{Cyc}^*} < \mathcal{WI}_{\Pi_P^*}$. So, which strategy should one choose for a given graph $\mathcal{G}$ patrolled by R agents?

[33] TSP_GA has been developed by Joseph Kirk. For more information, it is openly available at: http://www.mathworks.com/matlabcentral/fileexchange/13680

In this section, two partitioning-based strategies to address the first question are presented. The second question is addressed later in the evaluaion section 6.3.

As seen before, in a partition-based strategy, each agent r follows a minimum cost closed walk π_r in the subgraph induced by partition P_r in $\mathcal{G}$. Thereby, the worst idleness on each partition is given by (6.3): $\mathcal{WI}_{\pi_r} = \mathcal{L}(\pi_r)$. Note that all partitions are disjoint, hence each vertex is only visited by one robot. Since all partitions are patrolled in parallel, the worst idleness on $\mathcal{G}$, considering unitary agent's speed without lack of generality, is given by the maximum length of any tour π_r:

$$\mathcal{WI}_{\Pi_P} = \texttt{max}\, \mathcal{L}(\pi_r). \tag{6.6}$$

Classical graph-partitioning is a NP-hard problem, which is usually applied in parallel computing and clustering applications [Hendrickson and Kolda, 2000]. For high performance in such systems, regions should be identically sized, *i.e.*, each partition should ideally have the same number of vertices $|P_1| \simeq ... \simeq |P_R|$. Moreover, the interface between them should be small, *i.e.*, the edges that connect different partitions should have minimal cost. While this may yield a satisfactory solution for the patrol partitioning problem considered herein, it is necessary to consider two fundamental differences.

Firstly, due to (6.6), instead of identically sized partitions, here the aim is to obtain regions with balanced cost $\mathcal{L}(P_1) \simeq ... \simeq \mathcal{L}(P_R)$ so as to minimize the partition tour π_r with maximal cost and consequently, $\mathcal{WI}_{\Pi_P}$. Furthermore, since the edges between partitions are not traversed by any robot, the cut should ideally be conducted on long edges to minimize each robot's closed walk. Hence, the following problem is defined.

Problem 2 (Min-Max Cost Closed Walk)**.** *Given a generic graph $\mathcal{G}$, find a set of disjoint partitions $\mathbf{P} = \{P_1, ..., P_R\}$ in $\mathcal{G}$ such that:*

$$\mathbf{P}^* = \underset{\mathbf{P}}{\texttt{argmin}}\, (\texttt{max}\, \mathcal{L}(\pi_r)). \tag{6.7}$$

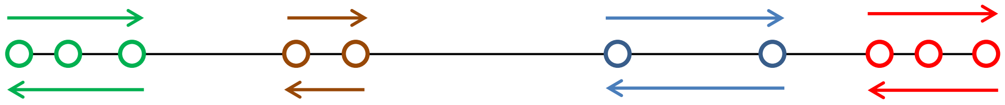


Figure 6.2: Four optimal closed walks on a chain graph.

Theorem 3 (Computational Complexity)**.** *The min-max cost closed walk problem is NP-hard.*

Proof. When $R = 1$, this problem is equivalent to find the minimal cost closed walk, which in its turn, is equivalent to the Traveling Salesman Problem (see Theorem 2). Since the TSP is a NP-hard problem, by restriction the min-max cost closed walk problem is NP-hard. ∎

Similarly as before, in order to solve Problem 2, one algorithm with known performance ratio is tested, as well as an evolutionary heuristic technique.

For the first partitioning-based algorithm, a previously known result is applied. The authors in [Pasqualetti et al., 2012a] have proposed an optimal min-max cost closed walk partition for the particular case of a chain graph, *e.g.*, Fig 6.2, which was called an "Optimal Left-Induced partition". By extending this result to generic graphs, the approximation method presented in Algorithm 6.3 has been proposed.

Algorithm 6.3: Left-Induced Partition on Generic Graphs.

i) Find an open walk, with at most $2|\mathcal{V}|-4$ edges, that visits all $v_i \in \mathcal{V}$ of $\mathcal{G}$.

ii) Construct a chain graph Γ, equivalent to the open walk in i).

iii) Compute the optimal left-induced partition with R agents in Γ.

iv) In order to create a strategy Π_Γ, assign a partition P_r to each of the R agents and have them patrolling each region back and forth (as shown in Fig 6.2).

Remark 2 (Performance Ratio)**.** *It has been shown that Algorithm 6.3 leads to the following performance guarantee wrt the optimal solution* [Pasqualetti et al., 2012a]*:*

$$\mathcal{WI}_{\Pi_\Gamma} \leq 8\left(\frac{|V|-2}{|V|}\right)\eta\,\mathcal{WI}_{\Pi^*}, \tag{6.8}$$

$$\textit{with:} \qquad \eta = \frac{\texttt{max}|e_{i,j}|}{\texttt{min}|e_{i,j}|}.$$

Note that such an open walk in $\mathcal{G}$ (step *i*) can be obtained from a MST, by starting from any leaf of the tree and stopping when all vertices have been visited (without returning to the initial one). Nonetheless, in the implementation described in this work, the result given by Algorithm 6.2 has been considered instead, which corresponds to a TSP tour on a complete graph $\mathcal{G}_C$. Clearly, an open walk with equal or inferior cost can be obtained by removing the longest edge of the TSP tour in $\mathcal{G}_C$ and translating it into an open walk in $\mathcal{G}$. Moreover, having at most $2|\mathcal{V}|-4$ edges, the performance bounds indicated by Remark 2 remain.

Despite the performance guarantees of Algorithm 6.3, given the high dependence on η, the previous algorithm is rarely expected to reach an optimal solution. Thus, an additional partitioning-based evolutionary[34] heuristic is proposed to solve the multi-robot patrol problem. This is described in Algorithm 6.4.

Vertex swaps in step *iii)* correspond to a mutation mechanism on the current solution. Additionally, the concept of "survival of the fittest" is applied, as the best solution found is kept during run time. In step *i)*, METIS algorithm is used as the method for multi-way partitioning [Karypis et al., 1998][35]. Furthermore, it is necessary to dimension the exploration factor Φ. A low value will not let the

[34] Note that although algorithm 6.4 belongs to the class of evolutionary algorithms, it cannot be considered a "genetic" approach, since no population larger than one solution is considered; and as a consequence, elitism, natural selection and crossover operators are not applied.

[35] METIS multi-way partitioning is available in the "MESHPART" toolbox for Matlab: `www.cerfacs.fr/algor/Softs/MESHPART`

Algorithm 6.4: Evolutionary Heuristic to approximate the Min-Max Cost Closed Walk Problem.

```
i) Compute a set $\mathbf{P}$ with $R$ initial partitions using a classical multi-way graph-partitioning approach on $\mathcal{G}$ and start a counter iter=0.
ii) Initially set global_best $= \mathcal{L}(P_n)$ as the length of the partition with maximal cost $P_r = \texttt{argmax}\,\mathcal{L}(P_{init})$.
iii) Swap a vertex from $P_r$ with a random neighbor partition $P_m$.
iv) Make sure $P_r$ stays connected. Otherwise, randomly keep one of the disjoint parts and swap all the others to $P_m$.
v) Assign $P_r = P_m$, if $|P_m| > 1$ and $P_m$ has not been used before. Otherwise choose randomly for $P_r$ a partition that has not been chosen as $P_r$ before.
vi) If $\texttt{max}\,\mathcal{L}(P_i) > \Phi\ \texttt{max}\,\mathcal{L}(P_{init})$ or all partitions have already been used as $P_r$, reset $P_r = \texttt{argmax}\,\mathcal{L}(P_{init})$.
vii) Save solution global_best $= \texttt{max}\,\mathcal{L}(P_i)$ if $\texttt{max}\,\mathcal{L}(P_i) <$ global_best. Increment iter.
viii) Repeat steps iii - vii, while iter $\leq$ MaxSteps.
xix) Build $\Pi_{EHP}$ by considering the set of partitions that generated global_best and use Algorithm 6.2 to compute a minimum cost closed walk for each agent $r$ in $P_r$.
```

approach explore the search space conveniently and may fall into local minima. Extreme high values will make the approach spend to much time generating solutions with low quality. In the considered experiments, it was used $2.5 \leq \Phi \leq 5.0$ and MaxSteps ≤ 15000.

In the next section, it is presented a discussion of $\mathcal{WI}$ performance results using this partitioning-based strategy and the three other aforementioned strategies applied in graphs with different connectivity and teams of patrolling agents with different sizes. This way, it is possible to estimate the theoretical performance of classical algorithms for multi-robot patrolling, without running an entire patrolling mission or simulation, and use them as an upper bound for estimation of the number of robots necessary to perform a patrolling task, while satisfying the constraint $\mathcal{WI} \leq \Omega$.

6.3 Performance Evaluation

In this section, the three usual benchmarking topological maps $\mathcal{G}_A$, $\mathcal{G}_B$ and $\mathcal{G}_C$ illustrated in Fig. 3.4, page 61, are employed once again in order to test all previously described strategies for classical deterministic multi-robot patrol. As stated before, each graph has distinct connectivity, and $\lambda_A < \lambda_B < \lambda_C$. Note, for example, that $\mathcal{G}_A$ has several dead-ends, *i.e.*, vertices with degree one. On the other hand $\mathcal{G}_C$ is the most connected of the three, with a maximum degree of 4. Despite being a highly connected graph in the context of a patrolling mission, $\mathcal{G}_C$ is far from being complete (each vertex would need to have degree 24) and may eventually be considered a sparse graph in different applications than multi-robot patrol.

As stated before, the methods presented to estimate performance are expected to be sensitive to graph topology. Additionally, even though cyclic-based and partitioning-based strategies should perform differently with the connectivity of $\mathcal{G}$, this evaluation is also aimed to understand whether one method can be superior to another according to team size. With that in mind, all approaches have been tested with $R \in [1, 20]$.

In essence, the implementation consists of building the routes for the MST Tour approximation (Π_{MSTt}), the heuristic for the TSP tour (Π_{HTSP}), the Left Induced partition-based strategy (Π_{Γ}) and the evolutionary partition-based heuristic (Π_{EHP}), and compute the worst idleness $\mathcal{WI}$ for an arbitrary R, using (6.4) for cyclic-based strategies and (6.6) for partitioning-based strategies. Furthermore, all methods used in the paper are made available to let the reader test other graph instances as desired[36].

[36] Matlab code of the four methods is available at: http://isr.uc.pt/~davidbsportugal/MRpatrol_toolbox

Table 6.1: $\mathcal{WI}$ performance results (in seconds), considering unitary speed and using the four described algorithms in Graph A with different team sizes.

	Graph A ($\mathcal{G}_A$)			
	Cyclic		Partitioning	
R	Π_{MSTt}	Π_{HTSP}	Π_{Γ}	Π_{EHP}
1	516.75	**507.75**	889.80	**507.75**
2	258.37	**253.87**	441.15	273.15
3	172.25	**169.25**	294.30	178.35
4	129.19	**126.94**	215.40	141.35
5	103.35	**101.55**	175.50	107.25
6	86.12	**84.62**	144.30	100.65
7	73.82	**72.53**	120.60	83.85
8	64.59	**63.47**	104.25	65.85
9	57.42	**56.42**	96.00	61.35
10	51.67	**50.77**	84.90	52.20
11	46.98	**46.16**	76.95	47.85
12	43.06	**42.31**	67.95	45.45
13	39.75	**39.06**	62.10	39.60
14	36.91	**36.27**	59.10	39.30
15	34.45	**33.85**	53.85	36.15
16	32.29	**31.73**	48.45	36.15
17	30.39	29.87	46.35	**28.20**
18	28.71	28.21	44.40	**27.90**
19	27.19	**26.72**	41.70	27.15
20	25.84	**25.39**	40.80	26.85

Table 6.2: $\mathcal{WI}$ performance results (in seconds), considering unitary speed and using the four described algorithms in Graph B with different team sizes.

	Graph B ($\mathcal{G}_B$)			
	Cyclic		Partitioning	
R	Π_{MSTt}	Π_{HTSP}	Π_{Γ}	Π_{EHP}
1	380.10	**313.65**	548.10	**313.65**
2	190.05	**156.82**	267.90	193.35
3	126.70	**104.55**	177.00	133.20
4	95.02	**78.41**	123.90	98.10
5	76.02	**62.73**	104.10	76.20
6	63.35	**52.27**	86.40	62.40
7	54.30	**44.81**	68.10	56.70
8	47.51	**39.21**	57.60	43.50
9	42.23	**34.85**	52.20	40.20
10	38.01	**31.36**	45.90	37.20
11	34.55	**28.51**	40.20	32.10
12	31.67	**26.14**	37.20	31.80
13	29.24	**24.13**	34.20	25.50
14	27.15	**22.40**	32.70	23.10
15	25.34	**20.91**	28.80	21.90
16	23.76	19.60	25.80	**19.20**
17	22.36	**18.45**	25.50	18.60
18	21.12	17.42	23.10	**16.80**
19	20.00	16.51	21.90	**13.80**
20	19.00	15.68	21.30	**12.90**

Tables 6.1, 6.2 and 6.3 present the overall results. Since one of the main goals of this work is to understand which class of strategy is more suited given a generic graph $\mathcal{G}$ and team size R, the best results over 25 trials for the partitioning-based strategies were saved. This is because Π_{Γ} depends on the choice of an open walk, which may differ in each trial and Π_{EHP} being an evolutionary algorithm may not always reach an optimal solution. As for cyclic-based strategies, this was not necessary because Π_{MSTt} always returns a spanning tree tour with minimal cost and Π_{HTSP}, as seen before, easily computes one optimal minimum cost closed walk in $\mathcal{G}_A$, $\mathcal{G}_B$ and $\mathcal{G}_C$, given enough iterations (typically $\simeq 1000$).

The prior evidence shown in these tables is that the performance of the cyclic-

Table 6.3: $\mathcal{WI}$ performance results (in seconds), considering unitary speed and using the four described algorithms in Graph C with different team sizes.

	Graph C ($\mathcal{G}_C$)			
	Cyclic		Partitioning	
R	Π_{MSTt}	Π_{HTSP}	Π_{Γ}	Π_{EHP}
1	273.60	**148.20**	273.60	**148.20**
2	136.80	**74.10**	136.80	79.80
3	91.20	**49.40**	91.20	57.00
4	68.40	**37.05**	68.40	45.60
5	54.72	**29.64**	45.60	34.20
6	45.60	**24.70**	45.60	34.20
7	39.09	**21.17**	34.20	22.80
8	34.20	**18.82**	34.20	22.80
9	30.40	**16.47**	22.80	22.80
10	27.36	**14.82**	22.80	22.80
11	24.87	**13.47**	22.80	22.80
12	22.80	**12.35**	22.80	22.80
13	21.05	**11.40**	**11.40**	**11.40**
14	19.54	**10.59**	11.40	11.40
15	18.24	**9.88**	11.40	11.40
16	17.10	**9.26**	11.40	11.40
17	16.09	**8.72**	11.40	11.40
18	15.20	**8.23**	11.40	11.40
19	14.40	**7.80**	11.40	11.40
20	13.68	**7.41**	11.40	11.40

strategy Π_{HTSP} is superior to all other methods in 90% of the configurations tested. This confirms that finding a minimum cost closed walk on the graph and having robots equally spaced is usually the most effective solution for the worst idleness multi-robot patrolling problem, in theory. In particular, it should be noticed that in $\mathcal{G}_C$, $\eta = 1$ since all edges have the same cost of 5.70 m. Thereupon, no other strategy was able to overcome Π_{HTSP}, which shows the potential of the approach when edges are balanced.

Besides, it is also important to refer that the team size R plays a fundamental role when choosing a classical multi-robot patrolling approach. Results show that when the number of robot grows, partitioning strategies tend to approximate the performance obtained by cyclic strategies, when $\eta > 1$, as in $\mathcal{G}_A$ and $\mathcal{G}_B$. In fact, it can be seen in those cases, that when $R \geq 10$, $\mathcal{WI}_{\Pi_{EHP}} \simeq \mathcal{WI}_{\Pi_{HTSP}}$ and even $\mathcal{WI}_{\Pi_{EHP}} < \mathcal{WI}_{\Pi_{HTSP}}$. Despite only being theoretically superior to optimal cyclic-based approaches with high R, in practice a partitioning-based strategy like Π_{EHP} presents a great advantage over Π_{HTSP}. Seeing as each robot patrols a disjoint subgraph of $\mathcal{G}$, robots do not cross the paths of each other and inter-robot coordination is not an issue. Yet, in a cyclic strategy, when a closed walk is computed, vertices may be repeated and robot interference is an issue. As a result, a mechanism must exist to avoid having robots visiting the same vertex at the same time. Such a mechanism will have impact on the worst idleness $\mathcal{WI}_{\Pi_{Cyc}}$ in the real-world, unless the considered closed walk does not repeat vertices. Indeed, the effect of robot interference has been shown in previous chapters of this thesis. Additionally, keeping robots evenly spaced presents a challenge in the real-world and is usually done in a centralized way. This clearly demonstrates the existing contrast between theory and practice in this field of research.

Results have also shown that the heuristics considered were able to outperform methods with known performance bounds, thus reaching solutions with higher quality. Additionally, despite the α-approximations reported in Theorem 1 and Remark 2, the performance of Π_{MSTt} and Π_{Γ} was consistently within a factor of at most 1.85 to the best solution. In contrast, the optimal solution was only reached by Π_{Γ} in one instance ($R = 13$ in $\mathcal{G}_C$).

As expected, the MST-tour approximation has closer performance to Π_{HTSP} when the connectivity of $\mathcal{G}$ decreases. On the other hand, the dependency of Π_{Γ} in η is evident, seeing as its best result was obtained in $\mathcal{G}_C$.

As indicated by the results for these environments, the cyclic strategy results in better $\mathcal{WI}$ patrol times in most cases. However, this may not be the only important consideration for optimizing the patrol. For longer running patrols, global resource usage of the robot team may also be an important consideration. For instance, it may be important to minimize the amount of fuel used by the robot team or the total distance covered. In the general case, partition strategies are expected to have lower overall resource usage than cyclic strategies for the same environment. This is because the patrol strategies are able to make graph cuts on the long edges in the graph, reducing the number of edges that must be traveled, while in the cyclic strategy, all robots must travel the entire route. Thus, the *Normalized Distance per Patrol* metric ς, is defined to be the total distance traveled by the full robot team to perform a single patrol, divided by the average number of vertices visited on the patrol, $\mathcal{C}$. For the partitioning case, this is given by:

$$\varsigma_P = \sum_{r=1}^{R} \frac{\mathcal{L}(\pi_r)}{\mathcal{C}}, \tag{6.9}$$

while for the cyclic case, the Normalized Distance per Patrol is given by:

$$\varsigma_{Cyc} = \frac{\mathcal{L}(\pi_{TSP}) \cdot R}{\mathcal{C}}. \tag{6.10}$$

Singularly, in this case it is verified that:

$$\mathcal{C} = \mathcal{C}_{TSP} \cdot R, \tag{6.11}$$

where $\mathcal{C}_{TSP}$ is the average number of vertices visited by a single robot in a cyclic closed walk. This property is the natural consequence of having all robots following the same route. For this reason, in this case ς is independent of the number of agents R in the mission:

$$\varsigma_{Cyc} = \frac{\mathcal{L}(\pi_{TSP})}{\mathcal{C}_{\mathcal{TSP}}}. \tag{6.12}$$

Accordingly, ς_{Cyc} is always the same, regardless of the number of robots in a cyclic patrol.

The normalized distances per patrol for each classical strategy are shown in Figure 6.3. In the partition case, robots visit only the vertices in their assigned partition. Consequently, the results indicate that the partition strategies result in

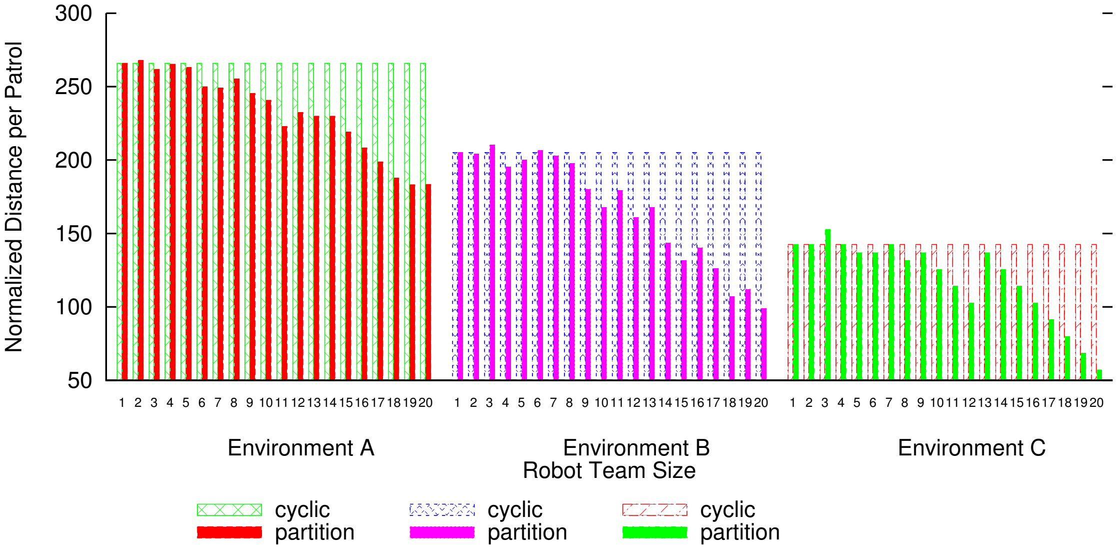


Figure 6.3: Normalized distances for the cyclic (Π_{HTSP}) and partition (Π_{EHP}) patrol strategies for each environment. The partition strategies are more resource efficient for larger robot teams.

lower normalized distances per patrol when the team size is greater than about 3 robots, and are therefore more resource efficient. This trend is present independently of the graph connectivity.

On a final note concerning real-world implementation, all the strategies tested rely on predefined trajectories for the robots that are computed prior to the mission start, potentially reaching an optimal solution. In some applications this may not be intended, for example in adversarial patrolling scenarios, where an intruder may apprehend the robots' routes and attack the system in an easier way. However, in situations such as cooperative cleaning of infrastructures, having near-optimal performance is highly desired.

Assuming constant unitary robot speed, the methods presented lead to an estimate of real-world results using teams of arbitrary number of robots. In particular, bold values in tables 6.1, 6.2 and 6.3, present upper bounds of performance, *i.e.*, lower bounds on $\mathcal{WI}$ values, which enable to answer the research question posed in the beginning of this chapter, by dimensioning R such that $\mathcal{WI} \leq \Omega$.

6.4 Summary

In this chapter, worst idleness performance estimation has been addressed in order to dimension teams of robots, so that every place in the environment is patrolled at least every Ω seconds. For that purpose, classical deterministic near-optimal strategies have been analyzed and four different methods were devised and tested to provide upper bound estimates on real-world results. With that goal, evaluation of performance of classical strategies has been done to understand, which approach is superior, being ultimately chosen to provide the intended upper bound.

Four distinct strategies, based on the concepts of cyclic tours and graph partitioning are compared and analyzed to further understand which approach suits best a given generic environment, with an arbitrary high number of robots. Considering an optimization criterion based on the worst idleness $\mathcal{WI}$, it has been demonstrated that generally, cyclic-based strategies generate solutions with high quality and should be selected when relatively small teams are used and/or the edge costs of the graph $\mathcal{G}$ are balanced. While partitioning-based strategies should be preferred otherwise. It was proven that graph connectivity is not the only parameter that should be considered in the selection of an optimal patrolling strategy, but also team size plays an important role

It was also shown that heuristic methods can effectively provide the intended estimates on typical real-world topological maps, as the achieved results were superior to those of algorithms with known performance bounds. Moreover, implementation issues on real-world robots were discussed and a useful testing tool is provided to the community for further tests with different graphs, so as to extend the results presented herein and infer about the optimality on any generic graph.

It becomes clear that the analytic analysis portrayed in this chapter can be an important tool to predict the outcome of real-world patrolling missions. Still, coordination of agents in the patrolling policy also plays an important role, as seen in previous chapters of this thesis, and one should also consider resource usage in real-world scenarios. For instance, distance traveled by agents has important implications in the robot's energetic autonomy.

Chapter 7

Overall Conclusions and Future Work

In this thesis, effectiveness and scalability in teams of cooperative mobile robots performing patrolling missions was deeply in focus. The patrolling problem was initially defined as a fundamental multi-robot task with important applications in the real-world, such as surveillance of infrastructures and automated cleaning. Furthermore, a taxonomy of the MRPP was proposed, where the main problem was divided in four key sub-problems: area patrol, perimeter patrol, adversarial patrol and pursuit-evasion. From that point on, area patrol was essentially studied based on an algorithmic approach with the main goal of effectively patrol all locations in indoor areas, independently of the number of robots in the team and the topology of the environment, as opposed to other challenges more related with handling abnormal situations, such as detecting or neutralizing alleged enemies in security tasks, which may represent a subsequent matter of study in the patrolling mission.

A thorough survey of the literature in multi-robot systems applied to cooperative patrol, area sweeping and graph coverage was also presented. The review of the state-of-the-art allowed to build up general knowledge and experience, as well as identifying issues and important questions in the area, thus capturing key research opportunities and providing new insights. A variety of concepts had been explored prior to this Ph.D. project and during the course of this work, in the past four years. Nevertheless, the problem enjoys high complexity, being proven as NP-Hard, and several gaps have been pointed out in the related work, of which stand out: scalability analysis, lack of experimental work and robustness to robot failures, hard assumptions and unrealistic simplifications, and the subsistence of deterministic approaches.

Overcoming the aforementioned weaknesses became the primary challenges in this thesis, as presented in section 2.3. Based on a topological conception of the environment represented as an undirected generic graph, which is acquired from a metric floor plan, the performance criterion, founded on the notion of idleness, was proposed in order to evaluate the different strategies described along this thesis.

Earlier, the implementation and evaluation of previously existing algorithms in a realistic simulation environment was considered in order to overcome the reported simplifications and investigating the potential of application of available methods in the literature, as well as confirming their performance and scalability. This represented a first step towards a common and unifying testing framework based on the well-known Robot Operating System (ROS). Besides, it was possible to extract which class of strategies are most suited according to graph connectivity and team size, and provide empirical data. Resorting to Analysis of Variance (ANOVA), it was shown that team size and connectivity represent parameters with imperative significance in the patrolling mission and it was concluded that meaningful approaches ought to consider adapting to all kinds of environments and explicitly dealing with multi-robot coordination. In fact, this proved to be a crucial aspect, as it was shown that agents' interference can inflict undesirable consequences on collective performance and prevent team scalability, which tends to decrease as more robots are added to the team.

Later on, preliminary distributed models for multi-robot patrol were described. It was shown that the portrayed Bayesian-inspired formalism is able to mixture different variables in a simple and effective way, being successfully accomplished and outperforming all algorithms tested in the benchmarking evaluation that was previously conducted. Significant aspects involved in patrolling missions were further covered, such as decision-making and agents' perception and communication. An extra effort has been made to implement a rigorous and transparent experimental testbed, producing an ambitious system to thoroughly test distributed algorithms, easily reproduce them and adapt the work that followed, so as to promote practical experimentation in this field. Accordingly, it was possible to experimentally validate theory in physical mobile robots, *i.e.*, prove the feasibility of the models proposed. On the upside, these have shown their effectiveness and ability to scale using simple communication mechanisms to cooperatively solve the problem, being robust to robot failures and even communication errors. On the downside, the preliminary model was not complete, since a uniform prior distribution was adopted and the likelihood distribution was hard-defined. As a result, the need to further extend it to a completely generalizable and scalable Bayesian decision framework was felt, strongly encouraged by the exceeding results obtained.

Following the above line of reasoning, an innovative probabilistic strategy for multi-robot patrolling using a team of autonomous mobile robots was derived. This new technique extended the preliminary approach, having the ability to deal with uncertainty and adapt to the system's state by remedying earlier concerns. Additionally, it contributes to the state-of-the-art with a rigorous solution to the long-standing issue of predictability and determinism in patrolling strategies tested in the real-world. Using the preceding experimental testbed in a large indoor infrastructure, successful validation with teams of six physical robots was made possible. The adaptability to constraints such as heterogeneous robots with different speed profile and robot failures was also addressed, coming as a consequence of continuously updating prior knowledge and employing concurrent Bayesian learning to sample the likelihood distribution of each agent. Furthermore, the approach can conveniently scale to a high number of robots due to the negligible bandwidth requirements and because it can effectively handle inter-robot coordination, exhibiting near-optimal results in real-world experiments and surpassing antecedent distributed approaches tested in simulations.

Finally, also regarding scalability, methods for automatic estimation of the optimal team size were designed. These methods consisted in computing estimated worst performance for teams of different size in a given generic environment. With that goal, the worst idleness criterion was adopted to define maximal time intervals, inside which all locations are guaranteed to be patrolled. In short, theoretical analysis on deterministic near-optimal solutions was conducted to extract upper bounds for general patrolling strategies. Classical approaches, namely partitioning-based strategies and TSP cyclic strategies, were examined, and computation based on graph-theory tools using any generic undirected environments provided estimates of the final outcome. According to temporal constraints, it was shown that is possible to estimate the minimal team size using approximation algorithms with known bounds or heuristic algorithms, thus guaranteeing a minimal visiting frequency to all vertices in the environment.

Generally, the author feels that all main objectives have been accomplished in the elaborated work. Nevertheless, on a self-critical note, some aspects could eventually be strengthened. For instance, experimental setup complexity increases with the addition of robots to the team, due to several factors, such as network interruptions, limited energy autonomy provided by batteries, supervising multiple robots simultaneously in a large environment without proper human assistance, and other logistic related issues. These factors have decisively limited the realization of several more trials and new experiments, preventing a more in-depth and extended statistical result analysis. All resources available were exploited in order to have mobile robot teams with maximal size patrolling a large real-world environment. Consequently, the experiments were limited to six physical platforms, even though it would be interesting to further extend team size in the real-world experiments and, as well, carry out experiments in larger and more complex environments.

Beyond the scope of this work, some aspects can eventually be explored in future research in the field. Different patrolling priorities can be considered if some locations need to be visited more often than others. Additionally, even though the work presented supports dynamic environments where robots avoid moving obstacles such as teammates or people, it could be interesting to investigate dynamic navigation graphs of the patrol area in the future. For example, assume that someone closes a door and therefore restricts access to a room; the topological map should be updated and propagated to all members of the team.

By the same token, for tasks which last for a long period of time, resource usage such as energetic autonomy could be considered, and algorithms that minimize the distance traveled by agents are advised. Also, in such situations, when robots are running low on battery, they could eventually drive to recharge stations, and strategies could be devised so that a minimal number R of robots in the team keep performing the patrolling task continuously in time. This brings up diverse new questions like how should other teammates compensate for the inactivity of the recharging robot(s) or how they should schedule their visits to the recharge station to ensure continuous mission execution, depending on the strategy implemented.

One can also think of the circumstance of intruder detection, *e.g.*, leveraging modern LIDARs to detect pedestrians or sensing unknown moving entities. Likewise, one can take advantage of other sensors and extend the decision-making process in the patrolling mission, with the incorporation of new variables in the formulated models in this thesis, according to the scenario of application. For instance, readings from a temperature sensor may be included in the model, guiding robots towards heat sources in the environment. Finally, preemption of patrolling decisions can also be investigated. One can think of a scenario where the robot may regret the original decision while executing it due to changes in some variables in the mission, or an hypothetical situation where the presence of a robot is requested by a teammate.

Bibliography

[ActivMedia, 2006] ActivMedia Robotics, *Pioneer 3 Operations Manual.* Version 3, Mobile Robots Inc., January 2006.

[Agmon et al., 2008] N. Agmon, S. Kraus and G. Kaminka. “Multi-Robot Perimeter Patrol in Adversarial Settings”. In Proceedings of the *IEEE International Conference on Robotics and Automation (ICRA'2008)*, pp. 2339-2345, Pasadena, California, U.S.A, May 19-23, 2008.

[Agmon et al., 2011] N. Agmon, G. Kaminka and S. Kraus, “Multi-Robot Adversarial Patrolling: Facing a Full-Knowledge Opponent”. In *Journal of Artificial Intelligence Research (JAIR)*, vol 42, pp. 887-916, 2011.

[Aguirre and Taboada, 2012] O. Aguirre, H. Taboada, “An Evolutionary Game Theory Approach for Intelligent Patrolling”. In *Procedia Computer Science*, Vol 12, Part II, pp. 140-145, Elsevier, 2012.

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

[Almeida et al., 2004] A. Almeida, G. Ramalho, H. Santana, P. Tedesco, T. Menezes, V. Corruble and Y. Chaveleyre, “Recent Advances on Multi-Agent Patrolling”. In *Advances in Artificial Intelligence, Proceedings of the Brazilian Symposium on Artificial Intelligence (SBIA 2004)*, Lecture Notes in Computer Science, Vol. 3171, pp 474-483, Springer-Verlag, 2004.

[Angluin and Valiant, 1979] D. Angluin and L. Valiant. “Fast Probabilistic Algorithms for Hamiltonian Circuits and Matchings”. In *Journal of Computer and System Sciences*, 155-193, 18(2), April 1979.

[Applegate et al., 2003] D. Applegate, W. Cook and A. Rohe, “Chained Lin-Kernighan for large traveling salesman problems”. In *INFORMS Journal on Computing*, vol. 15, pp. 82-92, 2003.

[Araújo et al., 2012] A. Araújo, D. Portugal, M. Couceiro, C. Figueiredo and R.P. Rocha, “TraxBot: Assembling and Programming of a Mobile Robotic Platform”.

In Proceedings of the 4^{th} *International Conference on Agents and Artificial Intelligence (ICAART'2012)*, pp. 301–304, Vilamoura, Algarve, Portugal, February 6-8, 2012.

[Araújo et al., 2013] A. Araújo, D. Portugal, M. S. Couceiro and R. P. Rocha, "Integrating Arduino-based Educational Mobile Robots in ROS". In Proceedings of the *13th International Conference on Autonomous Robot Systems and Competitions (Robotica 2013)*, pp. 8-13, Lisbon, Portugal, April 25-29, 2013.

[Baglietto et al., 2008] M. Baglietto, G. Cannata, F. Capezio and A. Sgorbissa, "PatrolGRAPH: a Distributed Algorithm for Multi-Robot Patrolling". In *Intelligent Autonomous Systems*, Vol. 10, pp. 415-424, IOS Press, 2008.

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

[Basilico et al., 2009] N. Basilico, N. Gatti, T. Rossi, S. Ceppi and F. Amigoni, "Extending Algorithms for Mobile Robot Patrolling in the Presence of Adversaries to More Realistic Settings". In Proc. of the *2009 IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'09)*, pp. 557-564, Milan, Italy, 2009.

[Banzi, 2011] M. Banzi, *Getting started with Arduino*, O'Reilly Media Inc., 2^{nd} edition, 2011.

[Beeson et al., 2005] P. Beeson, N. Jong, and B. Kuiper, "Towards Autonomous Topological Place Detection Using the Extended Voronoi Graph". In Proceedings of the *International Conference on Robotics and Automation (ICRA'2005)*, pp. 4373-4379, Barcelona, Spain, April, 2005.

[Blumenthal, 1953] L. M. Blumenthal, *Theory and Applications of Distance Geometry*, Chelsea Publishing Company, Bronx, New York, 1953.

[Bonabeau et al., 1999] E. Bonabeau, M. Dorigo and G Theraulaz, *Swarm intelligence: from natural to artificial systems.* Santa Fe Institute Studies in the Sciences of Complexity, Oxford University Press, 1999.

[Bondy and Monty, 1976] J. Bondy and U. Monty, *Graph Theory with Applications.* The Macmillan Press Ltd., 5th Edition, Elsevier Science Publishing Co. Inc, U.S.A., 1976.

[Bradski and Kaehler, 2008] G Bradski and A Kaehler, *Learning OpenCV: Computer vision with the OpenCV library.* O'Reilly Media, 2008.

[Bruyninckx, 2001] H. Bruyninckx, "Open robot control software: the OROCOS project". In Proceedings of the *IEEE International Conference on Robotics and Automation (ICRA'2001)*, Vol. 3, pp. 2523-2528, Seouls, Korea Rep., May 21-26, 2001.

[Cabrita et al., 2010] G. Cabrita, P. Sousa, L. Marques and A. de Almeida, “Infrastructure Monitoring With Multi-Robot Teams”. In Proceedings of the *International Conference on Intelligent Robots and Systems (IROS’10)*, Workshop on Robotics for Environmental Monitoring, Taipei, Taiwan, October 18-22, 2010.

[Cannata et al., 2011] G. Cannata, A. Sgorbissa, “A Minimalist Algorithm for Multirobot Continuous Coverage”. In *IEEE Transactions on Robotics*, 27(2), April, 2011.

[Chevaleyre, 2004] Y. Chevaleyre, “Theoretical analysis of the multi-agent patrolling problem”. In Proceedings of the *2004 IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT’04)*, pp. 302-308, Beijing, China, September 20-24, 2004.

[Christofides, 1975] N. Christofides, *Graph theory: An algorithmic approach (Computer science and applied mathematics)*. Academic Press, Inc. Orlando, FL, USA, 1975.

[Chu et al., 2007] 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 *IEEE International Conference on Tools with Artificial Intelligence (ICTAI’07)*, Vol. 1, 442-449, Paris, France, October 29-31, 2007.

[Chudoba et al., 2011] J. Chudoba, J. Faigl, M. Kulich, T. Krajník, K. Košnar and L. Přeučil, “A Technical Solution of a Robotic E-Learning System in the SyRoTek Project”. In Proc. of the *3rd International Conference on Computer Supported Education (CSEDU 2011)*, Noordwijkerhout, The Netherlands, 6-8 May, 2011.

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

[Couceiro et al., 2012] M. Couceiro, C. Figueiredo, D. Portugal, R. P. Rocha and 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.

[Couceiro et al., 2013] M. S. Couceiro, D. Portugal and R. P. Rocha, “A Collective Robotic Architecture in Search and Rescue Scenarios”. In Proceedings of the *28th ACM Symposium on Applied Computing (SAC 2013)*, Special Track on Cooperative Multi-Agent Systems and Applications, 64-69, Coimbra, Portugal, March 18-22, 2013.

[Diankov and Kuffner, 2008] R. Diankov and J. Kuffner, “OpenRAVE: A Planning Architecture for Autonomous Robotics”. Technical Report *CMU-RI-TR-08-34*, Robotics Institute, Pittsburgh, PA, USA, July, 2008.

[Diestel, 2005] R. Diestel. *Graph Theory*, Third Edition, Graduate Texts in Mathematics, Volume 173, Springer-Verlag Heidelberg, 2005.

[Dijkstra, 1959] E. W. Dijkstra, “A Note on Two Problems in Connection with Graphs”. In *Numerische Math*, 269-271, Vol. 1, 1959.

[Elmaliach et al., 2007] Y. Elmaliach, N. Agmon and G. Kaminka, “Multi-Robot Area Patrol under Frequency Constraints”. In Proceedings of the *2007 IEEE International Conference on Robotics and Automation (ICRA'07)*, 385-390, Rome, Italy, April 10-14, 2007.

[Elor and Bruckstein, 2010] Y. Elor and A. Bruckstein, “Autonomous Multi-Agent Cycle Based Patrolling”. In *Swarm Intelligence, Proceedings of the 7th International Conference on Swarm Intelligence (ANTS'2010)*, Lecture Notes in Computer Science, Vol. 6234, pp 119-130, Springer, 2010.

[Euler, 1736] L. Euler, “Solutio problematis ad geometriam situs pertinentis”. In *Comment. Academiae Sci. I. Petropolitanae*, 8, pp. 128.140, 1736.

[Fazli et al., 2010] P. Fazli, A. Davoodi, P. Pasquier and A. Mackworth. “Fault-Tolerant Multi-Robot Area Coverage with Limited Visibility”. In Proceedings of the 2010 *IEEE International Conference on Robotics and Automation (ICRA'10)*, Search and Pursuit/Evasion in the Physical World: Efficiency, Scalability, and Guarantees Workhop, Anchorage, Alaska, USA, May 3-8, 2010.

[Fazli, 2013] P. Fazli, A. Davoodi and A.K. Mackworth, “Multi-robot repeated area coverage”. In *Autonomous Robots*, 34(4), pp. 251-276, Springer Science, May 2013.

[Fiedler, 1973] M. Fiedler, “Algebraic connectivity of Graphs”. In *Czechoslovak Mathematical Journal*, 23(98), 1973.

[Fox et al., 2006] D. Fox, J. Ko, K. Konolige, B. Limketkai, D. Schulzz and B. Stewart, “Distributed Multi-robot Exploration and Mapping”. In *Proceedings of the IEEE*, 94(7), 1325-1339, July, 2006.

[Furukawa et al., 2006] T. Furukawa, F. Bourgault, B. Lavis and 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.

[Gabriely and Rimon, 2001] Y. Gabriely and E. Rimon, “Spanning-tree based coverage of continuous areas by a mobile robot”. In *Annals of Mathematics and Artificial Intelligence*, Kluwer Academic Publishers, Vol. 31, pp. 77-98, 2001.

[Garey and Johnson, 1979] M.R. Garey and D.S. Johnson, “Computers and Intractability”. Springer, 1979.

[Ge and Fua, 2005] S. S. Ge and C. Fua, “Complete Multi-Robot Coverage of Unknown Environments with Minimum Repeated Coverage”. In Proceedings of the 2005 *IEEE International Conference on Robotics and Automation (ICRA'2005)*, Barcelona, Spain, April, 2005.

[Gerkey et al., 2003] B. Gerkey, R. Vaughan and A. Howard. “The Player/Stage Project: Tools for Multi-Robot and Distributed Sensor Systems”. In Proceedings of the *Intelligent Conference on Advanced Robotics (ICAR'2003)*, 317-323, Coimbra, Portugal, July 2003.

[Gries and Schneider, 1993] D. Gries and F. Schneider. *A Logical Approach to Discrete Math*, Monographs in Computer Science, Springer Verlag, 1993.

[Grisetti et al., 2006] G. Grisetti, C. Stachniss and W. Burgard, “Improved Techniques for Grid Mapping with Rao-Blackwellized Particle Filters”. In *IEEE Transactions on Robotics*, 2006.

[Guo et al., 2007] Y. Guo, L. Parker and R. Madhavan, “Collaborative Robots for Infrastructure Security Applications”. In *Studies in Computational Intelligence (SCI)*, Springer-Verlag Berlin Heidelberg, pp. 185-200, Vol. 50, April 22, 2007.

[Hendrickson and Kolda, 2000] B. Hendrickson and T. G. Kolda, “Graph partitioning models for parallel computing”. In *Parallel Computing*, 26 (12), 1519-1534, Elsevier, November, 2000.

[Hwang et al., 2009] K. Hwang, J. Lin and H. Huang, “Cooperative Patrol Planning of Multi-Robot Systems by a Competitive Auction System”. In Proceedings of the *ICROS-SICE International Joint Conference*, Fukuoka, Japan, August 18-21, 2009.

[Imaging Source, 2013] DFK 21BF04 FireWire CCD Color Camera, Imaging Source. Available: http://www.theimagingsource.com/en_US/products/cameras/firewire-ccd-color/dfk21bf04 (16/09/2013).

[Iocchi et al., 2011] L. Iocchi, L. Marchetti and D. Nardi, “Multi-Robot Patrolling with Coordinated Behaviours in Realistic Environments”. In Proceedings of the *International Conference on Intelligent Robots and Systems (IROS'2011)*, pp. 2796-2801, San Francisco, CA, USA, September 25-30, 2011.

[Ishiwaka et al., 2003] Y. Ishiwaka, T. Sato and Y. Kakazu, “An approach to the pursuit problem on a heterogeneous multiagent system using reinforcement learning”. In *Robotics and Autonomous Systems (RAS)*, Elsevier, 43(4), pp. 245-256, 2003.

[Jackson, 2007] J. Jackson, “Microsoft robotics studio: A technical introduction. In *IEEE Robotics and Automation Magazine*, Dec. 2007.

[Jain et al., 1999] A. Jain, M. Murty and P. Flynn, “Data clustering: a review”. In *ACM Computing Surveys*, Vol. 31, No. 3, 264-323, September 1999.

[Jansen and Nielsen, 2007] F. Jansen and T. Nielsen, *Bayesian Networks and Decision Graphs*, 2nd Edition, Springer Verlag, 2007.

[Jensen et al., 2011] E. Jensen, M. Franklin, S. Lahr and M. Gini, “Sustainable Multi-Robot Patrol of an Open Polyline”. In Proc. of the *2011 IEEE International Conference on Robotics and Automation (ICRA'11)*, Shanghai, China, May 9-13, 2011.

[Julian et al., 2012] B. Julian, M. Angermann, M. Schwager, D. Rus, “Distributed Robotic Sensor Networks: An Information Theoretic Approach”. In the *International Journal of Robotics Research (IJRR)*, August 2012.

[Karypis et al., 1998] G. Karypis and V. Kumar, “A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs”. In *Journal of Scientific Computing*, Society for Industrial and Applied Mathematics (SIAM), pp. 359-392, 20(1), 1998.

[Keskin et al., 2012] B. B. Keskin, S. Li, D. Steil, S. Spiller, “Analysis of an integrated maximum covering and patrol routing problem”. In *Transportation Research Part E: Logistics and Transportation*, Vol. 48, pp. 215-232, Elsevier, 2012.

[Koenig and Howard, 2004] N. Koenig and A. Howard, “Design and use paradigms for Gazebo, an open-source multi-robot simulator”. In Proceedings of the *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2004)*, pp. 2149-2159, Sendai, Japan, 28 Sep.-2 Oct., 2004.

[Korte and Vygen, 2007] B. Korte and J. Vygen, “Combinatorial Optimization: Theory and Algorithms, 4th Edition, Series Algorithmics and Combinatorics, Vol. 21, Springer Verlag, 2007.

[Kruskal, 1956] J.B. Kruskal, “On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem”. In Proc. of the *American Mathematical Society*, 7(1), pp. 48-50, February, 1956.

[Machado, 2002] A. Machado, *Patrulha Multiagente: Uma Análise Empírica e Sistemática.* M.Sc. Thesis, Centro de Informática, Universidade Federal de Pernambuco (UFPE), Recife, Brasil, 2002. (In Portuguese)

[Machado et al., 2003] A. Machado, G. Ramalho, J. Zucker and A. Drogoul, “Multi-Agent Patrolling: an Empirical Analysis of Alternative Architectures”. In *Multi-Agent-Based Simulation II*, Lecture Notes in Computer Science, Vol. 2581, pp. 155-170, Springer-Verlag, 2003.

[Marder-Eppstein et al., 2010] E. Marder-Eppstein, E. Berger, T. Foote, B. Gerkey and K. Konolige. “The Office Marathon: Robust Navigation in an Indoor Office Environment”. In Proceedings of the *2010 IEEE International Conference on Robotics and Automation (ICRA’2010)*, pp. 300-307, Anchorage, Alaska, USA, May 3-8, 2010.

[Marier et al., 2009] J. Marier, C. Besse and B. Chaib-draa, “A Markov Model for Multiagent Patrolling in Continuous Time”. In Proceedings of the *16th International Conference on Neural Information Processing (ICONIP’2009)*, pp. 648-656, Bangkok, Thailand, Dec. 1-5, 2009.

[Marier et al., 2010] J. Marier, C. Besse and B. Chaib-draa, “Solving the Continuous Time Multiagent Patrol Problem”. In Proceedings of the *2010 IEEE International*

Conference on Robotics and Automation (ICRA'10), Anchorage, Alaska, USA, May, 2010.

[Marino et al., 2009] A. Marino, L. Parker, G. Antonelli, F. Caccavale, "Behavioral control for multi-robot perimeter patrol: a finite state automata approach". In Proceedings of the *IEEE International Conference on Robotics and Automation (ICRA'09)*, pp. 831-836, Kobe, Japan, May, 2009.

[Marino et al., 2012] A. Marino, G. Antonelli, A. P. Aguiar and A. Pascoal, "A new approach to multi-robot harbour patrolling: theory and experiments". In Proceedings of the *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems* (IROS'2012), Vilamoura, Portugal, Oct. 7-12, 2012.

[Marjovi et al., 2009] A. Marjovi, J. Nunes, L. Marques and A. Almeida. "Multi-Robot Fire Searching in Unknown Environment". In Proceedings of the *7th International Conference on Fields and Service Robots*, Cambridge, Massachusetts, U.S.A., July 14-16, 2009.

[Menezes et al., 2006] T. Menezes, P. Tedesco and G. Ramalho, "Negotiator Agents for the Patrolling Task". In *Advances in Artificial Intelligence, Proceedings of the 2nd International Joint Conference, 10th Ibero-American Conference on AI and 18th Brazilian AI Symposium (IBERAMA-SBIA 2006)*, Lecture Notes in Computer Science, Vol. 4140, pp. 48-57, Springer, 2006.

[Metta et al., 2006] G. Metta, P. Fitzpatrick, L. Natale, "Yarp: Yet another robot platform". In *International Journal on Advanced Robotics Systems (IJARS)*, 3(1), pp.43-48, 2006.

[Montemerlo et al., 2003] M. Montemerlo, N. Roy, S. Thrun, "Perspectives on Standardization in Mobile Robot Programming: The Carneggie Mellon Navigation (CARMEN) toolkit". In Proceedings of the *2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'2003).* Las Vegas, Nevada, October, 2003.

[Moreira et al., 2007] D. Moreira, L. Filho, P. Tedesco and G. Ramalho, "SimPatrol: Establishing a Testbed for Multi-agent Patrolling". In the Proceedings of the *6th Brazilian Symposium on Computer Games and Digital Entertainment*, São Leopoldo, Brazil, November 7-9, 2007.

[Murphy, 2004] R. Murphy, "Human-Robot Interaction in Rescue Robotics". In *IEEE Transactions on Systems, Man and Cybernetics - Part C: Applications and Reviews*, 34(2), pp. 138-153, May 2004.

[Murphy et. al, 2009] R. Murphy, J. Kravitz, S. Stover, R. Shoureshi, "Mobile robots in mine rescue and recovery". In *IEEE Robotics & Automation Magazine*, 16(2), pp. 91-103, 2009.

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

[Parker et. al, 2004] L. E. Parker, B. Kannan, F. Tang and M. Bailey, "Tightly-Coupled Navigation Assistance in Heterogeneous Multi-Robot Teams". In Proceedings of the *IEEE International Conference on Intelligent Robots and Systems (IROS 2004)*, Sendai, Japan, 28 Sep.-2 Oct., 2004.

[Parker, 2008] L.E. Parker, "Distributed Intelligence: Overview of the Field and its Application in Multi-Robot Systems". In *Journal of Physical Agents*, Special issue on Multi-robot systems, 2(2), pp. 5-14, March 2008.

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

[Pasqualetti et al., 2012b] F. Pasqualetti, J. Durham and F. Bullo, "Cooperative Patrolling via Weighted Tours: Performance Analysis and Distributed Algorithms". In *IEEE Transactions on Robotics*, 9, 1-8, 2012.

[Pippin et al., 2013] C. Pippin, H. Christensen and L. Weiss, "Performance Based Task Assignment in Multi-Robot Patrolling". In Proc. of the *2013 ACM Symposium on Applied Computing (SAC '13)*, Coimbra, Portugal, March 18-22, pp. 70-76, 2013.

[Pita et al., 2011] J. Pita, M. Tambe, C. Kiekintveld, S. Cullen and E. Steigerwald, "GUARDS-innovative application of game theory for national airport security". In Proceedings of the *22nd International Joint Conference on Artificial Intelligence (IJCAI'11)*, Vol. 3, pp. 2710-2715 Barcelona, Spain, July 16-22, 2011.

[Portugal, 2009] D. Portugal, *RoboCops: A Study of Coordination Algorithms for Autonomous Mobile Robots in Patrolling Missions*, Master of Science Dissertation, Faculty of Science and Technology, University of Coimbra, Portugal, September, 2009.

[Portugal and Rocha, 2010] 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)*, Special Track on Intelligent Robotic Systems, pp. 1271-1276, Sierre, Switzerland, March 22-26, 2010.

[Portugal et al., 2010] D. Portugal, C. Henggeler Antunes and R. Rocha, "A Study of Genetic Algorithms for Approximating the Longest Path in Generic Graphs". In Proceedings of the 2010 IEEE International Conference on Systems, Man and Cybernetics (SMC 2010), 2539-2544, Istanbul, Turkey, October 10-13, 2010.

[Portugal and Rocha, 2011a] D. Portugal and R. P. Rocha, "Partitioning Generic Graphs into *k* Balanced Subgraphs". In Proceedings of the *6th Iberian Congress On Numerical Methods in Engineering (CMNE 2011)*, Coimbra, Portugal, June 14-17, 2011.

[Portugal and Rocha, 2011b] D. Portugal and R. Rocha, "A Survey on Multi-Robot Patrolling Algorithms". In *Technological Innovation for Sustainability*, Proc. of the 2nd Doctoral Conference on Computing, Electrical and Industrial Systems (DoCEIS 11), Costa da Caparica, Lisbon, Portugal, Feb. 21-23, 2011, Vol. 349, pp. 139-146, Springer.

[Portugal and Rocha, 2011c] D. Portugal and R.P. Rocha, "On the Performance and Scalability of Multi-Robot Patrolling Algorithms". In Proceedings of the *9th International Symposium on Safety, Security and Rescue Robotics (SSRR'2011)*, pp. 50-55, Kyoto, Japan, November 1-5, 2011.

[Portugal and Rocha, 2012a] D. Portugal and 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)*, pp. 137-143, Vilamoura, Algarve, Portugal, February 6-8, 2012.

[Portugal and Rocha, 2012b] D. Portugal and R. P. Rocha, "Measuring Variables Effect to Statistically Model the Multi-Robot Patrolling Problem by means of ANOVA". In *Technological Innovation for Sustainability*, Proc. of the 3rd Doctoral Conference on Computing, Electrical and Industrial Systems (DoCEIS 12), Costa da Caparica, Lisbon, Portugal, Feb. 21-23, 2011, Vol. 349, pp. 199-206, Springer.

[Portugal and Rocha, 2012c] D. Portugal and R.P. Rocha, "Decision Methods for Distributed Multi-Robot Patrol". In Proceedings of the *10th International Symposium on Safety, Security and Rescue Robotics (SSRR'2012)*, College Station, Texas, USA, Nov. 8-11, 2012.

[Portugal et al., 2013] D. Portugal, M. Couceiro and R. P. Rocha, "Concurrent Bayesian Learners for Multi-Robot Patrolling Missions". In Proceedings of the *2013 IEEE International Conference on Robotics and Automation (ICRA 2013)*, Workshop on Towards Fully Decentralized Multi-Robot Systems: Hardware, Software and Integration, Karlsruhe, Germany, May 6-10, 2013.

[Portugal and Rocha, 2013a] D. Portugal and R. P. Rocha, "Scalable, Fault-Tolerant and Distributed Multi-Robot Patrol in Real World Environments". In Proceedings of the *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2013)*, Tokyo, Japan, Nov. 3-7, 2013.

[Portugal and Rocha, 2013b] D. Portugal and R.P. Rocha, "Applying Bayesian Learning to Multi-Robot Patrol". In Proceedings of the *11th International Symposium on Safety, Security and Rescue Robotics (SSRR'2013)*, Linköping, Sweden, Oct. 21-26, 2013. (Accepted)

[Portugal and Rocha, 2013c] D. Portugal and R. P. Rocha, “Retrieving Topological Information for Mobile Robots Provided with Grid Maps”. In *Agents and Artificial Intelligence*, Communications in Computer and Information Science (CCIS) series, Vol. 358, Springer-Verlag Berlin Heidelberg, pp. 204-217, 2013.

[Portugal and Rocha, 2013d] D. Portugal and R.P. Rocha, “Multi-Robot Patrolling Algorithms: Examining Performance and Scalability”. In *Advanced Robotics Journal*, Special Issue on Safety, Security, and Rescue Robotics, 27 (5), pp. 325-336, March, 2013.

[Portugal and Rocha, 2013e] D. Portugal and R.P. Rocha, “Distributed Multi-Robot Patrol: A Scalable and Fault-Tolerant Framework”. In *Robotics and Autonomous Systems (RAS)* Journal, Elsevier, 2013. (In Press)

[Portugal and Rocha, 2014a] D. Portugal and R.P. Rocha, “Cooperative Multi-Robot Patrol with Bayesian Learning”. In *Artificial Intelligence* Journal, Elsevier, 2013. (Submitted)

[Portugal et al., 2014b] D. Portugal, C Pippin, H. Christensen and R. P. Rocha, “Finding Optimal Routes for Multi-Robot Patrolling in Generic Graphs”. In Proceedings of the *IEEE International Conference on Robotics and Automation (ICRA’14)*, Hong Kong, May 31-June 5, 2014. (Submitted)

[Poulet et al., 2012] C. Poulet, V. Corruble and 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, Nov. 7-9, 2012.

[Prim, 1957] R. Prim, “Shortest connection networks and some generalizations”. In *Bell System Technical Journal*, Vol. 23, 1389-1401, 1957.

[Quigley et al., 2009] 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 Proceedings of the *IEEE International Conference on Robotics and Automation (ICRA’2009)*, Workshop On Open Source Software, Kobe, Japan, May 12-17, 2009.

[Rocha, 2005] R. Rocha, *Building Volumetric Maps with Cooperative Mobile Robots and Useful Information Sharing: a Distributed Control Approach based on Entropy*, Ph.D. Thesis, Faculty of Engineering of University of Porto, Portugal, October, 2005.

[Rosenfeld et al., 2006] A. Rosenfeld, G. Kaminka and S. Kraus, “A Study of Scalability Properties in Robotic Teams”. In *Coordination of Large-Scale Multiagent Systems*, pp. 27-51, Springer, 2006.

[Ruan et al., 2005] S. Ruan, C. Meirina, F. Yu, K.R. Pattipati and R.L. Popp, “Patrolling in A Stochastic Environment”. In Proceedings of the *10th International*

Command and Control Research and Technology Symposium, McLean, Virginia, USA, June 13-16, 2005.

[Rusu and Cousins, 2011] R. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)". In Proceeding of the *IEEE International Conference on Robotics and Automation (ICRA'2011)*, Shanghai, China, May 2011.

[Sak et al., 2008] T. Sak, J. Wainer and S. Goldenstein, "Probabilistic Multiagent Patrolling". In *Advances in Artificial Intelligence, Proceedings of the 19th Brazilian Symposium on Artificial Intelligence (SBIA 2008)*, Lecture Notes in Computer Science, Vol. 5249, pp. 124-133, Springer, 2008.

[Sampaio et al., 2010] P. Sampaio, G. Ramalho and P. Tedesco, "The Gravitational Strategy for the Timed Patrolling". In Proceedings of the *IEEE International Conference on Tools with Artificial Intelligence (ICTAI'10)*, 113-120, Arras, France, Oct. 27-29, 2010.

[Santana et al., 2004] H. Santana, G. Ramalho, V. Corruble and B. Ratitch, "Multi-Agent Patrolling with Reinforcement Learning". In Proceedings of the Third *International Joint Conference on Autonomous Agents and Multiagent Systems*, Vol. 3, pp. 1122-1129, New York, USA, 2004.

[Scheffé, 1959] H. Scheffé. *The Analysis of Variance*, John Wiley & Sons, New York, 1959.

[Schmickl et al., 2007] T. Schmickl, C. Möslinger and K. Crailsheim, "Collective Perception in a Robot Swarm". In *Swarm Robotics*, Lecture Notes in Computer Science, Vol. 4433, pp. 144-157, Springer, 2007.

[Sempé, 2004] F. Sempé, *Auto-organisation d'une collectivité de robots: Application à l'activité de patrouille en présence de perturbations*. Ph.D. Thesis, Université Pierre et Marie Curie, École Doctorale d'Informatique, Télécommunications et Electronique de Paris, France, January, 2004. (In French)

[Sempé and Drougoul, 2003] F. Sempé and 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, Nevada, USA, October, 2003.

[Sherman, 1992] T. Sherman, "Recent Results in Art Galleries". In Proceedings of the IEEE, 80(9), pp. 1384-1399, 1992.

[Smith et al., 2010] S. Smith and D. Rus, "Multi-Robot Monitoring in Dynamic Environments with Guaranteed Currency of Observations". In Proceedings of the *49th IEEE Conference on Decision and Control*, pp. 514-521, Atlanta, Georgia, USA, December 15-17, 2010.

[Stone and Veloso, 2000] P. Stone and M. Veloso, “Multiagent Systems: A survey from a machine learning perspective”. In *Autonomous Robots*, 8(3), pp. 345-383, July 2000.

[Stranders et al., 2013] R. Stranders, E.M. de Cote, A. Rogers and N.R. Jennings, “Near-optimal continuous patrolling with teams of mobile information gathering agents”. In *Artificial Intelligence*, Vol. 195, pp. 63-105, Elsevier, February 2013.

[Strom et al., 2010] J. Strom, R. Morton, K. Reilly and E. Olson, “Online Probabilistic Pursuit of Adversarial Evaders”. In Proceedings of the *2010 IEEE International Conference on Robotics and Automation (ICRA'10)*, Anchorage, Alaska, USA, May 3-8, 2010.

[Sweeney et al., 2003] J. Sweeney, H. Li and R. Grupen, “Scalability and schedulability in large, coordinated, distributed robot systems”. In Proceedings of the *IEEE International Conferance on Robotic Applications (ICRA 2003)*, pp. 4074-4079, Taipei, Taiwan, September 14-19, 2003.

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

[Vaughan, 2008] R. Vaughan, “Massively multi-robot simulation in Stage”. In *Journal of Swarm Intelligence*, Vol. 2, Issue 2-4, pp. 189-208, Springer, 2008.

[Vieira et al., 2009] M. Vieira, R. Govindan and G. Sukhatme, “Scalable and practical pursuit-evasion with networked robots”. In Intelligent Service Robotics, Springer-Verlag, Volume 2, Issue 4, pp. 247-263, October, 2009.

[Voronoi, 1908] G. Voronoi, “Nouvelles applications des paramètres continus à la théorie des formes quadratiques”. In *Journal für die Reine und Angewandte Mathematik* 134, pp. 198-287, 1908

[Webster's, 2013] *Webster's Online Dictionary*, September, 2013.
Available at: www.websters-online-dictionary.org

[Yanovski et al., 2003] V. Yanovski, I.A. Wagner and A.M. Bruckstein, “A Distributed Ant Algorithm for Efficiently Patrolling a Network”. In *Algorithmica*, Vol. 37, pp. 165-186, 2003.

[Zaman et al., 2011] S. Zaman, W. Slany and G. Steinbauer, “ROS-based Mapping, Localization and Autonomous Navigation using a Pioneer 3-DX Robot and their Relevant Issues”. In Proceedings of the *IEEE Saudi International Electronics, Communications and Photonics Conference (SIECPC'2011)*, Riad, Saudi-Arabia, 24-26 April, 2011.

[Zlot et al., 2005] R. Zlot and A. Stentz, "Complex task allocation for multiple robots". In Proceedings of the *IEEE International Conference on Robotics and Automation (ICRA'2005)*, pp. 1515-1522, Barcelona, Spain, 18-22 April, 2005.

Appendix A

List of Publications

Most of this thesis is essentially based on the following publications and achievements:

A.1 Peer-Reviewed Journal Articles

[4] **D. Portugal** and R.P. Rocha, “On Estimation of Team Dimensions for Multi-Robot Patrol”. In *Journal to be selected*, ongoing work.

[3] **D. Portugal** and R.P. Rocha, “Cooperative Multi-Robot Patrol with Bayesian Learning”. In *Artificial Intelligence* Journal, Elsevier, 2013. (Under Review)

[2] **D. Portugal** and R.P. Rocha, “Distributed Multi-Robot Patrol: A Scalable and Fault-Tolerant Framework”. In *Robotics and Autonomous Systems (RAS)* Journal, Elsevier, 2013. DOI: 10.1016/j.robot.2013.06.011.

[1] **D. Portugal** and R.P. Rocha, “Multi-Robot Patrolling Algorithms: Examining Performance and Scalability”. In *Advanced Robotics Journal*, Special Issue on Safety, Security, and Rescue Robotics, 27 (5), pp. 325–336, March, 2013. DOI: 10.1080/01691864.2013.763722.

A.2 Peer-Reviewed Book Chapters

[3] **D. Portugal** and R. P. Rocha, “Retrieving Topological Information for Mobile Robots Provided with Grid Maps”. In *Agents and Artificial Intelligence*, Communications in Computer and Information Science (CCIS) series, Vol. 358, Springer-Verlag Berlin Heidelberg, pp. 204–217, 2013. DOI: 10.1007/978-3-642-36907-0_14.

[2] **D. Portugal** and R. P. Rocha, “Measuring Variables Effect to Statistically Model the Multi-Robot Patrolling Problem by means of ANOVA”, In *Technological Innovation for Sustainability*, Proc. of the 3rd Doctoral Conference on Computing, Electrical and Industrial Systems (DoCEIS 12), Costa da Caparica, Lisbon, Portugal, Feb. 21–23, 2011, Vol. 349, pp. 199–206, Springer. DOI: 10.1007/978-3-642-28255-3_22.

[1] **D. Portugal** and R. Rocha, “A Survey on Multi-Robot Patrolling Algorithms”. In *Technological Innovation for Sustainability*, Proc. of the 2nd Doctoral Conference on Computing, Electrical and Industrial Systems (DoCEIS 11), Costa da Caparica, Lisbon, Portugal, Feb. 21–23, 2011, Vol. 349, pp. 139–146, Springer. DOI: 10.1007/978-3-642-19170-1_15.

A.3 International Awards

[1] IEEE SSRR’2011 Young Author’s Award for the paper “On the Performance and Scalability of Multi-Robot Patrolling Algorithms”, at the 9th IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR’2011), Kyoto, Japan, pp. 50–55, Nov. 1–5, 2011[37].

[37] More information available at: http://mrl.isr.uc.pt/?w=news_information&ID=10

A.4 Peer-Reviewed International Conferences

[10] **D. Portugal**, C Pippin, H. Christensen and R. P. Rocha, “Finding Optimal Routes for Multi-Robot Patrolling in Generic Graphs”, In Proceedings of the *IEEE International Conference on Robotics and Automation (ICRA’14)*, Hong Kong, May 31–June 5, 2014. (Under Review)

[9] **D. Portugal** and R.P. Rocha, “Applying Bayesian Learning to Multi-Robot Patrol”. In Proceedings of the *11th IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR’2013)*, Linköping, Sweden, Oct. 21–26, 2013.

[8] **D. Portugal** and R. P. Rocha, “Scalable, Fault-Tolerant and Distributed Multi-Robot Patrol in Real World Environments”. In Proceedings of the *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2013)*, Tokyo, Japan, Nov. 3–7, 2013.

[7] **D. Portugal**, M. Couceiro and R. P. Rocha, “Concurrent Bayesian Learners for Multi-Robot Patrolling Missions”, In Proceedings of the *2013 IEEE International Conference on Robotics and Automation (ICRA 2013)*, Workshop on Towards Fully Decentralized Multi-Robot Systems: Hardware, Software and Integration, Karlsruhe, Germany, May 6–10, 2013.

[6] **D. Portugal** and 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, Nov. 8–11, 2012. DOI: 10.1109/SSRR.2012.6523869.

[5] **D. Portugal** and 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)*, pp. 137–143, Vilamoura, Algarve, Portugal, February 6–8, 2012.

[4] **D. Portugal** and R.P. Rocha, “On the Performance and Scalability of Multi-Robot Patrolling Algorithms”. In Proceedings of the *9th IEEE International*

Symposium on Safety, Security and Rescue Robotics (SSRR'2011), pp. 50–55, Kyoto, Japan, November 1–5, 2011. DOI: 10.1109/SSRR.2011.6106761.

[3] **D. Portugal** and R. P. Rocha, "Partitioning Generic Graphs into k Balanced Subgraphs". In Proceedings of the *6th Iberian Congress On Numerical Methods in Engineering (CMNE 2011)*, Coimbra, Portugal, June 14–17, 2011.

[2] **D. Portugal**, C. Henggeler Antunes and R. Rocha, "A Study of Genetic Algorithms for Approximating the Longest Path in Generic Graphs". In Proceedings of the 2010 IEEE International Conference on Systems, Man and Cybernetics (SMC 2010), 2539-2544, Istanbul, Turkey, October 10-13, 2010. DOI: 10.1109/ICSMC.2010.5641920.

[1] **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)*, Special Track on Intelligent Robotic Systems, pp. 1271–1276, Sierre, Switzerland, March 22–26, 2010. DOI: 10.1145/1774088.1774360.

Appendix B

Collaborations

During the four years of these Ph.D. studies, there have been some collaborations with several other people. As a complement to the Ph.D. research, the author has co-supervised five Master Dissertation thesis on Electrical Engineering and Computers, by André Araújo, Amadeu Fernandes, Walter Miani (2011/2012), João M. Santos and João Martins (2012/2013), and is currently co-supervising two Master Dissertation studies in 2013/2014, of students Gonçalo Santos Martins and Francisco Ferrer Sales.

The author was also a Teacher monitor of practical classes at the Department of Electrical Engineering and Computers (DEEC) in three different semesters, namely of "Microprocessor Systems" during the second semester of 2009/2010, "Digital Systems Laboratory" during the first semester of 2010/2011 and also in the first semester of 2011/2012.

Furthermore, the author has been an active research member in the R&D project CHOPIN since April 2012, and has collaborated in the Ph.D. studies of colleague Micael Santos Couceiro. As a result of all these collaborations, the following publications have been obtained:

B.1 Peer-Reviewed Journal Articles

[5] A. Fernandes, M. S. Couceiro, **D. Portugal** and R.P. Rocha, “Ad Hoc Communication in Teams of Mobile Robots Using ZigBee Technology”. In *Journal to be selected*, ongoing work.

[4] M. S. Couceiro, **D. Portugal**, A. Araújo, and R. P. Rocha, “Fostering Human-Robot Cooperative Architectures for Search and Rescue Missions in Urban Fires”, In *Mathematics and Computers in Simulation*, Elsevier, 2014. (Under Review).

[3] A. Araújo, **D. Portugal**, M. S. Couceiro and R. P. Rocha, “Integrating Arduino-based Educational Mobile Robots in ROS”. In *Journal of Intelligent and Robotic Systems (JINT)*, Special Issue on Autonomous Robot Systems, Springer, 2014. (Under Review)

[2] M. S. Couceiro, **D. Portugal**, R. P. Rocha and N. M. Ferreira, “Marsupial Teams of Robots: Deployment of Miniature Robots for Swarm Exploration under Communication Constraints”, In *Robotica Journal*, Cambridge Unversity Press, 2013. (Under Review)

[1] M. S. Couceiro, **D. Portugal**, N. Gonçalves, R. P. Rocha, J. M. Luz, C. M. Figueiredo and G. Dias, “A Methodology for Detection and Estimation in the Analysis of the Golf Putting”, In *Journal of Pattern Analysis and Applications (PAA)*, 16 (3), pp. 459-474, Springer, August 2013. DOI: 10.1007/978-94-007-4722-7_4.

B.2 Peer-Reviewed Book Chapters

[1] J. M. Luz, M. S. Couceiro, **D. Portugal**, R. P. Rocha, H. Araújo and G. Dias, “Comparison of Classification Methods for Golf Putting Performance Analysis”, In *Computational Intelligence and Decision Making*, pp. 35–45, Dordrecht: Springer Netherlands, 2013. DOI: 10.1007/978-94-007-4722-7_4.

B.3 MSc. Dissertations as co-supervisor

[5] J. M. Santos, "SmokeNav – Simultaneous Localization and Mapping in Reduced Visibility Scenarios". Master of Science Dissertation, Department of Electrical Engineering and Computers, Faculty of Science and Technology, University of Coimbra, Portugal, September, 2013.
Supervised by: Prof. Dr. Rui P. Rocha and **D. Portugal**.

[4] J. A. Martins, "MRSLAM – Multi-Robot Simultaneous Localization and Mapping". Master of Science Dissertation, Department of Electrical Engineering and Computers, Faculty of Science and Technology, University of Coimbra, Portugal, September, 2013.
Supervised by: Prof. Dr. Rui P. Rocha and **D. Portugal**.

[3] W. Miani, "Arduino programming using Matlab and Simulink". Erasmus Program Master of Science Dissertation, Department of Electrical, Management and Mechanical Engineering, Faculty of Engineering, University of Udine, October, 2012.
Supervised by: Prof. Dr. Pier Luca Montessoro, Prof. Dr. Rui P. Rocha, **D. Portugal** and M. S. Couceiro.

[2] A. Fernandes, "Comunicação Ad Hoc em Equipas de Robôs Móveis Utilizando a Tecnologia ZigBee". Master of Science Dissertation, Department of Electrical Engineering and Computers, Faculty of Science and Technology, University of Coimbra, Portugal, September, 2012.
Supervised by: Prof. Dr. Rui P. Rocha, **D. Portugal** and M. S. Couceiro.

[1] A. Araújo, "ROSint – Integration of a mobile robot in ROS architecture". Master of Science Dissertation, Department of Electrical Engineering and Computers, Faculty of Science and Technology, University of Coimbra, Portugal, July, 2012.
Supervised by: Prof. Dr. Rui P. Rocha, **D. Portugal** and M. S. Couceiro.

B.4 Peer-Reviewed International Conferences

[8] J. M. Santos, **D. Portugal** and R. P. Rocha “An Evaluation of 2D SLAM Techniques Availabile in Robot Operating System”. In Proceedings of the *2013 International Symposium on Safety, Security and Rescue Robotics (SSRR 2013)*, Linköping, Sweden, Oct 21–26, 2013.

[7] R. P. Rocha, **D. Portugal**, M. Couceiro, F. Araújo, P. Menezes and J. Lobo, “The CHOPIN project: Cooperation between Human and rObotic teams in catastroPhic INcidents”. In Proceedings of the *2013 IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR 2013)*, Linköping, Sweden, Oct 21–26, 2013.

[6] A. Araújo, **D. Portugal**, M. S. Couceiro and R. P. Rocha, “ntegrating Arduino-based Educational Mobile Robots in ROS”. In Proceedings of the *13th IEEE International Conference on Autonomous Robot Systems and Competitions (Robotica 2013)*, 8–13, Lisbon, Portugal, April 25–29, 2013.

[5] M. S. Couceiro, **D. Portugal** and R. P. Rocha, “A Collective Robotic Architecture in Search and Rescue Scenarios”. In Proceedings of the *28th ACM Symposium on Applied Computing (SAC 2013)*, Special Track on Cooperative Multi-Agent Systems and Applications, 64–69, Coimbra, Portugal, March 18–22, 2013. DOI: 10.1145/2480362.2480377.

[4] M. S. Couceiro, C. M. 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/RSJ International Conference on Intelligent Robots and Systems (IROS 2012), pp. 4614–4619, Vilamoura, Algarve, October 7–12, 2012. DOI: 10.1109/IROS.2012.6385695.

[3] A. Araújo, **D. Portugal**, M. S. Couceiro, C. M. Figueiredo and R. P. Rocha, “Small and Compact Mobile Robots: Surveying and Comparing Platforms”. In Proceedings of the *Automatics International Conference (AUTOMATICS 2012)*, 349-358, Sozopol, Bulgaria, June 1–4, 2012.

[2] A. Araújo, **D. Portugal**, M. S. Couceiro, C. M. Figueiredo and 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)*, 301-304, Vilamoura, Algarve, Portugal, February 6–8, 2012.

[1] J. M. Luz, M. S. Couceiro, **D. Portugal**, R. P. Rocha, H. AraÃžjo and G. Dias, "Comparison of Classification Methods for Golf Putting Performance Analysis". In Proceedings of the *International Symposium on Computational Intelligence for Engineering Systems (ISCIES 2011)*, Coimbra, Portugal, November 16-18, 2011.

Appendix C

Web Material

The following is a list of web links related to the author's work during the course of this thesis.

Author's Videos	http://isr.uc.pt/~davidbsportugal/videos
Author's Website	http://isr.uc.pt/~davidbsportugal
CHOPIN Project	http://chopin.isr.uc.pt
DEEC Website	http://www.uc.pt/fctuc/deec
ISR Website	http://www.isr.uc.pt
MRL Website	http://mrl.isr.uc.pt
Prof. R. P. Rocha Website	http://www.deec.uc.pt/~rprocha
ROS@ISR Event	http://wiki.ros.org/Events/ROSatISRCoimbra
ROS MRL Robots Stack	http://wiki.ros.org/mrl_robots
ROS Patrol Stack	http://wiki.ros.org/patrol

ROS Patrolling Sim Package	http://wiki.ros.org/patrolling_sim
Robotic Tools Workshop (RTW)	http://mrl.isr.uc.pt/projects/socialrobot/index.php?option=com_content&view=article&id=72
Robotic Sensing Workshop (RSW)	http://mrl.isr.uc.pt/projects/socialrobot/index.php?option=com_content&view=article&id=73