

Micael Santos Couceiro

Evolutionary Robot Swarms Under Real-World Constraints

Tese de Doutoramento na área científica de Engenharia Electrotécnica e de Computadores, especialidade de Automação e Robótica, orientada pelo Professor Doutor Rui Paulo Pinto da Rocha e co-orientada pelo Professor Doutor Nuno Miguel Fonseca Ferreira apresentada ao Departamento de Engenharia Electrotécnica e de Computadores da Faculdade de Ciências e Tecnologia da Universidade de Coimbra.

Setembro de 2013



UNIVERSIDADE DE COIMBRA



FCTUC

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

EVOLUTIONARY ROBOT SWARMS UNDER REAL-WORLD CONSTRAINTS

by

Micael S. Couceiro
MS.c.

A Thesis submitted to the Electrical and Computer Engineering Department of the Faculty of Sciences and Technology of the University of Coimbra in partial fulfilment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

© Micael Couceiro, 2013
University of Coimbra

All rights reserved. This Thesis may not be reproduced in whole or in part, by photocopy or other means, without the permission of the author.

Supervision

EVOLUTIONARY ROBOT SWARMS UNDER REAL-WORLD CONSTRAINTS

by

Micael S. Couceiro
MS.c.

This thesis was prepared under the Supervision of:

Doctor Rui Paulo Pinto da Rocha
Assistant Professor
Electrical and Computer Engineering Department
Faculty of Sciences and Technology, University of Coimbra (UC)

And the Co-Supervision of:

Doctor Nuno Miguel Fonseca Ferreira
Assistant Professor
Electrical Engineering Department
Engineering Institute of Coimbra (ISEC), Polytechnic Institute of Coimbra (IPC)

Committee

EVOLUTIONARY ROBOT SWARMS UNDER REAL-WORLD CONSTRAINTS

by

Micael S. Couceiro
MS.c.

Supervisory Committee

Doctor Aníbal Traça de Carvalho Almeida, University of Coimbra (FCTUC) (**Chairman**)

Doctor Luís Miguel Parreira e Correia, University of Lisbon (FCUL)

Doctor António Paulo Gomes Mendes Moreira, University of Porto (FEUP)

Doctor Lino José Forte Marques, University of Coimbra (FCTUC)

Doctor Rui Paulo Pinto da Rocha, University of Coimbra (FCTUC) (**Supervisor**)

*To all who believed in me:
family, friends, girlfriend and professors.*

“While human ingenuity may devise various inventions to the same ends, it will never devise anything more beautiful, nor more simple, nor more to the purpose than nature does, because in her inventions nothing is lacking and nothing is superfluous.”

Leonardo Da Vinci in “Thoughts on Art and Life”

Acknowledgments

I want to thank a very special group that gave me some words of advice, encouragement, or even just some subliminal messages, that I hereby share with the reader.

Among these people I would like to highlight my supervisor Prof. Rui Rocha who advised me and, above all, encouraged me throughout this time – it would have been rather difficult to achieve this goal without his regular intervention and meticulous reading club sessions.

“Your Ph.D. is like your house. You need to have a very solid and structured one; nevertheless, it also needs to be cosy and extremely comfortable for your guests.” Rui Rocha

To my co-supervisor Prof. Nuno Ferreira who always supported me and gave me every opportunity to go further – more than just providing the solution is to show the way.

“You need to consider the Ph.D. as a marathon where it’s all about endurance over speed. If you run that extra mile, you may not get ‘till the end of it.” Nuno Ferreira

To Prof. Patricia Vargas that thoroughly supervised my work during the three months I was at Heriot-Watt University – studying abroad is the opportunity of a lifetime.

“I see you as a Ph.D. already; only your viva is missing.” Patricia Vargas

To Prof. Tenreiro Machado for his sharp and culminating proof readings – to maintain such a balance between agreeableness and conscientiousness should be a standard to follow.

“A premature disclosure of your research may put you in a lot of troubles.”

Tenreiro Machado

To my friend Carlos Figueiredo who participated in the development of the robotic platforms and helped me fulfilling the exhaustive experiments – working together since our BSc we developed the most hilarious projects and overcame the most difficult obstacles.

“Hear the advice of others and follow yours.” Carlos Figueiredo

To my friend Miguel Luz for the exchange of know-how that turned out to be crucial and decisive in this and other projects over the last decade – the exchange of knowledge is essential for a greater mobility in both academic and scientific worlds.

“ehh, alibe?” Miguel Luz

To my friend Fernando Martins who tirelessly supported me and gave me different perspectives when facing certain problems – the multidisciplinary team work provides a new vision that needs to be further explored.

“If it was easy, it would have been done already.” Fernando Martins

To my colleagues in the Mobile Robotics Laboratory of the Institute of Systems and Robotics from the Faculty of Sciences and Technology of University of Coimbra – in particular to David Portugal and André Araújo for their scientific and technical endeavour worth following.

“Artista, keep that spirit up!” David Portugal

To my colleagues at *RoboCorp* from the Engineering Institute of Coimbra – in particular to Filipe Clemente, Gonçalo Dias and Prof. Rui Mendes who all gave me scientific and motivational support, as well as the necessary conditions to fulfil some of the experiments.

“Good workers always think they should work even more.” Filipe Clemente

To all other friends across the world – in particular to Monika Ivanova, Pedram Ghamisi, Pedro Mendes and Rúben Rocha who stood by my side at so many different levels and that were there when I needed the most.

“Thank you for being that perfect example of someone who follows his dreams no matter what.”

Monika Ivanova

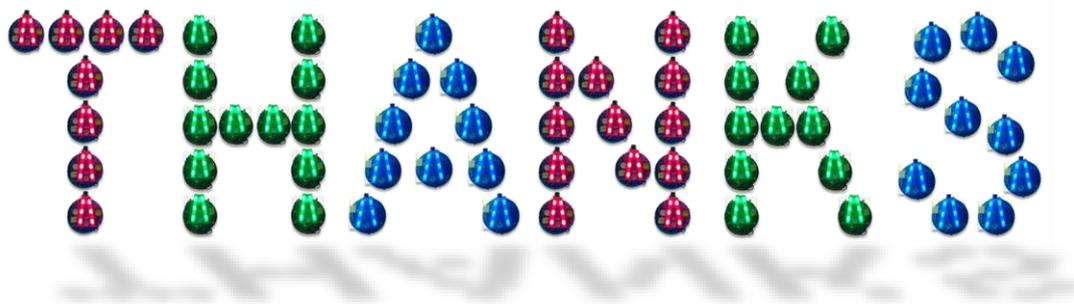
I also want to thank to my parents, brother and girlfriend since they gave me the opportunity to get here – without their financial and emotional involvement it would have been impossible to be where I am now.

“One of the things I praise is that, though you complain how hard it is, you’re always successful.”

Andria Lopes

At last, I would like to acknowledge the several institutes, namely, the Institute of Systems and Robotics from University of Coimbra (ISR-UC), the Engineering Institute of Coimbra from the Polytechnic Institute of Coimbra (ISEC-IPC) and the School of Mathematical and Computer Sciences from Heriot-Watt University (MACS-HW), for the resources made available to me.

For the help, support and, especially, the patience, to all of you...



This work was supported by a PhD scholarship (*SFRH/BD /73382/2010*) and the research project *CHOPIN (PTDC/EEA-CRO/119000/2010)* from the Institute of Systems and Robotics from University of Coimbra (*ISR-UC*) under regular funding by the Portuguese Foundation for Science and Technology (*FCT*). The Institute of Telecommunications (*IT-Covilhã*), *RoboCorp* from the Engineering Institute of Coimbra and Robotics Lab from Heriot-Watt University also partially provided funding for some necessary equipment and participation in conferences.

EVOLUTIONARY ROBOT SWARMS UNDER REAL-WORLD CONSTRAINTS

by

Micael Couceiro

Abstract

Over the past decades, many scientists and engineers have been studying nature's best and time-tested patterns and strategies. Within the existing biological architectures, *swarm societies* revealed that relatively unsophisticated agents with limited capabilities, such as ants or birds, were able to cooperatively accomplish complex tasks necessary for their survival. Those simplistic systems embrace all the conditions necessary to survive, thus embodying cooperative, competitive and adaptive behaviours. In the never-ending battle to advance artificial manmade mechanisms, computer scientists simulated the first swarm behaviour designed to mimic the flocking behaviour of birds in the late eighties. Ever since, many other fields, such as *robotics*, have benefited from the *fault-tolerant* mechanism inherent to *swarm intelligence*.

The area of research presented in this Ph.D. Thesis focuses on *swarm robotics*, which is a particular domain of *multi-robot systems (MRS)* that embodies the mechanisms of swarm intelligence into robotics. More specifically, this Thesis proposes a complete swarm robotic solution that can be applied to real-world missions. Although the proposed methods do not depend on any particular application, search and rescue (*SaR*) operations were considered as the main case study due to their inherent level of complexity. Such operations often occur in highly dynamic and large scenarios, with harsh and faulty conditions, that pose several problems to *MRS* applicability. This Thesis focuses on these problems raising new challenges that cannot be handled appropriately by simple adaptation of state-of-the-art swarm algorithms, planning, control and decision-making techniques.

The contributions of this Thesis revolve around an extension of the *Particle Swarm Optimization (PSO)* to *MRS*, denoted as *Robotic Darwinian Particle Swarm Optimization (RDPSO)*. The *RDPSO* is a distributed swarm robotic architecture that benefits from the dynamical partitioning of the whole swarm of robots by means of an evolutionary *social exclusion* mechanism based on *Darwin's survival-of-the-fittest*. Nevertheless, although currently applied solely to the *RDPSO* case study, the applicability of all concepts herein proposed is not restricted to it, since all parameterized swarm robotic algorithms may benefit from a similar approach.

The *RDPSO* is then proposed and used to devise the applicability of novel approaches. The fundamentals around the *RDPSO* are introduced by focusing on robots' dynamics, obstacle avoidance, communication constraints and its evolutionary properties. Afterwards, taking the initial deployment of robots within the environment as a basis for applying swarm robotics systems into real-world applications, the development of a realistic deployment strategy is proposed. For that end, the population of robots is hierarchically divided, wherein larger support platforms autonomously deploy smaller exploring platforms in the scenario, while considering communication constraints and obstacles. After the deployment, a way of ensuring a fault-tolerant multi-hop mobile ad hoc communication network (*MANET*) is introduced to explicitly exchange information needed in a collaborative real-world task execution. Such strategy not only considers the maximum communication range between robots, but also the minimum signal quality, thus refining the applicability to real-world context. This is naturally followed by a deep analysis of the *RDPSO* communication system, describing the dynamics of the communication data packet structure shared between teammates. Such procedure is a first step to achieving a more scalable implementation by optimizing the communication procedure between robots. The highly dynamic characteristics of real-world applications motivated us to ultimate the *RDPSO* development with an adaptive strategy based on a set of context-based evaluation metrics.

This thesis contributes to the state-of-the-art in swarm robotics with novel algorithms for real-world applications. All of the proposed approaches have been extensively validated in benchmarking tasks, in simulation, and with real robots. On top of that, and due to the limitations inherent to those (*e.g.*, number of robots, scenario dimensions, real-world constraints), this Thesis further contributes to the state-of-the-art by proposing a macroscopic model able to capture the *RDPSO* dynamics and, to some extent, analytically estimate the collective performance of robots under a certain task. It is the author's expectation that this Ph.D. Thesis may shed some light into bridging the reality gap inherent to the applicability of swarm strategies to real-world scenarios, and in particular to *SaR* operations.

Keywords: *Cooperation; Swarm Robotics; Evolutionary Algorithms; Survival-of-the-Fittest; Mobile Ad hoc Communication Networks; Fault-Tolerance; Performance Analysis and Estimation; Search and Rescue.*

ENXAMES DE ROBÔS EVOLUTIVOS SOB CONSTRANGIMENTOS DO MUNDO REAL

by

Micael Couceiro

Resumo

Nas últimas décadas, vários cientistas e engenheiros têm vindo a estudar as estratégias provenientes da natureza. Dentro das arquiteturas biológicas, as sociedades que vivem em enxames revelam que agentes simplistas, tais como formigas ou pássaros, são capazes de realizar tarefas complexas usufruindo de mecanismos de cooperação. Estes sistemas abrangem todas as condições necessárias para a sobrevivência, incorporando comportamentos de cooperação, competição e adaptação. Na “batalha” sem fim em prol do progresso dos mecanismos artificiais desenvolvidos pelo homem, a ciência conseguiu simular o primeiro comportamento em enxame no final dos anos oitenta. Desde então, muitas outras áreas, entre as quais a robótica, beneficiaram de mecanismos de *tolerância a falhas* inerentes da *inteligência coletiva de enxames*.

A área de investigação deste estudo incide na *robótica de enxame*, consistindo num domínio particular dos sistemas robóticos cooperativos que incorpora os mecanismos de inteligência coletiva de enxames na robótica. Mais especificamente, propõe-se uma solução completa de robótica de enxames a ser aplicada em contexto real. Nesta ótica, as operações de busca e salvamento foram consideradas como o caso de estudo principal devido ao nível de complexidade associado às mesmas. Tais operações ocorrem tipicamente em cenários dinâmicos de elevadas dimensões, com condições adversas que colocam em causa a aplicabilidade dos sistemas robóticos cooperativos. Este estudo centra-se nestes problemas, procurando novos desafios que não podem ser ultrapassados através da simples adaptação da literatura da especialidade em algoritmos de enxame, planeamento, controlo e técnicas de tomada de decisão.

As contribuições deste trabalho sustentam-se em torno da extensão do método *Particle Swarm Optimization (PSO)* aplicado a sistemas robóticos cooperativos, denominado de *Robotic Darwinian Particle Swarm Optimization (RDPSO)*. O *RDPSO* consiste numa arquitetura robótica de enxame distribuída que beneficia do particionamento dinâmico da população de robôs utilizando mecanismos evolucionários de *exclusão social* baseados na *sobrevivência do mais forte* de Darwin. No entanto, apesar de estar assente no caso de estudo do *RDPSO*, a aplicabilidade dos conceitos aqui propostos

não se encontra restrita ao mesmo, visto que todos os algoritmos parametrizáveis de enxame de robôs podem beneficiar de uma abordagem idêntica.

Os fundamentos em torno do *RDPSO* são introduzidos, focando-se na dinâmica dos robôs, nos constrangimentos introduzidos pelos obstáculos e pela comunicação, e nas suas propriedades evolucionárias. Considerando a colocação inicial dos robôs no ambiente como algo fundamental para aplicar sistemas de enxames em aplicações reais, é assim introduzida uma estratégia de colocação de robôs realista. Para tal, a população de robôs é dividida de forma hierárquica, em que são utilizadas plataformas mais robustas para colocar as plataformas de enxame no cenário de forma autónoma. Após a colocação dos robôs no cenário, é apresentada uma estratégia para permitir a criação e manutenção de uma rede de comunicação móvel *ad hoc* com tolerância a falhas. Esta estratégia não considera somente a distância entre robôs, mas também a qualidade do nível de sinal rádio frequência, redefinindo assim a sua aplicabilidade em cenários reais. Os aspetos anteriormente mencionados estão sujeitos a uma análise detalhada do sistema de comunicação inerente ao algoritmo, para atingir uma implementação mais escalável do *RDPSO* a cenários de elevada complexidade. Esta elevada complexidade inerente à dinâmica dos cenários motivaram a ultimar o desenvolvimento do *RDPSO*, integrando para o efeito um mecanismo adaptativo baseado em informação contextual (*e.g.*, nível de atividade do grupo).

Face a estas considerações, o presente estudo pode contribuir para expandir o estado-da-arte em robótica de enxame com algoritmos inovadores aplicados em contexto real. Neste sentido, todos os métodos propostos foram extensivamente validados e comparados com alternativas, tanto em simulação como com robôs reais. Para além disso, e dadas as limitações destes (*e.g.*, número limitado de robôs, cenários de dimensões limitadas, constrangimentos reais limitados), este trabalho contribui ainda para um maior aprofundamento do estado-da-arte, onde se propõe um modelo macroscópico capaz de capturar a dinâmica inerente ao *RDPSO* e, até certo ponto, estimar analiticamente o desempenho coletivo dos robôs perante determinada tarefa.

Em suma, esta investigação pode ter aplicabilidade prática ao colmatar a lacuna que se faz sentir no âmbito das estratégias de enxames de robôs em contexto real e, em particular, em cenários de busca e salvamento.

Palavras-Chave: *Cooperação; Robótica de Enxame; Algoritmos Evolucionários; Sobrevivência do mais forte; Redes de Comunicação Ad hoc; Tolerância a Falhas; Análise e Estimação de Desempenho; Busca e Salvamento.*

Contents

Supervision	iii
Committee	v
Acknowledgments	ix
Abstract	xi
Resumo	xiii
Contents	xv
List of Figures	xxi
List of Tables	xxix
List of Algorithms	xxxi
List of Symbols	xxxiii
List of Abbreviations	xxxvii
1. Introduction	1
1.1 Features and Assumptions	2
1.1.1 Main Application: Search and Rescue (<i>SaR</i>)	2
1.1.2 Multi-Robot System (<i>MRS</i>): Swarm Robotics	3
1.1.3 Architecture: Distributed	3
1.1.4 Inter-Robot Communication: Mobile Ad hoc Network (<i>MANET</i>)	4
1.2 Research Question	4
1.2.1 Subsidiary Question 1: Autonomous Deployment of Robots	5
1.2.2 Subsidiary Question 2: Communication in Faulty Environments	5
1.2.3 Subsidiary Question 3: Efficient Sharing of Information	6
1.2.4 Subsidiary Question 4: Adaptability to Dynamic Environments	6
1.2.5 Subsidiary Question 5: Estimation of a Provable Performance	7
1.3 Approach	7
1.3.1 Subsidiary Approach 1: Autonomous Deployment of Robots	8
1.3.2 Subsidiary Approach 2: Communication in Faulty Environments	8
1.3.3 Subsidiary Approach 3: Efficient Sharing of Information	9
1.3.4 Subsidiary Approach 4: Adaptability to Dynamic Environments	9
1.3.5 Subsidiary Approach 5: Estimation of a Provable Convergence	9
1.4 Outline	9
2. Background and Motivation	13
2.1 Philosophy	13
2.1.1 The Essence of Cooperation	14
2.1.2 Robots: The Rise of a New Society?	17
2.1.3 The Role of Communication	19
2.1.4 The Particular Domain of Swarm Robotics	20

2.2	Related Work	23
2.2.1	Search and Rescue (<i>SaR</i>) Procedure	24
2.2.2	Search and Rescue <i>R&D</i> Projects – The <i>CHOPIN</i> Case Study	26
2.2.3	Mobile Ad hoc Networks (<i>MANETs</i>) in Faulty Environments	29
2.2.4	Initial Deployment and its Influence on Task Completion	36
2.3	Summary	41
3.	Preliminaries	45
3.1	Theoretical	45
3.1.1	Fractional Calculus	45
3.1.2	Jury-Marden’s Theorem	47
3.1.3	Fuzzy Logic	48
3.1.4	Semi-Markov Chain	50
3.1.5	Z-Transform	51
3.2	Technological	52
3.2.1	Scouts (Swarm Robots)	53
3.2.2	Rangers (<i>Marsupial</i> Robots)	58
3.2.3	Implementation of <i>MANETs</i>	60
3.2.4	<i>MRSim</i>	65
3.2.5	<i>Webots</i>	67
3.3	Summary	68
4.	Robotic Darwinian <i>PSO</i>	69
4.1	From Optimization to Robotics	70
4.2	Fractional Order Convergence	76
4.2.1	Memory Complexity	77
4.3	Obstacle Avoidance	80
4.4	Ensuring <i>MANET</i> Connectivity	81
4.4.1	Problem Statement	82
4.4.2	General Approach	82
4.5	“Punish-Reward” Mechanism	86
4.5.1	Socially Active Subgroups	87
4.5.2	Socially Excluded Subgroups	88
4.6	<i>RDPSO</i> Outline	90
4.6.1	Low Level Control Architecture	90
4.6.2	Attractive/Repulsive Components	92
4.6.3	Stagnation Avoidance	93
4.6.4	Algorithm	94
4.7	Experimental Results	94
4.7.1	Numerical Simulations	94
4.7.2	Real-World Experiments	101
4.8	Discussion	105
4.9	Summary	106

5. Deployment and Fault-Tolerance	109
5.1 Initial Deployment	110
5.1.1 Problem Statement	112
5.1.2 General Approach	112
5.1.3 Fault-Tolerance Generalization	115
5.1.4 Rangers' Computational Requirements	119
5.2 Fault-Tolerance Assessment	119
5.2.1 Problem Statement	120
5.2.2 Fault-Tolerance Generalization	120
5.2.3 Scouts' Computational Requirements	123
5.3 Experimental Results	123
5.3.1 Real-World Experiments	123
5.3.2 Scalability Evaluation through Simulations	131
5.4 Discussion	136
5.5 Summary	137
6. Communication Optimization	139
6.1 Sharing Information in the <i>RDPSO</i>	141
6.1.1 Ensuring Connectivity	141
6.1.2 Converging to the Optimal Solution	142
6.1.3 Avoiding Sub-Optimality	144
6.2 Routing Protocol	144
6.2.1 Ad hoc On-demand Distance Vector (<i>AODV</i>)	145
6.2.2 <i>RDPSO</i> based <i>AODV</i>	146
6.3 Experimental Results	150
6.3.1 Real-World Experiments	151
6.3.2 Scalability Evaluation through Simulation	159
6.4 Discussion	161
6.5 Summary	162
7. Parameterization and Adaptability	165
7.1 Convergence Analysis	167
7.1.1 Problem Statement	167
7.1.2 General Approach	168
7.1.3 Robot Constraints	170
7.1.4 Outline	172
7.2 Context-Based Evaluation Metrics	172
7.2.1 Exploitation vs Exploration	173
7.2.2 Cognition vs Socialization	176
7.2.3 Obstacles Susceptibility	179
7.2.4 Connectivity Susceptibility	181
7.2.5 Outline	184
7.3 <i>Fuzzified</i> Systematic Parameter Adjustment	184

7.4	Experimental Results	189
7.4.1	Numerical Simulations	189
7.4.2	Dynamic Environments in Simulation	195
7.5	Discussion	200
7.6	Summary	202
8.	Benchmark	203
8.1	Theoretical Comparison	203
8.1.1	Extended Particle Swarm Optimization (<i>EPSO</i>)	204
8.1.2	Physically-embedded Particle Swarm Optimization (<i>PPSO</i>)	205
8.1.3	Glowworm Swarm Optimization (<i>GSO</i>)	206
8.1.4	Aggregations of Foraging Swarm (<i>AFS</i>)	207
8.1.5	Theoretical comparison	209
8.2	Experimental Results	212
8.2.1	Simulation Experiments	212
8.2.2	Real-World Experiments	217
8.3	Discussion	222
8.4	Summary	224
9.	Macroscopically Modelling	225
9.1	Problem Formulation	227
9.2	Features and Assumptions	227
9.2.1	Simulation setup	227
9.2.2	Robot's features	228
9.2.3	World features	229
9.2.4	Algorithm's features	230
9.3	Evolutionary Rates	231
9.3.1	Robots' communication interference rate	232
9.3.2	Obstacles' encountering rate	233
9.3.3	Mission-related detection rate	238
9.3.4	Social exclusion rate	241
9.3.5	Social inclusion rate	242
9.4	Simplified Macroscopic Models	245
9.4.1	Socially Active Subgroups	246
9.4.2	Socially Excluded Subgroup	250
9.5	Evolutionary Macroscopic Model	252
9.5.1	Steady-state analysis	254
9.5.2	Preliminary Results	256
9.6	Experimental Results	259
9.7	Discussion	262
9.8	Summary	263
10.	Conclusion	265
10.1	Fulfilment of the Objectives	265

10.1.1	Subsidiary Objective 1: Autonomous Deployment of Robots	265
10.1.2	Subsidiary Objective 2: Communication in Faulty Environments	266
10.1.3	Subsidiary Objective 3: Efficient Sharing of Information	266
10.1.4	Subsidiary Objective 4: Adaptability to Dynamic Environments	267
10.1.5	Subsidiary Objective 5: Estimation of a Provable Convergence	268
10.2	Main Contributions and Achievements	268
10.2.1	New Platforms and Tools	268
10.2.2	Applicability of Mathematical Concepts	269
10.2.3	Benchmark on Swarm Robotics	271
10.3	Prospects	271
10.3.1	Future Work	271
10.3.2	Further Expectations	276
	References and Bibliography	281
	Publications Relevant to the Thesis	303
	Refereed Scientific Journals	303
	Book Chapters	304
	Refereed Conferences	304
	Technical Reports	306

List of Figures

Figure 2.1. Maslow's Pyramid.	14
Figure 2.2. <i>Stigmergy</i> between ants: a) travelling a <i>pheromone</i> path between the nest and the food; b) an obstacle interrupts the path; c) the ants create two different paths in order to overcome the obstacle; d) a new <i>pheromone</i> path is created around the obstacle - the shortest path.	15
Figure 2.3. Adapted 3C Model.	16
Figure 2.4. Cooperation in robotics: a) 3D mapping with two robots (Rocha, Dias, & Carvalho, 2005); b) Robotic football - <i>RoboCup Legged Robots League</i> (Veloso, Uther, Fujita, Asada, & Kitano, 1998); c) Pursuit of multiple targets with multiple unmanned ground vehicles (<i>UGV</i>) and one unmanned aerial vehicles (<i>UAV</i>) (Vidal, Shakernia, Kim, Shim, & Sastry, 2002); d) Two robotic manipulators moving an object (Ferreira, 2006; Takahashi, Ise, Konno, Uchiyama, & Sato, 2008).	18
Figure 2.5. Example of a <i>SaR</i> real application. a) Basement garage of a shopping centre; b) underground fire.	22
Figure 2.6. General flowchart of a <i>SaR</i> (robotic) operation.	25
Figure 3.1. The <i>eSwarBot</i> robot.	54
Figure 3.2. Evaluation of several platforms for swarm robotics using radar graphs. a) <i>eSwarBot</i> (Couceiro M. S., Figueiredo, Luz, Ferreira, & Rocha, 2011); b) <i>e-puck</i> (Mondada, et al., 2009); c) <i>SRV-1 Blackfin Surveyor</i> (Cummins, Azhar, & Sklar, 2008); d) <i>TraxBot</i> (Araújo, Portugal, Couceiro, Figueiredo, & Rocha, 2012); e) <i>Mindstorms NXT Lego</i> (Bagnall, 2007); f) <i>marXbot</i> (Bonani, et al., 2010); g) <i>Hemisson</i> ; h) <i>IdMind Circular GT</i> ; i) <i>Bot'n'Roll ONE</i>	56
Figure 3.3. The <i>e-puck</i> robot.	57
Figure 3.4. The <i>TraxBot</i> robot platform.	58
Figure 3.5. The <i>TraxBot Conveyor Kit</i> loaded with 5 <i>eSwarBots</i>	60
Figure 3.6. Electrical modification of <i>XBee Series 2</i> from <i>Digi International</i> to provide the <i>RSSI</i> signal output.	61
Figure 3.7. Measured <i>RSSI</i> versus distance from two robots located in the experimental scenario.	62
Figure 3.8. System architecture of the <i>e-puck</i> robot equipped with <i>Gumstix Overo COM</i>	64
Figure 3.9. Multi-Robot Simulator (MRSim). Illustration of one trial with 10 robots performing the collective mapping of an unknown scenario under the influence of the <i>RDPSO</i> algorithm (Couceiro, Portugal, & Rocha, 2013).	66
Figure 4.1. Convergence of the robot toward the solution changing the differential derivative r	79

Figure 4.2. Sensing function $gxnt$ represented as the relation between the analog output voltage of distance sensors and the distance to obstacle. a) Sensor Sharp GP2Y0A21YK IR - monotonically decreasing sensing function $gxnt$; b) Sensor Sonaswitch Ultrasound - monotonically increasing sensing function $gxnt$	80
Figure 4.3. Illustrative example of obstacle avoidance behaviour of a robot. a) $\rho_3 \ll \min\rho_1, \rho_2$; b) $\rho_3 \gg \min\rho_1, \rho_2$	81
Figure 4.4. Illustration of a <i>MANET</i> topology of a subgroup. Dashed lines represent the maximum distance $dmax$ between each pair of robots and the bold arrows represent the force vectors that ensure <i>MANET</i> connectivity.	85
Figure 4.5. Control architecture of the <i>RDPSO</i> with <i>LLC</i> of a robot n	91
Figure 4.6. Geometrical Illustration of the <i>RDPSO</i> using a subgroup of two robots.	92
Figure 4.7. Sequence of a <i>MRS</i> exploration using the <i>RDPSO</i> algorithm (Couceiro, Rocha, & Ferreira, 2013a).....	93
Figure 4.8. Virtual scenario with obstacles and robots divided into 5 subgroups.	96
Figure 4.9. Estimated marginal means of the final global solution.	100
Figure 4.10. Estimated marginal means of the runtime (number of iterations).....	101
Figure 4.11. Experimental setup. a) Enclosed arena with 2 subgroups (different colours); b) Virtual representation of the target distribution.	102
Figure 4.12. Frame sequence showing the <i>RDPSO</i> performance on a population of 12 robots (some robots may be outside camera's range). a) The population is initially divided into two swarms – green and red – deployed in a spiral manner; b) The swarms independently search for the brighter site taking into account a maximum communication distance of 1.5 meters between robots of the same swarm; c) One robot from the red and green swarm finds the sub-optimal and optimal solution, respectively; d) As the red swarm does not improve, some robots are excluded, thus being added to the socially excluded subgroup (white swarm); e) Since the green swarm has improved, it is able to call new members from the socially excluded subgroup; f) Finally, the green swarm proliferates calling all the previously excluded robots that were unable to improve their solution. Note that robots do not all converge the optimal solution as they try to maintain a distance of $dmax$ between them.....	103
Figure 4.13. Performance of the algorithms changing the number of robots N in the population: a) $dmax = 0.5 meters$; b) $dmax = 1.5 meters$	104
Figure 5.1. Initial deployment of the <i>RDPSO</i> algorithm of a population of robots divided in 3 subgroups of 3 robots each.	111
Figure 5.2. <i>EST</i> deployment in a k -connected <i>MANET</i>	115
Figure 5.3. Relation between d and $dmax$ varying the <i>MANET</i> connectivity k	116
Figure 5.4. Computational outputs of the <i>EST</i> (left) and random (right) deployment of 10 scouts over a given scenario endowed with obstacles.	118
Figure 5.5. Relation between d and $dmax$ varying the <i>MANET</i> connectivity k	118

Figure 5.6. Illustration of a <i>MANET</i> topology. The dashed lines represent the vector Qnt for each robot, thinner arrows represent the force vectors Qn, it regarding chosen neighbour i , and larger arrows represent the resulting force vectors $\chi 4t$	122
Figure 5.7 Experimental setup. a) Arena with 3 subgroups (different colours) of 5 <i>eSwarBots</i> each deployed by 3 <i>TraxBot</i> (one for each subgroup); b) Virtual representation of the target distribution.....	124
Figure 5.8. Frame sequence showing the <i>EST</i> deployment strategy on a population of 15 scouts and 3 rangers. a) The population of scouts is initially divided into three subgroups – red, green and blue – each subgroup loaded on top of a different ranger (one of the rangers is outside camera’s field-of-view); b) Each ranger randomly chooses the first position to deploy the first scout of each subgroup; c) The rangers will deploy the other successive scouts considering the previously deployed ones while avoiding obstacles; d) After deploying all the scouts from one subgroup, the ranger in charge of such deployment broadcasts a message to start the mission.	125
Figure 5.9 Performance of the <i>RDPSO</i> under two different deployment strategies. Coloured zones correspond to the interquartile range of the best solution of the 20 trials for each different deployment.	126
Figure 5.10 Diversity of solutions at the instant scouts start their mission.....	127
Figure 5.11 Trajectory of scouts under the <i>EST</i> approach in the 16 th trial (dark objects corresponds to obstacles). a) The configuration evolves starting with a spiral-like deployment; b) The scouts try to maintain a signal quality of -50dBm between their nearest neighbour while subgroups evolve (grey scouts correspond to the socially excluded subgroup); c) One scout from the green subgroup had an energy depletion due to the exhaustive experiments, thus partitioning its subgroup; d) As the top green scout thinks its subgroup is not improving, it got socially excluded.....	128
Figure 5.12 Performance of the <i>RDPSO</i> under the <i>EST</i> approach with and without fault tolerance. Coloured zones correspond to the interquartile range of the best solution of the 20 trials for each different deployment.	129
Figure 5.13 Simulation experiments in a 20 × 10 meters indoor scenario (sports pavilion): a) <i>WiFi</i> communication propagation b) $NT = 15$; c) $NT = 30$; d) $NT = 60$	132
Figure 5.14 Performance of the <i>RDPSO</i> under different deployment strategies and number of scouts. Coloured zones correspond to the interquartile range of the best solution of the 20 trials for each different deployment. a) $NT = 15$ (the pattern regions are the representation of the real experiments from Figure 5.9); b) $NT = 30$; c) $NT = 60$	134
Figure 5.15 Illustration of the area covered by 3 subgroups of 5 scouts each using the <i>EST</i> approach. Each different coloured region represents the area covered by a subgroup. Grey regions represent the intersection between areas covered by different subgroups.	135
Figure 5.16 Ratio of covered area for a population of 15, 30 and 60 scouts grouped into three subgroups.....	136
Figure 6.1. General communication packet structure for a subgroup of Ns robots.	141

Figure 6.2. Communication packet structure that allows robots in maintaining the <i>MANET</i> k -connectivity within their subgroup of N_s robots.	142
Figure 6.3. Communication packet structure that allows robots from active subgroups to cooperatively converge to the solution. This packet is only sent if a robot improves its best cognitive solution.	143
Figure 6.4. Ratio between the number of useful messages and the total number of messages retrieved from the experimental evaluation on section 4.7.2.	143
Figure 6.5. <i>RDPSO</i> based <i>AODV</i> routing protocol. Red bolder lines between robots represent that there exists a possible link between them but that the <i>AODV</i> protocol is unaware of. a) The robots start connected by means of the <i>EST</i> initial deployment strategy, thus enforcing the <i>MANET</i> connectivity of the whole subgroup (section 5.1). The node discovery and route discovery allows to retrieve the <i>ID</i> of all robots and build the routes between them (blue thin lines). b) After a while, robot 2 improves and tries to broadcast its new solution and position to the whole network. However, as robot 2 is unable to communicate with robot 6 by means of the route previously built using <i>AODV</i> , a new route discovery needs to be sent (red thick lines). c) Using the <i>RDPSO</i> based <i>AODV</i> will allow robot 2 to choose the neighbour that is near robot 6, <i>i.e.</i> , robot 3, that will forward the message to its destination, <i>i.e.</i> , robot 6.	149
Figure 6.6. Virtual representation of the target distribution over day under different lighting conditions for the scenario depicted in Figure 5.7.	152
Figure 6.7. Packet delivery ratio within robots from the same subgroup as a function of the subgroup size.	154
Figure 6.8. Routing overhead within robots from the same subgroup.	155
Figure 6.9. Average hop count within robots from the same subgroup.	156
Figure 6.10. Evolution of robotic subgroups over a trial of 360 seconds. a) Population size; b) Number of local broadcasts; c) Number of global broadcasts.	157
Figure 6.11. Normalized temporal average number of local and global broadcasts.	159
Figure 6.12. Ratio between the number of packets exchanged using the “optimized” <i>RDPSO</i> and the “regular” <i>RDPSO</i> over the number of iterations in a population of 60 robots.	160
Figure 7.1. Global and particular attraction domain of the asymptotic stability of the <i>RDPSO</i>	172
Figure 7.2. Experimental setup to evaluate the exploration/exploitation capabilities of a subgroup of two robots.	174
Figure 7.3. Centre-of-Mass trajectories in phase space of a subgroup of 2 robots.	175
Figure 7.4. Experimental setup to evaluate the cognition/socialization between two robots of the same subgroup.	177
Figure 7.5. Distance between robots in phase space to evaluate the relation between ρ_1 and ρ_2	177
Figure 7.6. Experimental setup to evaluate the obstacle susceptibility of a robot.	180
Figure 7.7. Distance from the worst performing robot to the obstacle in phase space.	180

Figure 7.8. Experimental setup to evaluate the connectivity between two robots from the same subgroup.	182
Figure 7.9. Distance between robots in phase space.	182
Figure 7.10. Fuzzy Logic System to dynamically adapt the parameters of the <i>RDPSO</i> algorithm.	185
Figure 7.11. General membership function for each input.	185
Figure 7.12. Membership functions to quantify the consequents for each coefficient.	186
Figure 7.13. Set of <i>IF-THEN-ELSE</i> fuzzy rules do control robots' behaviour based on contextual information.	187
Figure 7.14. Parameters' evolution under a hypothetical situation.	188
Figure 7.15. Virtual scenario with obstacles and 25 robots divided into 5 swarms. a) $f1x, y$; b) $f2x, y$; and c) $f3x, y$	190
Figure 7.16. <i>RDPSO</i> evaluation changing the number of robots NT for each objective function and set of parameters: a) $f1x, y$ and S_1 ; b) $f1x, y$ and S_2 ; c) $f2x, y$ and S_1 ; d) $f2x, y$ and S_2 ; e) $f3x, y$ and S_1 ; and f) $f3x, y$ and S_2	192
Figure 7.17. Estimated marginal means of the <i>RDPSO</i> performance for the: a) final solution using $f1x, y$; b) final solution using $f2x, y$; c) final solution using $f3x, y$; d) runtime using $f1x, y$; e) runtime using $f2x, y$; f) runtime using $f3x, y$	194
Figure 7.18. Planar motion of $F1x, y$ peaks based on Forced Duffing Oscillator. a) $t = 0$; b) $t = 150$; c) $t = 300$; d) $t = 450$ iterations.	196
Figure 7.19. Performance of the non-adaptive and adaptive <i>RDPSO</i> under a dynamic environment for: a) $NT = 25$ robots; b) $NT = 50$ robots; c) $NT = 100$ robots.	198
Figure 7.20. Heat maps representation of robot's trajectory in a dynamic environment. The blue arrows indicate the trajectory of the sub-optimal and optimal solutions. a) Non-adaptive $NT = 25$; b) Adaptive $NT = 25$; a) Non-adaptive $NT = 50$; b) Adaptive $NT = 50$; a) Non-adaptive $NT = 100$; b) Adaptive $NT = 100$	199
Figure 8.1. Median of the exploration ratio η_{expt} over the 500 iteration for each method. a) $NT, dmax = 10, 30$; b) $NT, dmax = 10, 100$; c) $NT, dmax = 20, 30$; d) $NT, dmax = 20, 100$; e) $NT, dmax = 30, 30$; f) $NT, dmax = 30, 100$	214
Figure 8.2. <i>AUC</i> of the exploration ratio η_{expt} over the 500 iteration for each method. a) $NT, dmax = 10, 30$; b) $NT, dmax = 10, 100$; c) $NT, dmax = 20, 30$; d) $NT, dmax = 20, 100$; e) $NT, dmax = 30, 30$; f) $NT, dmax = 30, 100$	216
Figure 8.3. Experimental setup in an arena of 2.0×1.8 meters. a) Virtual representation in <i>Webots</i> ; b) Real representation; b) Representation of the sound distribution.	219
Figure 8.4. Representation of the rescue success of the <i>RDPSO</i> , <i>AFS</i> and <i>GSO</i> algorithms. Each marker corresponds to a different trial under a different algorithm. As closer the markers are from the origin $0, 0$, the fastest robots were able to find the victims. Markers located on the border lines of the 300 seconds means that only 1 victim was found during that trial.	221

Figure 9.1. Illustration of the normalization process for the scenario depicted in Figure 3.9. Step 1) Create a grid of points identifying obstacles' location uniformly distributed within the unit square based on the density of obstacles; Step 2) Map the obstacles location to a unit circle; and Step 3) Dilate the obstacles considering robot's obstacle detection range R_w	237
Figure 9.2. Social exclusion γ_{exct} and inclusion γ_{inct} rates for a population of $NT = 15$ robots.	244
Figure 9.3. Evolution of the average number of robots in each different state over time using a socially active subgroup of $N_s = 15$ robots.	246
Figure 9.4. Finite state semi-Markov model of socially active subgroups.	246
Figure 9.5. Steady-state analysis of socially active subgroups on both simulated and analytical results using the proposed macroscopic model. a) number of robots exploring the scenario; b) number of robots avoiding obstacles; and c) number of robots constrained by communication interference.	249
Figure 9.6. Finite state semi-Markov model of socially excluded subgroups.	250
Figure 9.7. Steady-state analysis of the socially excluded subgroup on both simulated and analytical results using the proposed macroscopic model. a) number of robots exploring the scenario; b) number of robots avoiding obstacles; and c) number of robots constrained by communication interference.	251
Figure 9.8. Finite state semi-Markov model of the complete evolutionary <i>RDPSO</i> algorithm. The \times superscript identifies the states and transition probabilities associated to the socially excluded subgroups.	253
Figure 9.9. Evolution of the average number of socially excluded robots in each different state over time. The final values (horizontal lines) retrieved from equation (9.49) are represented by the subscript *.....	257
Figure 9.10. Proportion of the explored area over time, A_{et} , for both simulated experiments and macroscopic estimation.....	259
Figure 9.11. Comparison between the normalized best intensity of light sensed by the swarm of robots over time and the macroscopic estimation A_{et}	261
Figure 10.1. Examples of spiral-based three-dimensional deployment of <i>UAVs</i> . a) <i>conical spiral</i> ; b) <i>rhumb line</i>	272
Figure 10.2. Illustrative sequence of a swarm of two snake-like nanorobots finding a target (<i>e.g.</i> , cancer cell) using a simplified version of the <i>RDPSO</i> algorithm. a) Mission start; b) One of the nanorobots is able to find its target while the other is still exploring.	273
Figure 10.3. <i>Protocooperation</i> in nature: a) Hermit crab and sea anemone; b) Ants and aphids; c) Buffalo and Egyptian plover birds; d) Humans and dolphins.	277
Figure 10.4. An example application of using a heterogeneous robot team to cooperatively fulfil a <i>SaR</i> mission with first responders. Small low-cost robots represented by the <i>e-pucks</i> could be used as scouts to explore the environment by benefiting from swarm robotic algorithms such as the <i>RDPSO</i> herein proposed. Humanoid robots represented by the <i>NAO</i> platforms could be	

used to walk over debris due to their higher mobility over the wheeled platforms and to build a bridge between human first responders and the robotic agents by benefiting from *HMI* algorithms (Casper & Murphy, 2003). The *UAVs* represented by the *AR.Drone* quadcopters could be used to significantly increase the coverage area of the rescuing operation and reach places that *UGVs* cannot (*e.g.*, buildings), thus increasing the mission success (Julian, Angermann, Schwager, & Rus, 2012). Both *UAVs* and humanoid robots could deploy the smaller scouts following the same principles previously addressed in this Thesis.....279

List of Tables

Table 2.1. Summary of works focusing on the initial deployment and inter-robot communication problems.	42
Table 3.2. Main <i>eSwarBot</i> features.	55
Table 3.3. Main e-puck features.	57
Table 3.4. Main <i>TraxBot</i> features.	59
Table 4.1. “Punish-Reward” <i>RDPSO</i> Rules.	88
Table 4.2. Typical maximum communication distances of the <i>WiFi</i> , <i>ZigBee</i> and <i>Bluetooth</i>	97
Table 4.3. <i>RDPSO</i> parameters obtained by trial-and-error and used in numerical simulations.	97
Table 4.4. Tukey’s <i>HSD Post Hoc Test</i> to the maximum communication distance d_{max}	99
Table 4.5. Tukey’s <i>HSD Post Hoc Test</i> to the total number of robots NT	99
Table 4.6. <i>RDPSO</i> parameters obtained by trial-and-error and used in its first evaluation with physical robots.	103
Table 5.1. <i>RDPSO</i> parameters obtained by trial-and-error and used in a larger experiment with physical robots.	125
Table 5.2. Average and standard deviation of communication data.	129
Table 6.1. Communication cost.	152
Table 7.1. Two sets of <i>RDPSO</i> parameters.	189
Table 7.2. <i>RDPSO</i> parameters.	190
Table 7.3. Multivariate test for the number of robots.	193
Table 7.4. Multivariate test for the set of parameters.	193
Table 8.1. Summary of swarm foraging algorithms used in the benchmarking study.	209
Table 9.1. Inputs of the robot model.	229
Table 9.2. Inputs of the world model.	230
Table 9.3. Inputs of the <i>RDPSO</i> algorithm.	230
Table 9.4. Average delay of robots within each state.	231
Table 9.5. Inputs of the <i>eSwarBot</i> robot model.	260
Table 9.6. Inputs of the real world model.	260

List of Algorithms

Algorithm 4.1. Traditional <i>PSO</i> Algorithm	71
Algorithm 4.2. <i>DPSO</i> Algorithm	74
Algorithm 4.3. Ensuring subgroup <i>s</i> network connectivity.	84
Algorithm 4.4. <i>RDPSO</i> Algorithm	95
Algorithm 5.1. Initial deployment <i>EST</i> algorithm for ranger <i>n</i>	119
Algorithm 5.2. <i>k</i> -fault-tolerance algorithm for scout <i>n</i>	122
Algorithm 6.1. Sharing information within the <i>RDPSO</i>	151
Algorithm 7.1. Dynamic <i>FDO</i> function generator.	197
Algorithm 8.1. <i>EPSO</i> algorithm for robot <i>n</i>	205
Algorithm 8.2. <i>PPSO</i> algorithm for robot <i>n</i>	206
Algorithm 8.3. <i>GSO</i> algorithm for robot <i>n</i>	207
Algorithm 8.4. <i>AFS</i> algorithm for robot <i>n</i>	208

List of Symbols

<i>Symbol</i>	<i>Description</i>
GENERAL NOTATION	
t	Discrete time [s]
A	Area of the scenario [m^2]
ϖ	Dimensionality of the physical space ($\varpi = 2$ for planar problems), $\varpi \in \mathbb{N}$
RDPSO NOTATION	
v_n	Velocity vector of robot n , $v_n \in \mathbb{R}^\varpi [m \cdot s^{-1}]$
x_n	Position vector of robot n , $x_n \in \mathbb{R}^\varpi [m]$
w_n	Inertial influence on robot n , $w_n \in \mathbb{R}^\varpi [m \cdot s^{-1}]$
ρ_1	Local best (cognitive) coefficient, $\rho_1 \in \mathbb{R}$
ρ_2	Global best (social) coefficient, $\rho_2 \in \mathbb{R}$
ρ_3	Obstacle avoidance coefficient, $\rho_3 \in \mathbb{R}$
ρ_4	Enforcing network communication coefficient, $\rho_4 \in \mathbb{R}$
χ_1	Best position of the robot regarding its local (cognitive) solution, $\chi_1 \in \mathbb{R}^\varpi [m]$
χ_2	Best position of the robot regarding its global (social) solution, $\chi_2 \in \mathbb{R}^\varpi [m]$
χ_3	Best position of the robot regarding its obstacle avoidance, $\chi_3 \in \mathbb{R}^\varpi [m]$
χ_4	Best position of the robot regarding its network topology, $\chi_4 \in \mathbb{R}^\varpi [m]$
r_i	Random vectors, $r_i \in \mathbb{R}^\varpi$, $i = 1,2,3,4, sp$
$f(x_n[t])$	Sensed solution of robot n at position x_n , $f(x_n[t]) \in \mathbb{R}$
Δx_{max}	Maximum travelled distance between iterations, $\Delta x_{max} \in \mathbb{R} [m]$
v_{max}	Maximum velocity vector of robot n , $v_{max} \in \mathbb{R} [m \cdot s^{-1}]$
d_{max}	Maximum communication distance, $d_{max} \in \mathbb{R} [m]$
q_{min}	Minimum signal quality, $q_{min} \in \mathbb{R} [dBm]$
R_w	Obstacle sensing radius, $R_w \in \mathbb{R} [m]$
N_T	Population of robots, $N_T \in \mathbb{N}_0$
N_s	Total number of robots in subgroup s , $N_s \in \mathbb{N}_0$
N_I	Initial number of robots in a subgroup, $N_I \in \mathbb{N}_0$
N_{min}	Minimum number of robots required to form a subgroup, $N_{min} \in \mathbb{N}_0$

N_{max}	Maximum number of robots in a subgroup, $N_{max} \in \mathbb{N}_0$
N_s^I	Initial number of subgroups, $N_s^I \in \mathbb{N}_0$
N_s^{min}	Minimum number of subgroups, $N_s^{min} \in \mathbb{N}_0$
N_s^{max}	Maximum number of subgroups, $N_s^{max} \in \mathbb{N}_0$
N_s^{kill}	Punishing counter, $N_s^{kill} \in \mathbb{N}_0$
SC_s	Stagnancy counter, $SC_s \in \mathbb{N}_0$
SC_{max}	Stagnancy threshold, $SC_{max} \in \mathbb{N}_0$
$g(x_n[t])$	Monotonic and positive sensing function from position $x_n[t]$, $g(x_n[t]) \in \mathbb{R}$
p_{sp}	Probability of spawning a subgroup with the predefined number of robots N_I , $p_{sp} \in \mathbb{R}$
L	Link matrix, $L \in \mathbb{R}^{N_s \times N_s}$
A_c	Adjacency matrix, $A_c \in \mathbb{B}^{N_s \times N_s}$
$C^{(N_k)}$	Multi-hop connectivity matrix of order N_k , $C^{(N_k)} \in \mathbb{N}_0^{N_s \times N_s}$
C_B, C_{break}	Binary and break connectivity matrices, $C_B, C_{break} \in \mathbb{B}^{N_s \times N_s}$
\mathcal{L}	Laplacian matrix, $\mathcal{L} \in \mathbb{R}^{N_s \times N_s}$
Δ	Valency matrix, $\Delta \in \mathbb{R}^{N_s \times N_s}$

LOW LEVEL CONTROL

δ	Encoders-wheel resolution of robots, $\delta \in \mathbb{R}$
τ_{rev}	Number of pulses from revolution of the wheel, $\tau_{rev} \in \mathbb{N}_0$
$dist$	Euclidean distance, $dist \in \mathbb{R} [m]$
$pulses$	Number of pulses necessary for the robot to travel a distance of $dist$, $pulses \in \mathbb{N}_0$
R_{wheel}	Radius of robot's wheel, $R_{wheel} \in \mathbb{R} [m]$
R_{robot}	Radius of the robot, $R_{robot} \in \mathbb{R} [m]$
h_n	Euclidean distance for the robot n to travel, $h_n \in \mathbb{R} [m]$
θ_n^d	Absolute orientation of the robot n , $\theta_n^d \in \mathbb{R} [deg]$
θ_n	Rotation that robot n needs to perform, $\theta_n \in \mathbb{R} [deg]$
τ_n^{d1}	Number of steps necessary for robot n to rotate, $\tau_n^{d1} \in \mathbb{N}_0$
τ_n^{d2}	Number of steps necessary for robot n to move forward, $\tau_n^{d2} \in \mathbb{N}_0$

FRACTIONAL CALCULUS NOTATION

α	Fractional order coefficient, $\alpha \in \mathbb{C}$
Γ	Gamma function
T	Sampling period, $T \in \mathbb{N} [s]$

r	Truncation order, $r \in \mathbb{N}_0$
$\Delta^d x[t]$	Integer “direct” discrete difference of signal $x[t]$, $d \in \mathbb{N}_0$
$D^\alpha[x[t]]$	Approximate discrete time Grünwald–Letnikov fractional difference of order α of the discrete signal $x[t]$

DEPLOYMENT AND FAULT-TOLERANCE

k	Connectivity of the network, $k \in \mathbb{N}$
φ_n	The total angle φ_n of robot n when it is initially deployed, $\varphi_n \in \mathbb{R}$ [<i>degrees</i>]
$x_0^{i,i+1}$	Centre of the spiral based on the deployment of robots i and $i + 1$, $x_0^{i,i+1} \in \mathbb{R}^\varpi$ [m]
$\varphi_0^{i,i+1}$	Orientation of the spiral based on the deployment of robots i and $i + 1$, $\varphi_0^{i,i+1} \in \mathbb{R}$ [<i>deg</i>]
$x_{i+2}^d[0]$	Desired initial position to deploy robot $i + 2$, $x_{i+2}^d[0] \in \mathbb{R}^\varpi$ [m]
N_n	Number of “available” robots that still did not choose robot n
$\vec{Q}_n[t]$	Force vector of robot n towards the k chosen neighbours, $ \vec{Q}_n[t] = \min(k, N_n)$

PARAMETERIZATION AND ADAPTABILITY NOTATION

$\mathcal{A}, \mathcal{A}_p$	Global and particular attraction domains
x_n^*	Equilibrium point (position) from which robot n converges, $x_n^* \in \mathbb{R}^\varpi$ [m]
v_s	Centre-of-mass velocity of swarm s , $v_s \in \mathbb{R}^\varpi$ [$m \cdot s^{-1}$]
A_s	Swarm activity of swarm s , $A_s \in \mathbb{R}$
S_n	Level of socialization of robot n (robot socialization), $S_n \in \mathbb{R}$
O_n	Level of susceptibility to obstacles of robot n (robot avoidance), $O_n \in \mathbb{R}$
d_{nm}	Distance between robot n and its nearest neighbour m , $d_{nm} \in \mathbb{R}$ [m]
P_n	Level of proximity of robot n to its nearest neighbour (robot proximity), $P_n \in \mathbb{R}$
C_s	Level of connectivity of the swarm s , $C_s \in \mathbb{R}$
γ_{FDO}	Damping coefficient
ω_{FDO}	Restoring force coefficient
ε_{FDO}	Non-linearity coefficient in the restoring force
Γ_{FDO}	Amplitude of the periodic driving force
Ω_{FDO}	Frequency of the periodic driving force

SEMI-MARKOV MACROSCOPIC MODELLING NOTATION

N_e	Number of robots in Search state
N_w	Number of robots in Obstacles Avoidance state
N_r	Number of robots in Communication Interference state
\bar{v}	Average velocity of robots [$m \cdot s^{-1}$]
R_r	Communication interference radius [m]
A_a	Useful area of the scenario [m^2]
ρ_w	Density of obstacles
T_e	Delay spent on the Search (Wandering) states [s]
T_w	Delay spent on the Obstacle Avoidance state [s]
T_r	Delay spent on the Communication Interference state [s]
T_{exc}	Delay spent as socially excluded [s]
T_{inc}	Delay spent as socially active [s]
γ_e	Mission-related detection rate
γ_w	Obstacles' encountering rate
γ_r	Robots' communication interference rate
γ_{exc}	Social exclusion rate
γ_{inc}	Social inclusion rate
p_e	Transition probability between Search sub-state
p_w	Transition probability between Search and Obstacle Avoidance states
p_r	Transition probability between Search and Communication Interference states
p_{exc}	Transition probability between socially active and excluded robots
p_{inc}	Transition probability between socially excluded and active robots

List of Abbreviations

<i>Abbreviation</i>	<i>Description</i>
<i>AFS</i>	Aggregations of Foraging Swarm
<i>AODV</i>	Ad hoc On-demand Distance Vector
<i>API</i>	Application Programming Interface
<i>CHOPIN</i>	Cooperation between Human and rObotic teams in catastroPhic INcidents
<i>CLT</i>	Central Limit Theorem
<i>DE</i>	Difference Equation
<i>DPSO</i>	Darwinian Particle Swarm Optimization
<i>EPSO</i>	Extended Particle Swarm Optimization
<i>eSwarBot</i>	Educative Swarm Robot
<i>EST</i>	Extended Spiral of Theodorus
<i>FDO</i>	Forced Duffing Oscillator
<i>GPS</i>	Global Positioning System
<i>GSO</i>	Glowworm Swarm Optimization
<i>ISEC-IPC</i>	Engineering Institute of Coimbra from the Polytechnic Institute of Coimbra
<i>ISR-UC</i>	Institute of Systems and Robotics from University of Coimbra
<i>LLC</i>	Low-Level Control
<i>LRF</i>	Laser Range Finder
<i>MACS-HWU</i>	School of Mathematical and Computer Sciences from Heriot-Watt University
<i>MANET</i>	Mobile Ad hoc Network
<i>MANOVA</i>	Multivariate Analysis of Variance Analysis
<i>MRL</i>	Mobile Robotics Laboratory
<i>MRS</i>	Multi-Robot System
<i>MRSim</i>	Multi-Robot Simulator
<i>PFSM</i>	Probabilistic Finite State Machines
<i>PPSO</i>	Physically-embedded Particle Swarm Optimization
<i>PSO</i>	Particle Swarm Optimization
<i>R&D</i>	Research and Development
<i>RDPSO</i>	Robotic Darwinian Particle Swarm Optimization
<i>ROS</i>	Robot Operating System

<i>RSSI</i>	Received Signal Strength Indicator
<i>SaR</i>	Search and Rescue
<i>SLAM</i>	Simultaneous Localization and Mapping
<i>TCP/IP</i>	Transmission Control Protocol and Internet Protocol
<i>UAV</i>	Unmanned Aerial Vehicle
<i>UGV</i>	Unmanned Ground Vehicle
<i>USAR</i>	Urban Search and Rescue
<i>V-REP</i>	Virtual Experimentation Platform, Coppelia Robotics

Introduction

PHYSICAL and chemical processes were the origin of all life on Earth more than 2.5 billion years ago. As far as human knowledge goes, those processes contributed to the natural evolution of nowadays species and their survival through several ages filled with catastrophic events. The reasons behind such survivability of species, although not fully explained, fall into the evolutionary cycle of *mutation*, *selection* and *replication*. However, assuming that this cycle is an individual process without inter and intra-species interactions is simply wrong – the selection mechanism itself falls into the principles of *coevolution*¹. To certify this, most palaeontologists believe that non-avian dinosaurs lacked the complex cooperation of existing mammal pack hunters such as wolves (Lewis, 2001). In other words, the dinosaurs were unable to cope, through cooperation, with the competition from mammals and the changing climate. Looking at the facts, this is, unquestionably, a highly interesting phenomenon. Note that although dinosaurs were massively extinct due to the *Cretaceous-Palaeogene extinction event* around 66 million years ago, even way before that they started to succumb. Remarkably, the only survivors born in the *Late Triassic* were some few types of birds and most insects – species known for their intrinsic cooperative, and even competitive, behaviour observed in nowadays flocks and swarms. The neologism combining *cooperation* and *competition* is often known as *coopetition* (Tsai, 2002) and nourishes the principles of natural selection, also known as *Darwin's survival-of-the-fittest* (Darwin, 1872).

Those principles, along with the way nature copes with the difficulties of life, brought forth the research towards *nature inspired*, more widely known as *biologically inspired*, manmade designs. From biologically inspired robots mimicking birds' kinematics (Couceiro, Luz, Figueiredo, & Ferreira, 2012) to complex collective aggregation of robots mimicking swarms of insects (Parker, Schneider, & Schultz, 2005), robotics has benefitted the most from biologically inspired evolution over the past few years.

Based on those principles, this Ph.D. Thesis describes the results of a three years research by proposing a complete biologically inspired swarm robotic solution that can be applied to real-world

¹ According to *Merriam-Webster* dictionary, coevolution may be defined as “*the evolution involving successive changes in two or more ecologically interdependent species (as of a plant and its pollinators) that affect their interactions*”.

missions, mainly focusing on search and rescue (*SaR*) operations due to their inherent level of complexity. This introductory chapter aims at providing the reader with an overview of the Thesis by first introducing several features and assumptions which this work is based on. Afterwards, the thesis statement is presented, thus describing the key research question and the associated subsidiary questions. This is naturally followed by outlining an account of how these research questions will be addressed. At last, we present the organisation of this document.

1.1 Features and Assumptions

Asserting robotics for real-world operations is a tough challenge in the current state-of-the-art. Therefore, they are usually clustered based on several features and assumptions defined by the type and application of the robotic strategy, among others. The present section describes the features and assumptions which the strategies presented in this Thesis hold on to. Note that some concepts introduced in these sections will only be further described later on.

Multi-Robot Systems (MRS), a particular case of *Multi-Agent Systems*, have been a field of research in constant progress in recent years since it has the potential to effectively assist humans in multiple relevant real-world application domains, such as catastrophic incidents (Murphy, et al., 2008) or nanomedicine (Al-Hudhud, 2012). Although most laboratorial applications have been well-studied in the past (Benkoski, Monticino, & Weisinger, 1991), the use of *MRS* to fulfil such real-world missions has not yet received the proper attention. Nonetheless, *MRS* still offer several advantages over single solutions, or even humans, within such applications. Besides providing a natural fault-tolerance mechanism, the use of multiple robots becomes especially preferable when the area is either hazardous or inaccessible to humans, such as performing *search and rescue (SaR)* of victims in catastrophic scenarios (Suarez & Murphy, 2011).

1.1.1 Main Application: Search and Rescue (*SaR*)

Although the solution presented in this Thesis could be applied to any real-world application that could be seen as an optimization problem, *SaR* operations will be constantly mentioned throughout the work so as to evaluate the proposed approach side-by-side with state-of-the-art alternatives. Those applications are regularly classified by some specific features that pose several problems, especially in terms of inter-robot communication, in which we can highlight the following ones (Murphy, et al., 2008):

- Large – Dimensions of real applications do not stick to laboratorial *testbeds*, going from hundreds of square meters (*e.g.*, manmade disasters) to hectares of land (*e.g.*, natural disasters);
- Highly dynamic – Scenarios change over time, either due to agents' mobility and actions or due to external factors (*e.g.*, collapses, fires, floods, earthquakes, explosions);
- Harsh and Faulty – Abundant presence of abrasive dust, water, chemical substances and a wide range of obstacles in the scenarios;

It is with these real-world problems in mind that *swarm robotics* is seen as the most promising class of *MRS*.

1.1.2 Multi-Robot System (*MRS*): Swarm Robotics

This Thesis will focus on swarm robotic techniques applied to *SaR* tasks. Those techniques offer several major benefits over the conventional search techniques, such as the robustness of the swarm to individual units failure or run-time addition of new units, the scalability of emergent behaviours to swarms of different sizes, the leveraging of self-organization principles of environmental noise and individual differences, and the synergetic effect whereby the work of the swarm is greater than the sum of the work by the individual units (known as *superlinearity*) (Floreano & Mattiussi, 2008) – a concept shared by other fields, such as complex systems.

On these grounds, other examples of potential applications for swarm robotics include military missions, unmanned space exploration, environmental data collection, and others. However, regardless the application type, all of them require two particular features: a *distributed architecture* and *explicit inter-robot communication*.

1.1.3 Architecture: Distributed

This work will focus on distributed solutions without a central task allocator. This is evident within swarm robotics context in which tasks are inherently distributed in space, time, or functionality and, as such, this section could have been neglected. Nevertheless, it should be noted that some works still emphasize on centralized architectures (Li, Alvarez, De Pellegrini, Prabhakaran, & Chlamtac, 2007), thus moving away from the fully distributed nature inherent to the principles of collective intelligence. In practice, centralized swarm architectures are computationally expensive and unsuitable as a large number of robots usually generates very dynamic behaviours that a centralized controller cannot handle (Sahin E. , 2005). Also, centralized architectures lack robustness as the failure of the centralized entity may compromise the performance of the whole *MRS* (Parker L. E., 2008a).

1.1.4 Inter-Robot Communication: Mobile Ad hoc Network (*MANET*)

Although many different swarm robotic algorithms have been proposed recently in the literature with different communication paradigms, this work will focus on the ones that benefit from explicit communication, such as the work recently proposed by Kernbach *et al.* (Kernbach, et al., 2013). In algorithms under explicit communication, robots need to be able to explicitly exchange information within a network path using some sort of a medium (*e.g.*, wireless communication). Despite such requirement, the choice of explicit communication over alternatives, such as *stigmergy*², relies on the application domain of realistic applications such as *SaR*. According to the current state-of-the-art in this field, robotic technology is used, almost exclusively, to assist and not to substitute human responders (Murphy, 2004). Hence, multiple mobile robots can take advantage of parallelism to reduce the time required to fulfil the mission, while explicitly providing important data about the site (*e.g.*, contextual information), whether accessible or inaccessible for human agents. To do so, they need to be endowed with an explicit communication medium.

However, while cooperative architectures usually assume a reliable pre-existent communication network to support teamwork (Rocha R. P., 2006a), in many real-world situations, robots have to move to complete their tasks while maintaining communication among themselves without the aid of a communication infrastructure. This is a typical scenario from real-world applications such as *SaR*, among others (*e.g.*, hostile environments, disaster recovery, battlefields, space), in which the communication infrastructure may be damaged or missing. In such situations, inter-robot communication must be explicitly dealt with within a mobile ad hoc network (*MANET*), since its interruption may imply impaired team performance, loss of robots and, in the worst case, mission failure.

1.2 Research Question

Considering what was previously stated, as their counterpart biological inspirational models, such as swarms of ants or bees, swarm robotics' potential advantages are not only related with space and time distribution, as *MRS* in general are, but also with its robustness to flaws given the naturally inherent distributed nature and emergent collective behaviour of simplistic robots (Beni, 2004; McLurkin J. D., 2004). However, this also raises several issues regarding the applicability of swarm robotics to real-world situations one needs to overcome, thus leading to the following *key research question*:

² *Stigmergy* is a form of self-organization that comprises a mechanism of indirect coordination between agents or actions, usually observed in insects such as ants. This will be better explained further ahead in the next chapter.

“Is it possible to fully accomplish real-world missions, such as search and rescue, by means of a swarm robotic solution?”

Due to the generality of the research question at stake, one will fragment the key problem into five subsidiary problems. As such, let us introduce each one of the subsidiary problems encountered during the implementation of swarm robotic approaches into real-world problems.

1.2.1 Subsidiary Question 1: Autonomous Deployment of Robots

The literature shows evidence that swarm algorithms, either applied to simple optimization problems or to complex robotic applications, are, in essence, chaotic systems (Liu & Abraham, 2009). As a consequence, they are extremely susceptible to changes in the initial conditions (Gleick, 1987). Moreover, as if this was not enough, swarm strategies usually perform better the more scattered throughout the workspace agents are (Beni, 2004). In real-world robotic applications, this presents itself as an even more complex problem due to communication constraints inherent to non-existent infrastructures. Therefore, to perform a given cooperative task such as *SaR*, robots need to explicitly exchange information between themselves, while creating and maintaining a *MANET*. Therefore, in practice, robots should be initially positioned in the scenario, in a strategic scattered fashion, to not only guarantee the success of the swarm, but to also ensure that robots are able to establish a *MANET*. Moreover, and to also guarantee the accuracy of the deployment strategy, this should be carried out in an autonomous fashion, *i.e.*, without human assistance. This leads to the following *subsidiary question 1*:

“How to initially deploy the robots throughout the scenario, in an autonomous fashion, so as to guarantee that they are scattered enough and, at the same time, create a Mobile Ad hoc Network?”

1.2.2 Subsidiary Question 2: Communication in Faulty Environments

Considering the need of having a *MANET*, as stated above, one needs to ensure that it is pervasive against the harsh conditions observed in real-world missions (*e.g.*, wide range of obstacles in the scenarios) (Derbakova, Correll, & Rus, 2011). As such, robots should be able to not only create a *MANET* between themselves, but to also ensure that the *MANET* remains connected throughout the mission. As those networks comprise of a large number of distributed nodes (*i.e.*, robots), multi-hop paths should be established, *i.e.*, robots need to perform the roles of both hosts and routers. Moreover, given the harsh conditions of real-world applications such as *SaR*, fault-tolerance strategies need to

be considered to prolong the *MANET* lifetime and prevent loss of connectivity. This leads to the following *subsidiary question 2*:

“How to ensure that robots can maintain a pervasive multi-hop Mobile Ad hoc Network under faulty environments?”

1.2.3 Subsidiary Question 3: Efficient Sharing of Information

The emerging collective behaviour of swarm systems depends, not only, on the local cognitive solution of each agent, but also on the global social solution of the swarm. As such, it is necessary that each robot maintains a sufficient and consistent level of awareness about the mission assigned to the swarm and about its teammates' location and solution. This situation awareness may allow us to overcome complex problems decomposition, while keeping a certain level of robustness and reliability (Rocha R. P., 2006b; Hsieh, Kumar, Cowley, & Taylor, 2008). Nevertheless, it needs to be done while minimizing the number of inter-robot exchanged messages so as to reduce the communication overhead, the power consumption and the loss of packets. For that reason, robots should efficiently share the information in a distributed manner via the self-created communication channel to foster inter-agent cooperation without jeopardizing the success of the mission. This leads to the following *subsidiary question 3*:

“How to efficiently share information between robots without decreasing the collective performance of the swarm?”

1.2.4 Subsidiary Question 4: Adaptability to Dynamic Environments

Real-world applications, such as *SaR*, are known for their highly dynamic characteristics caused by human interaction, weather conditions, fires, explosions and other phenomena. Most of those changes are not considered and, consequently, solutions are proposed for specific controlled or structured tasks without dynamic interferences (Saikishan & Prasanna, 2010; Pasqualetti, Durham, & Bullo, 2012). As we aim at applying swarm robotics to a real-world context, such assumptions do not hold. For that reason, robots need to be capable of adapting their behaviour based on the retrieved agent-based, mission-related and environmental contextual information. Such information may be assessed either at the individual or the collective level, and should tune robots' behaviours toward an improved convergence. This may not only avoid the stagnation of the swarm into sub-optimal solutions, but also enable tracking dynamic sources, such as plume tracking problems (*e.g.*, gas leak) (Marques, Nunes, & Almeida, 2006). This leads to the following *subsidiary question 4*:

“How to retrieve contextual information and adapt robot’s behaviour to dynamic problems so as to improve the collective performance of the swarm?”

1.2.5 Subsidiary Question 5: Estimation of a Provable Performance

Proving that a given swarm robotic architecture will succeed on a given mission has been a subject neglected by most researchers on swarm robotics (Agassounon, Martinoli, & Easton, 2004). The stochasticity inherent to swarm robotic algorithms makes it difficult to predict the teams’ performance under specific situations and, henceforth, almost impossible to synthesize the most rightful configuration (*e.g.*, to determine the adequate number of robots in a team). As such, most works have been trying to answer those questions by means of exhaustive experimentation and trial-and-error strategies. To go against this trend, and contrarily to the previous subsidiary questions which may find their answers in experimentation and engineering, this question requires a mathematical and analytical approach to synthesize swarm robotic teams. This leads to the following *subsidiary question 5*:

“How to accurately estimate the collective performance of a swarm of robots under a given task?”

1.3 Approach

The scientific goal of this Ph.D. project is to answer the aforementioned questions and, as a result, cross the reality gap around the applicability of swarm strategies to real-world scenarios. The research questions stated in the previous section are sequentially addressed in this Thesis by progressively following the document structure.

Generally speaking, this work aims to demonstrate the recent progress being made in enabling *MRS*, in particular swarm robotic teams, to operate reliably in faulty communication environments, by developing a proof of concept of cooperative mobile robots using *MANETs*, wherein mobile robots act as relay nodes (multi-hop connection) using off-the-shelf technology. As a first step, this Thesis calls upon the historical background and state-of-the-art, by getting inspiration from a broad sampling of the research currently ongoing in the field of distributed mobile robot systems, namely swarm robotics with real-world applicability. Since the behaviour-based paradigm for swarm robotics is rooted in biological inspirations, it is found instructive and motivating to examine the social characteristics of insects and animals in general, and to apply these findings on the design of *MRS* (chapter

2). All the herein proposed techniques get inspiration from several mathematical tools and are implemented and experimentally validated by both simulation and physical robots (chapter 3), with the perspective of technology transfer and the establishment of an increased value in this field.

More specifically, the answer to the key research question relies on a complete swarm robotic solution, herein denoted as *Robotic Darwinian Particle Swarm Optimization (RDPSO)*; a *Particle Swarm Optimization (PSO)* (Kennedy & Eberhart, 1995) based evolutionary approach adapted to real-world *MRS*. The fundamentals around the *RDPSO* will be introduced in chapter 4 in which special attention will be given to the translation between optimization problems and real-world applications, by focusing on robots' dynamics, obstacle avoidance, communication constraints and, as a key component, the evolutionary properties. These evolutionary properties endow the *RDPSO* with the ability to dynamically partition the swarm of robots into subgroups. As such, the *RDPSO* comprises of several dynamical subgroups, in which each one corresponds to an effective cooperative *MRS* and, as a consequence, to a single *MANET*.

The other chapters further extend the *RDPSO* to real-world applications by sequentially answering each one of the five subsidiary questions.

1.3.1 Subsidiary Approach 1: Autonomous Deployment of Robots

Taking the initial deployment of robots within the environment as a basis for applying swarm systems into real-world applications, chapter 5 reports the development and evaluation of a realistic, autonomous and fault-tolerant deployment strategy. For that end, the population of robots is hierarchically divided, wherein larger support platforms sequentially and autonomously deploy smaller exploring platforms in the scenario, while considering communication constraints and obstacles. This ensures that the *MANET* can be established at the beginning of the mission, wherein every deployed robot acts as a network node.

1.3.2 Subsidiary Approach 2: Communication in Faulty Environments

After the deployment, still in chapter 5, a way of ensuring an adjustable level of connectivity between robots, to explicitly exchange information needed in collaborative real-world task execution, is introduced. This chapter studies how fault-tolerance can be addressed within the *RDPSO* by benefiting from attractive and repulsive forces so as to locally maintain the connectivity of the multi-hop *MANET*. Moreover, the strategy not only considers the maximum communication range between robots, but also the minimum signal quality, thus refining its applicability to real-world context.

1.3.3 Subsidiary Approach 3: Efficient Sharing of Information

Given the intrinsic distributed architecture and simplicity of swarm agents, chapter 6 analyses the architecture and characteristics of the *RDPSO* communication system, thus describing the dynamics of the communication data packet structure shared between teammates. Such procedure will be the first step to achieve a more scalable implementation of the *RDPSO* by optimizing the communication procedure between robots. Secondly, the so far adopted communication reactive routing protocol is extended based on the *RDPSO* concepts, so as to reduce the communication overhead within swarms of robots.

1.3.4 Subsidiary Approach 4: Adaptability to Dynamic Environments

The highly dynamic characteristics of real-world applications motivated chapter 7 to ultimate the *RDPSO* design by further extending it and adapting the behaviour of robots based on a set of context-based evaluation metrics. Those metrics are then used as references so as to systematically adjust the *RDPSO* parameters, thus improving its convergence rate, susceptibility to obstacles and communication constraints. All the previous features, including this last one, are evaluated side-by-side with state-of-the-art alternatives in chapter 8, under both simulation and real experiments.

1.3.5 Subsidiary Approach 5: Estimation of a Provable Convergence

In the course of the research process, special attention was given to the implementation and validation of the proposed approaches by means of numerical methods, computer simulations and real experiments with multiple, and different, physical robotic platforms. On top of that, and due to the limitations inherent to each one of those (*e.g.*, number of robots, scenario dimensions, real-world constraints), chapter 9 reports the concluding contribution of this Thesis by proposing a macroscopic model able to capture the *RDPSO* dynamics and, to some extent, analytically estimate the collective performance of robots under a certain task.

1.4 Outline

The Thesis is organized in ten chapters divided by several sections each. After this introductory chapter, chapter 2 provides the reader with a historical background, highlighting the importance of cooperative systems, and also surveying thoroughly the most relevant state-of-the-art related with cooperative multi-robot and multi-agent systems, mainly focusing on bio-inspired techniques applied to real world applications. The purpose is to go deeply into the motivation and main issues related with

cooperative robots and *MANETs*. Note, however, that this chapter will not encompass all of the state-of-the-art this Thesis relies on. Due to the diversity of the content, the state-of-the-art was split into the several chapters to contextualize each chapter's contributions and to motivate the reader.

Chapter 3 presents the relevant mathematical concepts, theorems and definitions, as well as the technological tools used throughout this work. In brief, this chapter brings together the necessary background to support the reader by centralizing all the preliminaries into a single chapter.

Chapter 4 introduces the core of this Thesis – the *Robotic Darwinian Particle Swarm Optimization (RDPSO)* and, as a consequence, it corresponds to a significant portion of this Thesis. First, the relevant swarm techniques such as *PSO* and the *Darwinian PSO (DPSO)* are introduced to the reader. Afterwards, a set of strategies is introduced to go from the typical optimization method to a swarm robotics approach, thus considering real-world constraints such as obstacles and communication. After fully describing the *RDPSO* and all of its components, numerical and experimental results with real-world robots are presented to fully show the advantages of the proposed strategy. This chapter is a refined compilation of several papers that were a successive improvement of the *RDPSO* first presented on the *IEEE International Symposium on Safety, Security, and Rescue Robotics* (Couceiro, Rocha, & Ferreira, 2011a), and further extended in the *IEEE/RSJ International Conference on Intelligent Robots and Systems* (Couceiro, Rocha, Figueiredo, Luz, & Ferreira, 2012) and in the journal of *Advanced Robotics* (Couceiro, Rocha, & Ferreira, 2013a).

Chapter 5 advances into progressively turning the *RDPSO* into a more appropriate swarm robotic algorithm for real-world applications domain. This is followed by a natural extension of the previous chapter to maintain the *MANET* connectivity during the mission with fault-tolerance capabilities. Both strategies are first evaluated under a large scenario with up to 15 physical robots and subsequently further evaluated under computational experiments with a larger number of robots. A major part of this chapter was submitted to the *Robotics and Autonomous Systems* (Couceiro M. S., Figueiredo, Rocha, & Ferreira, 2014 (In Press)) and presented at the *IEEE International Conference on Robotics and Automation* (Couceiro, Rocha, & Ferreira, 2013b).

Chapter 6 further analyses the communication complexity of the *RDPSO* by studying its architecture and characteristics, thus describing the dynamics of the communication data packet structure shared between teammates. Secondly, the Ad hoc On-demand Distance Vector (*AODV*) reactive routing protocol is extended based on the *RDPSO* concepts, so as to reduce the communication overhead within swarms of robots. The content in this chapter was recently published in the *IEEE Congress on Evolutionary Computation* (Couceiro, Rocha, Ferreira, & Vargas, 2013) and further described in the *Robotica Cambridge Journal* (Couceiro, Fernandes, Rocha, & Ferreira, 2013 (In Press)).

Chapter 7 goes a step further into extending the *RDPSO* with adaptability features. The idea is to first present an attraction domain, based on stability theory, wherein the *RDPSO* parameters may be defined to ensure the convergence of robots. Afterwards, the parameters are subjected to fuzzy rules so as to overcome the real-world inherent unpredictability and dynamic phenomena based on contextual information retrieved by the robots. This adapted version is compared with the non-adaptive one to extract evidences of the obtained improvements. This chapter is substantially based on the papers published at *Robotics and Autonomous Systems* (Couceiro, Machado, Rocha, & Ferreira, 2012) and the *Journal of Intelligent and Robotic Systems* (Couceiro M. S., Martins, Rocha, & Ferreira, 2013a (Under Review)).

Chapter 8 presents a survey on multi-robot search inspired by swarm intelligence, thus classifying and discussing the theoretical advantages and disadvantages of the existing studies. To that end, the most attractive techniques are evaluated and compared with the *RDPSO* fully described along the previous chapters. This chapter presents experiments conducted to benchmark five state-of-the-art algorithms for cooperative exploration tasks. This was a major part of the work developed at *Robotics Lab* from the *School of Mathematical and Computer Sciences* at *Heriot-Watt University (MACS-HWU)*, United Kingdom, under the supervision of Professor Patricia Vargas, which resulted in a publication submitted to the *Robotics and Autonomous Systems* (Couceiro, Vargas, Rocha, & Ferreira, 2014).

Chapter 9 proposes a semi-Markov macroscopic model able to capture the *RDPSO* dynamics and, to some extent, estimate analytically the collective performance of robots under a certain task. The model is explored step-by-step throughout the chapter and compared to its microscopic counterpart by experimental means. This was the last effort to come to a closure of this Ph.D. work that was submitted in the *International Journal of Robotics Research* (Couceiro M. S., Martins, Rocha, & Ferreira, 2013b (Under Review)).

Chapter 10 outlines this Ph.D. Thesis pointing out some short-term and futuristic long-term perspectives of the future work that may be developed.

Background and Motivation

EXISTING cooperation in the various societies (*e.g.*, ants, bees, plants, humans and others) inspired researchers to place a considerable amount of effort in developing robots that are able to cooperatively perform tasks. This chapter highlights the importance of cooperation in societies (section 2.1) and presents earlier works in Multi-Robot Systems (*MRS*), focusing on the interest in biologically inspired cooperative systems and the issues related to the applicability of swarm robotic techniques to real-world missions such as *SaR* (section 2.2).

The key points of the concepts described in this section will be used to determine the best strategy to be addressed in this work.

2.1 Philosophy

“*Man is a natural animal and, inevitably, selfish*” has been the beginning of all the discussions about capitalism since the early stages reinforced by the powerful and seemingly scientific notion of “*survival of the fittest*”. Charles Darwin defended that in what turned out to be one of the most important works in the history of science: *The Origin of Species* (Darwin, 1872). The theory of natural selection defended by Darwin concluded that not all organisms, at birth, had the same survival abilities and that only those which adapted better to the environment survived. Put in a less complicated way, Darwin believed that the evolution of species was like the “*law of the jungle*”, where only the brightest would survive and evolve, while all the rest disappear or hardly survive. Darwin’s theory has been applied mainly on a biological level, but over the years, it turned out to be applicable also to the economic and social competition.

After all, this is not so far from reality as the main critics of Darwinism tried to prove. Men, like any other living being, hierarchically divide their needs. The psychologist Abraham Maslow (Maslow, 1943) developed the theory of motivation by illustrating a hierarchy of needs that man seeks to satisfy. These requirements are typically represented in the form of the *Maslow’s Pyramid* (Figure 2.1).



Figure 2.1. Maslow's Pyramid.

As we can see, according to the previous model, men have to meet their physiological and safety needs before entering the levels of interpersonal and intergroup relations. We can see that this makes sense and, ultimately, there is a certain analogy in nature. Wolves hunt in groups to obtain a better performance in order to get enough food for the whole group. However, if something unforeseen happens (*i.e.*, more efficient predators or such in large groups encounters) each element in the pack of wolves feels the need to escape and survive alone. A more common example for us (humans) is the simple fact that we need to eat and rest before we can accomplish the tasks which we are responsible for in our society.

2.1.1 The Essence of Cooperation

However, there is probably a feature that at the time did not occur to Darwin and Maslow: the survival of a particular member of a society may depend upon the cooperation with other members of this society, or even other societies.

Kevin Foster (Foster & Xavier, 2007), who has taken further the work of William Hamilton, proved that there are situations of cooperation between individuals which do not fit the basic principle of Darwin, arguing that altruism is a way that nature has to assert itself. In June 2008, Kevin Foster said, at the *Institute of Molecular Pathology and Immunology* in the *University of Porto*, Portugal, that cooperation is everywhere: “*The genes have joined in the genomes, the cells work together in multicellular organisms and animals cooperate in societies*”.

Many other examples in the various societies of our world demonstrate the importance of cooperation in nature and how it can be essential.

Thousands of years ago, King Solomon, who was a student of nature, observed the humble ant and wrote: “Go to the ant, you sluggard; consider its ways and be wise! It has no commander, no overseer or ruler, yet it stores its provisions in summer and gathers its food at harvest” (Dean, 1913). In fact, ants are a perfect example of cooperation, diligence and order. In addition to working together and helping each other, ants seem to be able to find their paths (the nest to a food source and back, or just get around an obstacle) with relative ease, despite being virtually blind (Figure 2.2). Several studies have found that in many cases this capacity is the result of the interaction of chemical communication between ants (a substance called *pheromone*³) and the emergent phenomenon caused by the presence of many ants. This is the concept of *stigmergy* (Aras, Dutech, & Charpillet, 2004). This mechanism is so efficient that there are computational algorithms that use this principle. Such is the case of the heuristic principle *Ant System* that simulates the behaviour of a group of ants working together to solve an optimization problem by using simple communications (Dorigo & Stützle, 2004), and the case of *Brood Sorting* (group selection) used in swarms of robots (Wilson, Melhuish, Sendova-Franks, & Scholes, 2004). Those and other similar principles can be seen in other optimization algorithms, such as genetic algorithms, evolutionary strategies and the well-known *PSO* initially proposed by Kennedy and Eberhart (Kennedy & Eberhart, 1995).

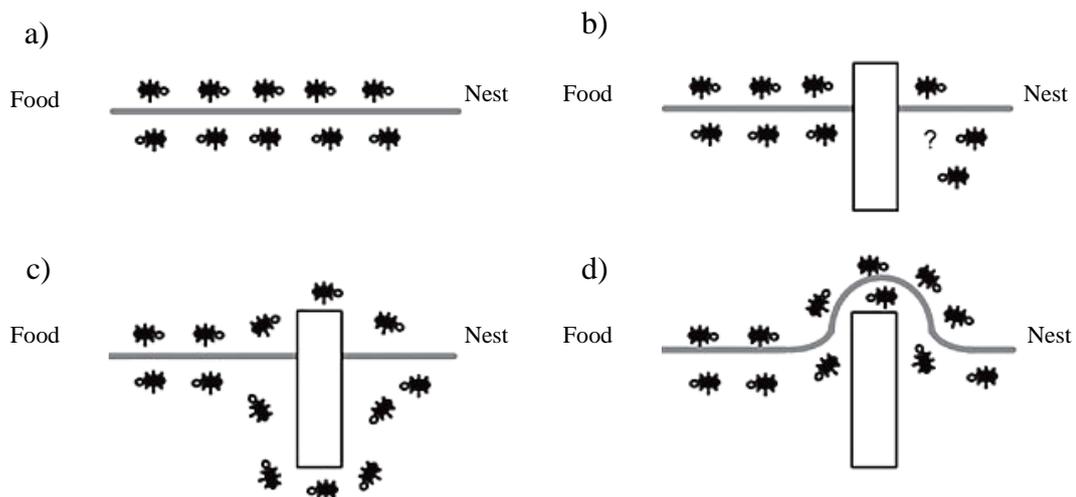


Figure 2.2. *Stigmergy* between ants: a) travelling a *pheromone* path between the nest and the food; b) an obstacle interrupts the path; c) the ants create two different paths in order to overcome the obstacle; d) a new *pheromone* path is created around the obstacle - the shortest path.

³ According to *Merriam-Webster* dictionary, *pheromone* may be defined as “a chemical substance that is usually produced by an animal and serves especially as a stimulus to other individuals of the same species for one or more behavioural responses”.

Another interesting engineering example based on biological cooperation is reflected in the flight of pelicans. Researchers have discovered that pelicans flying in formation earn extra boost when compared to the ones flying forward, resulting in a 15% reduction in the heart rate. In order to validate this concept, a group of engineers prepared a flight test with electronic equipment that enabled the pilot to keep the plane at a distance of 90 m (with a small tolerance of 30 cm) over the plane that was ahead. As such, the plane suffered an air resistance 20% lower and it consumed 18% less fuel. These results can be used in military or civilian planes, but also in the concept of robotics to improve the dynamics of flying robots to monitor forest fires (Martínez-de Dios, Merino, Caballero, Ollero, & Viegas, 2006) or biologically inspired robots for spying (Couceiro M. S., Figueiredo, Fonseca Ferreira, & Tenreiro Machado, 2009).

In fact, we can see mutual support at all levels of life, from microbes to man, and between related or different species. However, when we speak about cooperation we should, instead, speak about *cooperative systems*, wherein we cannot classify the group as the sum of contributions (“*zero-sum game*”) but yet as the interception of all contributions (“*non-zero-sum game*”) (Colman, 1995). The 3C Model of *communication*, *coordination* and *cooperation* adapted from Ellis *et al.* presented in Figure 2.3 (Ellis, Gibbs, & Rein, 1991) shows a good schematic of cooperative systems.

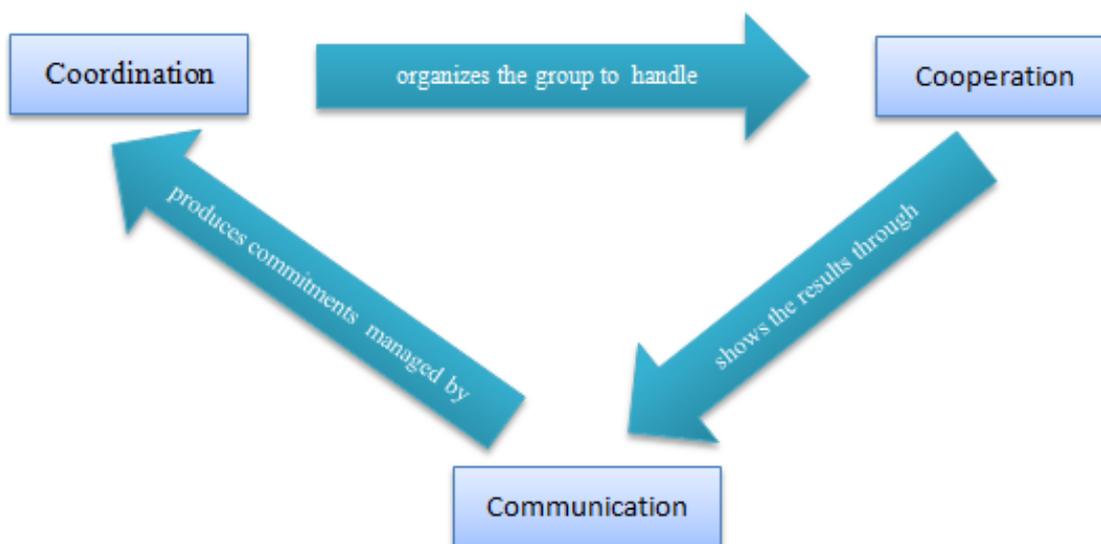


Figure 2.3. Adapted 3C Model.

The cooperation is one of the essential tools in cooperative systems since, without the collaboration between different members of a particular group or society, cooperative systems cannot survive. On the other hand, to cooperate, communication is essential between group members and this communication must be familiar to all of them. The coordination also plays an important part in

cooperative systems, since it organizes the group to prevent that communication and cooperation efforts are lost and that tasks are performed in the correct order, at the correct time and meeting the constraints and objectives.

This and other similar models related to the dynamics and cooperation in groups have been studied in several areas, including computer science (Borghoff & Schlichter, 2000) (Fuks, Raposo, Gerosa, & Pereira de Lucena, 2003) and robotics (Cao, Fukunaga, & Kahng, 1997) (Jung, 1998) (Rocha R. P., 2006a).

2.1.2 Robots: The Rise of a New Society?

Inspired by the results of the existing cooperation in several biological societies (*e.g.*, ants, bees, plants, humans), researchers have placed a great emphasis on developing robots that can cooperate with each other and perform multiple tasks. However, the following question arises:

“Does it make sense to think of a group of robots as a cooperative system or even a society?”

Given the context of this work the answer can only be one: Yes, it makes sense.

Remembering the earlier definition of cooperative systems in which these are based on the interception of the contribution of each member (*i.e.*, robot), if there is a group of robots cooperating to perform a given task, they need to communicate⁴ with each other in order to coordinate their actions and obtain the desired result. This is the concept of Multi-Robot System (*MRS*) (Parker L. E., 1994) that offer a countless number of advantages similar to the benefits of cooperative systems in other societies that may be described in the following key factor: *time*.

The time is, without any doubt, the most relevant variable to every single earthling since the early days to the present. Time does not stop and requires a careful management – the loss of time is represented as one of the biggest fears of nowadays society. One way to circumvent the limitations inherent to the concept of time is to perform simultaneous procedures (parallelism). This is valid for all biological creatures and even non-biological such as robots: if there are multiple robots instead of one they can act on multiple places at the same time (*i.e.*, spatial distribution) and they can perform multiple tasks simultaneously (*i.e.*, temporal distribution).

⁴ Note that communication does not need to be explicit – robots can use implicit communication (*e.g.*, typically in robot swarms) or simply observe the teammates' actions. This will be better explained in section 2.1.3.

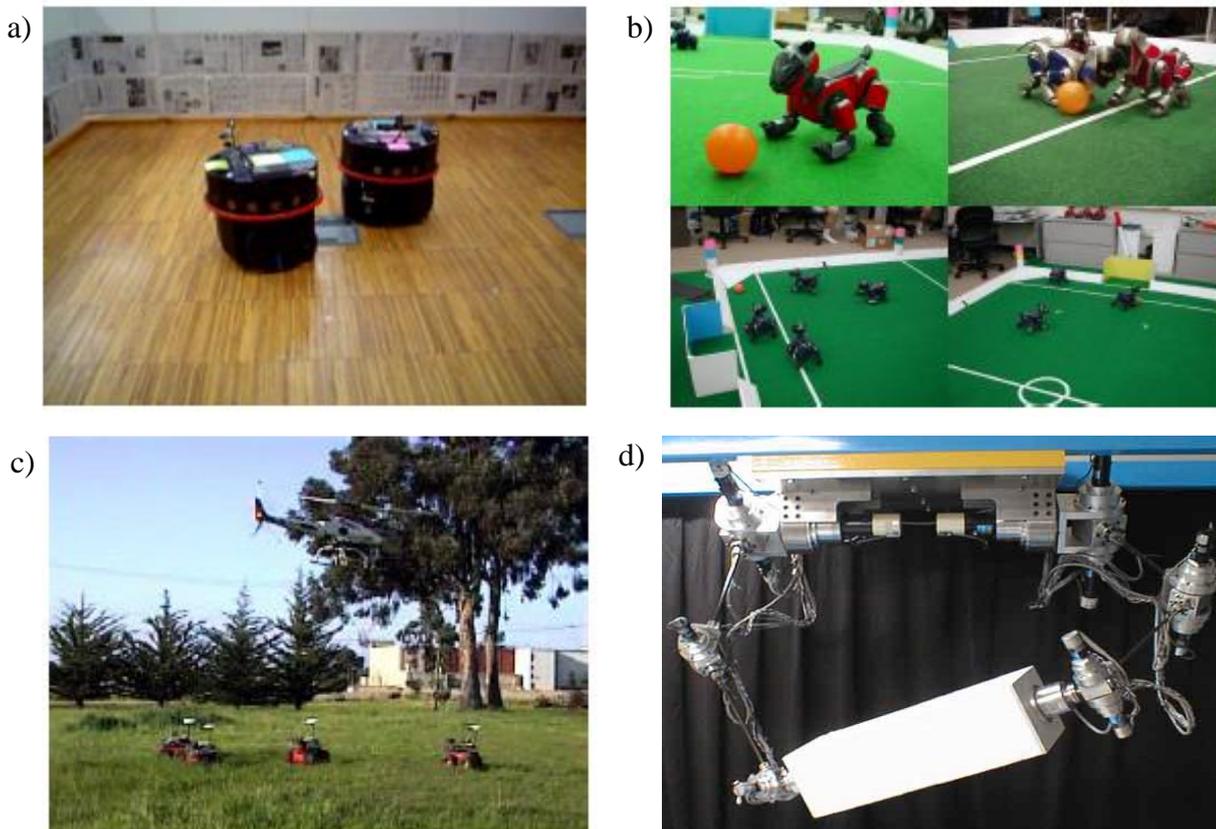


Figure 2.4. Cooperation in robotics: a) 3D mapping with two robots (Rocha, Dias, & Carvalho, 2005); b) Robotic football - *RoboCup Legged Robots League* (Veloso, Uther, Fujita, Asada, & Kitano, 1998); c) Pursuit of multiple targets with multiple unmanned ground vehicles (*UGV*) and one unmanned aerial vehicles (*UAV*) (Vidal, Shakernia, Kim, Shim, & Sastry, 2002); d) Two robotic manipulators moving an object (Ferreira, 2006; Takahashi, Ise, Konno, Uchiyama, & Sato, 2008).

However, there are certain characteristics that robots must possess in order to belong to a *MRS* (Rocha R. P., 2006a):

- They need to be able to interact with dynamic environments (and possibly unstructured ones);
- They need to be able to react (specific situations) and take deliberative actions (usual processing);
- They need to be able to perform tasks without supervision (autonomous);
- They may have other features such as being sociable and be able to adapt and learn.

This last point takes us to *MRS* as a society. The definition of society is a group of individuals living together, having some kind of organization and tasks division and interacting with each other. All of the groups mentioned above (*i.e.*, men, ants, wolves, birds and even robots) represent different societies.

Even though this concept of robot society and robots living in our society started with the fiction books of Isaac Asimov (Asimov, 1982), it gained more and more emphasis on the real world and roboticists believe that the society of robots will emerge over the next ten years⁵. Some scientific studies have been paying special attention to cooperation between robots and humans, known as human-machine interaction (*HMI*), or human-robot interaction. Real-world applications that require human interaction, such as *SaR* operations, are good examples to show *HMI* potentialities. For instance, in the *World Trade Center* September 11, 2001 tragedy, it was possible to achieve a first step towards an interaction between human rescuers and robots by means of *explicit communication* (Casper & Murphy, 2003).

2.1.3 The Role of Communication

Communication is a major part of our daily lives since it is the act of passing or sharing information between individuals. Because we live in a complex society, we depend upon communication to help our lives run more smoothly – communication may be seen as the “*glue*” that holds society together. However, as biological societies are so different in nature (*e.g.*, swarms of insects or flocks of birds), communication also comes in many forms and involves the use of signals (*e.g.*, sound, look, motion). Every animal, from the smallest insect to humans, uses some type of communication since the ability to share information is an important survival tool. Although they have different lifestyles than humans, other animals use communication for many of the same reasons, such as getting food, staying safe, finding a mate, and protecting territory.

In *MRS* the objective of communication is very similar – usually, in order to accomplish a given task (*e.g.*, finding an object), robots must share information (*e.g.*, about what they are sensing). Yet, more sharing requires more resources (*i.e.*, time, sensory effort and communication bandwidth) since the amount of information that must be shared determines how well coordinated a task is (Klavins, 2002; Rooker & Birk, 2006).

A fundamental assumption in *MRS* research is that globally coherent and efficient solutions sometimes can be achieved through the interaction of robots lacking complete global information (Parker L. E., 2008b). However, achieving these globally coherent solutions typically requires robots to obtain a minimal information about their teammates’ states or actions. To this end, robots need to benefit from their communication capacities.

⁵ www.inovacaotecnologica.com.br/noticias/noticia.php?artigo=robos-industriais-tem-tres-desafios-para-participar-da-sociedade-robotizada

The way robots communicate can be divided into basically three most common techniques:

- Implicit communication “*through the world*” (*i.e.*, *stigmergy*) – robots sense the effects of their teammates’ actions through the effects they leave on the environment (Deneubourg, Goss, Sandini, Ferrari, & Dario, 1990; Kube & Zhang, 1993; Beckers, Holland, & Deneubourg, 1994; Mataric, 1995; Onn & Tennenholtz, 1997; Werger, 1999);
- Passive action recognition – robots use sensors to directly observe the actions of their teammates (Huber & Durfee, 1995);
- Explicit (intentional) communication – robots directly and intentionally communicate relevant information to their teammates through some active means (*e.g.*, radio) (Asama, Ozaki, Matsumoto, Ishida, & Endo, 1992; Jennings, 1995; Tambe, 1997; Parker L. E., 1998).

From the three techniques described above, the use of explicit communication is the most appealing method because of its directness and ease with which robots can become aware of the actions and/or goals of their teammates or even human teams. The major uses of explicit communication in multi-robot teams are to synchronize actions, exchange information, and to negotiate between robots. Furthermore, explicit communication is a way of dealing with the hidden state problem, in which limited sensors cannot distinguish between different states of the world that are important for task performance.

However, explicit communication also shows some limitations in terms of fault-tolerance and reliability, because it typically depends upon a noisy, limited-bandwidth communication channel that may be unable to continually connect all members of the robot team (Shannon, 1949). Consequently, approaches that make use of explicit communications must also provide mechanisms to handle communication failures and lost messages.

2.1.4 The Particular Domain of Swarm Robotics

Endowing robots with communication capabilities significantly increases the range of *MRS* applicability. However, choosing the most rightful cooperative architecture within *MRS* for realistic applications, on which this Thesis is sustained on, such as *SaR*, remains a challenge (Murphy, et al., 2008). Due to its potential, *MRS* have begun to rapidly branch into several other domains and the most recent one has been denoted as *swarm robotics* (Dorigo & Sahin, 2004), a domain that naturally evolved from the concepts of swarm intelligence mostly applied to computational problems such as optimization (Reynolds, 1987). As such, and similar to optimization problems in which one can distinguish exhaustive methods from biologically-inspired ones, *MRS* within *SaR* applications face the same dilemma: either decide on an exhaustive technique, in which robots sweep all the areas (Murphy, 2004),

or mimic simple local control rules of the several existing biological societies (*e.g.*, ants, bees, birds) to stochastically search the scenario (Floreano & Mattiussi, 2008). This last one is a typical feature of most *swarm robotics* algorithms (Dorigo & Sahin, 2004).

Given the advantages so far described in this chapter, the use of *MRS*, in particular swarm robotics, to overcome the real-world issues described in the literature and summarized in section 1.1, has been a common trend over the last few years (Dorigo, et al., 2011). However, the following question could still be considered:

“Can we really ensure that a biologically inspired behaviour will satisfy the requirements of the desired real-world mission?”

Several recent works have been trying to answer this question but still none has reached a final decision. For instance, Suarez and Murphy presented a broad description of more than 50 papers on animal foraging, making the analogy to *SaR* applications (Suarez & Murphy, 2011). Most works presented in this survey suggest that robots should divide the whole environment into patches, as many animals do (*e.g.*, bees), and then search within such patches. Nevertheless, and even as stated by the authors, victims, or other points of interest such as fire outbreaks, can appear anywhere. Hence, the difficulty in subdividing a search environment and defining patches within unknown scenarios still remains. The same authors claim that *SaR* robotics should focus on exhaustive search as the motivation is different from animal foraging – while animals attempt to maximize their net energy level to stay alive, robots must find victims in a search area or determine that there are none to be found. However, although biologically inspired optimization methods may seem unsuitable for *SaR* robotics at first, there are some specific applications in which one can foresee their use like, for instance, in urban fires. Urban fires are probably the most frequent catastrophic incidents in urban areas, requiring a prompt response because of life endangerment in highly populated zones and the high risk of fire propagation to buildings and parked cars in the vicinity. An urban fire in a large basement garage often frequented by people and containing inflammable materials, like in a basement garage of a shopping mall, would make for a particularly challenging *SaR* application because of the confined nature of the environment (*cf.*, Figure 2.5). As the fire evolves, the space becomes rapidly filled with smoke, with very hard visibility and an unbreathable and toxic atmosphere, which is dangerous for both victims and first responders. Moreover, victims prone to such atmosphere would be unable to survive more than 10 to 20 minutes. Therefore, the use of exhaustive search strategies within this context would be simply unfeasible.



Figure 2.5. Example of a *SaR* real application. a) Basement garage of a shopping centre; b) underground fire.

Nevertheless, despite the advantages of biological solutions, there is still a considerably large number of alternatives which one can divide into two domains: *i*) non-swarm algorithms; and, in opposition, *ii*) swarm algorithms. In general, the main differences between both are the same as observed in nature between non-swarm societies (*e.g.*, wolf packs) and swarm societies (*e.g.*, beehives), and are summarized as follows:

- **Agent** – While non-swarm societies usually comprise of complex agents capable of acting individually, swarm agents are limited, *i.e.*, in terms of robotics, the sensorial and actuation systems, as well as processing and communication units are usually poor and, as a consequence, of lower cost;
- **Population** – Typically, the population of swarm societies is considerably larger than non-swarm ones, in such a way that the removal or addition of new members within the swarm society does not affect significantly the collective performance, *i.e.*, in terms of robotics, removing some group members in a non-swarm system may strongly decrease the collective performance.
- **Architecture** – In general, swarm societies do not have a centralized agent to command all other agents since they act upon simple local control rules from which the global behaviour of the system emerges, *i.e.*, in terms of robotics, although this principle of distributed architecture is not an exclusive one in swarm strategies, while mandatory on this domain, it is not mandatory to any other domain.

The choice between either of the domains once again falls within the application itself. As previously stated, the loss of robots is expected under such realistic applications due to the harsh conditions at stake. As such, the adage “*simpler is better*” has been considered as particularly suitable to

classify rescue robotics (Murphy, et al., 2008). In other words, the failure of a single agent, or sub-group of agents, should not jeopardize the success of the mission and, at a different level, the budget of intervention teams. A curious fact is that, while the *ADEPT SEEKUR* outdoor ground robot currently costs around 80.000€, the *Ascending Technologies Hummingbird* flying robot costs around 4.000€, ideal for real outdoor applications with multiple robots (Julian, Angermann, Schwager, & Rus, 2012).

In spite of these facts, this work settles around swarm intelligence as a means to fulfilling real applications such as *SaR*. The biological world abounds in collective swarm phenomena that have important adaptive functions, ranging from coordinated movement, to nest building, and all the way to communication (Floreano & Mattiussi, 2008). The principles of self-organization are appealing for explaining biological collective phenomena where the resulting structures and functionalities greatly exceed in complexity the perceptual, physical, and cognitive abilities of the participating organisms. Examples of biological self-organization include the construction of beehives, the foraging strategies of ants, and the regulation of colony life in social insects. In all of those cases, the resulting structure emerges from the collective work of individual organisms that execute simple behaviours based on local information and do not possess a global plan of the end result or a central coordinator. As military units adapted those principles for *swarming operations*⁶, we intend to redirect and adapt those biological properties to real-world swarm robotic applications. This is a far-reaching challenge and this Thesis gives a step forward towards a full solution.

To do so, let us first survey the state-of-the-art to which this work is bounded. Once again, note that only the main facets related to this Thesis will be reflected in the following section. Due to the variety of the work *per se*, each chapter will have a brief introductory chapter, some more than others, that will play as both motivation and discussion of prior works.

2.2 Related Work

The creation of artificial devices with life-like characteristics has been pursued over the last two millennia, beginning, as so many things in our modern world did, in Ancient Greece. More recently, mainly in the last two decades, a significant progress in applied computing and robotics occurred through the application of principles derived from the study of biology. The navigation of groups of

⁶ *Swarming operations* are a type of military behaviour where autonomous, or semi-autonomous, units of action attack an enemy from several directions and then regroup.

robots, especially swarm robots, has been one of the fields that has benefited from biological inspiration (Bonabeau, Dorigo, & Theraulaz, 1999).

Nevertheless, until recently, swarm robotics applicability has been kept far from real-world missions such as *SaR*. To cross the reality gap inherent to *MRS* and, in particular, swarm robotics, many initiatives^{7,8} and worldwide research and development (*R&D*) efforts have been focusing on the full-scale deployment of such systems in real-world application environments. This chapter essentially revolves around *SaR* applications as they cover most of the challenges encountered in real-world applications. Therefore, let us first present the typical *SaR* procedure.

2.2.1 Search and Rescue (*SaR*) Procedure

According to the several meetings engaged with the *Bombeiros Sapadores de Coimbra*⁹, a Portuguese Fire Department, in the context of the *Cooperation between Human and rObotic teams in catastroPhic INcidents (CHOPIN) R&D* project¹⁰ described further ahead, much activities in *SaR* applications are in line with most of the other military operations. The enquiries show that, after receiving a call reporting a sinister, predefined firefighting teams, including a rapid intervention team, are informed and depart for the site (deployment of agents). Afterwards, the *SaR* mission begins (mission execution) with the following phases¹¹:

- Reconnaissance;
- Rescuing;
- Establishment of means of action;
- Firefighting and protection;
- Aftermath;
- Monitoring.

These phases are not necessarily sequential. For instance, if victims are found or their existence is found to be highly probable during the reconnaissance phase, rescuing becomes a priority and proceeds in parallel with the rest of the reconnaissance phase.

During the entire process, the communication once again plays a fundamental role. At the very beginning of the mission, even while teams are being deployed in the scenario, an exclusive radio

⁷ <https://sites.google.com/site/icra2012manyrobotsystems/>

⁸ <http://www.robocup.org/>

⁹ <http://sapadoresdecoimbra.no.sapo.pt>

¹⁰ <http://chopin.isr.uc.pt/>

¹¹ http://www.bvpacodesousa.pt/downloads/Manuais_ENB/Combate%20Inc.%20Urbanos%20-%20X.pdf

frequency communication channel is created for that specific *SaR* mission. In Portugal, this channel is used for exchanging information between the agents assigned to this particular mission and is usually defined by the *Comando Distrital de Operações de Socorro* (*Central Command of Rescue Operations*).

By considering these insights, one can outline the general procedures of a realistic *SaR* (robotic) operation as Figure 2.6 depicts.

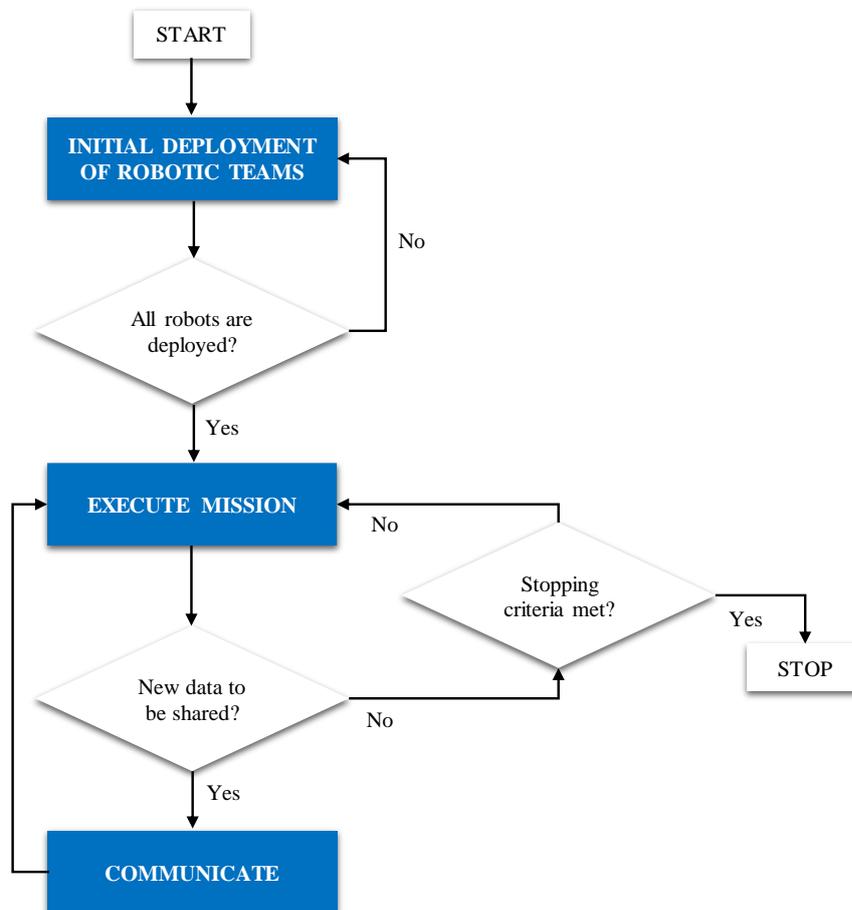


Figure 2.6. General flowchart of a *SaR* (robotic) operation.

This macroscopic scheme, although quite simplistic, highlights the three central activities one should expect from *SaR* robotic teams: *i*) the initial deployment of exploring robots in the scenario; *ii*) the autonomous mission execution (*e.g.*, coverage, flocking, foraging, among others); and *iii*) the information exchange about the mission progress and any points of interests (*e.g.*, victims' location, fire outbreaks, among others) with other robots or first responders.

Considering these key procedures highlighted in Figure 2.6, let us now survey how researchers have been working on each of those three central activities. The initial deployment problem, the mission execution and communication under faulty environments are research aspects usually studied

separately. Therefore, and to facilitate the context of this work, a disjointed analysis will be carried out. For organizational purposes, and following the priorities given by researchers to each one of the three topics, let us start by presenting the most pertinent *R&D* projects around *SaR* robotic missions.

2.2.2 Search and Rescue *R&D* Projects – The *CHOPIN* Case Study

Considering the benefits of *MRS* cooperative systems, in which explicit communication plays a central role, several *R&D* projects have been proposed over the past years, with the particular ambition to revolutionize *SaR* tasks. While several research works ultimately seek a high degree of robot autonomy in real world rescue missions, according to (Murphy, 2004) “*only a few [proposed robots] have actually participated in a rescue or been allowed to operate on site after a disaster for testing purposes; all of these have been teleoperated*”. This is still the case nowadays due to the exceeding *SaR* missions demands, as well as time constraints, and the lack of trust in full robot autonomy and their efficient performance, since the former can arise ethical and legal issues about the risk involved for human lives. However, to counteract this trend, recent research work has been conducted on remote interfacing and easy control of robots (Kim, et al., 2010). Taking this one step further, in Doroodgar *et al.* (Doroodgar, Ficocelli, Mobedi, & Nejat, 2010) a semi-autonomous robot architecture in rescue tasks was proposed. The robot is able to learn and make decisions whether autonomous or human control should be used to perform the tasks in a quicker and more efficient manner, considering the minimization of stress and burden placed on the operator. Nevertheless, in this Thesis, we intend to promote the complete autonomy of agents to assist human agents in the field and avoid, or at least minimize, any type of teleoperation.

An example is the work of Liao *et al.* (Liao, Hollinger, Djughash, & Singh, 2006) that presented a *MRS* for estimating firefighter positions in dangerous environments using only ranging sensors. A cooperative control algorithm that uses a decentralized cost map-based approach, decides how robots actively position themselves by reducing the uncertainty related to the position estimation of a moving firefighter. The authors use the *Geometric Dilution of Precision* as a performance metric and state their motivation as to offer guidance and avoid the risk for additional human lives through the deployment of a robot team alongside a firefighter. Despite not being tested on physical *MRS*, two-dimensional simulations present promising results.

In the *GUARDIANS EU FP6 R&D* project, a swarm of robots is assumed to be deployed in a large warehouse filled with smoke, toxic gases and inflammable materials to support a team of human firefighters (Gancet, et al., 2010). Aiming at a minimal additional mental or communication load to the agents involved, a minimalist light array visor solution, which is embedded in the firefighter’s helmet, has been proposed for human-swarm interactions. Based on the sensed data, the swarm of

robots recommends a direction to proceed and computes the firefighter's pose and direction, presenting this information through the light array. Furthermore, a formation algorithm for swarm of robots that follows human agents has also been developed in the context of this project (Marjovi, Marques, & Penders, 2009). Swarm agents react to human actions, while providing guidance information to firefighters. Another major aspect of this project was the development of a risk assessment system and environmental odour mapping based on topological *Simultaneous Localization and Mapping (SLAM)*. Cooperative localization based on *Ad hoc WiFi Received Signal Strength Indicator (RSSI)* was also discussed. In spite of the project being finished, there is no report of actual implementation results of such system or the full integration of all modules developed by each involved partner.

The *NIFTi EU FP7* project has been mainly focused on developing systems for human-robot teaming in urban search and rescue (*USAR*) scenarios (Kruijff, et al., 2012). Having a real-life tunnel accident as example case and communicating through a multi-modal team interface and spoken dialogue, the authors describe a complex socio-technical system to promote joint exploration of human-robot teams in accident sites. Working closely with rescue services to define requirements and evaluate the proposed system, this project follows a user-centric design approach, mimicking human understanding and operational procedure. Situational awareness plays a key role in the system, to join robot and human sensing and, similarly to other projects, investment in human-robot interaction is conducted to facilitate the process. Real-world results show that there is still a long way to go, although the authors highlight that the robot's autonomous behaviour must be transparent to promote collaboration. Recently, a promising approach to semantic mapping for interaction between a robot and three-dimensional landmarks in a *SaR* environment has been proposed based upon ontological knowledge (Keshavdas & Kruijff, 2012). The approach uses probabilistic inferences about three-dimensional structures, maintaining quantitative maps, in order to promote efficiency and transparency to human eyes. The authors have used *Support Vector Machines* to form models of high-visibility robots poses to inspect crashed cars. However, it has only been tested in a simulation environment, thus bypassing much of the uncertainty associated with real-world environments.

Making a broad contribution to apply swarm robotics into real-world applications, the *SMAVNET R&D* project was proposed (Hauert, Zufferey, & Floreano, 2009). This project aimed at using swarms of *unmanned aerial vehicles (UAVs)* in disaster areas to create and maintain a communication network for first responders. The motivation behind flying robots was their mobility factor and the benefit of providing line-of-sight communication. For that purpose, the authors developed specific low-cost outdoor platform (around 250€), as well as human-swarm protocols and interfaces to allow non-experts to easily and safely operate large groups of robots. The controllers were inspired by the be-

haviour of ants (Hauert, Winkler, Zufferey, & Floreano, 2008), making the analogy between the optimal deployment and maintenance of *pheromone* paths leading to food sources, with the deployment and maintenance of communication pathways between all agents (*i.e.*, first responders and robots).

More recently, the *CHOPIN R&D* project was proposed to exploit the human-robot symbiosis in the development of human rescuers' support systems for *USAR* missions. This *R&D* project funded by the *Portuguese Foundation for Science and Technology (FCT)* started on April 2012 by a team of which the author of this Thesis is a member, from the *Mobile Robotics Lab (MRL)* of the *Institute of Systems and Robotics* from *University of Coimbra (ISR-UC)*, Portugal, led by Professor Rui Rocha, supervisor of this Thesis.

One of the main catastrophic scenarios being used for proof of concept in the *CHOPIN* project is the occurrence of fire outbreaks in large basement garages. An example of such typical scenario was previously presented in Figure 2.5 (section 2.1.4) and later modelled for simulation purposes in Figure 3.9 (section 3.2.4). In this use case, the project focuses on utilizing a fleet of cooperative *unmanned ground vehicles (UGVs)* to cooperatively explore a basement garage where the fire is progressing, thus identifying the localization of fire outbreaks and victims. One of the first approaches to fulfilling the objectives of this project was to divide that kind of application into two operations wherein robotics could be more useful (see the previous section for the remaining operations): *i) reconnaissance*; and *ii) rescuing* (Couceiro, Portugal, & Rocha, 2013). In both phases, and as previously stated, this kind of scenario usually poses radio propagation difficulties to the response teams, whose members usually wear a radio emitter/transmitter to communicate by voice. Often under these noisy scenarios, the communication is only possible with teammates located in the line-of-sight. Therefore, and since a wireless communication computer network may be absent or damaged, robotic agents may have to deploy and maintain a mobile ad hoc wireless network (*MANET*) in order to support the interaction between the human team and the robotic team. In the *reconnaissance* phase, the mission consisted of a team of robots that arrived at the scenario via a common entry and spread out to explore and map the unknown area, signaling possible points of interest, such as victims and fire outbreaks. After a certain degree of confidence in the built representation of the scenario, the *rescuing* phase consisted of having the team of robots inspecting the area in a coordinated way visiting all points of interest, looking for remaining victims, possible changes in the scenario and the evolution of fire outbreaks.

In *CHOPIN*'s first scientific publication presented at the 28th *Symposium On Applied Computing*, the *reconnaissance* phase was handled using the *RDPSO* algorithm proposed in this Thesis (Couceiro, Portugal, & Rocha, 2013).

Given the considerations and benefits from swarm inspired approaches presented in section 2.1.4, and further observed in the strategies adopted by the *R&D* projects herein described, as the *CHOPIN* project, this Thesis will contribute towards a further growth of swarm robotics' applicability. As such, the *CHOPIN R&D* case study will be used throughout this work to evaluate certain features of the herein proposed approaches.

2.2.3 Mobile Ad hoc Networks (*MANETs*) in Faulty Environments

As one may conclude from the previous *R&D* projects, communication constitutes one of the most important utilities for more effective cooperation between the robots and improved robust performance. It has been consensual that the development of robot teams for *SaR* missions requires that robots have to be able to maintain the communication network among themselves without the aid of a pre-existent infrastructure. Besides that, robots also need to be able to guarantee the connectivity during the mission in order to explicitly exchange information within multi-hop network paths, so as not to unnecessarily restrict the team's range (Miller, 2001).

MANETs have attracted much attention in the past years, since they allow the coordination and cooperation between agents belonging to a given *MRS*. Since these networks can be set up at any place and time and do not require any pre-existing communication infrastructure, potential applications of such systems include military missions, unmanned space exploration and data collection. For instance, *MANETs* can be used to enable the next generation of battlefield applications envisioned by the military, including situation awareness systems for manoeuvring war fighters and remotely deployed unmanned robotic networks. However, such networks typically comprise of a large number of distributed nodes (*i.e.*, robots) that organize themselves into multi-hop wireless networks. Therefore, robots may cooperate and route messages to each other, *i.e.*, robots can perform the roles of both hosts and routers. Usually, a node corresponds to a robot with embedded processor, low-power radio, which is typically battery operated. In order for *MANETs* to be cost effective, the on-board processing and wireless communication capabilities and the battery power of each robot are highly limited. Consequently, the nodes are prone to both damages from their environment and inner failures such as battery demise. Also, in most *MRS* real-world applications, robots are deployed in either hostile or inaccessible environments, and it is impractical or infeasible to repair or replenish energy by replacing batteries. Moreover, since robots are mobile, the topology of the distributed networks is time varying and the strength of the connection can change rapidly in time or even disappear completely, thus increasing the challenges of *MANETs* design.

One of the earliest works studying the effects of communication on multi-agent systems was presented by Dudek *et al.* (Dudek, Jenkin, Milios, & Wilkes, 1995), where the effect of two-way,

one-way, and completely implicit communication and sensing in a leader-follower task was considered. Other works often assumed constant communication ranges and/or relied on line-of-sight maintenance for communication (Winfield, 2000) (Arkin & Diaz, 2002). Other examples include Pereira *et al.* (Pereira, Das, Kumar, & Campos, 2003) and Sweeney *et al.* (Sweeney, Grupen, & Shenoy, 2004), where decentralized controllers for concurrently moving toward goal destinations, while maintaining relative distance and line-of-sight constraints, were respectively presented. Anderson *et al.* discussed the formation of communication relays between any pair of robots using line-of-sight (Anderson, Simmons, & Goldberg, 2003). Although coordination strategies that rely on line-of-sight maintenance might significantly improve each agent's ability to communicate, it has been proved through simulation by Thibodeau *et al.* that line-of-sight maintenance strategy is often not necessary and may potentially be too restrictive (Thibodeau, Fagg, & Levine, 2004). In this Thesis, we will not limit the problem to line-of-sight communication as nowadays technology makes it possible to take advantage of multi-hop communication, wherein non-neighbour robots can communicate by mean of intermediary robots acting as relay nodes.

Hsieh *et al.* presented an experimental study of strategies for maintaining end-to-end communication links for tasks such as surveillance, reconnaissance, and target search and identification, where team connectivity is required for situational awareness (Hsieh, Kumar, Cowley, & Taylor, 2008). The authors show experimental results obtained using a multi-robot *testbed* in representative urban environments. Similarly, Michael *et al.* work proposed strategies to guarantee connectivity in a team of robots (Michael, Zavlanos, Kumar, & Pappas, 2009). The authors proposed a gradient-descent control law that preserves the system connectivity by ensuring that links to neighbours are maintained. The algorithm requires limited local information and communication between agents to determine the addition or deletion of network links through distributed consensus and market based auctions. A new graph structure called *Separated Distance Graph* was also proposed in the work of Brüggemann and Schulz (Brüggemann & Schulz, 2010). This strategy allows to specify a large class of constraints, like ensuring that each individual robot never departs from the group by more than a certain distance, thus maintaining a formation in which every robot can be seen by its neighbours while keeping up the wireless communication within a *MRS*. However, experimental results show that the algorithm is not optimal with respect to the number of robots required to solve a task. The same hard restriction to maintain the connectivity of *MANETs* will be adopted in this Thesis. However, we aim at using measures inherent to explicit communication networks (*e.g.*, *RSSI*) instead of passive action recognition such as computer vision techniques.

Behaviour-based strategies have also been considered to maintain *MANET* connectivity. For instance, the authors in (Arrichiello, Chiaverini, & Fossen, 2006) and, successively, in (Arrichiello, et

al., 2009) and (Arrichiello, Heidarrson, Chiaverini, & Sukhatme, 2010) presented the extension of the *Null-Space-based Behavioural* approach to control a group of marine vehicles to execute missions, such as formation control in the presence of water current, cooperative target visiting with communication constraints, and cooperative caging of floating objects. This is a promising approach since it would be possible to merge different behaviours and actions with different priorities in order to define the final motion directives of robots. However, the design choices concerning the organization of the behaviours in terms of priority represents a high level of complexity since these choices derive from practical considerations related to both the mission objective and the hardware or software characteristics of the *MRS*. The *RDPSO* herein proposed merges the main task (*e.g.*, finding a victim) with other subtasks (*e.g.*, avoiding obstacles or maintaining *MANET* connectivity), by considering adaptive weights based on contextual information.

Going a step further, a full solution to maintain the network connectivity was presented by Tardioli *et al.* (Tardioli, Mosteo, Riazuelo, Villarroel, & Montano, 2010). The work proposed a multi-robot cooperative motion control technique based on a *virtual spring-damper* model to prevent communication network splits, a task allocation algorithm that takes advantage of the network link information to ensure autonomous mission and a network layer capable of sustaining hard real-time traffic and changing topologies. The authors' approach was based on maintaining multi-hop routes between nodes of sufficient quality in order to avoid the network becoming disconnected. For that particular reason, a measure of the communication link quality was considered (Tardioli & Villarroel, 2007). Based on this measure, robot movements were constrained if necessary. This approach is more reliable than the traditional approaches where the restriction is based on the communication range such as Sheng *et al.* (Sheng, Yang, Tan, & Xi, 2006) and Mosteo *et al.* (Mosteo, Montano, & Lagoudakis, 2008). The robots' navigation in Tardioli *et al.* (Tardioli, Mosteo, Riazuelo, Villarroel, & Montano, 2010) was based on a spring-damper mechanism, clustering robots in a flexible formation, with one robot being the leader of the formation and other robots being the slaves. This kind of approach incorporates the management of the system dynamics in real situations, dealing with dynamic behaviour of robots. However, the use of spring-damper systems introduces constraints that traditional allocation methods do not face. Similar and simpler methodologies that also take into consideration the dynamic behaviour of robots could be applied such as elastic bands by Menezes *et al.* (Menezes, 1999) or fuzzy systems (Zhong & Zhou, 2012). In this Thesis, we benefit from adaptive attractive and repulsive forces to maintain the connectivity of the *MANET* by means of a fuzzy system.

Although previous works focus on the main characteristics of *MANETs* and maintaining their connectivity, they do not handle one of the most common issues: fault-tolerance. Fault-tolerance is the ability of a system to deliver a desired level of functionality in the presence of faults (Demirbas,

2004). In the context of communication networks, even if the condition of the hardware is appropriate for the task, the communication between nodes is affected by many real-world factors, such as signal strength, antenna angle, obstacles, weather conditions, and interference. In *MRS*, fault-tolerance has been commonly summarized by the word *biconnectivity*, meaning that each pair of nodes (*i.e.*, robots) in the network has at least two disjoint routes between them. Therefore, the failure at any single node does not partition the network. Despite the positive results provided by biconnected networks, most of the works introduce complementary strategies, such as attractive forces, as explored in this Thesis, redundancy, or power adaptation transmission to maintain *MANET* connectivity. Furthermore, in wireless sensor networks, the authors even generalized the biconnectivity feature to multi-connectivity, *aka*, k -connectivity, $k \in \mathbb{N}$, in which the communication is not disrupted even when up to $k - 1$ nodes fail (Cheriyān, Vempala, & Vetta, 2003) (Bahramgiri, Hajiaghayī, & Mirrokni, 2006). However, only a few works in *MRS*, such as Cornejo and Lynch (Cornejo & Lynch, 2010) and the one herein presented, addressed k -fault-tolerance, since most of them only considers its particular case of $k = 2$ (*i.e.*, biconnectivity).

For instance, Tuan *et al.* presented a solution for making robots aware of *MANET* connectivity by selecting a reference robot that would allow them to indirectly communicate with other robots from the same *MRS* (Tuan, Bouraqadi, Moraru, Stinckwich, & Doniec, 2009). In order to do so, robots needed to broadcast information with their neighbours to build their connectivity table. Moreover, authors extended their algorithm to improve fault-tolerance by using a biconnected *MANET*. This is an interesting solution but, since the *MANET* topology is highly dynamic in real situations, robots would need to constantly refresh their connectivity table since neighbours may be unreachable at a given moment or new ones may appear. Nevertheless, authors used connectivity tables to ensure *MANET* connectivity considering a safe-moving zone around robots in which neighbours need to move in order to allow direct communication. This was validated in obstacle-free simulated experiments using a number of robots from 5 to 10, starting near the same location and showing that, as the number of robots increases, the exploration time decreases, yet leading to a faster partitioning of the network. Another work from the same authors considered multi-robot exploration as distributed constraint satisfaction problems (Doniec, Bouraqadi, Defoort, Le, & Stinckwich, 2009). In this work, the authors benefit from the algorithm presented in their previous work (Tuan, Bouraqadi, Moraru, Stinckwich, & Doniec, 2009) to allow robots to construct their connectivity table, thus considering *MANET* connectivity as a constraint of the *MRS*. Therefore, the problem is then solved based on connectivity tables and current positions of robots in order to obtain their new positions. The approach was validated in a simulated environment (*NetLogo*) with changing obstacles features (*e.g.*, density

and size). Robots started at the same location of the environment with a distance between them inferior to the communication range. As expected, adding robots improves the speed of the exploration to a certain limit – when robots are too many, they interfere with each other and spend more time avoiding each other than exploring. Also, and as expected, a larger communication range improves the exploration since robots are less constrained in their movements and can cover a larger area of exploration. Authors claim that the initial positions of robots in the environment have a minor influence on the exploration (Doniec, Bouraqadi, Defoort, Le, & Stinckwich, 2009). However, they do not perform any experiments with different initial positions; they all start at the bottom right corner of the environment. Although many of those works served as a guide during the experimental evaluation of the herein proposed strategies, some restrictions were overcome and substantially taken into account, such as the relevance of obstacles' density in the environment and the initial position problem.

In Pei *et al.* (Pei, Mutka, & Xi, 2010), a *Connectivity and Bandwidth-Aware eXploration* was proposed taking into account bandwidth information, thus dividing the problem of *MANET* connectivity into: *i*) frontier node placement; *ii*) relay node placement with routing path selection; and *iii*) the match of each robot with its target position. Despite the addition of bandwidth-constrained relay node placement which little prior work considers, the authors present a heuristic solution to maintain the connectivity between robots while considering obstacles. Authors' strategy is evaluated in a simulated environment comparing its exploration efficiency and communication quality with the distributed *Sensor-based Random Graph* from Franchi *et al.* (Franchi, Freda, Oriolo, & Vendittelli, 2009) and centralized *Possible Moves Sampling* from Rooker and Birk (Rooker & Birk, 2007). For both metrics, their algorithm outperformed the other two, especially when increasing the number of robots. However, it would be expected that increasing the number of robots would somehow constrain robots movements or communications. More recently, Mi *et al.* (Mi, Yang, & Liu, 2011b) presented a distributed and hybrid connectivity restoration framework to restore the connectivity of the disconnected networks. The authors' algorithm was divided into two layers. The first one consisted of using a *Distributed Connectivity Restoration Algorithm* to constantly monitor neighbour status and select the best available agents to restore the connectivity of the network. The second layer consisted of using a potential function-based motion controller to drive the selected agents to certain destinations in order to restore the connectivity of the network, while keeping the network from further partitioning and inter-agent collisions. Therefore, in order to compute the *MANET* connectivity, a robot needs to receive information from all of its neighbours. Also, to avoid being disconnected from the network, each robot's movement is restricted to a specific range around its neighbour. The authors use repulsive and attractive forces to avoid collisions and maintain connectivity between robots. The framework was evaluated using a variety of simulations that considered connectivity restoration subject to

failure of a single agent and to multiple simultaneous agent failures. The algorithm was able to restore the connectivity of mobile networks subject to a single or multiple simultaneous agent failures. Similarly, the same authors, in a subsequent work, presented a hybrid coverage enhancement system to reconfigure an initially compact multi-agent system to achieve a certain level of coverage with guaranteed connectivity and collision avoidance between robots (Mi, Yang, & Liu, 2011a). The authors mainly focus on finding the redundant communication links necessary to reorganize the *MANET* topology. Once again, the evaluation is done by using simulated experiments in two and three-dimensional coverage tasks. Results showed that both communication and collision avoidance between robots are granted. However, besides considering an easier problem than multi-robot exploration in real applications (*i.e.*, coverage) as this Thesis addresses, the authors considered obstacle-free scenarios.

Sabattini *et al.* (Sabattini, Chopra, & Secchi, 2011b) proposed a decentralized control strategy to maintain *MANET* connectivity in *rendezvous* and formation control tasks. The main idea of the authors consisted of ensuring a lower bound on the estimate of the *second smallest eigenvalue* λ_2 , thus guaranteeing that the actual value of λ_2 is strictly greater than zero. The control strategy was evaluated using simulations varying the number of robots from 3 to 20 with randomly chosen initial positions. In both *rendezvous* and formation control tasks, robots were able to maintain the *MANET* connectivity. The work of Abichandani *et al.* presented a decentralized approach to generate velocity profiles for a group of robots while considering collision avoidance and communication connectivity between them (Abichandani, Benson, & Kam, 2011). Despite being a decentralized approach, a central communication station was considered to ensure that robots would communicate the trajectory data to all other robots in the group (*i.e.*, broadcast) and keep individual clocks synchronized. The authors used *Friis* equation (Friis, 1946) to obtain the received signal power, thus determining whether the robots are in communication range of each other. However, one could benefit from the *RSSI* signal since most wireless equipment already provides this feature and *Friis* equation may fail under unknown environmental variables (*e.g.*, obstacles). The authors also claimed that the decision order of robots, *i.e.*, the order in which each robot solves its planning problem, has a great influence over the algorithm performance. Therefore, the authors proposed to reassign different decision orders to improve the overall solutions. However, they do not present any reassigning strategy or when such reassignment should occur. Note that this level of organization and coordination between swarm robots is not addressed in this Thesis, as the main idea is to obtain such a collective performance with minimal decision-making from each individual robot.

In Derbakova *et al.* (Derbakova, Correll, & Rus, 2011), it is presented a set of several algorithms for repairing connectivity in mobile networks. The authors experimentally prove with simulated and real experiments that the proposed strategies were robust to individual and group node failure and

were able to restore the network connectivity. However, despite the decentralized nature of the algorithms, robots were only considered to be disconnected if they were unable to reach a designated gateway within a multi-hop network. In other words, all robots need to be aware of the gateway location in order to repair the network disconnections. Likewise, the work of Reddy and Veloso (Reddy & Veloso, 2011) also presented a similar problem in which a team of mobile robots needs to be able to maintain the connectivity between a static gateway and other mobile targets, denoting this task as *Target Tethering*. The authors presented an approach for inferring the physical layout of the network using *RSSI* data from multiple nodes. This is quite a complex task since it is difficult to find a simple function that models the relationship between *RSSI* values and physical distance, especially in indoor environment (Kotz, Newport, & Elliott, 2003). Moreover, they also classified the target motion patterns using the *RSSI* data with *Markov Decision Processes* and multi-agent reinforcement learning to fulfil target tethering. Despite the interesting results with both simulated and real platforms, this approach relies on the decisions of a central node (head node). The work of Casteigts *et al.* (Casteigts, Albert, Chaumette, Nayak, & Stojmenovic, 2010) proposed a self-deployment strategy considering biconnectivity, coverage, diameter and quantity of movements required to complete the deployment. To that end, the authors combined spring forces with angular forces, which strive to reduce the angles formed by pairs of angularly consecutive neighbours. Strategically, the authors also studied the best way to combine both forces considering their impact on each particular situation, thus adapting their weight depending on agent-based contextual information (*e.g.*, inter-robot distance). Simulation results showed that the authors' algorithm achieved biconnectivity in more than 90% of the cases for large populations of robots (between 100 and 200). In this Thesis, the assumption of a centralized node that all robots are aware of their location is not considered. All agents are considered to act in a distributed way.

Alternatively to the previously described works in which the main objective is to continuously ensure *MANET* connectivity, the work of Hollinger and Singh (Hollinger & Singh, 2010) presented the idea of periodic connectivity, where the network must regain connectivity at a fixed interval. The advantages inherent to this strategy can be observed in situations in which it may be desirable to break the connectivity, thus decreasing the number of robots and the information exchanged between robots of the same sub-network. Afterwards, when the network regains connectivity, it is possible to communicate the information gathered by the sub-network. The authors' algorithm outperformed other algorithms that require continual connectivity in simulated and real experiments. However, they did not consider unknown or dynamic environments. Despite dissimilarities with our work, a similar concept of network partitioning will be adopted in this work. However, contrarily to Hollinger and Singh (Hollinger & Singh, 2010) work, we will focus on an aperiodic connectivity since one cannot predict

when a robot (or subgroup of robots) will regain their connectivity due to algorithm's stochasticity. Nevertheless, the same advantages in reducing the number of robots within a sub-network remains in the work presented in this Thesis.

From the herein discussed studies, one may conclude that, although explicit communication significantly extends the applicability of *MRS* to real-world applications, and swarm robotics for that matter, it also adds up a significant constraint and complexity to the system. Moreover, one of the major concerns transversal across the previously described studies, is that all robots should have an initial deployment that favours the communication between robots during the mission. Henceforth, next section describes the initial deployment problem and the strategies presented so far in the literature.

2.2.4 Initial Deployment and its Influence on Task Completion

The initial deployment of mobile robots has not been fully addressed yet and only a few studies evaluating its relevance have been conducted. The deployment problem considers the number of needed robots for a specific situation (*e.g.*, objective, scenario, constraints) and their initial locations (Couceiro M. S., Figueiredo, Portugal, Rocha, & Ferreira, 2012). For instance, in a *SaR* mission, robots need to move in a catastrophic scenario in order to find survivors. When robots are transported to the catastrophe site, they need to be deployed. The deployment problem is to decide how many robots and where they will be initially located before performing the mission using their control strategy (*e.g.*, coverage, herding, formation and others). Despite the lack of works studying the initial deployment effect on the performance of the robotic team, a wrong decision about the number of robots and their initial location may greatly jeopardize the mission. For instance, one of the first works that addressed the effect of different initial deployments was presented by Correll *et al.* (Correll, Bachrach, Vickery, & Rus, 2009). The authors evaluated their coverage algorithm using both centralized and random initial deployments, thus concluding that the algorithm convergence was slower using a random initial deployment but tended to lead to better overall coverage for sparse topologies.

In fact, this is in line with most of the works presented in the literature, wherein a random initial deployment scattering the robots throughout the scenario is adopted (Kazadi, 2003) (Groß, O'Grady, Christensen, & Dorigo, 2011) (Xue & Zeng, 2008) (Kloetzer & Belta, 2006) (Lee, Nishimura, Tatara, & Chong, 2010). The work of Kazadi extended the sensory capability of plume tracking systems using swarms of robots deployed in the proximities of a common starting point (Kazadi, 2003). However, the author did not go to any lengths when exploring the plume tracking effectiveness for any other initial deployment strategies. Alternatively, the authors from Groß *et al.* presented a strategy to assign starting points and orientations of robots within circles of different radius around a prey (Groß,

O'Grady, Christensen, & Dorigo, 2011). Hence, using a team of 16 robots, the authors assign 16 different positions and 4 different orientations which are randomly assigned at each trial. Despite the apparent advantages of this deployment strategy in this context, no other strategies were evaluated, thus being hard to predict if the number of unsuccessful trials depicted in the work is somehow related to the initial deployment of robots. Xue and Zeng (Xue & Zeng, 2008) and, similarly, Kloetzer and Belta (Kloetzer & Belta, 2006), proposed a robotic swarm algorithm in which the initial positions and velocities of robots were randomly generated within an area limited to one corner near the origin of coordinates of the workspace. Despite the fact that both works considered a maximum allowed distance between robots, the proposed initial deployment strategy did not take into account any sort of communication constraints. More recently, in the work of Lee *et al.* (Lee, Nishimura, Tatara, & Chong, 2010), a three-dimensional deployment strategy was explored. As the previous works, the authors focus on a deployment strategy in which the initial distribution of all robots is arbitrary and their positions are distinct. The main difference with other works resides in the fact that robots autonomously move in a three-dimensional space instead of a planar scenario. Such strategy could be useful for coordinated formation flight and reconfiguration of UAVs (Hattenberger, Lacroix, & Alami, 2010). Therefore, the authors provide a coverage and connectivity strategy using a self-configuration process to enable robots to form a three dimensional tetrahedron shape.

Notwithstanding the positive results inherent to a random deployment, in real situations, it is necessary to ensure several constraints of the system. For instance, if the network supports multi-hop connectivity, this kind of constraints may significantly increase the complexity of the random distribution since it would depend not only on the communication constraints, but also on the number of robots and their position. Moreover, random deployment may cause an unbalanced deployment therefore increasing the number of necessary robots and energy depletion.

Recently, several more efficient deployment strategies taking into account a given contextual information have been addressed in other works such as (Schwager, Rus, & Slotine, 2010) and (Macwan, Nejat, & Benhabib, 2011). The authors in Schwager *et al.* (Schwager, Rus, & Slotine, 2010) focused on multi-robot deployment as an optimization problem. Although they did not address the initial deployment problem directly, the authors introduced a *mixing function* that describes how the information from different robots is combined, hence turning the convergence to a final configuration. As a prerequisite, the computation of the proposed deployment strategy requires a given robot to be aware of the states of all other robots in the network, thus requiring a fully connected network communication topology. Furthermore, the algorithm presented by Schwager *et al.* (Schwager, Rus, & Slotine, 2010) requires prior information about the environment such as the sensor function and the geometry of the environment. Alternatively, the work of Macwan *et al.* (Macwan, Nejat, &

Benhabib, 2011) proposed a method to carry out the optimal deployment of robots on wilderness *SaR* scenarios. To that end, the authors introduced the concept of *iso-probability* curves to represent the probabilistic information about the targets' location within the search area at any given time. Therefore, the number of robot subgroups, as well as the number of robots within each subgroup, was defined based on the number, shape and position of the *iso-probability* curves. The authors presented an illustrative example in which robots were able to find a given target after 4635 seconds. However, the performance of the algorithm greatly depends on several initial conditions, such as *a priori* knowledge about the target (*e.g.*, subject age, terrain type). Also, it is noteworthy that their proposed algorithm can only be computed if the last known position coordinates of a target are given. Nevertheless, in most situations, the last known position or other contextual information is not known and *a priori* assumptions need to be considered. For instance, in the case of the *World Trade Center* towers, one could confine the search to the basement since fire rescue personnel would have headed there (Suarez & Murphy, 2011). However, although this may work in some particular cases, it is a hard assumption that we do not consider in this Thesis, *i.e.*, the scenario is considered to be unknown, unstructured and without any associated *a priori* information.

Another work with many intersections to the one herein described was presented by Howard *et al.* (Howard, Mataric, & Sukhatme, 2002a). The exploring robots deploy themselves in the unknown environment in an incremental way and secure line-of-sight contact with teammates. Robots have the ultimate goal of mapping the environment while using teammates as landmarks. A greedy deployment algorithm is presented, which aims at maximizing the coverage area by exploring robots. The work has been tested using four *Pioneer 2DX* mobile robots equipped with *SICK* laser range finders (*LRFs*).

The previously described works must represent a large portion of research in which the initial deployment problem has been addressed. Nevertheless, none specifies how robots are deployed within a scenario – most of the works assume that robots are manually deployed or they simply “appear” in the desired initial locations. However, despite the usefulness and scientific accomplishment of all previous works, one needs to cross the reality gap inherent to *MRS* deployment, thus presenting complete solutions to robotic applications and applying the conducted researches on multiple physical robots. These are often called *marsupial systems* (Murphy, 2000).

The term *marsupial* comes, once again, from nature. According to the *Merriam-Webster* dictionary, a *marsupial* is “*an order of mammals comprising kangaroos, wombats, bandicoots, opossums, and related animals that do not develop a true placenta and that usually have a pouch on the abdomen of the female which covers the teats and serves to carry the young*”. This phenomenon in which a certain living being (*e.g.*, kangaroos) is able to carry tiny new-borns have fostered the development of heterogeneous multi-robot systems with such principle in mind.

Ferworn *et al.* defined a *marsupial* robotic operation as the “*delivery of robotic services through the explicit physical interaction of two or more robots employed cooperatively*” (Ferworn, et al., 2006). Exercises in training facilities were conducted with diverse teams of different *marsupial* robots in three specific scenarios. These robots were equipped with a variety of different sensors like cameras, audio devices, grippers, among others. The authors concluded that *marsupial* heterogeneous robots provided functionalities beyond what each robot could deliver individually. Janssen and Papanikolopoulos (Janssen & Papanikolopoulos, 2007) underlined the challenge implied in maintenance and mechanisms of *marsupial* robotics. Rather than performing real-world experiments, they alternatively presented an abstraction of the modelling using a *Player/Stage* simulation interface (Gerkey, Vaughan, & Howard, 2003), provided with a custom extension.

Depending on the requirements of the application at hand, many solutions for the deployment problem are possible. In their view, the design of *marsupial* robots should answer crucial questions like:

“*How many robots should the mechanism carry? What sensors does the mechanism have? And can it charge the batteries of the robots that it is carrying?*”

Exploiting the strengths of heterogeneous robots, an example of a more realistic approach was presented in Rybski *et al.* in which the authors divided the population of real robots into two different platforms: *rangers* and *scouts* (Rybski, et al., 2000). The rangers were built as large platforms used to transport the scouts over distances of several kilometres and deploy them rapidly over a large area. The scouts, on the other hand, were built as small and expendable robotic platforms used to sense the environment, act on their sensing, and report their findings. Despite the innovative work, the authors did not focus on the cooperation and communication between robots. Also, the deployment strategy was accomplished through a launcher system equipped on the ranger that was able to throw the scouts up to a range of 30 meters. However, in most cooperative applications in unknown and unstructured scenarios (*e.g.*, *SaR* missions) this would require robots to be able to measure the relative distance between themselves or to be equipped with global positioning systems (*GPS*) for an efficient processing of the exchanged information. In the work of Dellaert *et al.* (Dellaert, et al., 2002), a pioneering *marsupial* system of large wheeled robots deploying a team of small legged robots for victim-localization in *USAR* scenarios has been presented. This was motivated by the inability to reach remote locations using solely the main robot. Larger robots were equipped with odometry, a *Sick LRF*, an omnidirectional camera, wireless communication and were able to transport up to four legged

robots, which were endowed with a single camera and wireless communication as well. Beyond effective mechanisms for the deployment of smaller robots, the authors also underlined the absence of reliable wireless communications as a handicap of these systems in real-world applications but did not propose any strategy to overcome it.

Other works have focused on deploying robots in a unique and compact unloading location (Mei, Lu, Hu, & Lee, 2005) (Mei, Lu, Hu, & Lee, 2006) (Niccolini, Innocenti, & Pollini, 2010) (Howard, Mataric, & Sukhatme, 2002b) (Sahin, et al., 2008) (Bartolini, Calamoneri, Fusco, Massini, & Silvestri, 2008). In the works of Mei *et al.* (Mei, Lu, Hu, & Lee, 2005) (Mei, Lu, Hu, & Lee, 2006), the authors presented a coverage task in which the deployment problem involves determining the number and size of robot groups that need to be unloaded from a carrier, and the initial robots' location. A solution that can cover the deployment area within the maximum coverage time allowed is iteratively determined by varying the number and sizes of groups based on heuristics. In order to compute their algorithm, the authors assume that the density of obstacles is available. In addition, besides only considering a scanline deployment strategy, the authors assume having a unique unloading location for the whole team of robots. In other words, the carrier (*e.g.*, autonomous mobile robot) transports the robots into the field and the exploring robots need to autonomously move from the unloading location to their individual starting locations. The authors in Niccolini *et al.* (Niccolini, Innocenti, & Pollini, 2010) described an approach for the deployment of a swarm of heterogeneous autonomous vehicles based on descriptor functions. Similarly to what is presented in this Thesis and in Howard *et al.* work (Howard, Mataric, & Sukhatme, 2002b), each robot is treated as an agent of the network, in which repulsive forces are computed as a function of the distance between agents, to spread the network throughout the environment. Nevertheless, there is a certain similarity to other works such as Sahin *et al.* (Sahin, et al., 2008) and Bartolini *et al.* (Bartolini, Calamoneri, Fusco, Massini, & Silvestri, 2008), where the authors chose an initial deployment in which robots start from a compact configuration. Although this kind of initial deployment strategy works well when the main purpose is to spread the robots within area coverage scenarios, no other deployment strategy was taken into consideration by the authors. Also, despite being similar to the deployment of military units, thus simplifying the deployment strategy, it requires exploring robots to find energy-efficient paths to avoid jeopardizing the success of the mission.

To further extend the potential application of *marsupial* systems, *USAR human-marsupial* robot teams have been also explored by Murphy *et al.* (Murphy, Ausmus, Bugajska, & Ellis, 1999; Murphy, 2000). Similarly to kangaroo societies, the team members were divided in three roles: *Human*, *Dispensing Agent* (*aka, mother*) and *Passenger Agent* (*aka, daughter*). The *mother* robot provides not

only transportation to the *daughters*, but also power (*aka, food*) and help. The latter refers to communicating suggestions, warnings or to rescue the *daughters*, which are responsible for exploring remote locations and are equipped with a camera, a microphone, two headlights and a video transmitter to send images directly to the human. The role of the human rescuers is to supply decision making capabilities, remotely speak with victims and collect information about their state, number of victims, location and presence of hazards, like gas leaks. Heuristics were proposed for the deployment of micro-rovers.

2.3 Summary

This chapter started by motivating the reader while justifying the choices considered throughout this work. The state-of-the-art in which this work falls into was then exploited, thus mainly focusing on three activities that one should consider while applying *MRS* to *SaR* missions: *i*) the initial deployment problem; *ii*) the mission execution; and *iii*) inter-robot communication under faulty environments. Although the mission execution problem has been a central topic investigated in the literature, the initial deployment of robots and the inter-robot communication under faulty environment have not received the proper attention.

For that particular reason, and to summarize the content from the previous two sections, let us introduce Table 2.1. The solutions presented to overcome the initial deployment problem were divided into deploying the robots at a *common* starting point (or region), with *random* locations, considering *contextual* information and under a realistic approach based on *marsupial* systems. On the other hand, to maintain inter-robot communication, the solutions were divided into the use of *heuristics*, *behaviour-based* approaches, mathematical *modelling* of certain phenomena (*e.g.*, radio frequency propagation) and considering a *fault-tolerance* policy necessary for real-world applications. Note that although some of the works may fall within more than one class of each activity (*e.g.*, benefit from both behaviour-based strategies and heuristics to maintain inter-robot communication), we only consider the main contribution of each work.

The inner grey region in the middle of the table aggregates the main features wherein this work may be classified. In other words, the herein proposed initial deployment strategy relies on a marsupial system that combines a geometric-based approach with random properties with contextual-information (*e.g.*, quality of the network). Moreover, the fault-tolerant *MANET* connectivity is ensured by benefiting from the behaviour-based *RDPSO* algorithm with a systematic adaptive mechanism that considers a set of context-based evaluation metrics.

Table 2.1. Summary of works focusing on the initial deployment and inter-robot communication problems.

		INITIAL DEPLOYMENT				
		Common	Random	Contextual	Marsupial	Not considered
MAINTAIN COMMUNICATION	Heuristics	(Winfield, 2000) (Brüggemann & Schulz, 2010) (Hsieh, Kumar, Cowley, & Taylor, 2008)	(Pereira, Das, Kumar, & Campos, 2003) (Anderson, Simmons, & Goldberg, 2003)	(Michael, Zavlanos, Kumar, & Pappas, 2009) (Howard, Mataric, & Sukhatme, 2002a)		(Arkin & Diaz, 2002) (Sweeney, Grupen, & Shenoy, 2004) (Thibodeau, Fagg, & Levine, 2004)
	Behaviour-Based	(Niccolini, Innocenti, & Pollini, 2010)				(Arrichiello, Chiaverini, & Fossen, 2006) (Arrichiello, Heidarsson, Chiaverini, & Sukhatme, 2010)
	Modelling	(Tardioli, Mosteo, Riazuelo, Villarroel, & Montano, 2010)	(Lee, Nishimura, Tatara, & Chong, 2010)			
	Fault-Tolerance	(Doniec, Bouraqadi, Defoort, Le, & Stinckwich, 2009) (Mi, Yang, & Liu, 2011b) (Abichandani, Benson, & Kam, 2011)	(Derbakova, Correll, & Rus, 2011) (Pei, Mutka, & Xi, 2010)			(Tuan, Bouraqadi, Moraru, Stinckwich, & Doniec, 2009) (Sabattini, Chopra, & Secchi, 2011b) (Reddy & Veloso, 2011) (Casteigts, Albert, Chaumette, Nayak, & Stojmenovic, 2010)
	Not considered	(Mei, Lu, Hu, & Lee, 2006) (Bartolini, Calamoneri, Fusco, Massini, & Silvestri, 2008)	(Correll, Bachrach, Vickery, & Rus, 2009) (Kazadi, 2003) (Kloetzer & Belta, 2006)	(Schwager, Rus, & Slotine, 2010) (Macwan, Nejat, & Benhabib, 2011)	(Ferworn, et al., 2006) (Rybski, et al., 2000) (Dellaert, et al., 2002) (Murphy, Ausmus, Bugajska, & Ellis, 1999; Murphy, 2000)	

Despite all previously addressed accomplishments, either regarding exploration algorithms, techniques to maintain the *MANET* connectivity or strategies to carry out the initial deployment of swarm robots, there is still a long, and even multifaceted, way to go towards a complete swarm robotic solution for real-world *SaR*. Although this Thesis proposes a methodological way of achieving a first complete swarm solution, it does not do so without first introducing some preliminary mathematical concepts and presenting the hardware and software considered.

Preliminaries

ROBOTICS research, as many other technoscientific fields, requires a deep interdisciplinary knowledge from basic sciences and specialized disciplines, such as mathematics and electronics. This chapter introduces some preliminary concepts, options and rationales behind those, from both theoretical (or mathematical) (section 3.1) and technological (section 3.2) perspectives, to pave the way towards a better understanding of the contributions presented in the following chapters.

3.1 Theoretical

This Thesis benefits from a wide range of mathematical tools in which, on several occasions, it goes back and forth requiring the same definitions and theorems to outline new results. This section introduces all the necessary theoretical concepts that one may require to better understand the proposed approaches.

3.1.1 Fractional Calculus

Fractional calculus has attracted the attention of several researchers (Sabatier, Agrawal, & Machado, 2007; Ortigueira & Machado, 2003), being applied in various scientific fields, such as engineering, computational mathematics, fluid mechanics, among others. In brief, fractional calculus can be considered as a generalization of integer-order calculus, thus accomplishing what integer-order calculus cannot. As a natural extension of the integer (*i.e.*, classical) derivatives, fractional derivatives provide an excellent instrument for the description of memory and hereditary properties of real-world processes and systems (such as robotic systems). The concept of *Grünwald–Letnikov* fractional differential is presented by the following definition.

Definition 3.1. (Machado, et al., 2010) *Let Γ be the gamma function defined as:*

$$\Gamma(k) = (k - 1)! \tag{3.1}$$

The signal $D^\alpha[x(t)]$ given by

$$D^\alpha[x(t)] = \lim_{h \rightarrow 0} \left[\frac{1}{h^\alpha} \sum_{k=0}^{+\infty} \frac{(-1)^k \Gamma(\alpha+1) x(t-kh)}{\Gamma(k+1) \Gamma(\alpha-k+1)} \right], \quad (3.2)$$

is said to be the **Grünwald–Letnikov fractional derivative of order α** , $\alpha \in \mathbb{C}$, of the signal $x(t)$.

An important property retrieved after equation (3.2) is that the fractional-order derivative requires an infinite number of terms, while an integer-order derivative just implies a finite series. Therefore, integer derivatives are “local” operators while fractional derivatives have, implicitly, a “memory” of all past events. However, the influence of past events decreases over time.

The formulation in (3.2) inspires a discrete time *Grünwald–Letnikov* formulation presented by the following definition.

Definition 3.2. (Machado, et al., 2010) *The signal $D^\alpha[x[t]]$ given by*

$$D^\alpha[x[t]] = \frac{1}{T^\alpha} \sum_{k=0}^r \frac{(-1)^k \Gamma[\alpha+1] x[t-kT]}{\Gamma[k+1] \Gamma[\alpha-k+1]}, \quad (3.3)$$

where T is the sampling period and r is the truncation order. This is the **approximate discrete time Grünwald–Letnikov fractional difference of order α** , $\alpha \in \mathbb{C}$, of the discrete signal $x[t]$.

The series presented in (3.3) can be implemented by a rational fraction expansion which leads to a superior compromise in what concerns the number of terms versus the quality of the approximation. Nevertheless, since this study will consider fractional calculus’ applicability to improve the convergence of robots toward a given solution considering their dynamics, the simple series approximation is adopted (*cf.*, section 4.2).

That being said, it is possible to extend an integer discrete difference, *i.e.*, classical discrete difference, to a fractional-order one, using the following definition.

Definition 3.3. (Ostalczyk, 2009) *The classical integer “direct” discrete difference of signal $x[t]$ is defined as follows:*

$$\Delta^d x[t] = \begin{cases} x[t] & , d = 0 \\ x[t] - x[t-1] & , d = 1, \\ \Delta^{d-1} x[t] - \Delta^{d-1} x[t-1], & d > 1 \end{cases} \quad (3.4)$$

where $d \in \mathbb{N}_0$ is the order of the integer discrete difference. Hence, one can extend the integer-order $\Delta^d x[t]$ assuming that the fractional discrete difference satisfies the following inequalities:

$$d - 1 < \alpha < d. \quad (3.5)$$

The features inherent to fractional calculus make this mathematical tool well suited to describe many phenomena, such as irreversibility and chaos, because of its inherent memory property. In this line of thought, the dynamic phenomenon of a robot's trajectory configures a case where fractional calculus tools fit adequately.

3.1.2 Jury-Marden's Theorem

The *Jury-Marden's stability criterion* is a method that allows defining an attraction domain that represents the stability of a linear discrete time system by analysis of the coefficients of its characteristic polynomial (Barnett, 1983). In brief, if the system poles are located inside the unit circle centred at the origin, then the system fulfils Jury-Marden's theorem and it is considered to be stable. This can be formulated by the following theorem.

Theorem 3.1. (Barnett, 1983) *Consider the real polynomial:*

$$p(y) = a_0 y^n + a_1 y^{n-1} + \dots + a_{n-1} y + a_n, a_0 > 0.$$

Construct an array having initial rows:

$$\begin{aligned} \{c_{11}, c_{12}, \dots, c_{1,n+1}\} &= \{a_0, a_1, \dots, a_n\}, \\ \{d_{11}, d_{12}, \dots, d_{1,n+1}\} &= \{a_n, a_{n-1}, \dots, a_0\}, \end{aligned}$$

and subsequent rows defined by:

$$\begin{aligned} c_{\beta\gamma} &= \begin{vmatrix} c_{\beta-1,1} & c_{\beta-1,\gamma+1} \\ d_{\beta-1,1} & d_{\beta-1,\gamma+1} \end{vmatrix}, \beta = 2, \dots, n+1, \\ d_{\beta\gamma} &= c_{\beta, n-\gamma-\beta+3}. \end{aligned}$$

All roots of the polynomial $p(y)$ have modulus less than one if and only if $d_{21} > 0, d_{\tau 1} < 0$ ($\tau = 3, 4, \dots, n+1$).

This work will benefit from such stability theory tool to overcome the parameterized complexity of the proposed *RDPSO* (cf., section 7.1).

3.1.3 Fuzzy Logic

Fuzzy logic was introduced in 1965 by Zadeh at the *University of California, Berkeley*, to deal with and represent uncertainties (Zadeh, 1965). Despite the several possible approaches to implement an online auto-tuning system, fuzzy logic seems to be the more adequate to proceed as a multiple criteria analysis tool. The strength of fuzzy logic is that uncertainty can be included into the decision process. Vagueness and imprecision associated with qualitative data can be represented in a logic way using linguistic variables and overlapping membership functions in the uncertain range. Traditionally, a fuzzy system is a static nonlinear mapping between its inputs and outputs (*i.e.*, it is not a dynamic system). The term “*fuzzy*” refers to the method’s ability of dealing with imprecise or vague information. Although the input information to the system may be imprecise, the results of fuzzy analysis are not. The field of fuzzy logic has a solid foundation of research that allows for meaningful application of its principles. Fuzzy logic allows to accurately describe a control system in linguistic terms in order to define the relationship between the input information and the output action instead. This linguistic definition is different from other fields of analysis that would use complex mathematical equations.

Fuzzy sets and fuzzy logic are used to quantify the meaning of linguistic variables, values and rules that are specified accordingly within the scope of the study. This is the translation from the rules of language to the rules of mathematics. Fuzzy sets need membership functions which are mathematical equations that can take certain shapes. Reasonable functions are often linear functions, such as triangular or trapezoidal functions because of their simplicity and efficiency when considering computational issues. Depending on the application and user, many different membership functions may be used.

In order to process the input to get the output reasoning there are six steps involved in the creation of a rule based fuzzy system:

1. Identify the inputs and their ranges and name them;
2. Identify the outputs and their ranges and name them;
3. Create the degree of fuzzy membership function for each input and output;
4. Construct the rule base that the system will operate under;
5. Decide how the action will be executed by assigning strengths to the rules;
6. Combine the rules and defuzzify the output.

The inputs and outputs are “*crisp*”, *i.e.*, they are real numbers, not fuzzy sets. Fuzzy sets are used to quantify the information in the rule-base, and the inference mechanism operates on fuzzy sets. Hence, it must be specified how the fuzzy system will convert its numeric inputs into fuzzy sets so that they can be used by the fuzzy system, this process is called *fuzzification*. This process involves the use of certain defined rules. To specify rules for the rule-base, a linguistic description is used. Hence, linguistic expressions are needed for the inputs and outputs and the characteristics of the inputs and outputs. For instance, temperatures are not always given in °C or °F, but in linguistic terms like cold, warm or hot. Since linguistic values are not precise representations of the underlying quantities they are describing, linguistic rules are not precise either. They are simply abstract ideas about how to achieve good control that could mean different things to different people. However, they are at a level of abstraction that humans are often comfortable with in terms of specifying how to control a certain process.

The problem in applying this is that the appropriate fuzzy operator may not be known. For this reason, fuzzy logic usually uses ***IF-THEN-ELSE*** rules. Rules are usually expressed in the form:

IF *variable* **IS** *property* **THEN** *action_1* **ELSE** *action_2*

The decision-making-logic determines how the fuzzy logic operations are performed and together with the knowledge base determine the outputs of each fuzzy ***IF-THEN-ELSE*** rules. Afterwards, many fuzzy implication methods that can be chosen. *Mamdani-Minimum* inference method is one of the most often used to represent the ***AND*** connective. With this method, in order to perform *fuzzification*, the minimum value is selected among the available variables. After the fuzzy reasoning, there is a linguistic output variable which needs to be translated into a crisp value. This process is called *defuzzification*. This is equivalent to translating the output from the fuzzy domain back into the crisp domain. Some *defuzzification* methods tend to produce an integral output considering all the elements of the resulting fuzzy set with the corresponding weights, *e.g.*, *Centre-of-Gravity (COG)*. Other methods take into account just the elements corresponding to the maximum points of the resulting membership functions (*e.g.*, *Centre-of-Maximum*) (Shaw, 1998).

This simple decision-making reasoning makes *fuzzified* systems as one of the most commonly used in the literature, from classical control problems (Couceiro, Ferreira, & Machado, 2012) and all the way to more unusual problems such as decision-making systems to prevent zombie outbreaks (Couceiro M. S., Figueiredo, Luz, & Delorme, 2014 (In Press)). In this Thesis, fuzzy logic will be used to adapt the behaviour of swarm robots based on the contextual information retrieved by them (*cf.*, section 7.3).

3.1.4 Semi-Markov Chain

Semi-Markov processes were firstly proposed by Levy in 1954 (Levy, 1954). In semi-Markov chains, as in Markov chains, processes are described by a set of states whose transitions are governed by a transition probability matrix P . Nevertheless, in semi-Markov chains, the time between transitions may be random. Consequently, the amount of time spent in any state is random and may be described by a probability density that can be a function of both the state of occupancy and the states to which transitions can occur.

The *Markovian* property is formally introduced in Definition 3.4 (Tijms, 2003).

Definition 3.4. (Tijms, 2003) *The future probabilistic behaviour of the process depends only on the present state of the process and is not influenced by its past history. This is called the Markovian property.*

By following this property, one can describe the semi-Markov decision process through the following definition (Tijms, 2003).

Definition 3.5. (Tijms, 2003) *A controlled dynamic system is called a semi-Markov decision process when the following property is satisfied: if at a decision epoch the action \mathbf{a} is chosen in state \mathbf{I} , then the time until the next decision epoch and the state at that epoch, depends only on the present state \mathbf{I} and the subsequently chosen action \mathbf{a} and are, thus, independent of the past history of the system.*

State-to-state transitions within a semi-Markov chain, as any other probabilistic model representation, is described as a probability density. Whereas the purpose in this Thesis is to model robot swarms at the macroscopic level, such transition will depend on the interaction probabilities of a robot with other teammates, with the environment and the mission itself (*cf.*, chapter 9). Such transitions will be time-variant. Hence, to avoid rounding approximations, the model was parameterized considering a discretization interval of $\Delta t = 1$ s. Any other discretization interval could be considered as long as one would take it into account during the definition of the transition probabilities.

That being said, from constant rates over the time step, it is easy to calculate the corresponding transition probabilities for a small time interval $T_j > \Delta t$, $T_j \in \mathbb{N}$, given the corresponding encountering rate for a state j , γ_j , (Agassounon, Martinoli, & Easton, 2004):

$$p_j = \int_t^{t+T_j} \gamma_j dt = \gamma_j T_j, \quad (3.6)$$

One of the most interesting features of *Markovian* models is the information one can obtain by analysing the system at equilibrium, *i.e.*, the long-term behaviour usually known as steady-state behaviour. Such steady-state analysis makes it possible to retrieve several performance parameters such as throughput, delay, loss probability, among others. It is, however, noteworthy that semi-Markov systems at steady-state do not settle down to one particular state. The *Markovian* steady-state regime means that the probability of being in any state will not change.

One of the most common approaches to find the steady-state behaviour of the semi-Markov chain with more than 2 states relies on the *Z*-transform (Gebali, 2008). Hence, we hereafter establish some useful properties of the *Z*-transform that will be needed in this Thesis.

3.1.5 Z-Transform

The *Z*-transform method is one of the most, if not the most, suitable for solving and analysing linear discrete equations (*DEs*) and discrete systems (Elaydi, 2005). Let us present some basic definitions.

Definition 3.6. (Elaydi, 2005) *Let $x[n]$ be a sequence, which is identically zero for negative integers n (*i.e.*, $x[n] = 0$ for $n = -1, -2, \dots$). The *Z*-transform of a signal $x[n]$ is defined as*

$$X(z) = Z(x[n]) = \sum_{j=0}^{\infty} x[j]z^{-j}, \quad (3.7)$$

wherein z is a complex number.

The set of numbers z in the complex plane for which series (3.7) converges is called the region of convergence of $x(z)$. This brings us to the following definition.

Definition 3.7. (Elaydi, 2005) *Let $R = \lim_{j \rightarrow \infty} \left| \frac{x[j+1]}{x[j]} \right|$. The number R is called the radius of convergence of series (3.7). Hence, the series (3) converges in the region $|z| > R$ and diverges for $|z| < R$.*

Remark 3.1. (Elaydi, 2005) *If $R = 0$, the *Z*-transform $X(z)$ converges everywhere with the possible exception of the origin. On the other hand, if $R = \infty$, the *Z*-transform diverges everywhere.*

Definition 3.6 and Definition 3.7 allow presenting the following two necessary properties.

Theorem 3.2. (Linearity property) *Let $X(z)$ be the Z-transform of $x[n]$ with radius of convergence R_1 , and let $Y(z)$ be the Z-transform of $y[n]$ with radius of convergence R_2 . Then, for any complex numbers α, β we have*

$$Z[\alpha x[n] + \beta y[n]] = \alpha X(z) + \beta Y(z), \text{ for } |z| > \max(R_1, R_2). \quad (3.8)$$

Theorem 3.3. *Let R be the radius of convergence of $X(z)$. If $x(-i) = 0$ for $i = 1, 2, \dots, k$, then:*

(i) *Right-shifting:*

$$Z(x[n - k]) = z^{-k} X(z), \text{ for } |z| > R.$$

(ii) *Left-shifting:*

$$Z(x[n + k]) = z^k X(z) - \sum_{j=0}^{k-1} x[j]z^{k-j}, \text{ for } |z| > R.$$

After obtaining the Z-transform of a given system, one can analyze its steady-state, *i.e.*, equilibrium, by employing the *final value theorem*:

Theorem 3.4. *If $x[t]$ is a sequence with Z-transform $X(z)$ then the “final value” of $x[t]$ is given by*

$$\lim_{t \rightarrow \infty} x[t] = \lim_{z \rightarrow 1} (z - 1)X(z).$$

After having introduced the necessary theoretical background, let us now present the technology used throughout this work.

3.2 Technological

Following the same lines as before, this section devotes itself on presenting some technological tools, such as the robotic platforms and simulators used throughout this work.

3.2.1 Scouts (Swarm Robots)

Although the term “*scout*” will be used now and then, in the absence of it while mentioning “robots” one should consider those as scouts. As this work focus on swarm robotics, scouts are merely a necessary term to separate them from their supportive *marsupial* platforms denoted as “*rangers*” that do not fall within the description of swarm robots.

The following requirements, sorted by relevance, can be expected from robots (or scouts) to be used in swarm robotic systems:

- Cost – scouts should be as cheap as possible since most swarm teams may have dozens or hundreds of robots (Arvin, Samsudin, & Ramli, 2009);
- Energy Autonomy – scouts should have a long battery life since the swarm may need to operate long enough for the collective behaviour to emerge and the goal to be reached (Parker, Schneider, & Schultz, 2005);
- Communication – scouts have to support wireless communication such as in the form of ad hoc networks (Parker, Schneider, & Schultz, 2005);
- Sensory System – scouts should be equipped with some form of sensing capability to allow interaction among robots as well as with their environment (Dorigo, et al., 2004);
- Processing – scouts need to be able to process information about other robots and the environment (*e.g.*, sensing data) (Parker, Schneider, & Schultz, 2005).

Furthermore, there is an evident similarity between swarm robots and the so called minimalism-based user created robots (Park & Kim, 2011). The concept of minimizing the cost (minimal cost) is one of the essential properties of swarm robots and was already stated. However, both concepts of minimizing the number of components (minimal intricacy) and the required development time to complete the entire process (minimal development) can be considered basic features in the development of swarm robots. Since the cost is directly related with the robot design, a reduced intricacy in sensing, control and motion is required. In addition, since swarm robotics deals with many robots, the development time of each unit must be reduced, thus allowing a fast extension of the swarm team.

Following that line of thought, as swarm robots, scouts should be small, easily deployable and able to sense the environment. There were two platforms playing this role throughout this Thesis: the *Educational Swarm Robot eSwarBots* developed at MRL from ISR-UC and *RoboCorp* at the *Engineering Institute of Coimbra* from the *Polytechnic Institute of Coimbra (ISEC-IPC)* in the context of this work (Couceiro M. S., Figueiredo, Luz, Ferreira, & Rocha, 2011), and the well-known *e-puck* developed at the *École Polytechnique Fédérale de Lausanne* (Mondada, et al., 2009).

The *eSwarBot* is a differential robot with a diameter of 126 mm , a height of 100 mm and a weight of approximately 600 g , being small enough to avoid an inadequate increase of the size of test arena, and yet big enough not to limit the expandability of the robot nor to increase the cost of the swarm robots due to components miniaturization. These robots are ideal for studying emergent behaviour and self-organization in bio-inspired societies (*i.e.*, swarm robotics).

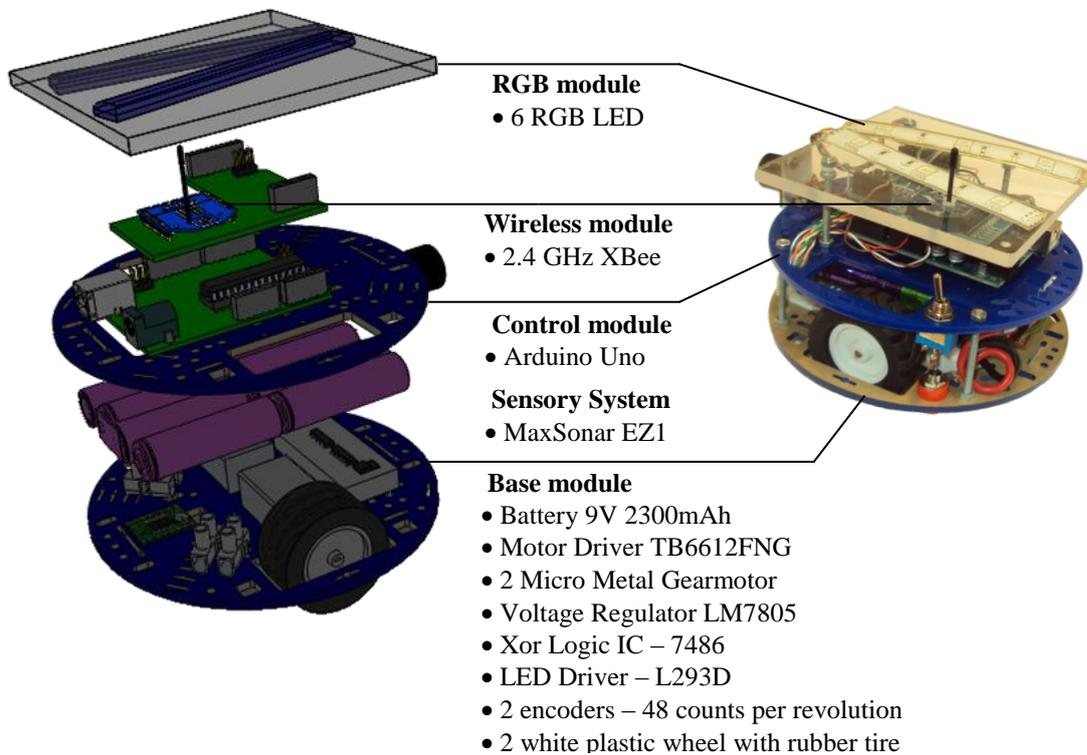


Figure 3.1. The *eSwarBot* robot.

The *eSwarBot* platform (Figure 3.1) is a differential ground platform with an *Arduino Uno* processing unit. Although the platform presents a limited odometric resolution of 3.6 degrees while rotating and 2.76 mm when moving forward, its low cost (around 175 €) and high energetic autonomy (average run time around 3 hours) allows performing experiments with a large number of robots.

As it can be seen in Figure 3.1, *eSwarBots* are equipped with *RGB-LEDs* that allow representing a wide range of different colours, which account for different subgroups (or smaller swarms). Some relevant specifications of the *eSwarBot* are presented in Table 3.2.

Compared to other low-cost solutions in the market, the feature that clearly excel on the *eSwarBots* platforms is its capability of creating *MANETs* with multi-hop capabilities using *ZigBee* technology (this will be described with some more detail in section 3.2.3).

Table 3.2. Main *eSwarBot* features.

Features	Description
Size	∅ 126 mm
Cost	≈ 175 €
Battery autonomy	≈ 3 hours
Wireless Communication	<i>ZigBee</i> Networks
Wired Communication	<i>USB</i>
Sensing capabilities	1 ultrasound sensor 1 <i>LDR</i> light sensor
Actuation capabilities	2 48 <i>steps/rev</i> DC motors with encoders Strip of RGB <i>LEDs</i>
Expansion Capabilities	All <i>Arduino</i> Shields: Ethernet / Internet access <i>WiFi</i> capability <i>GPS</i> Camera ...

The radar graphs from Figure 3.2 were designed by quantifying and normalizing the features of several known platform in terms of: 1) Cost; 2) Energy Autonomy; 3) Communication; 4) Sensory System; and 5) Processing (Couceiro M. S., Figueiredo, Luz, Ferreira, & Rocha, 2011). The cost and energy autonomy were easily quantified based on the price of the platform and the number of working hours until it runs out of power, respectively. The communication was evaluated based on the wireless technology used taking into account that *ZigBee* is the one recommended by the authors for swarm robotics. The sensory system was quantified based on the number and heterogeneity of sensors and encoders' resolution. Finally, the processing was evaluated based on the processing power of the control unit.

The other platform used for swarm exploration purposes during this work was the *e-puck*. The *e-puck* depicted on Figure 3.3 was officially presented for the first time in a scientific paper on 2009 (Mondada, et al., 2009) with a project that started with the *École Polytechnique Fédérale de Lausanne* as collaboration between the *Autonomous Systems Lab*, the *Swarm-Intelligent Systems group* and the *Laboratory of Intelligent System*. Since then, many other works have been benefiting from the *e-puck* platform due to its price/features qualities (Table 3.3), with especial attention given to its sensing capabilities (*cf.*, Figure 3.2b).

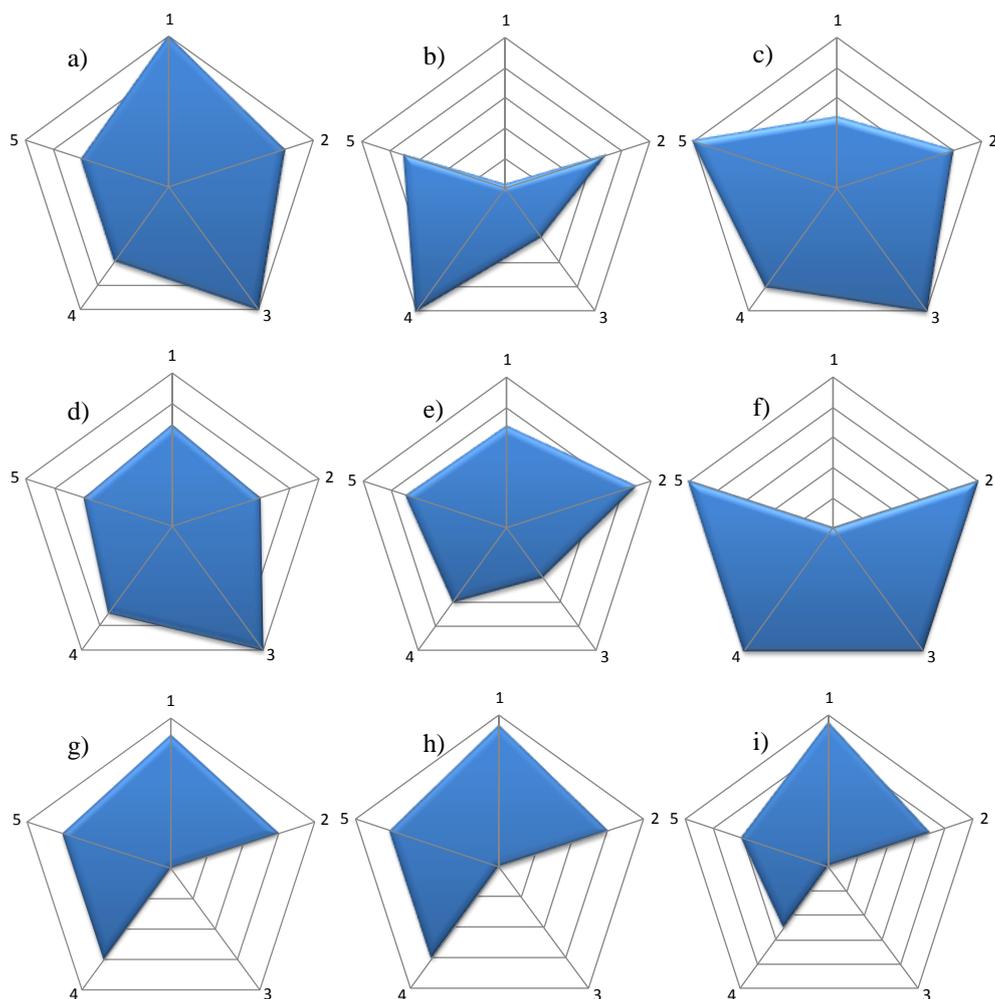


Figure 3.2. Evaluation of several platforms for swarm robotics using radar graphs. a) *eSwarBot* (Couceiro M. S., Figueiredo, Luz, Ferreira, & Rocha, 2011); b) *e-puck* (Mondada, et al., 2009); c) *SRV-1 Blackfin Surveyor* (Cummins, Azhar, & Sklar, 2008); d) *TraxBot* (Araújo, Portugal, Couceiro, Figueiredo, & Rocha, 2012); e) *Mindstorms NXT Lego* (Bagnall, 2007); f) *marXbot* (Bonani, et al., 2010); g) *Hemisson*¹²; h) *IdMind Circular GT*¹³; i) *Bot'n'Roll ONE*¹⁴.

¹² http://ftp.k-team.com/hemisson/Manuel_En/Manual_Hemisson.pdf

¹³ <http://www.idmind.pt/en/education/images/Circular%20GTIng.pdf>

¹⁴ http://botnroll.com/product.php?id_product=67

Figure 3.3. The *e-puck* robot.

As it is possible to see on Table 3.3, the *e-puck* provides a high level of applicability with a large number of sensors and acting capabilities (Mondada, et al., 2009). Moreover, its low-cost and small desktop size, together with the open source feature, makes the *e-puck* as one of the most widely used robotic platforms in laboratory context. Although it still presents some limitations, those may be overcome by adding turrets (extension boards) for an extra cost that may endow the robots with higher processing power or other features (*Linux Gumstix Overo turret* or *Fly-Vision omnidirectional vision turret*). How *e-pucks* communicates between themselves is briefly described in section 3.2.3.

Table 3.3. Main *e-puck* features.

Features	Description
Size	$\varnothing 70\text{ mm}$
Cost	$\approx 230\text{ €}$
Battery autonomy	$\approx 3\text{ hours}$
Wireless Communication	<i>Bluetooth</i> (UART1)
Wired Communication	<i>RS232</i> (UART2)
Sensing capabilities	8 infrared proximity sensors 1 3D accelerometer 1 640×480 <i>CMOS</i> colour camera 3 microphones
Actuation capabilities	2 1000 steps/rev stepper motors 1 Speaker 8 red light emitting diodes (<i>LED</i>)
Expansion Capabilities	Omnidirectional vision Ground sensors Colour <i>LED</i> communication <i>ZigBee</i> communication <i>Linux</i> extension ...

3.2.2 Rangers (*Marsupial Robots*)

The term “*ranger*”, or *marsupial* robot, was introduced in this work to easily distinguish swarm robotic platforms for exploration (the so-called scouts, or simply, robots) and the support platforms for deployment. Rangers act as supporting platforms that need to carry the team rapidly into place and deploy the scouts. They must be extremely robust and be able to transport multiple scout platforms and process the sensor data, acting as coordinators of the team during the deployment phase. Therefore, *TraxBot* platforms (Figure 3.4) were adopted as rangers, being suitable for both outdoor and indoor operation with high autonomy and low cost (around 470 €). These platforms have also been recently developed at *MRL* from *ISR-UC* (Araújo, Portugal, Couceiro, Figueiredo, & Rocha, 2012; Araújo, 2012).

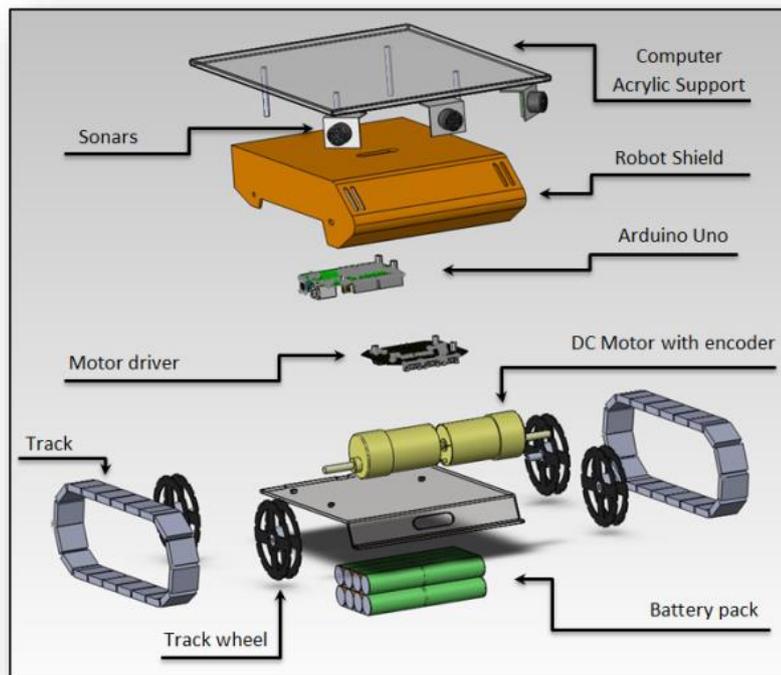


Figure 3.4. The *TraxBot* robot platform.

The *TraxBot* platform is a differential drive system built upon the *Traxster II Robot educational Kit*, equipped with 2 *DC* gearhead motors with quadrature wheel encoders and rubber tracks. The *TraxBot* can reactively avoid obstacles with a maximum range of approximately 6 meters using three *Maxbotix* Sonars *MB1300* mounted below the top acrylic support, as seen in Figure 3.4. Some other specifications are presented in Table 3.4.

Giving the robustness of the *TraxBot* (*i.e.*, aluminum and stainless steel chassis with high power *DC* motors), an extension conveyor kit has been built to support 5 *eSwarBots* on top of the platform

(Figure 3.5), thus forming a marsupial system. The conveyor's structure was built in order to promote proper adherence of the scouts' wheels and a stepper motor is connected to the *TraxBot* to allow convenient sliding and deployment of the scouts. A major part of the conveyor was built with equipment from damaged printers (*e.g.*, conveyor pulleys, stepper motors, gearboxes). It is noteworthy that, in real *SaR* applications, *i.e.*, real unknown and uneven terrain, a cleated conveyor belt (*i.e.*, partitioned upper layer) could be used to improve the capability of the ranger to keep the scouts on top of it.

Table 3.4. Main *TraxBot* features.

Features	Description
Size	203 × 229 mm
Cost	≈ 470 €
Battery autonomy	≈ 2 hours
Wireless Communication	<i>ZigBee</i> Networks
Wired Communication	<i>USB</i>
Sensing capabilities	3 ultrasound sensors
Actuation capabilities	2 624 steps/rev DC motors with encoders
Expansion Capabilities	All <i>Arduino</i> Shields Netbook support on top

The *TraxBot Conveyor Kit* illustrated on Figure 3.5, even being entirely made of lightweight aluminium, it increases the original weight of the *TraxBot* platform to 4.2 Kg (*i.e.*, unladed weight) and 7.1 Kg at full load (*i.e.*, with 5 *eSwarBots* on top), being able to support up to a maximum weight of 4.5 Kg without suffering from any sliding effect on the driving pulley. Nevertheless, this is more than enough as the *TraxBot* mobile platform is unable to efficiently rotate when carrying a weight of approximately 5.0 Kg above the unladed weight.

The single feature that rangers should have in common with scouts is their communication technology and protocol, so as to endow robotic team with explicit communication networks under the same medium.

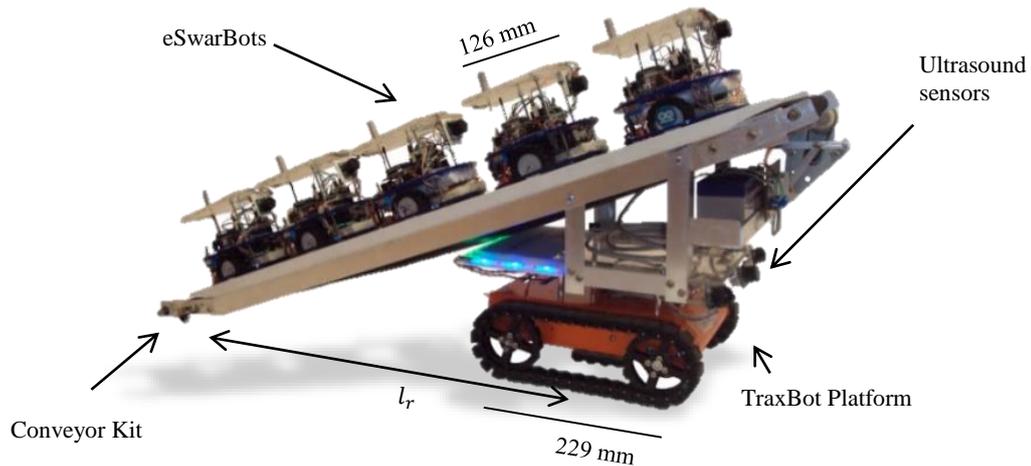


Figure 3.5. The *TraxBot Conveyor Kit* loaded with 5 *eSwarBots*.

3.2.3 Implementation of *MANETs*

MANETs can be implemented using several wireless technologies, such as *Bluetooth*, *ZigBee*, or *WiFi*. The definition itself does not imply any restrictions to the implementing devices. In this work, inter-robot communication to define the initial position of *eSwarBots* as scouts was carried out using *ZigBee* 802.15.4 wireless protocol¹⁵, while communication between *e-pucks* was carried out using *WiFi* 802.11 from *Linux Gumstix Overo turret*.

Regarding *eSwarBots* and *TraxBots*, they were all endowed with *XBee* modules¹⁶ that communicate with the microcontroller via *Serial Peripheral Interface*. *XBee Series 2* is based on *ZigBee/802.15.4* silicon from *Freescale*®. Its 802.15.4 firmware feature set makes it ideal for point-to-point, peer-to-peer and point-to-multipoint (star or mesh) topologies. Hence, these modules are a suitable solution for swarm robotics since they present power consumption near $10 \mu A$ when in sleep mode and $50 mA$ while sending and receiving data. Furthermore, since swarms may be formed by more than dozens of robots (*i.e.*, nodes), the *ZigBee* protocol is the most adequate option since it can theoretically support up to 65536 network nodes. Although the *XBee Series 2* modules allow a maximum communication range of approximately 30 meters in indoor/urban environments (Couceiro M. S., Figueiredo, Luz, Ferreira, & Rocha, 2011), the signal quality of the received data is highly susceptible to obstacles and other phenomena (*e.g.*, communication reflection and refraction), thus resulting in the loss of packets as the inter-robot distance increases. In fact, preliminary experiments to test the *XBee* modules on a large indoor scenario endowed with obstacles showed that the connectivity

¹⁵ <http://www.zigbee.org>

¹⁶ http://ftp1.digi.com/support/documentation/90000982_A.pdf

starts failing above 10 meters (*cf.*, Figure 3.7). Therefore, to allow a more realistic and conservative approach, the connectivity between robots was maintained using the received signal quality. To that end, the *XBee* modules were modified in order to provide the *RSSI* signal output (*cf.*, Figure 3.6).

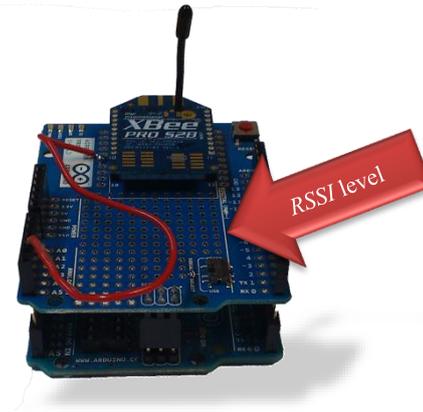


Figure 3.6. Electrical modification of *XBee Series 2* from *Digi International*¹⁷ to provide the *RSSI* signal output.

This *RSSI* output is available as a pulse width modulation signal of 120 *Hz* where the duty cycle *DC* varies accordingly to the signal level relative to the receiver sensitivity as it follows:

$$PWM_{\%} = \frac{295 + (17.5 \times dBm_above_sensitivity)}{10.24}, \quad (3.9)$$

For instance, a 30% duty cycle (*i.e.*, 1.5 *V*) is equivalent to approximately the receiver sensitivity of -94 *dBm*. In order to choose a minimum signal threshold that would ensure the *MANET* connectivity, Figure 3.7 presents the relation between the *RSSI* and the distance between two robots randomly wandering in a large indoor scenario endowed with obstacles while sending 30 periodic messages every 2 seconds to each other at each different distance. The *RSSI* vs the inter-robot distance was represented using a boxplot chart, in which the ends of the blue thicker lines and the circle in between correspond to the first and third quartiles and the median values, respectively. The numbers on top of each set of measures correspond to the number of messages received at each different distance.

¹⁷ http://alumni.ipt.pt/~lrafael/manual_XBee_Series2_OEM_RF-Modules_ZigBee.pdf

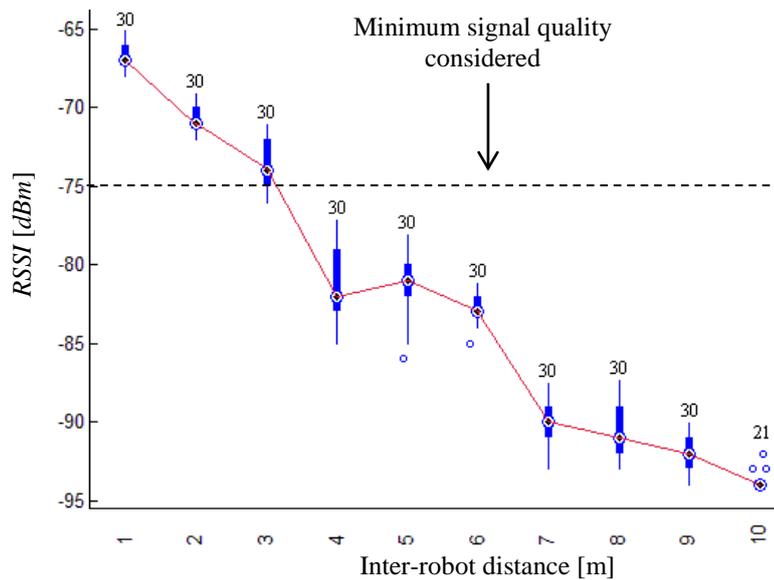


Figure 3.7. Measured RSSI versus distance from two robots located in the experimental scenario.

As expected, in an indoor scenario endowed with obstacles, the signal quality is not proportional to the inter-robot distance. In fact, the inverse relationship between distance and signal quality considered in many works does not match reality since the propagation model is more complex, *i.e.*, the signal depends not only on the distance but also on the multiple paths from walls and other obstacles. Moreover, for a distance above 10 meters, a robot is only able to receive approximately $2/3^{\text{rd}}$ of the messages. Therefore, to avoid the possible loss of packets due to the distance between robots, the minimum allowed receiver power was set to -75 dBm , *i.e.*, for distances of approximately 3 meters. This allows avoiding the possible loss of packets due to low levels of signal quality and, at the same time, carry out experiments in limited size scenarios.

Regarding the inter-robot communication using the *e-pucks*, and due to the limitations of the standard *e-puck* for swarm applications that require explicit inter-robot communication, it was necessary to accomplish some engineering work around the platforms. This engineering process was divided into two steps that took place at the Robotics Lab from the *School of Mathematical and Computer Sciences at Heriot-Watt University (MACS-HWU)* under the supervision of Professor Patricia Vargas:

- Bridging the gap between *Webots* simulator (Michel, 2004) and the real *e-puck* platforms by improving the already existing features and adding new ones (first item from Technical Reports section at the end of this document¹⁸);

¹⁸ Technical Report 1 available at http://www2.isr.uc.pt/~micalcouceiro/media/ReportFinal_1.pdf.

- Endowing the *e-puck* platforms with *WiFi* communication (second item from Technical Reports section¹⁹).

The first part of the work mainly consisted on using the *Webots* not only as a simulator but also as a developer tool for real robot experimentation. Programming *e-pucks* by using *Webots* has been consensual in most research using *e-puck* robotic platforms. Either more directed for education purposes as the work of (Guyot, Heiniger, Michel, & Rohrer, 2011) or all the way to swarm applications as (Cianci, Raemy, Pugh, & Martinoli, 2006), it has been consensual that the combination between *Webots* and *e-pucks* is appropriated to the further development of robotic applications. One of the main reasons regarding such consensus resides in the compatibility between both simulation and real-world experiments. Nevertheless, such compatibility is still an on-going work and, therefore, a deeper contribution from the community is required. This work gave its contribution by changing the whole compilation architecture of *Webots* for a faster cross-compiling development. Moreover, three important functionalities that were erstwhile inaccessible were also included: access to the microphones, the speaker and, most importantly, *Bluetooth* inter-robot communication. Microphones can extend *e-puck*'s sensing capabilities on both real world and simulations toward optimization problems, *e.g.*, find audio sources emulating victims in a search-and-rescue (*SaR*) scenarios. Combining the speaker with the microphones may allow to create a communication network with the ability to detect the direction of teammates or promote human-robot interaction. At last, and as most *MRS* require it, the *Bluetooth* radio link allows inter-robot explicit communication necessary to share information and bring forth the necessary cooperation to fulfil a given task (see the first item at Technical Reports section). However, even despite these accomplishments, *e-puck*'s *Bluetooth* interface was not able to completely satisfy the requirements of this work. As the *Bluetooth* interface was never designed for inter-robot communication, it presented some drawbacks, such as the high required time of around 19.8 ± 4.5 seconds to establish a *serial port profile* link between pairs of robots, thus being unsuitable for experiments that require a high rate of shared information between different robots. Also, due to the limitations of *e-puck*'s *Bluetooth* module, it was impossible to create a wireless network for *MANET* creation. In other words, the use of *e-puck* platforms on *MRS* was still confined to systems with sporadic explicit communication between robots.

¹⁹ Technical Report 2 available at http://www2.isr.uc.pt/~michaelcouceiro/media/ReportFinal_2.pdf.

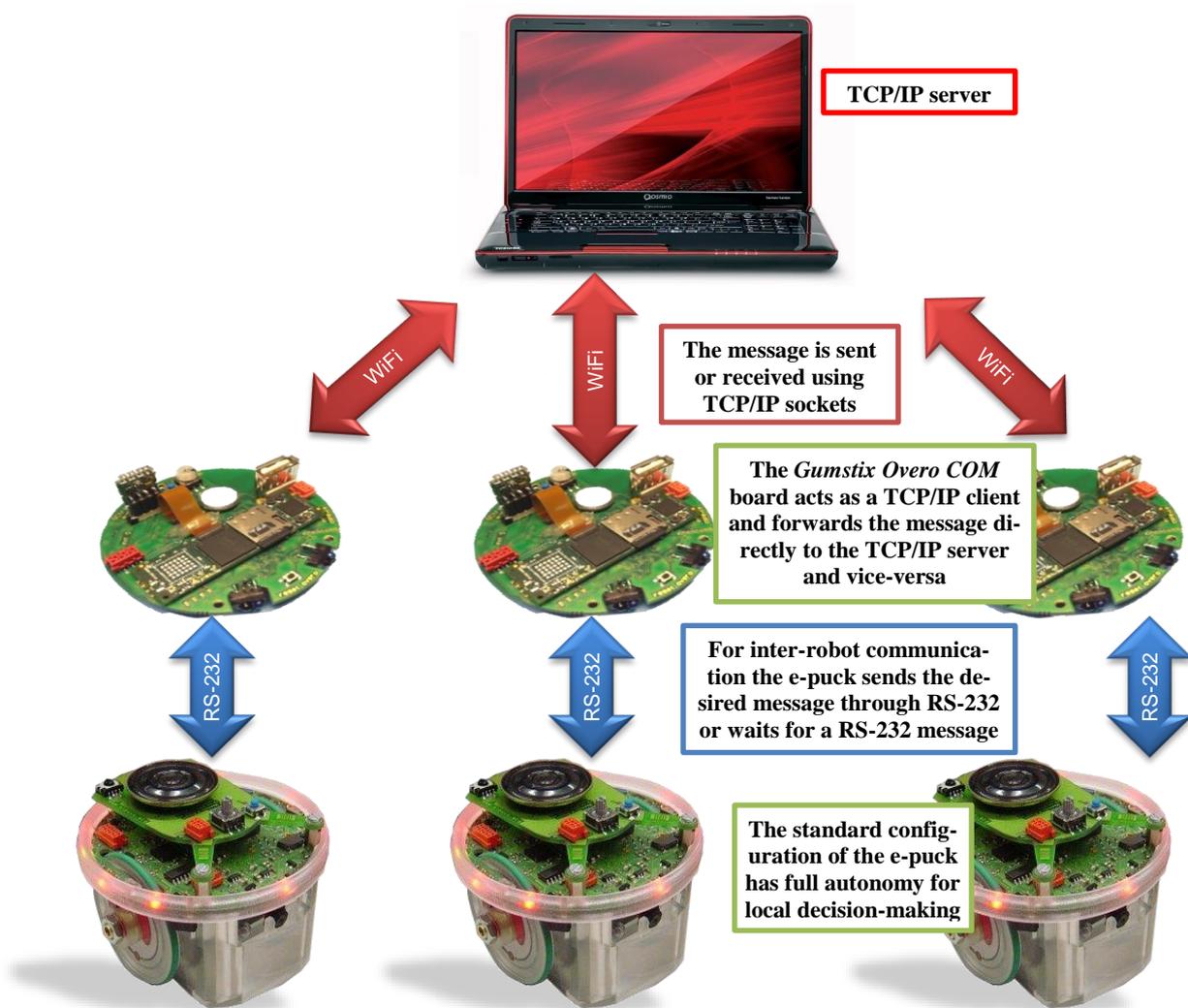


Figure 3.8. System architecture of the *e-puck* robot equipped with *Gumstix Overo COM*.

As a result, these limitations were only overcome by equipping the *e-puck* with the *Gumstix Overo COM* turret²⁰. This turret endows the *e-puck* with higher processing capabilities under a *Linux* embedded environment and *WiFi* communication. Despite the advantages that this turret brings, the community have not paid the necessary attention so far. Therefore, part of the developed work aimed to bring some more insights about the *turret* and explain how inter-robot communication may be achieved using *WiFi* technology. Moreover, this was accomplished by sticking with *Webots* architecture in such a way that one may fulfil the same *MRS* experiments on both virtual and real world scenarios using *WiFi* communication (see the second item on Technical Reports section). This extension allowed to benefit from *TCP/IP* sockets for inter-robot communication (Figure 3.8). To that end, a global *TCP/IP* server to serve all *TCP/IP* socket clients (*i.e.*, *e-pucks*) concurrently was developed

²⁰ http://www.gctronic.com/doc/index.php/Overo_Extension

in *MatLab*. This server was created to manage all *TCP/IP* socket connections and forward the messages to the correct *e-puck*(s). Moreover, it was also created with the purpose on emulating constraints from wireless networks such as the maximum distance between one-hop robots and the maximum number of hops in the network. Although many other properties of phenomena could be emulated on the server side, only those two were contemplated so far. Nevertheless, one can easily extend the software presented in the second item from the Technical Reports section with other features, such as radio propagation modelling, delays or even abrupt losses of communication. Nevertheless, due to scalability purposes and budget, robotic simulators are still required to evaluate larger teams of robots.

3.2.4 *MRSim*

Simulators play an important role in the development of new robotic platforms, algorithms and architectures (Staranowicz & Mariottini, 2011). However, the usefulness of a simulation environment grows with the consistency between simulation and real world results. In robotics, such consistency needs to be handled by the adequate implementation of several real-world features such as sensor noise, robot dimensions and actuators nonlinearities. Nevertheless, when one goes from single robot applications to cooperative *MRS*, the level of complexity significantly grows. Besides the previously described phenomena, one needs to consider the several mechanisms that may foster such cooperation as, naturally, the communication between robots (Parker L. E., 2008b). This gap between reality and simulation, although mitigated over the years in laboratorial context or for single robot applications, as it is the case of the well-known *Unified System for Automation and Robot Simulation (USARSim)* (Carpin, Lewis, Wang, Balakirsky, & Scrapper, 2007), is still far from contemplating all real-world features inherent to *SaR* applications and, at the same time, scale with the number of robots.

The *Multi-Robot Simulator (MRSim)*²¹ was initially created to evaluate the simulation experiments from this Ph.D. Thesis in 2012. Since then, it has been successively improved considering several real-world phenomena such as radio frequency (*RF*) propagation. *MRSim* is an evolution of the *Autonomous mobile robotics toolbox SIMROBOT (SIMulated ROBOTs)* previously developed for an obsolete version of *MatLab*²². The simulator was completely remodelled for the newer *MatLab* version and new features were included, such as mapping and inter-robot communication. Besides that, *MRSim* also allows adding a monochromatic *bitmap* as a planar scenario changing its properties (*e.g.*, obstacles, size, among others), adding features of each swarm robotic technique (*e.g.*, robotic

²¹<http://www.mathworks.com/matlabcentral/fileexchange/38409-mrsim-multi-robot-simulator-v1-0>

²²http://www.uamt.feec.vutbr.cz/robotics/simulations/amrt/simrobot_en.html

population, maximum communication range, among others) and edit robots' model (*e.g.*, maximum velocity, type of sensors, among others). This simulator was first evaluated in the context of the *CHOPIN* project, to compare decentralized and centralized versions of both *RDPSO* for exploration purposes (Couceiro, Portugal, & Rocha, 2013).

Figure 3.9 depicts the *MRSim* interface with a simulation trial with robots using the *RDPSO* algorithm to collectively explore the whole scenario of a large basement garage environment – the *ISR-UC* garage with an area of $A = 2975 \text{ m}^2$. Note that this area may not correspond to the area of the real garage. This scenario will be one of the case studies considered throughout this Thesis.

To fulfil all the requirements of the *CHOPIN* project, the simulator was further enhanced with real-world features typical from *SaR* missions in urban fires, namely:

- Fire spreading (Ohgai, Gohnai, Ikaruga, Murakami, & Watanabe, 2005);
- Victims' behaviour (Heliövaara, 2007), firefighter behaviour (Abbasi, Hossain, Hamra, & Owen, 2010);
- Radio frequency communication (Luca, Mazzenga, Monti, & Vari, 2006);
- Voice propagation (Herman, 2007).

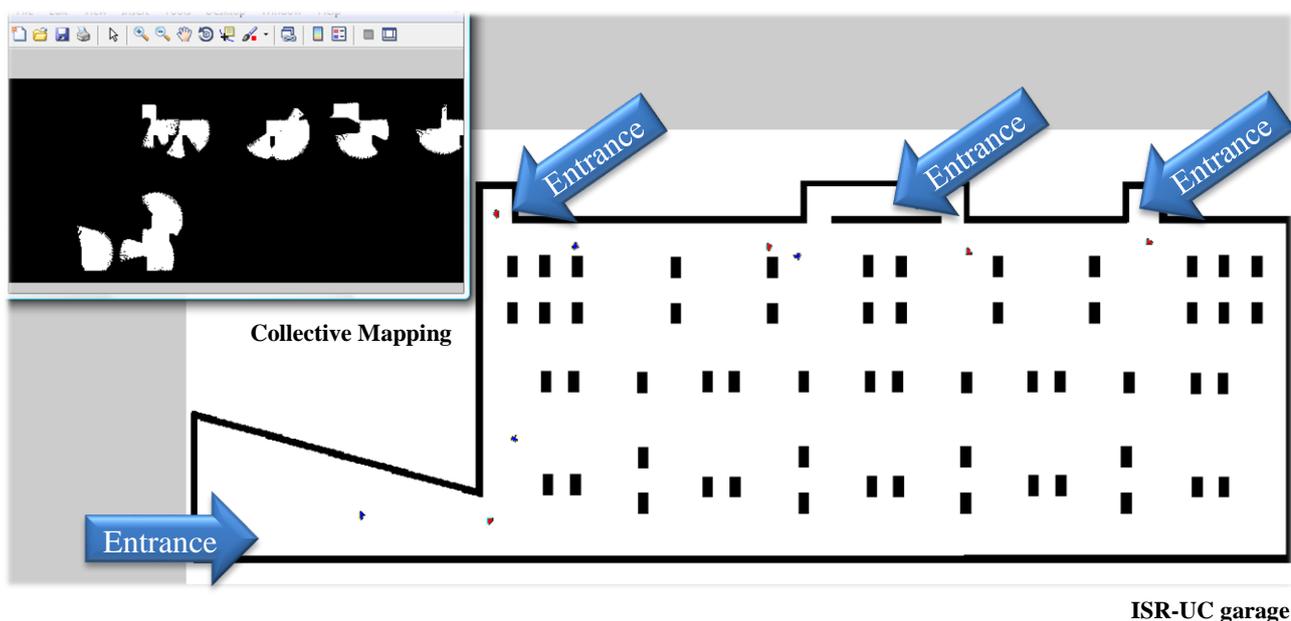


Figure 3.9. Multi-Robot Simulator (*MRSim*). Illustration of one trial with 10 robots performing the collective mapping of an unknown scenario under the influence of the *RDPSO* algorithm (Couceiro, Portugal, & Rocha, 2013).

The third item from the Technical Reports section presents some more insights around *MRSim*²³.

²³ Technical Report 3 available at <http://www2.isr.uc.pt/~michaelcouceiro/media/help/helpMRSim.htm>.

3.2.5 *Webots*

The correspondence between simulation and reality has been highly discussed in the literature and several simulation environments featuring a certain level of portability²⁴ have been proposed such as the *Virtual Experimentation Platform Coppelia Robotics (V-REP)* (Freese, Singh, Ozaki, & Matsuhira, 2010), the well-known *Willow Garage Robotic Operating System (ROS)* (Quigley, et al., 2009) and, one of the oldest, the *Player/Stage* (Gerkey, Vaughan, & Howard, 2003). Despite these recent accomplishments, the environment that still presents a higher compatibility between real and virtual *e-pucks* is still *Webots* from *Cyberbotics*. This is mainly due to the close relation between *e-puck* and the education field that is the main target of *Cyberbotics*. Hence, this is frequently seen as a win-win relationship to train beginners to use *Webots* and consequently increase *e-puck*'s employment (and so the *e-puck* community). In fact, the number of papers using both *Webots* and *e-puck* real robots greatly exceed any other *simulator+platform combo*. For instance, Rohrer's report present a full curriculum describing the link between the *e-puck* and *Webots*, going from a totally beginner audience to expert skilled programmers (Rohrer, 2008).

Developing a cross-compilation system allows to compile the *Webots* controller for the embedded processor of the real robot. Hence, the source code for the *Webots* virtual robot is executed on the real robot itself, and there is no need to have a permanent connection between the robot and the remote control system. In brief, the idea is to generate a machine code file from the same program as the one in simulation, typically written in *C/C++* languages, and to upload it on the *e-puck* in a transparent fashion for users. *Webots* then includes the source code, and eventually low level assembly files, of robots Application Programming Interface (*API*) being compatible for *e-puck* and the *Hemisson* robots. Therefore, in the case of *e-puck* robots, the cross-compilation system requires the source code necessary to compile the *e-puck* firmware. This requires rewriting many of the *Webots* include files to be specific to *e-puck* platform. In other words, for the purposes of this Ph.D. work, some source files were created to be used as a replacement for the *Webots API*, but running on the real *e-puck*. The first item from the Technical Reports section presents some more insights on the work around *Webots* simulator.

In this work, *Webots* simulator will mainly be used to program the *e-puck* platforms and to calibrate all experiments with them through simulations.

²⁴ In the context of robotic simulation, portability means that the code written for virtual agents is portable to real platforms (Staranowicz & Mariottini, 2011).

3.3 Summary

This chapter introduced the necessary preliminaries by giving a brief overview on the several mathematical tools, robotic platforms and simulator environments used throughout this work. The reader is advised to resort from this chapter whenever the concepts introduced later on require a specific mathematical tool or hardware feature.

As previously stated, this work is devoted on presenting a full swarm robotic solution for real-world applications such as *SaR*. As a starting point, next chapter introduces the core of this Thesis: the *Robotic Darwinian Particle Swarm Optimization (RDPSO)*.

Robotic Darwinian *PSO*

FOLLOWING the many examples of behaviour-based biological collective architectures presented in the literature, researchers have been proposing ever-improving designs of novel swarm robotic algorithms. This area of research, belonging to swarm intelligence, studies large collections of relatively simple agents that can collectively solve problems that are far too complex for a single agent, or that can display the robustness and adaptability to environmental variation displayed by biological agents (Bonabeau, Dorigo, & Theraulaz, 1999) (Beni, 2004). Starting with optimization problems and all the way to robotic applications, those tools have been proved to be robust to many of the drawbacks inherent to more complex and exhaustive methods, namely the factor time previously discussed (*cf.*, section 2.1.2).

This chapter presents the core of this Thesis by methodically describing the *Robotic Darwinian Particle Swarm Optimization (RDPSO)*. It is, however, noteworthy that, although all subsequent chapters revolves around the *RDPSO* algorithm herein proposed, their methods, tools and insights can, and should, be applied to other swarm robotic algorithms.

This chapter is divided into eight main contributions:

- i)* A short survey around Particle Swarm Optimization (*PSO*)-based algorithms is presented, focusing on the description of the Darwinian *PSO* proposed by (Tillett, Rao, Sahin, Rao, & Brockport, 2005), which may be considered as the foundation (or optimization method counterpart) of the *RDPSO* algorithm (section 4.1);
- ii)* It is then introduced the first step towards a realistic robotic implementation of the *RDPSO*, by benefiting from fractional calculus concepts to consider robots' dynamical properties (section 4.2);
- iii)* Afterwards, the existence of obstacles in robots' sensing range is considered by proposing a new objective function (section 4.3);
- iv)* As a real-world requirement, and a central part of this Thesis, communication constraints are also considered by resorting to attractive and repulsive forces so as to ensure the *MANET* connectivity (section 4.4);
- v)* As a final base property of the *RDPSO*, a “punish-reward” mechanism to emulate natural selection, so as to avoid stagnation, is proposed by getting inspiration on the concepts of social exclusion and inclusion (section 4.5);

- vi) To encapsulate all the previous concepts, the *RDPSO* algorithm is formally presented and illustrative examples are provided (section 4.6);
- vii) Finally, based on the previously described properties, the proposed *RDPSO* algorithm is evaluated using both numerical and physical experiments (section 4.7).

Sections 4.8 and 4.9 outline the discussion and main conclusions of this chapter.

4.1 From Optimization to Robotics

The navigation of groups of robots, especially swarm robots, has been one of the fields that has benefited from biological inspiration (Bonabeau, Dorigo, & Theraulaz, 1999). As many other things in robotics, the advances were first introduced and evaluated in the context of computer agents. One of the first applicability of those methods started with optimization tools with the well-known *Particle Swarm Optimization (PSO)* previously mentioned. The original *PSO* was developed by Kennedy and Eberhart in 1995 and it is based on social and computer sciences (Kennedy & Eberhart, 1995). The *PSO* basically takes advantages on the swarm intelligence concept, which is the property of a system whereby the collective behaviours of unsophisticated agents that are interacting locally with their environment, create coherent global functional patterns (Valle, Venayagamoorthy, Mohagheghi, Hernandez, & Harley, 2008). Imagine a flock of birds wherein each bird cries at an intensity proportional to the amount of food that it finds at its current location. At the same time, each bird can perceive the position of neighbouring birds and can tell which of the neighbouring birds emits the loudest cry. There is a good chance that the flock will find a spot with the highest concentration of food if each bird simply follows a trajectory that combines three directions (Floreano & Mattiussi, 2008):

- i) Keep flying in the same direction;
- ii) Return to the location where it found the highest concentration of food so far;
- iii) Move towards the neighbouring bird that cries the loudest.

In the traditional *PSO*, candidate solutions (*e.g.*, birds) are called particles. These particles travel through the search space to find an optimal solution, by interacting and sharing information with neighbour particles, namely their individual best solution (local best) and computing the neighbourhood best. Also, in each step of the procedure, the global best solution obtained in the entire swarm is updated. Using all of this information, particles realize the locations of the search space where success was obtained, and are guided by these successes. In each step of the algorithm (Algorithm 4.1), a fitness function is used to evaluate the particle success. To model the swarm, each particle n moves in a multidimensional space according to position (x_t^n) and velocity (v_t^n) values which are highly dependent on local best (\tilde{x}_t^n), neighborhood best (\tilde{n}_t^n) and global best (\tilde{g}_t^n) information:

$$v_{t+1}^n = wv_t^n + \rho_1 r_1 (\tilde{g}_t^n - x_t^n) + \rho_2 r_2 (\tilde{x}_t^n - x_t^n) + \rho_3 r_3 (\tilde{n}_t^n - x_t^n), \quad (4.1)$$

$$x_{t+1}^n = x_t^n + v_{t+1}^n. \quad (4.2)$$

Coefficients w , ρ_1 , ρ_2 and ρ_3 assign weights to the inertial influence, the global best, the local best and the neighbourhood best when determining the new velocity, respectively. Typically, the inertial influence is set to a value slightly less than 1. ρ_1 , ρ_2 and ρ_3 are constant integer values, which represent “*cognitive*” and “*social*” components. However, different results can be obtained by assigning different influences for each component. For example, several works do not consider the neighbourhood best and ρ_3 is set to zero. The parameters r_1 , r_2 and r_3 are random vectors with each component generally a uniform random number between 0 and 1. The intent is to multiply a new random component per velocity dimension, rather than multiplying the same component with each particle’s velocity dimension. Depending on the application and the characteristics of the problem, tuning all these parameters properly may lead to better results. However, this is a problem associated with the branch of *parameterized complexity* and far from being completely solved (Downey & Fellows, 1999).

In the beginning, particles’ velocities are set to zero and their position is randomly set within the boundaries of the search space (Algorithm 4.1). The local, neighbourhood and global bests are initialized with the worst possible values, taking into account the nature of the problem. For instance, in a cost problem where the objective is to minimize the fitness function, particles are initialized with a large value (tending to infinity). There are other few parameters that need to be adjusted:

- i) Population size – very important to optimize to get overall good solutions in acceptable time;
- ii) Stopping criteria – it can be a predefined number of iterations without getting better results, or other criteria depending on the problem.

Initialize swarm (Initialize x_t^n , v_t^n , \tilde{x}_t^n , \tilde{n}_t^n and \tilde{g}_t^n)

Loop:

```

for all particles
  Evaluate the fitness of each particle
  Update  $\tilde{x}_t^n$ ,  $\tilde{n}_t^n$  and  $\tilde{g}_t^n$ 
  Update  $v_{t+1}^n$  and  $x_{t+1}^n$ 
end

```

until stopping criteria (convergence)

Algorithm 4.1. Traditional *PSO* Algorithm

The *PSO* reveals an effect of implicit communication between particles (similar to broadcasting) by updating neighbourhood and global information, which affects the velocity and consequent position of particles. Also, there is a stochastic exploration effect due to the introduction of the random multipliers (r_1 , r_2 and r_3). Given its simplicity in terms of implementation, and reduced computational and memory complexity, the *PSO* has been successfully used in many applications such as robotics (Tang, Zhu, & Sun, 2005) (Pires, Oliveira, Machado, & Cunha, 2006) (Couceiro, Mendes, Fonseca Ferreira, & Tenreiro Machado, 2009) (Couceiro, Luz, Figueiredo, & Ferreira, 2012), electric systems (Alrashidi & El-Hawary, 2006) and sport sciences (Couceiro, Luz, Figueiredo, Ferreira, & Dias, 2010).

However, a general problem with the *PSO*, as many other optimization algorithms, is that of becoming trapped in a sub-optimal solution, such that it may work well on one problem but yet fail on another problem. In order to overcome this problem, many authors have suggested other adjustments to the parameters of the *PSO* algorithm combining *fuzzy logic* where the inertia weight w is dynamically adjusted using fuzzy **IF-THEN-ELSE** rules (Shi & Eberhart, 2001; Liu, Abraham, & Zhang, 2007) or *Gaussian* approaches where the inertia constant w is no longer needed and the acceleration constants ρ_1 , ρ_2 and ρ_3 are replaced by random numbers with Gaussian distributions (Secret & Lamont, 2003). More recently, Pires *et al.* used *fractional calculus* to control the convergence rate of the *PSO* (Pires, Machado, Cunha, & Mendes, 2010). The authors rearrange the original velocity equation (4.1) in order to modify the order of the velocity derivative. The work of Pires *et al.* (Pires, Machado, Cunha, & Mendes, 2010) was the foundation for the methodology presented in this Thesis. Nevertheless, more than to simply control the convergence rate of robots, its main purpose is to consider robots' dynamics (next section).

Many authors have considered incorporating evolutionary properties, such as *selection*, *mutation* and *crossover*, as well as the *differential evolution*, into the *PSO* algorithm (Valle, Venayagamoorthy, Mohagheghi, Hernandez, & Harley, 2008). The main goal of those works was to increase the diversity of the population by either preventing the particles to move too close to each other and collide (Blackwell & Bentley, 2002) (Krink, Vesterstrom, & Riget, 2002) or to self-adapt parameters such as the constriction factor, acceleration constants (Miranda & Fonseca, 2002), or inertia weight (Lovbjerg & Krink, 2002). The fusion between *Genetic Algorithms* and the *PSO* originated an improved version which combines the advantages of swarm intelligence and a natural selection mechanism in order to increase the number of highly evaluated agents, while decreasing the number of lowly evaluated agents at each iteration step (Chia-Feng, 2004). Similar to this last one, a differential evolution operator has been proposed to improve the performance of the *PSO* algorithm in two different ways. The first one (Zhang & Xie, 2003) applies the differential evolution operator to the particle's best position

to eliminate the particles falling into sub-optimal solution, while the second one (Kannan, Slochanal, & Padhy, 2004) applies it to find the optimal parameters (inertia and acceleration constants).

In search of an ever-improving model of natural selection using the *PSO* algorithm, the Darwinian Particle Swarm Optimization (*DPSO*) was first formulated by (Tillett, Rao, Sahin, Rao, & Brockport, 2005), in which many *subgroups* (or smaller swarms) of test solutions may exist at any time. Note that the word “*subgroup*” will be used throughout this Thesis to identify clusters of robots within the whole swarm (*i.e.*, population). In a few words, we could state that subgroups are a particular case of swarms, wherein, contrarily to swarms, there is a predefined maximum number of robots allowed to form each subgroup.

In the *DPSO*, each subgroup individually performs just like an ordinary *PSO* with some rules governing the collection of subgroups that are designed to simulate natural selection. This natural selection mechanism, or *Darwinian* principle of *survival-of-the-fittest*, enhance the ability of the *PSO* to escape from sub-optimality. The idea is to run many simultaneous parallel *PSO* algorithms, each one being a different subgroup, on the same test problem, while a simple selection *competitive* mechanism is applied. When a search tends to a local optimum, the search in that area is simply discarded and another area is searched instead. In this approach, at each step, subgroups that get better are rewarded (extend particle life or spawn a new descendent) and subgroups which stagnate are punished (reduce subgroup life or delete particles). To analyse the general state of each subgroup, the fitness of all particles is evaluated and the neighbourhood and individual best positions of each of the particles are updated. If a new global solution is found, a new particle is spawned. A particle is deleted if the subgroup fails to find a fitter state in a defined number of steps (Algorithm 4.2). In brief, particles within the same subgroup cooperates while different subgroups compete toward the same goal and with cross-beneficial properties, thus leading to the concept of *cooperation* (Tsai, 2002).

Some simple rules are followed to delete a subgroup, delete particles, and spawn a new subgroup and a new particle: *i*) when the subgroup population falls below a minimum bound, the subgroup is deleted; and *ii*) the worst performing particle in the subgroup is deleted when a maximum threshold number of steps (search counter SC_{max}) without improving the fitness function is reached. After the deletion of the particle, instead of being set to zero, the counter resets to a value that tends to the threshold number, according to:

$$SC_s = SC_{max} \left[1 - \frac{1}{N_s^{kill+1}} \right], \quad (4.3)$$

with N_s^{kill} being the number of particles deleted from the subgroup s over a period in which there was no improvement in the fitness. For a subgroup to spawn a new subgroup, the subgroup must not have any particle ever deleted and the maximum number of subgroups must not be exceeded. Still, the new subgroup is only created with a small probability of:

$$p_{sp} = \frac{r_{sp}}{NS}, \quad (4.4)$$

wherein r_{sp} is a random number between 0 and 1 and NS the number of subgroups. This factor avoids the creation of newer subgroups when there is a large number of subgroups in existence. The parent subgroup is not affected and half of the parent's particles are selected at random for the child subgroup and the other half of the particles of a random member of the subgroup collection are also selected. If the subgroup initial population number is not obtained, the rest of the particles are randomly initialized and added to the new subgroup. A particle is spawned whenever a subgroup achieves a new global best and the maximum defined population of a subgroup has not been reached. All these rules belong to the “*punish-reward*” mechanism of the *DPSO*.

Initialize all subgroups (Initialize x_t^n , v_t^n , \bar{x}_t^n , \bar{v}_t^n and \bar{g}_t^n)

Loop (**Main Program**):

```

| For each subgroup in the collection
|   | Evolve the subgroup (goto: Evolve Swarm Algorithm)
|   | Allow the subgroup to spawn
|   | Delete “failed” subgroups
| end

```

until stopping criteria (convergence)

Function (**Evolve Swarm Algorithm**):

```

| for all particles in the subgroup
|   | Evaluate the fitness of each particle
|   | Update  $\bar{x}_t^n$ ,  $\bar{v}_t^n$  and  $\bar{g}_t^n$ 
|   | Update  $v_{t+1}^n$  and  $x_{t+1}^n$ 
|   | If subgroup gets better
|   |   | Reward subgroup: spawn particle: extend subgroup life
|   | If subgroup has not improved
|   |   | Punish subgroup: possibly delete particle: reduce subgroup life
| end

```

return

Algorithm 4.2. *DPSO* Algorithm

Like the *PSO* algorithm, a few parameters of the *DPSO* also need to be adjusted to run the algorithm efficiently (continuing the numbered list from above):

- iii) Initial subgroup population N_I ;
- iv) Maximum and minimum subgroup population N_{max} , N_{min} ;
- v) Initial number of subgroups N_S^I ;
- vi) Maximum and minimum number of subgroups N_S^{max} , N_S^{min} ;
- vii) Stagnancy threshold SC_{max} .

In estimation problems previously studied in (Couceiro, Luz, Figueiredo, Ferreira, & Dias, 2010), segmentation and classification methods compared in (Ghamisi, Couceiro, Benediktsson, & Ferreira, 2012), both *DPSO* and the fractional-order variation of it proposed in (Couceiro, Ferreira, & Machado, 2011) have been successfully compared with the *PSO* and many other exhaustive and evolutionary methods depicting a superior performance.

In this Thesis, we try to go a step forward by adapting the version of the *DPSO* to *MRS*, denoting it as Robotic *DPSO* (*RDPSO*). Any other *PSO* variant could be adapted to *MRS* exploration. However, the *DPSO* was chosen since it is an evolutionary algorithm that extends the *PSO* using natural selection to enhance the ability to escape from sub-optimal solutions. Moreover, it does that without significantly increasing the computational cost of the traditional *PSO*. As such, just like in *MRS* where groups of robots interact to accomplish their goals, the *DPSO* use groups of interacting virtual agents in order to achieve its optimization. However, real *MRS* present several constraints that need to be considered. Contrarily to virtual agents, robots are designed to act in the real world where obstacles need to be taken into account. Also, and since that in certain environments or applications the communication infrastructure may be damaged or missing, the self-spreading of autonomous mobile nodes of a mobile ad hoc network (*MANET*) over a geographical area needs to be considered. For instance, the development of robot teams for surveillance or rescue missions in unstructured and unknown environments require that robots have to be able to maintain communication among them without the aid of a communication infrastructure.

Since the herein proposed *RDPSO* approach is an adaptation of the *DPSO* to real mobile robots, four general features are (so far) proposed:

1. An improved inertial influence based on fractional calculus concept taking into account convergence dynamics;
2. Integration of an obstacle avoidance behaviour to avoid collisions;
3. A way to enforce multi-hop network connectivity to ensure that the *MANET* remains connected throughout the mission;

4. A novel robots' "punish-reward" mechanism to emulate the deletion and creation of particles in the original algorithm;

Therefore, to model a robotic swarm, each robot n , moves in a multidimensional space. For the sake of simplicity, let us rearrange the previous equations from (4.1)-(4.2), describing The *RDPSO* with the following discrete equation (*DE*) system:

$$v_n[t + 1] = w_n[t]v_n[t] + \sum_{i=1}^4 \rho_i r_i (\chi_i[t] - x_n[t]), \quad (4.5)$$

$$x_n[t + 1] = x_n[t] + v_n[t + 1], \quad (4.6)$$

wherein $w_n[t]$ and ρ_i , $i = 1,2,3,4$, assign weights to the inertial influence, the local best (cognitive component), the global best (social component), the obstacle avoidance component and the enforcing communication component when determining the new velocity, with $\rho_i > 0$. As before, r_i are random vectors where in each component is generally a uniform random number between 0 and 1. $v_n[t]$ and $x_n[t]$ represents the velocity and position vector of robot n , respectively. While $\|v_n[t]\|$ is limited to the maximum allowed velocity of v_{max} for robots, *i.e.*, $\|v_n[t]\| \leq v_{max}$, $x_n[t]$ depends on the scenario dimensions. $\chi_i[t]$ represents the best position of the cognitive, social, obstacle and *MANET* matrix components. The cognitive $\chi_1[t]$ and social components $\chi_2[t]$ are the commonly presented in the classical *PSO* algorithm (*cf.*, equation (4.1)). $\chi_1[t]$ represents the local best position of robot n while $\chi_2[t]$ represents the global best position of robot n . The size of the vectors (ϖ) depends on the dimensionality \mathbb{R}^ϖ of the physical space being explored, *e.g.*, $\varpi = 2$ for planar problems.

Since the other features are novel, they are further explored in the following sections. Let us start by describing the inertial component $w_n[t]$ that, contrarily to most traditional methods in which it is simply proportional to the inertial influence, the *RDPSO* uses *fractional calculus* (Podlubny, 1999), to describe the dynamic phenomenon of a robot's trajectory that depends on past events (Couceiro M. S., Martins, Rocha, & Ferreira, 2012).

4.2 Fractional Order Convergence

As previously presented in section 3.1.1, fractional calculus has been drawing researchers' attention for the last two decades, being rediscovered and applied in an increasing number of fields. This section proposes a new method to control the convergence rate of the *RDPSO* algorithm based on Pires *et al.* fractional-order approach to the traditional *PSO* (Pires, Machado, Cunha, & Mendes,

2010). A previous work presented a similar approach conducted to the traditional *DPSO* to optimization problems in (Couceiro, Ferreira, & Machado, 2011).

By manipulating equations (4.5) and considering $w = 1$, one can describe $v_n[t + 1] - v_n[t]$ as the discrete version of the fractional difference of order $\alpha = 1$, *i.e.*, the first order integer difference $\Delta^d v_n[t + 1]$ (*cf.*, Definition 3.3). Assuming $T = 1$ and based on (Pires, Machado, Cunha, & Mendes, 2010) work, the following expression can be defined (Definition 3.2):

$$D^\alpha[v_n[t + 1]] = \sum_{i=1}^4 \rho_i r_i (\chi_i[t] - x_n[t]). \quad (4.7)$$

Based on fractional calculus concepts and Definition 3.3, the order of the velocity derivative can be generalized to a real number $0 < \alpha < 1$, thus leading to a smoother variation and a longer memory effect. Therefore, considering the discrete time fractional differential presented on Definition 3.2, one can define the inertial component $w_n[t]$ from equation (4.5) as:

$$w_n[t]v_n[t] = - \sum_{k=1}^r \frac{(-1)^k \Gamma[\alpha+1]v[t+1-kT]}{\Gamma[k+1]\Gamma[\alpha-k+1]}. \quad (4.8)$$

being Γ the gamma function, α the fractional coefficient and r the truncation order.

In other words, the next position of a given robot depends, not only, on its current position, but also on the previous r positions. Equation (4.8) allows then to fit the dynamic phenomena of a robot's trajectory because of its inherent memory property, thus controlling the robot convergence.

4.2.1 Memory Complexity

Adding memory to the *RDPSO* algorithm allows improving the convergence rate of robots since each robot will have the information about its preceding actions. Nevertheless, the computational requirements increase linearly with r , *i.e.*, the *RDPSO* present a $\mathcal{O}(r)$ memory complexity *per* robot. Moreover, it is noteworthy that this kind of optimization or foraging algorithms presents a higher performance as the number of robots increase. Hence, robots should be as simple and low-cost as possible (*i.e.*, swarm robots) which are usually memory limited (see section 3.2.1).

Therefore, the truncation of equation (4.8) will depend on the requirements of the application and the features of the robot. For instance, for the *eSwarBot* platforms previously presented in section 3.2.1, a $r = 4$ leads to results of the same type than for $r > 4$. Although one could consider the processing power as the main reason to such a limited number of terms, the kinematical features of

the platform and the mission requirements also needs to be considered in such a way that one can present the following result.

Proposition 4.1. *Let δ and Δx_{max} be the encoders-wheel resolution of robots and the maximum travelled distance allowed between iterations, i.e., $\Delta x_{max} = x_n[t - 1] - x_n[t]$, respectively. If τ is the minimum natural number that verifies the following inequality:*

$$-\frac{(-1)^\tau \Gamma[\alpha+1] \Delta x_{max}}{\Gamma[\tau+1] \Gamma[\alpha-\tau+1]} < \delta, \quad (4.9)$$

then the RDPSO equation (4.8) should be truncated based on δ and Δx_{max} , in $r = \tau - 1$.

Proof: Let us consider the example of a differential drive robot (e.g., *eSwarBot*). A differential drive robot consists of two independently driven wheels and, usually, a free wheel for stability (e.g., caster wheel). For navigation purposes, the driven wheels are usually equipped with encoders that provide odometry measures. Hence, the major odometry parameter of such mobile robot to drive forward is the radius of the wheels R_{wheel} and the number of pulses from revolution of the wheel τ_{rev} . The kinematical equation of a differential drive robot, while moving forward, can be defined as:

$$pulses = \tau_{rev} \times \frac{dist}{2\pi \times R_{wheel}}, \quad (4.10)$$

where *pulses* is the number of pulses necessary for the robot to travel a distance of *dist*. Defining *pulses* = 1 we can obtain the minimum distance that a robot can travel at each iteration, i.e., the resolution δ . Hence, an increment of the distance lower than δ would be unfeasible for the robot to travel. Also, one may observe through equation (4.8), that the relevance of past events, i.e., the $v[t + 1 - kT]$ term, reduces over time. In other words, from a given term $r = \tau - 1$, the relevance of all previous events before it would be irrelevant as the robot would be unable to travel with such accuracy. ■

To clarify the previous result, let us consider the following example.

Example 4.1. *Considering the eSwarBot platform, a resolution of $\delta = 2.76$ mm is obtained for a single pulse, taking into account that $R_{wheel} = 21.09$ mm and the combination between encoders-*

wheel provides $\tau_{rev} = 48$ pulses/revolution. Let us consider a maximum travelled distance between two iterations of $\Delta x_{max} = 0.1$ m, i.e., the robot cannot travel more than 0.1 m without any update of the information. Figure 4.1 presents the computation of each term of equation (10). As one may observe, a term of $r = 4$ would be enough to represent the RDPSO dynamics in such conditions as the 5th term returns an increment of 2.73 mm. In other words, the algorithm would present similar results for $r \geq 4$.

As *eSwarBots* would be the robotic platforms most widely used throughout this work, one will only consider the first $r = 4$ terms of the fractional discrete difference in (4.8), yielding:

$$v_n[t + 1] = \alpha v_t^n + \frac{1}{2} \alpha (1 - \alpha) v_{t-1}^n + \frac{1}{6} \alpha (1 - \alpha) (2 - \alpha) v_{t-2}^n + \frac{1}{24} \alpha (1 - \alpha) (2 - \alpha) (3 - \alpha) v_{t-3}^n + \sum_{i=1}^4 \rho_i r_i (\chi_i[t] - x_n[t]). \quad (4.11)$$

It is, however, noteworthy that a similar analysis should be conducted if one intends to use different robotic platforms. The same analysis was carried out with *e-puck* platforms (see section 3.2.1) obtaining the same truncation number ($r = 4$).

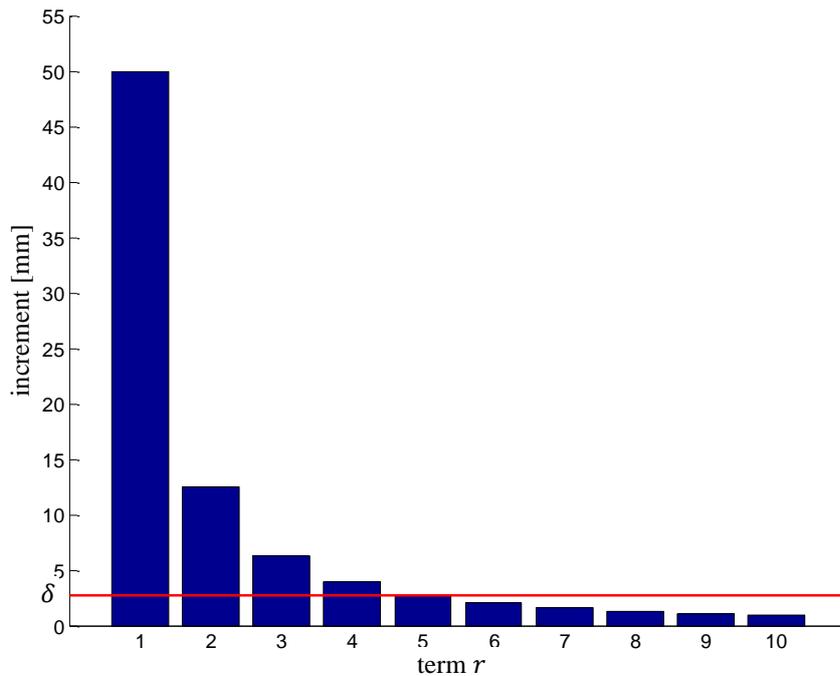


Figure 4.1. Convergence of the robot toward the solution changing the differential derivative r .

4.3 Obstacle Avoidance

In short, the *RDPSO* algorithm tries to minimize a cost function, or maximize a fitness function depending on the mission objective. For instance, if we have a gas leak, robots running the *RDPSO* algorithm try to maximize the sensed gas at each iteration. The approach proposed in this chapter seeks to create a new cost or fitness function in such a way that it would guide the robot to perform the main mission (*e.g.*, find the gas leak) while avoiding obstacles.

When a robot needs to move from any arbitrary start position to any target position in the environment, it must be able to avoid both static and dynamic obstacles (Williams & Wu, 2010). For this purpose we assume that each robot is equipped with sensors capable of sensing the environment for obstacle detection within a finite sensing radius R_w . A monotonic and positive *sensing function* $g(x_n[t])$ is then defined. This function depends on the sensing information, *i.e.*, distance from the robot to obstacle. Note that in most situations, as it can be observed in Figure 4.2, the *sensing function* $g(x_n[t])$ can be represented as the relation between the analog output voltage of distance sensors (*e.g.*, sonars) and the distance to the detected object.

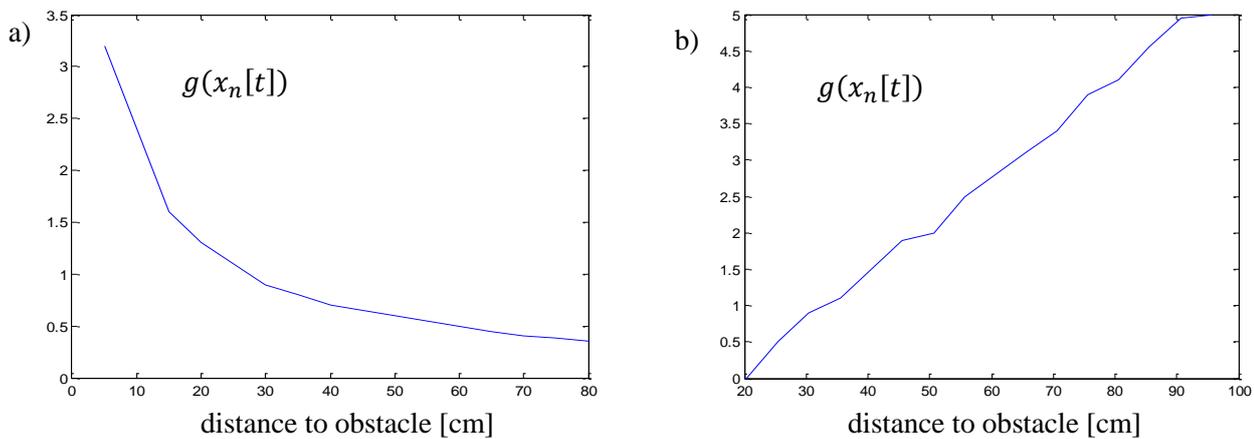


Figure 4.2. Sensing function $g(x_n[t])$ represented as the relation between the analog output voltage of distance sensors and the distance to obstacle. a) Sensor Sharp GP2Y0A21YK IR - monotonically decreasing sensing function $g(x_n[t])$; b) Sensor Sonaswitch Ultrasound - monotonically increasing sensing function $g(x_n[t])$.

Variable $\chi_3[t]$ is then represented by the position of each robot that optimizes the monotonically decreasing or increasing $g(x_n[t])$. In other words, when a robot does not sense any obstacle at time t , the best position that optimizes $g(x_n[t])$ is constantly updated and equal to the current position $x_n[t]$. Afterward, if the robot detects an obstacle inside its sensing range, the best position that optimizes $g(x_n[t + 1])$ is not updated, thus creating an attractive force towards its last best position

$x_n[t]$. While in a free-obstacle environment ρ_3 can be set to zero, in real-world scenarios, obstacles need to be taken into account and the value of ρ_3 depends on several conditions related with the main objective (*i.e.*, minimizing a cost function or maximizing a fitness function) and the sensing information (*i.e.*, monotonicity of the *sensing function* $g(x_n[t])$). Furthermore, the relation between ρ_3 and the other weights, namely ρ_1 , ρ_2 and ρ_4 , depends on the susceptibility of each robot to the main objective and the obstacle avoidance behaviour (Figure 4.3). For instance, a $\rho_3 \ll \min(\rho_1, \rho_2)$ may lead to a faster convergence to the solution making the team of robots to accomplish the main objective faster, but it may also lead to obstacle collisions (Figure 4.3a) while a $\rho_3 \gg \min(\rho_1, \rho_2)$ may increase the performance of obstacle avoidance but may lead to a slower convergence of the main objective (Figure 4.3b).

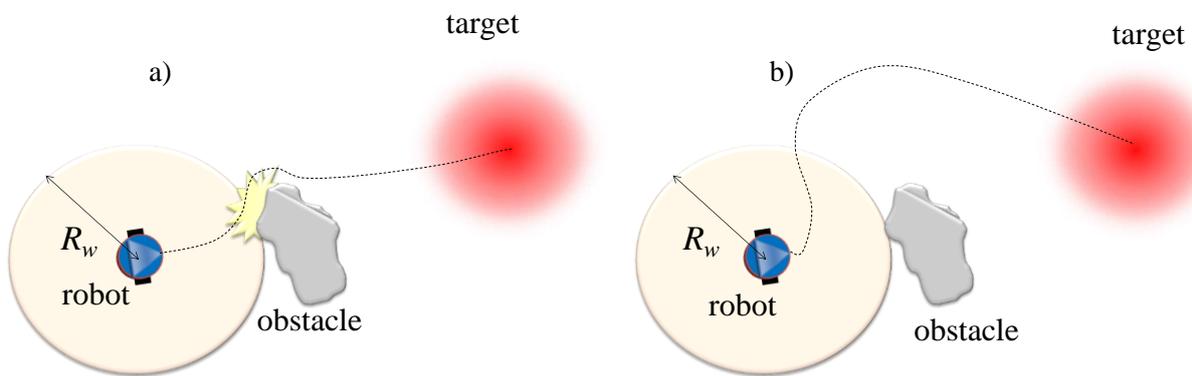


Figure 4.3. Illustrative example of obstacle avoidance behaviour of a robot. a) $\rho_3 \ll \min(\rho_1, \rho_2)$; b) $\rho_3 \gg \min(\rho_1, \rho_2)$.

Besides considering obstacle avoidance in real world applications, communication constraints also need to be considered.

4.4 Ensuring *MANET* Connectivity

It has generally been assumed in *MRS* that each robot has the ability to communicate with any other robot with small consideration for the quality and performance of the wireless communication network. Although being valid in some particular situations, the assumption does not generally hold. Since robots may move apart to further areas, it is important to have a pervasive networking environment for communications among robots. Furthermore, without a pre-existent infrastructure, robots need to be able to act as intermediate nodes, *i.e.*, routers, in order to relay information from one point

to another, thus supporting multi-hop communication in a *MANET* (Miller, 2001). One way of ensuring that is by controlling robots' position in order to maintain the communication based on constraints such as maximum distance d_{max} or minimum signal quality q_{min} . The way network will be forced to preserve connectivity depends on communication characteristics (*e.g.*, multi-hop, biconnectivity) (Crispin, 2009). At this point, only the minimum necessary requirements to maintain *MANET* connectivity are considered. This topic will be further explored in next chapter.

4.4.1 Problem Statement

Consider a subgroup (or smaller swarm) of N_s robots where each robot is both an exploring agent of the environment and a mobile node of a *MANET* that performs packet forwarding, according to a paradigm of *multi-hop communication*. The goal is to ensure that robots explore an unknown environment, while ensuring that the *MANET* remains connected throughout the mission.

4.4.2 General Approach

Assuming that the network supports multi-hop connectivity, the communication between two end nodes (*i.e.*, robots) is carried out through a number of intermediate nodes whose function is to relay information from one point to another. Considering that nodes are mobile, it is necessary to guarantee the communication between all nodes. In the case where each robot corresponds to a node, in order to overcome the non-connectivity between them, the desired position $x_n[t + 1]$ must be controlled.

The connectivity between robots can be described by means of a *link matrix* $L = [l_{ij}] \in \mathbb{R}^{N_s \times N_s}$ for an N_s -robot network, wherein each entry represents the link between robot i and j . The link is defined accordingly with the users' preferences. The most common approaches include:

1. Calculating the l_{ij} values as functions of the distance between pairs of robots indicating the link distance between them (Sheng, Yang, Tan, & Xi, 2006);
2. Calculating the l_{ij} values as functions of the radio quality signal between pairs of robots indicating the link quality between them (Tardioli & Villarroel, 2007).

Trying to maintain the network connectivity by only taking into account the communication range d_{max} (approach 1) does not match reality since the propagation model is more complex – the signal depends not only on the distance but also on the multiple paths from walls and other obstacles (approach 2). However, in simulation, the communication distance is a good approach and it is easier to implement.

Depending on the chosen approach (1 or 2), an *adjacency matrix* $A_c = [a_{ij}] \in \mathbb{B}^{N_s \times N_s}$, in which \mathbb{B} represents the set of binary numbers (*i.e.*, 0 or 1), can be defined based on the maximum distance

or minimum radio quality signal between robots, respectively (Miller, 2001). The adjacency matrix, *i.e.*, one-hop connectivity matrix, where a 1 entry at (i, j) indicates a connection between robot i and j and a 0 entry at (i, j) indicates no connection between robot i and j , represents the neighbours of each node, *i.e.*, direct connection between robots.

$$a_{ij} = \begin{cases} 1, & \text{connection between node } i \text{ and } j \\ 0, & \text{no connection between node } i \text{ and } j \end{cases} \quad (4.12)$$

Note that the diagonal elements (*i.e.*, when $i = j$) of the adjacency matrix are set equal to 0. If the communication system supports the relay of messages to distant robots via intermediate robots, then multi-hop connections can be made. Using the hop distances, *i.e.*, the smallest numbers of hops to connect non-adjacent robots, the zero-valued off-diagonal entries in the adjacency matrix can be manipulated in order to create a multi-hop *connectivity matrix* $C^{(N_k)} = [c_{ij}^{(N_k)}] \in \mathbb{R}^{N_s \times N_s}$, for which the entry at (i, j) represents the minimum number of hops necessary to connect robot i and j , and N_k represents the iteration which varies with the number of hops the network can handle. $N_s - 1$ is the maximum number of possible hops. The connectivity matrix can then be defined as:

$$c_{ij}^{(N_k)} = \begin{cases} h, & \text{node } i \text{ connected to } j \text{ by } h \leq N_k \text{ hops} \\ 0, & \text{otherwise} \end{cases} \quad (4.13)$$

Note that the diagonal elements of the connectivity matrix ($i = j$) are set equal to 0. Furthermore, the adjacency matrix is the first iteration in calculating the connectivity matrix ($C^{(1)} = A_c$). When $N_k > 1$ (*i.e.*, for multi-hop connections) an auxiliary matrix $B^{(N_k)} = [b_{ij}^{(N_k)}] \in \mathbb{R}^{N_s \times N_s}$ is then calculated based on the iteration (number of hops):

$$b_{ij}^{(N_k)} = \begin{cases} 0, & c_{ij}^{(N_{k-1})} > 0 \\ K, & \sum_{v=1}^{N_s} c_{iv}^{(N_{k-1})} b_{vj}^{(N_{k-1})} > 0 \text{ and } c_{ij}^{(N_{k-1})} = 0 \end{cases} \quad (4.14)$$

Note that the diagonal elements ($i = j$) of the auxiliary matrix are set equal to 0 and ($B^{(1)} = A_c$). The connectivity matrix can now be calculated using the following equation:

$$C^{(N_k)} = C^{(N_{k-1})} + B^{(N_k)}. \quad (4.15)$$

After $N_s - 1$ iterations, $C^{(N_s-1)}$ represents the multi-hop network connectivity. The existence of zero elements (except diagonal elements) indicates no connection between robot i and j even using multi-hop. In this case it is necessary to implement an algorithm to ensure the complete connectivity of the network. One strategy is to define a *binary connectivity matrix* $C_B = [c_{Bij}] \in \mathbb{B}^{N_s \times N_s}$ wherein each non-zero element of the connectivity matrix matches the logic value 1.

$$c_{Bij} = \begin{cases} 1, & c_{ij}^{(N_s-1)} \neq 0 \\ 0, & c_{ij}^{(N_s-1)} = 0 \end{cases} \quad (4.16)$$

Performing an element-by-element multiplication between the link matrix and the logical inverse (binary *NOT*) of the binary connectivity matrix, yields a *break matrix* $C_{break} = [c_{breakij}] \in \mathbb{B}^{N_s \times N_s}$ containing the values that represent the break of connection between the robots. In the case where each robot corresponds to a robot, in order to overcome the non-connectivity between them, the desired position $x_n[t + 1]$ of each robot from equation (4.6) must be controlled since it influences the link matrix. In this chapter, the multi-hop connectivity matrix $C^{(N_s-1)}$ and auxiliary matrices (C_B and C_{break}) will only be used as information about the network topology (Algorithm 4.3).

Determines $N_s \times N_s$ link matrix L
Calculates $N_s \times N_s$ adjacency matrix A_c
Initialize $N_s \times N_s$ connectivity and auxiliary matrix $C^{(1)} = A, B^{(1)} = A_c$
For $N_k = 2$ to the longest hop ($N_k = N_s - 1$)
 For all robot pairs $(i, j) = (1, 1)$ to (N_s, N_s)
 If $i = j$ OR $c_{ij}^{(N_k)} > 0$
 | skip to the next robot pair
 If $\sum_{v=1}^{N_s} c_{iv}^{(N_k-1)} b_{vj}^{(N_k-1)} > 0$ AND $c_{ij}^{(N_k-1)} = 0$
 | $b_{ij}^{(N_k)} = N_k$
 $C^{(N_k)} = C^{(N_k-1)} + B^{(N_k)}$
Calculates $N_s \times N_s$ binary connectivity and break matrix C_B, C_{break}
If the connectivity depends on the distance/quality, find the minimum/maximum value of each line of link matrix L , excluding zeros and (i, j) pairs previously chosen
Computes equation (4.5)

Algorithm 4.3. Ensuring subgroup s network connectivity.

As Algorithm 4.3 describes, one way to ensure the full connectivity of the *MANET* is to “force” each robot to communicate with its nearest neighbour that has not chosen it as its nearest neighbour. Since the connectivity depends on the distance/signal quality, connectivity between robots may be ensured by computing the minimum/maximum value of each line of link matrix L , after excluding zeros and (i, j) pairs previously chosen. Therefore, the *MANET* component $\chi_4[t]$ is represented by

the position of the nearest neighbour increased by the maximum communication range d_{max} towards robot's current position. A larger ρ_4 may enhance the ability to maintain the network connected ensuring a specific range or signal quality between robots (an example of a link matrix can be observed in Figure 5.6). To better understand how the *MANET* connectivity principle works, please consider the topology depicted in Figure 4.4.

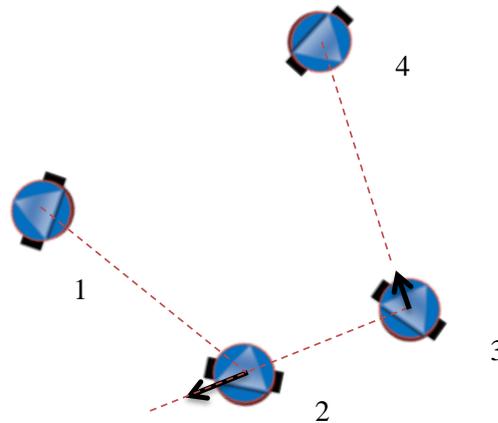


Figure 4.4. Illustration of a *MANET* topology of a subgroup. Dashed lines represent the maximum distance d_{max} between each pair of robots and the bold arrows represent the force vectors that ensure *MANET* connectivity.

As it may be perceived, robot 2 is the nearest neighbour of robot 1 and is at the correct distance d_{max} resulting in a null force connectivity vector. The nearest neighbour of robot 2 is robot 3 which is too close, thus resulting in a repulsive force at robot 2 in order to ensure d_{max} . Finally, the nearest neighbour of robot 3, that was not previously chosen, is robot 4 which is too far away, thus being affected by an attractive force toward robot 4.

Note that having multiple subgroups, which is inherent to the proposed *RDPSO* algorithm, enables a distributed approach because the network that was previously defined by the whole population of robots (swarm) is now divided into multiple smaller *MANETs* (one for each subgroup), thus decreasing the number of robots and the information exchanged between robots of the same network. In other words, robots interaction with other robots through communication is confined to local interactions inside the same subgroup, thus making *RDPSO* scalable to large populations of robots. The exchanged data concerning to the signal quality or robot's position allows the implicit processing of the *RDPSO* algorithm by the team in a distributed way. In other words, every robot needs to be aware of the position or signal quality of all other robots in the same subgroup in order to proceed to the next iteration of the algorithm. This is a limitation of the algorithm since all robots need to be equipped with a good odometry or localization systems (*e.g.*, *GPS*). An alternative to it would be

extending the *GPS* capabilities of some robots to non-*GPS* robots using strategies to find the teammates position under their visual range (Kulkarni & Venayagamoorthy, 2010). For instance, if robots are equipped with *LRFs*, retro-reflective markers can be used for recognition. Since the implementation of such strategies is out of scope of this Thesis, they will not be taken into account and it is considered that each robot knows its own position.

4.5 “Punish-Reward” Mechanism

Besides the biological inspiration that makes roboticists capture the underlying principles of living organisms and assimilate them into ever-improving robot forms, the social behaviours inherent to the several biological societies have been studied and applied to *MRS* showing interesting and promissory results (Balch & Hybinette, 2000).

In this work, it is addressed one of the most relevant social processes from human society: *the social exclusion*. The formal concept of social exclusion appeared in the 70’s in France referring to the “*rupture of the social bond*”, where socially excluded individuals or groups were unprotected by the French social security system, thus resulting in a lack of resources and inadequate access to services making it difficult to participate in the society (Scutella, Wilkins, & Horn, 2009). This concept has since grown, being taken up by most of Europe, and is currently used to refer to the range of dimensions which marginalize people and reduce their opportunities to engage in social or political life.

Burchardt proposes a more precise, and also close related with our approach, definition of social exclusion (Burchardt, 2000): “*an individual is socially excluded if he or she does not participate to a reasonable degree over time in certain activities of his or her society, and (a) this is for reasons beyond his or her control, and (b) he or she would like to participate*”. On the other hand, for a socially excluded individual or group to be accepted back in the society, *aka* social inclusion, the Brotherhood of St Laurence’s Executive Director Tony Nicholson has proposed the following definition (Nicholson, 2008): “*a social inclusion approach involves the building of personal capacities and material resources, in order to fulfil one’s potential for economic and social participation, and thereby a life of common dignity*”. In other words, social inclusion, the converse of social exclusion, is the affirmative action to change the circumstances and habits that lead to (or have led to) the social exclusion.

However, this is not an exclusive concept from the human race. These concepts of social exclusion and inclusion may also be found in nonhuman animals through *stigmatization*²⁵ processes such as:

- Territoriality (*e.g.*, fish, birds, reptiles and mammals) – exclusion of other members of the same species (*e.g.*, certain sex) from an area;
- Status hierarchies (*e.g.*, some bird species, lions, baboons, chimps) – individual at the top of the hierarchy excludes others from resources (*e.g.*, food, territory, mates);
- Social ostracism (*e.g.*, some fish species, lemurs, baboons, chimps) – prevent others from joining social group or forcing expulsion.

For instance, *Three-spined Sticklebacks* (fish) avoid others of the same specie with parasites while Grizzlies (bear) present a hierarchy-related behaviour that provides a mechanism that mutes the potential social costs of membership in stable aggregations (Craighead, Sumner, & Mitchell, 1995).

How this concept of socially exclusion (and inclusion) is used in this Thesis and what are the advantages inherent to it resides in the “*punish-reward*” mechanism previously described in the *DPSO* and that will now be adapted to *MRS*.

4.5.1 Socially Active Subgroups

As the original *DPSO*, the number of times a subgroup s is evolved without finding an improved objective was tracked with a search counter defined in equation (4.3), SC_s . In the proposed approach, N_s^{kill} represents the number of robots excluded from the subgroup s over a period of time in which there was no improvement in the subgroup’s objective function. If the subgroup’s search counter exceeds a maximum critical threshold, SC_{max} , the subgroup is punished by excluding the worst performing robot, which is added to a socially excluded subgroup. The worst performing robot is evaluated by the value of its objective function compared to other members in the same subgroup, *i.e.*, if the objective is to maximize the fitness function the robot to be excluded will be the one with the higher fitness value. If the number of robots falls below the minimum acceptable number of robots to form a subgroup N_{min} , the subgroup is punished by being dismantled and all the robots that belong to that subgroup are added to the socially excluded subgroup. On the other hand, if the subgroup improves its objective function, then it is rewarded with the best performing robot in the socially excluded subgroup. If a subgroup has been more often rewarded than punished, it has a small probability p_{sp} of spawning a new subgroup with the predefined number of robots N_f . However, contrarily

²⁵ Process where certain individuals are excluded from particular sorts of social interactions.

to the *DPSO* in which p_{sp} depended on the current number of active subgroups, since the *RDPSO* does not allow inter-subgroup communication (unless to forward messages between robots from the same subgroup), p_{sp} is rewritten as:

$$p_{sp} = r_{sp} \frac{N_s}{N_{max}}, \quad (4.17)$$

with r_{sp} being a random number between 0 and 1, N_s is the number of robots within subgroup s and N_{max} is the maximum number of allowed robots in a subgroup. In other words, if a socially active subgroup is constantly improving, it is able to create another socially active subgroup with an expected value of 0.5 when it is at full capacity, *i.e.*, $N_s = N_{max}$. Moreover, the group of robots forming this new subgroup will be the best performing robots within the socially excluded subgroup.

The *RDPSO* “punish-reward” rules are summarized in Table 4.1.

Table 4.1. “Punish-Reward” *RDPSO* Rules.

PUNISH	REWARD
If a socially active subgroup does not improve during a specific threshold SC_{max} (stagnancy counter $SC_s = SC_{max}$) and the number of robots is superior to N_{min} ($N_s > N_{min}$), then the subgroup is punished by socially excluding the worst performing robot	If a socially active subgroup improves and its current number of robots is inferior to N_{max} ($N_s < N_{max}$) and there is, at least, one socially excluded robot, then it is rewarded with the best performing socially excluded robot
If a socially active subgroup does not improve during a specific threshold SC_{max} (stagnancy counter $SC_s = SC_{max}$) and the number of robots is N_{min} ($N_s = N_{min}$), then the subgroup is punished by being dismantled, <i>i.e.</i> , all robots from that subgroup are socially excluded	If a socially active subgroup is not stagnated (stagnancy counter $SC_s = 0$) and there are, at least, N_l socially excluded robots, then it has a small probability p_{sp} of spawning a new socially active subgroup

4.5.2 Socially Excluded Subgroups

The key issue in this novel approach is the answer to the question:

“What does the robots of the socially excluded subgroup do?”

In fact, the answer is the same that we would give if asking about a group excluded from our society: they do not follow the rules imposed by the society and, henceforth, they do not directly contribute for it. In the context of this work, instead of searching for the objective function’s optimality (*i.e.*, the main activity of the society) like other robots in the active subgroups do, socially excluded

robots basically randomly wander in the scenario. Note, however, that they are always aware of their individual solution and the global solution of the socially excluded subgroup.

Nevertheless, the dynamic partitioning of the whole population of robots into multiple subgroups, despite the advantages inherent to it (*i.e.*, avoid sub-optimal solutions and improve scalability), still yields several issues that need to be solved.

When a robot is excluded from a subgroup, it needs to be able to find the other excluded robots. However, it may be unable to communicate with them. As a socially excluded robot, it will randomly wander in the scenario. As it moves, the robot will broadcast a message containing its current position. At this point, three possible situations may occur:

- i)* Any robot ear the message and the socially excluded robot will continue to randomly wander broadcasting the same message;
- ii)* If a robot from an active subgroup receives the message, it will forward it to any available robots in vicinities;
- iii)* If a robot from the socially excluded subgroup receives the message, it will answer with a message containing its current position.

If the second situation is verified and an excluded robot receives the forwarded message, it will answer with a message containing its current position. The active subgroup will then receive the answer, thus forwarding it to the robot that first broadcasted its position. If the third situation is verified, then the recently excluded robot is included in the subgroup of the excluded robots that replied to its message. However, this subgroup may not contain all excluded robots. Therefore, the behaviour of excluded robots will follow this cycle since they are unable to identify if all other excluded robots are “connected” – they always randomly wander, considering the position of the other nearby excluded robots, while broadcasting their current position.

The opposite situation also needs to be considered - when a socially active subgroup fulfils the necessary conditions to be rewarded with a robot or spawn a new subgroup (Table 4.1), then it broadcasts its award. Similarly as before, three situations may occur:

- i)* Any robot ear the message and robots within the socially active subgroup continue their mission while broadcasting their award;
- ii)* If a robot from an active subgroup receives the message, it will broadcast it.
- iii)* If a robot from the socially excluded subgroup receives the message, it will communicate it with the other connected excluded robots;

If the second situation is verified and an excluded robot receives the forwarded message, then the third situation occurs. Therefore, depending on the active subgroup reward, the best performing robot from this socially excluded subgroup will be added to the active subgroup or the best N_l robots

from this socially excluded subgroup will form a new subgroup (note that this will only happen if the number of robots within this socially excluded subgroup is equal or superior to N_I). It is noteworthy that the best performing robots from this socially excluded subgroup may not be the best performing robots from all socially excluded robots since this subgroup may not encompass all excluded robots. Also, if a subgroup is unable to be rewarded with a new robot before being punished, then both reward and punishment are cancelled out.

4.6 RDPSO Outline

Having presented all the *RDPSO* base mechanisms in the previous sections, let us now outline those principles by providing some illustrative examples and by formalizing the *RDPSO* algorithm.

4.6.1 Low Level Control Architecture

To start with, let us present how the *RDPSO* algorithm can be used on real robotic platforms. Although the *RDPSO* handles robots' dynamics using fractional calculus, all the computation is carried out by considering robots as particles, thus ignoring the hardware of the real platforms. In other words, the output is a position vector for each robot (*cf.* equations (4.6)). Depending on the robot kinematical and dynamical characteristics, this new position may be achieved from different ways. For instance, a holonomic robot (*e.g.*, omnidirectional drive system), which has the kinematic advantage of allowing continuous translation and rotation in any direction, can move to a desired position regardless of its orientation. However, a non-holonomic robot (*e.g.*, differential drive system) can first change its orientation to be aligned with the target and then move forward to the desired position (*i.e.*, *Brownian motion*).

In order to achieve a higher level of hardware abstraction, a *Low-Level Control (LLC)* was introduced in the control architecture (Antonelli, Arrichiello, & Chiaverini, 2010). In this work, and since all robots are non-holonomic, the *LLC* was designed for the specific structure of the differential-drive robot to make it turn and then follow the desired position vector received by the *RDPSO* (Figure 4.5).

As shown in Figure 4.5, the output of the *RDPSO* is given as a reference value to the *LLC* that considers the kinematical and dynamical characteristics of the robot, thus defining the commands to the actuators, *e.g.*, number of pulses for *DC* motors equipped with encoders.

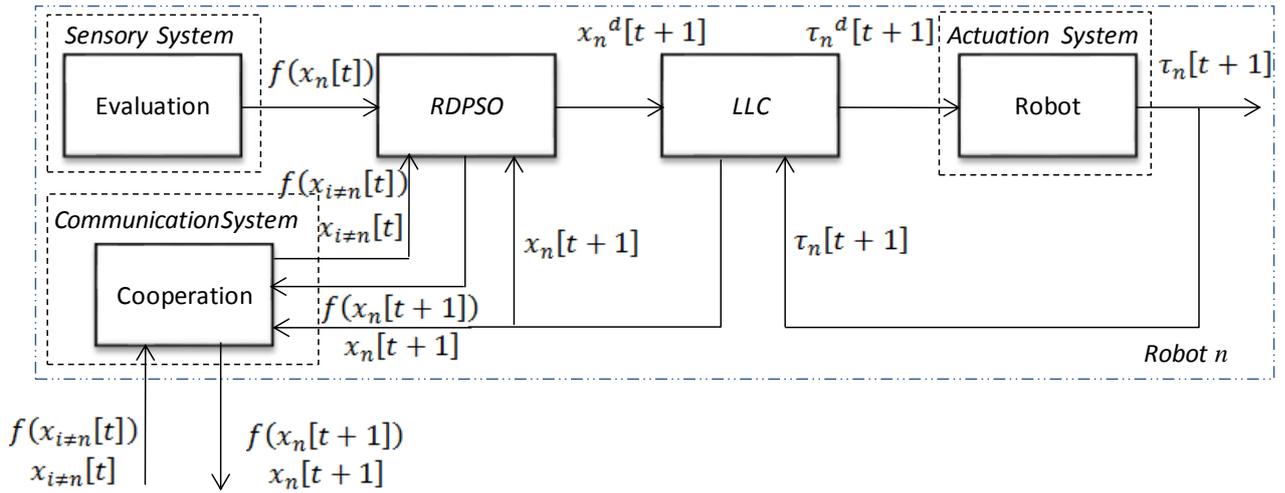


Figure 4.5. Control architecture of the *RDPSO* with *LLC* of a robot n .

This two-level control loop organization allows using the *RDPSO* with different types of robotic systems, neglecting their kinematical and dynamical characteristics. These aspects are considered by the *LLC* that needs to be properly designed for a specific robotic system. In other words, if the hardware changes, only the block *LLC* will need to be replaced.

To our specific situation, *i.e.*, differential-drive robots, the *LLC* receives the desired position $x_n^d[t+1]$, which corresponds to a cartesian position $[x_n^{d1}[t+1] \ x_n^{d2}[t+1]]^T$, and computes the inverse kinematic model based on the following equations:

$$h_n[t+1] = \sqrt{(v_n^1[t+1])^2 + (v_n^2[t+1])^2}, \quad (4.18)$$

$$\theta_n^d[t+1] = \text{atan2}(v_n^2[t+1], v_n^1[t+1]), \quad (4.19)$$

$$\theta_n[t+1] = \theta_n^d[t+1] - \theta_n^d[t], \quad (4.20)$$

wherein $v_n^1[t+1]$ and $v_n^2[t+1]$ are the elements of the vector $v_n[t+1]$ from equation (4.5). The function *atan2* in (4.19) is a variant of the trigonometric arctangent function, but accounts for the quadrant in which $\theta_n^d[t+1]$ lies. Note that, $\theta_n^d[t]$ corresponds to the previous computation of (4.19) and may be considered zero in the first iteration (*i.e.*, initial orientation of zero degrees). The output of the inverse kinematic model is represented by the rotation $\theta_n[t+1]$ that the robot needs to perform to be aligned with the desired position and the distance $h_n[t+1]$ it needs to travel to reach it.

The rotation $\tau_n^{d1}[t+1]$ and the forward movement $\tau_n^{d2}[t+1]$ of the differential-drive robot are defined by:

$$\tau_n^{d1}[t+1] = \tau_{rev} \cdot \frac{\theta_n[t+1]}{2\pi} \cdot \frac{R_{robot}}{R_{wheel}}, \quad (4.21)$$

$$\tau_n^{d2}[t+1] = \tau_{rev} \cdot \frac{h_n[t+1]}{2\pi} \cdot \frac{1}{R_{wheel}}, \quad (4.22)$$

wherein τ_{rev} is the total number of steps or pulses per revolution. The radius of the robot and the wheels are defined by R_{robot} and R_{wheel} , respectively. In order to improve the time response of the robot and the smoothness of its movement, a rotational threshold θ_T was introduced. Rotations $\theta_n[t+1]$ inferior to θ_T are then ignored and only the forward distance $h_n[t+1]$ is considered. Bearing in mind this assumption, and since a possible loss of steps or pulses may occur while executing the commands, *i.e.*, $\tau_n^1[t+1] \neq \tau_n^{d1}[t+1]$ or $\tau_n^2[t+1] \neq \tau_n^{d2}[t+1]$, a new real position is then recalculated and considered as the current position $x_n[t+1]$ of the robot. This new position and the corresponding value of the objective function $f(x_n[t+1])$ defined in this position (*i.e.*, sensed by the sensory system) needs to be shared between robots (*cf.*, *Communication System* in Figure 4.5) so that cooperation can emerge. To that end, this information is sent directly to the robots in the neighbourhood (one-hop nodes) and relayed to other robots based on a multi-hop ad hoc networking paradigm (section 4.4). The details about this process and how to optimize it will be introduced in chapter 6.

4.6.2 Attractive/Repulsive Components

Let us represent each *RDPSO* component from equation (4.5), by considering the geometrical illustration depicted in Figure 4.6.

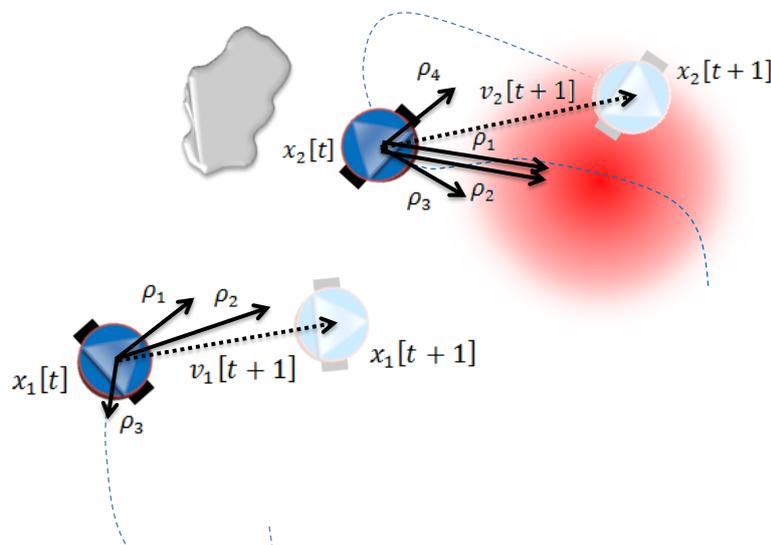


Figure 4.6. Geometrical Illustration of the *RDPSO* using a subgroup of two robots.

Considering a subgroup of two robots as Figure 4.6 depicts, the cognitive coefficient ρ_1 influences robots to improve their own individual solutions. In the case of robot 2, both cognitive ρ_1 and social coefficients ρ_2 influence toward the same position since it is the best performing robot of the subgroup. On the other hand, the social coefficient ρ_2 of robot 1 attracts it to the global best position found so far by robot 2. The obstacle susceptibility weight ρ_3 influences robots to move to a previous position in which obstacles were not detected within robot's range. As for the enforcing communication component ρ_4 , since robot 2 was the first to choose robot 1 as its nearest neighbour, being at a distance inferior to d_{max} , it is slightly repelled by it. It is noteworthy that the fractional coefficient α influences the next position of both robots $x_n[t + 1]$, $n = 1, 2$, with an inertial factor that considers their trajectory.

4.6.3 Stagnation Avoidance

To further understand *RDPSO* social dynamics, let us suppose a population divided into 3 subgroups of 3 robots each as Figure 4.7a depicts. If subgroup 1 and 2 (red and green robots, respectively) cannot improve their objective for SC_{max} iterations, they are punished by excluding the worst performing robot of each subgroup and adding them to the socially excluded subgroup (Figure 4.7b).

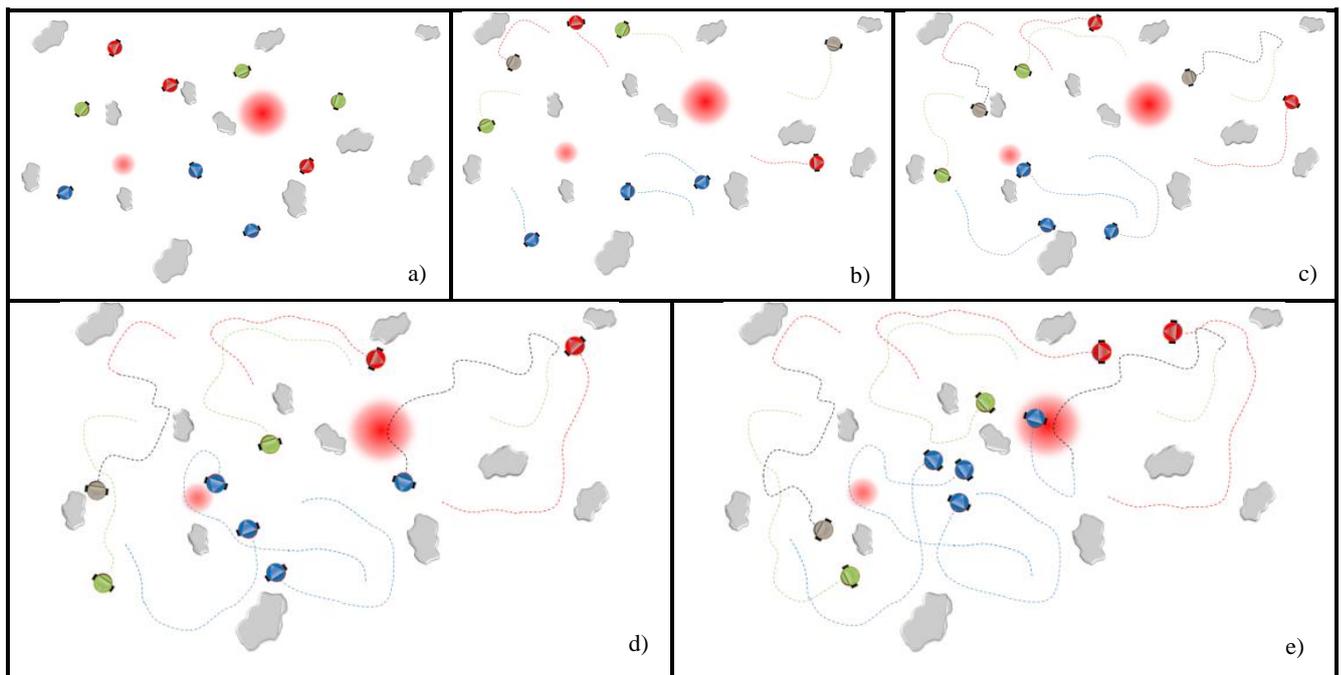


Figure 4.7. Sequence of a *MRS* exploration using the *RDPSO* algorithm (Couceiro, Rocha, & Ferreira, 2013a).

The socially excluded robots randomly wander in the scenario memorizing their individual best solution and the global best solution of the socially excluded group (Figure 4.7c). Subgroup 3 improves its solution, since it finds a local optimum, and it is rewarded with the best performing robot in the socially excluded group (Figure 4.7d). Finally, the new member of subgroup 3 communicates its best individual solution to the other members which is better than their best global solution inducing them to move toward this new solution (Figure 4.7e).

4.6.4 Algorithm

At least, and to complete the description of the *RDPSO* base mechanisms, let us summarize it with the following algorithm (Algorithm 4.4).

As a first evaluation of the scientific content provided in this Thesis, the next section provides experimental results retrieved from both numerical simulations and laboratorial experiments with physical mobile robots.

4.7 Experimental Results

To demonstrate the mechanisms of the herein proposed *RDPSO* swarm robotic distributed algorithm, a set of experimental results with multiple numerical simulated and real robots is presented.

4.7.1 Numerical Simulations

This section carries out a preliminary statistical analysis of the previously proposed algorithm in order to evaluate the relationship between the two vital variables within swarm systems with explicit communication: the population of robots and communication constraints (Mohan & Ponnambalam, 2009). In this section, the use of virtual agents in a numerical context instead of realistic robots (*i.e.*, without considering robots' dynamic and radio frequency propagation) was necessary to evaluate the *RDPSO* using statistically significant samples.

Robots were randomly deployed in the search space of 300×300 meters (area of $A = 90000 \text{ m}^2$) with obstacles randomly deployed at each trial (Figure 4.8). A *Gaussian* cost function $F(x, y)$ was defined where x and y -axis represent the planar coordinates in meters (Molga & Smutnicki, 2005).

```

Wait for information about initial pose  $\langle x_n[0], \varphi_n[0] \rangle$  and swarmID
Loop:
  If swarmID  $\neq 0$  // it is not an excluded robot
    Evaluate its individual solution  $h_n[t]$ 
    If  $h(x_n[t]) > h_{best}$  // robot has improved
       $h_{best} = h(x_n[t])$  // Section 2
       $\chi_1[t] = x_n[t]$ 
    Exchange information with teammates about the individual solution  $h_n[t]$  and current position  $x_n[t]$ 
    Build a vector  $H[t]$  containing the individual solution of all robots within swarmID
    If  $\max H[t] > H_{best}$  // subgroup has improved
       $H_{best} = \max H[t]$  // Section 2
       $\chi_2[t] = x_n[t]$ 
      If  $SC_s > 0$ 
         $SC_s = SC_s - 1$  // stagnancy counter
      If  $SC_s = 0$  // the subgroup can be rewarded
        If  $N_s < N_{max}$  and  $\text{rand}() \frac{1}{N_s^{kill+1}} > \text{rand}()$  // small probability of calling a new robot
          Broadcast the need of a new robot to any available excluded robot // Table 1
          If  $N_s^{kill} > 0$ 
             $N_s^{kill} = N_s^{kill} - 1$  // excluded robots counter
        If  $\text{rand}() \frac{N_s}{N_{max}} > \text{rand}()$  // small probability of creating a new subgroup
          Broadcast the possibility of creating a new subgroup to any available excluded robot // Table 1
          If  $N_s^{kill} > 0$ 
             $N_s^{kill} = N_s^{kill} - 1$  // excluded robots counter
      Else // subgroup has not improved
         $SC_s = SC_s + 1$  // stagnancy counter
        If  $SC_s = SC_{max}$  // punish subgroup
          If  $N_s > N_{min}$  // it is possible to exclude the worst performing robot
             $N_s^{kill} = N_s^{kill} + 1$  // excluded robots counter
             $SC_s = SC_{max} \left[ 1 - \frac{1}{N_s^{kill+1}} \right]$  // reset search counter
            If  $h_{best} = \min H[t]$  // this is the worst performing robot
               $swarmID = 0$  // exclude this robot
            Else // delete the entire subgroup
               $swarmID = 0$  // exclude this robot
        If  $g(x_n[t]) \geq g_{best}$  // maximize distance to obstacles
           $g_{best} = g(x_n[t])$  // Section 2.2
           $\chi_3[t] = x_n[t]$ 
         $[L_n, index_n] = \text{sort\_ascending}(L_{n,1:N_s})$  // sort the elements of line  $n$  from link matrix  $L$  in ascending order
        For  $i = 1: N_s$ 
          If  $index_n(i)$  has not yet chosen it as its nearest neighbour
             $\chi_4[t] = x_i[t] + d_{max} \frac{x_i[t] - x_n[t]}{\|x_i[t] - x_n[t]\|}$  // the position of the nearest neighbour increased by  $d_{max}$  toward  $x_n[t]$ 
            Communicate to robot  $i$  that it was chosen by robot  $n$  // Section 2.3
            break from For
         $v_n[t + 1] = w_n[t] + \sum_{i=1}^4 \rho_i r_i (\chi_i[t] - x_n[t])$  // equation 1
         $x_n[t + 1] = x_n[t] + v_n[t + 1]$  // equation 2
      Else // it is an excluded robot
        Wandering algorithm // e.g., (Bräunl, 2008)
        Evaluate its individual solution  $h_n[t]$ 
        If  $h(x_n[t]) > h_{best}$  // robot has improved
           $h_{best} = h(x_n[t])$ 
        Exchange information with teammates about the individual solution  $h_n[t]$  and current position  $x_n[t]$ 
        Build a vector  $H[t]$  containing the individual solution of all  $N_x$  robots within the excluded subgroup ( $swarmID = 0$ )
        If  $\max H[t] > H_{best}$ 
           $H_{best} = \max H[t]$ 
          If  $h_{best} = \max_{N_I} H[t]$  // this is one of the best  $N_I$  performing robot of the excluded subgroup
            If  $N_x \geq N_I$  and  $\text{rand}() \frac{N_x}{N_T} > \text{rand}()$  // small probability of creating a new subgroup
               $swarmID = swarmID\_new$  // include this robot in the new active subgroup
              Broadcast the need of  $N_I - 1$  robots to any available excluded robot // Table 1
            Else
              If receives information about the need of a new robot
                 $swarmID = swarmID\_received$  // include this robot in the active subgroup
                 $N_s = N_s + 1$ 
                Exchange information with teammates about  $N_s$ 
              If receives information about the need of creating a new subgroup
                 $swarmID = swarmID\_new$  // include this robot in a new active subgroup
                 $N_s = N_I$  // reset number of robots in the subgroup
                 $N_s^{kill} = 0$  // reset number of excluded robots
                 $SC_s = 0$  // reset search counter
  until stopping criteria (convergence/time)

```

Algorithm 4.4. RDPSO Algorithm

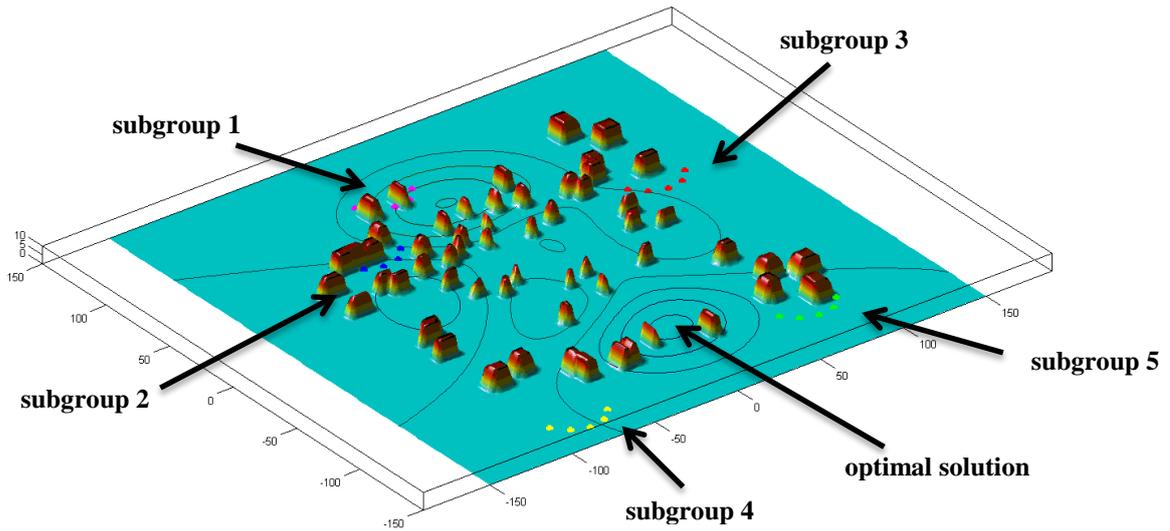


Figure 4.8. Virtual scenario with obstacles and robots divided into 5 subgroups.

In order to improve the interpretation of the algorithm performance, results were normalized in a way that the objective of robotic teams was to maximize $f(x, y)$, *i.e.*, minimize the original benchmark functions $F(x, y)$, thus finding the optimal solution of $f(x, y) = 1$, while avoiding obstacles and ensuring the *MANET* connectivity:

$$f(x, y) = \frac{F(x, y) - \max F(x, y)}{\min F(x, y) - \max F(x, y)}. \quad (4.23)$$

Since the *RDPSO* is a stochastic algorithm, every time it is executed it may lead to a different trajectory convergence. Therefore, multiple test groups of 100 trials of 300 iterations each were considered. Independently of the population of robots, it will be used a minimum, initial and maximum number of $N_s^{min} = 0$ (*i.e.*, all robots socially excluded), $N_s^I = 3$ and $N_s^{max} = 6$ socially active subgroups (represented by different colours in Figure 4.8), respectively. The maximum travelled distance between iterations was set as 0.5 meter, *i.e.*, $\Delta x_{max} = \max \|x_n[t + 1] - x_n[t]\| = 0.5$. Thus, robots moved in the 300×300 meters environment where their solution depended on the intensity of the *Gaussian* function at each (x, y) position. Note that while robots moved they needed to consider all the components within the *RDPSO* algorithm from the *DE* in (4.5)-(4.6). Although those preliminaries experiments did not consider realistic robot's dynamics, they still considered the fractional order convergence from section 4.2. Moreover, they needed to avoid obstacles according to section 4.3 and maintain the communication with teammates from the same subgroup according to section 4.4.

Regarding this last point, it is important to note that trying to maintain the network connectivity by only taking into account the communication range does not match reality since the propagation model is more complex – the signal depends not only on the distance but also on the multiple paths

from walls and other obstacles. However, in simulation, the communication distance is a good approach and it is easier to implement. Therefore, for the sake of simplicity and without the lack of generality, the maximum communication range was considered at this point. The maximum communication distance d_{max} will then vary depending on the chosen wireless protocol. Four conditions were described: *i*) Existence of a communication infrastructure (*i.e.*, without communication constraints $\equiv d_{max} \rightarrow \infty$); *ii*) *WiFi*; *iii*) *ZigBee*; and *iv*) *Bluetooth*. Table 4.2 depicts the maximum communication distance adapted from a comparison between the key characteristics of each wireless protocol in (Lee, Su, & Shen, 2007). The mean between the minimum and maximum range shown in (Lee, Su, & Shen, 2007) was considered as the maximum communication distance $d_{max} = \{10,55,100, \infty\}$.

Table 4.2. Typical maximum communication distances of the *WiFi*, *ZigBee* and *Bluetooth*.

	<i>No Limit</i>	<i>WiFi</i>	<i>ZigBee</i>	<i>Bluetooth</i>
d_{max} [m]	∞	100	55	10

The number of robots in the swarm varied from 3 to 33 robots with incremental steps of 6 robots, *i.e.*, $N_T = \{3,9,15,21,27,33\}$, in order to understand the performance of the algorithm while changing the population size and the maximum communication distance.

Table 4.3 summarizes the whole *RDPSO* configuration. Note that, so far, we do not hold any sort of knowledge regarding the *RDPSO* parameters and, as such, the parameters presented on Table 4.3, namely α and ρ_i , $i = 1,2,3,4$, were retrieved by trial-and-error based on exhaustive numerical simulations. Also note that the number of trials is considered for each different configuration, thus resulting in 2400 trials for the 24 pairwise combinations (swarm size and communication range).

Table 4.3. *RDPSO* parameters obtained by trial-and-error and used in numerical simulations.

<i>RDPSO</i> Parameter	Value
Number of trials	100
Time per trial [iterations]	300
N_T	{3,9,15,21,27,33}
N_s^{min}	0
N_s^I	3
N_s^{max}	6
SC_{max}	30
d_{max} [m]	{10,55,100, ∞ }
Δx_{max} [m]	0.5
α	0.5
ρ_1	0.2
ρ_2	0.4
ρ_3	0.8
ρ_4	0.8

Since these simulation experiments represent a search task, it is necessary to evaluate not only the completeness of the mission but also the speed. Therefore, the performance of the algorithm was evaluated through the analysis of the final global solution of the population and the runtime of the simulation. If the swarm could not find the optimal solution, the runtime was considered to be the simulation time (*i.e.*, 300 iterations).

The significance of the maximum communication distance and the number of robots (independent variables) on the global solution and the runtime (dependent variables) was analysed using a two-way Multivariate Analysis of Variance Analysis (*MANOVA*) after checking the assumptions of multivariate normality and homogeneity of variance/covariance. The assumption of normality of each of the univariate dependent variables was examined using univariate tests of *Kolmogorov-Smirnov* ($p\text{-value} < 0.05$). Although the univariate normality of each dependent variable was not verified, since the number of trials was over 30 (100), based on the *Central Limit Theorem (CLT)* (Pedrosa & Gama, 2004), the assumption of multivariate normality was validated (Pestana & Gageiro, 2008; Maroco, 2010). The assumption about the homogeneity of variance/covariance matrix in each group was examined with the *Box's M Test* ($M = 6465.13$, $F(69; 5368369.62) = 92.98$; $p\text{-value} = 0.001$). Although the homogeneity of variance/covariance matrices was not verified, the *MANOVA* technique is robust to this violation because all the samples have the same size (Maroco, 2010). The classification of the effect size, *i.e.*, measure of the proportion of the total variation in the dependent variable explained by the independent variable, was done according to Maroco (Maroco, 2010). This analysis was performed using *IBM SPSS Statistics* for a significance level of 5%.

The *MANOVA* revealed that the maximum communication distance had a small effect and significant on the multivariate composite (*Pillai's Trace* = 0.75; $F(6; 4752) = 30.974$; $p\text{-value} = 0.001$; *Partial Eta Squared* $\eta_p^2 = 0.038$; *Power* = 1.0). The number of robots also had a small effect and significant on the multivariate composite (*Pillai's Trace* = 0.080; $F(10; 4752) = 19.706$; $p\text{-value} = 0.001$; $\eta_p^2 = 0.04$; *Power* = 1.0). Finally, the interaction between the two independent variables had a small statistically significant effect on the multivariate composite (*Pillai's Trace* = 0.032; $F(30; 4752) = 2.55$; $p\text{-value} = 0.001$; $\eta_p^2 = 0.016$; *Power* = 1.0).

After observing the multivariate significance in the maximum communication distance and the number of robots, a univariate *ANOVA* for each dependent variable followed by the *Tukey's HSD Test* was carried out. For the maximum communication distance, the dependent variable final global solution presented statistically significant differences ($F(3, 2376) = 45.185$; $p\text{-value} = 0.001$; $\eta_p^2 = 0.054$; *Power* = 1.0) and the dependent variable runtime presented statistically significant differences ($F(3, 2376) = 53.683$; $p\text{-value} = 0.001$; $\eta_p^2 = 0.063$; *Power* = 1.0). For the number of robots, the dependent variable final global solution also presented statistically significant differences ($F(5, 2376)$

= 23.347; p -value = 0.001; $\eta_p^2 = 0.047$; $Power = 1.0$) and also the dependent variable runtime showed statistically significant differences ($F(5, 2376) = 39.816$; p -value = 0.001; $\eta_p^2 = 0.077$, $Power = 1.0$).

Using the *Tukey's HSD Post Hoc*, it was possible to verify where the differences between maximum distances of communication lied. Analysing the swarm's final solution and the runtime variables, it appears that there were statistically significant differences between experiments without communication constraints and experiments using the *WiFi* protocol, the *ZigBee* protocol and the *Bluetooth* protocol (Table 4.4).

Table 4.4. Tukey's *HSD Post Hoc Test* to the maximum communication distance d_{max} .

d_{max}	Final Solution	Runtime
<i>No Limit vs WiFi</i>	0.002*	0.854
<i>No Limit vs ZigBee</i>	0.001*	0.001*
<i>No Limit vs Bluetooth</i>	0.001*	0.001*
<i>WiFi vs ZigBee</i>	0.207	0.019*
<i>WiFi vs Bluetooth</i>	0.001*	0.001*
<i>ZigBee vs Bluetooth</i>	0.001*	0.001*

* The corresponding p -value for mean difference when it is significant at the 0.05 level

Table 4.5. Tukey's *HSD Post Hoc Test* to the total number of robots N_T .

N	Final Solution	Runtime
<i>3vs9</i>	1.000	0.861
<i>3vs15</i>	0.151	0.182
<i>3vs21</i>	0.001*	0.001*
<i>3vs27</i>	0.001*	0.001*
<i>3vs33</i>	0.001*	0.001*
<i>9vs15</i>	0.249	0.844
<i>9vs21</i>	0.001*	0.001*
<i>9vs27</i>	0.001*	0.001*
<i>9vs33</i>	0.001*	0.001*
<i>15vs21</i>	0.004*	0.001*
<i>15vs27</i>	0.001*	0.001*
<i>15vs33</i>	0.001*	0.001*
<i>21vs27</i>	0.842	0.654
<i>21vs33</i>	0.785	0.076
<i>27vs33</i>	1.000	0.845

* The corresponding p -value for mean difference when it is significant at the 0.05 level

It is noteworthy that the algorithm produces better solutions without communication constraints. Also, using *WiFi* protocol produces better solutions than using the *ZigBee* protocol and, on the other hand, this last one produces better solutions than the *Bluetooth* protocol as expected. In fact, using the *Bluetooth* protocol proves to be the worst communication protocol to employ.

Analysing both the final global solution of the team and the runtime variables, it appears that there were statistically significant differences between a population inferior to 15 robots and a population superior to 21 robots, not showing statistically significant differences for a population between 3 to 15 robots and 21 to 33 robots (Table 4.5). Note that the worst result was obtained using 3 robots, which cannot be considered significantly worse than using 9 or even 15 robots. This may be relevant since the increase in the number of robots result in an increase in the cost of the solution.

To strengthen the conclusions from Table 4.4 and Table 4.5, Figure 4.9 and Figure 4.10 depict the estimated marginal means for both final global solution and runtime, respectively. These figures illustrates how the performance of the *RDPSO* is affected under pairwise combinations between the swarm population and the communication technology.

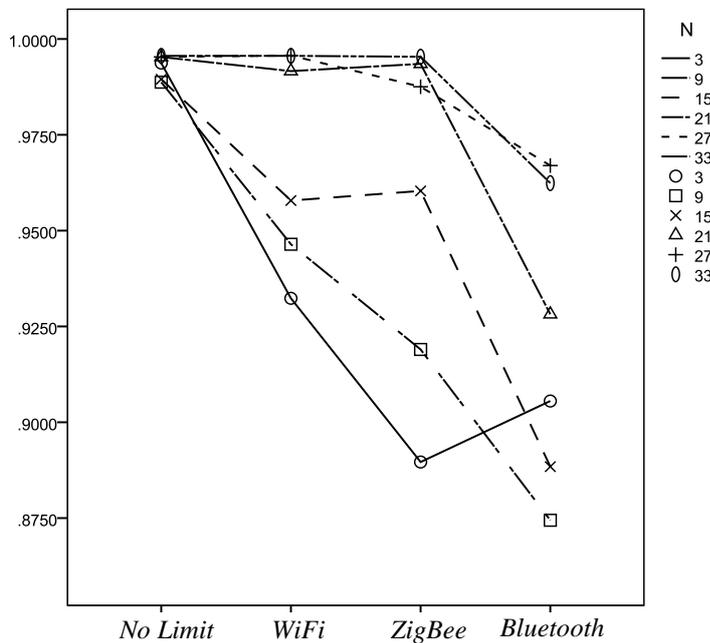


Figure 4.9. Estimated marginal means of the final global solution.

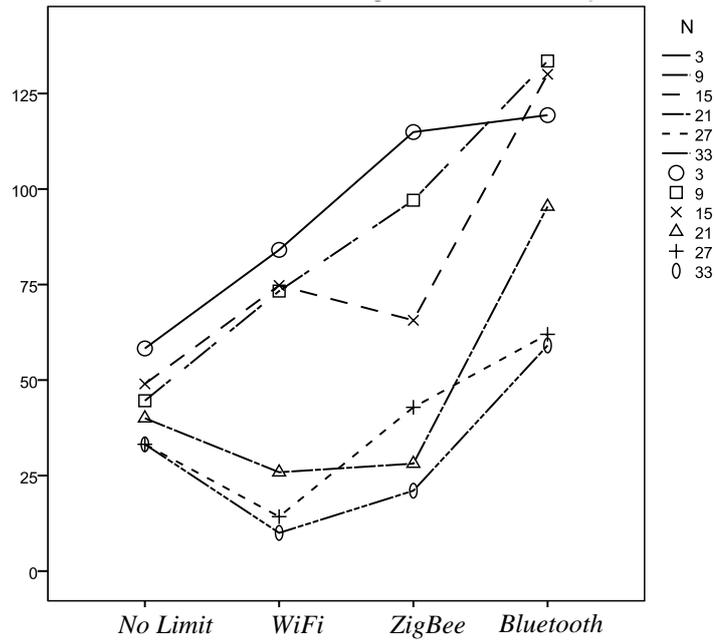


Figure 4.10. Estimated marginal means of the runtime (number of iterations).

A video of these numerical experiments is provided to have a general overview of the *RDPSO* dynamics²⁶.

Having studied the *RDPSO* main mechanisms in theory and numerically evaluated it, let us present experiments carried out with physical mobile robots.

4.7.2 Real-World Experiments

In this section, it is explored the effectiveness of using the *RDPSO* on swarms of real robots, while performing a collective foraging task with local and global information under communication constraints. Multiple test groups of 20 trials of 180 seconds each were considered.

The *eSwarBot* (Educational Swarm Robot) was the platform used to evaluate the algorithm (*cf.*, section 3.2.1). Its low cost and high energy autonomy allowed to perform experiments with up to 12 robots, with $N_T = \{4, 8, 12\}$. The *RGB-LEDs* on top of the *eSwarBots* were used to identify their subgroup using different colours. Independently from the population of robots, it was used a minimum, initial and maximum number of $N_s^{min} = 1$, $N_s^I = 3$ and $N_s^{max} = 6$ socially active subgroups (represented by different colours in Figure 4.11a), respectively. The maximum travelled distance between iterations was set to 0.1 meters, *i.e.*, $\Delta x_{max} = 0.1$.

²⁶ http://www2.isr.uc.pt/~michaelcouceiro/media/RDPSO_numerical.wmv

All of the experiments were carried out in a 2.55×2.45 meters scenario ($A \approx 6.25 \text{ m}^2$). The experimental environment (Figure 4.11a) was an enclosed arena containing two sites represented by illuminated spots uniquely identifiable by controlling the brightness of the light. Despite being an obstacle-free scenario, the robots themselves act as dynamic obstacles – a maximum number of 12 robots correspond to a population density of approximately 2 robots.m^{-2} . As previously mentioned in section 3.2.1, each *eSwarBot* possesses overhead light sensors (*LDR*) that allows it finding candidate sites and measuring their quality. The brighter site (optimal solution) was considered better than the dimmer one (sub-optimal solution), and so the goal of the swarm robots was to collectively choose the brighter site. The intensity values $F(x, y)$ represented in Figure 4.11b were obtained sweeping the whole scenario with a single robot in which the light sensor was connected to a 10-bit analog input. The maximum intensity values obtained (hot colour in Figure 4.11b) was found between 800 and 860.

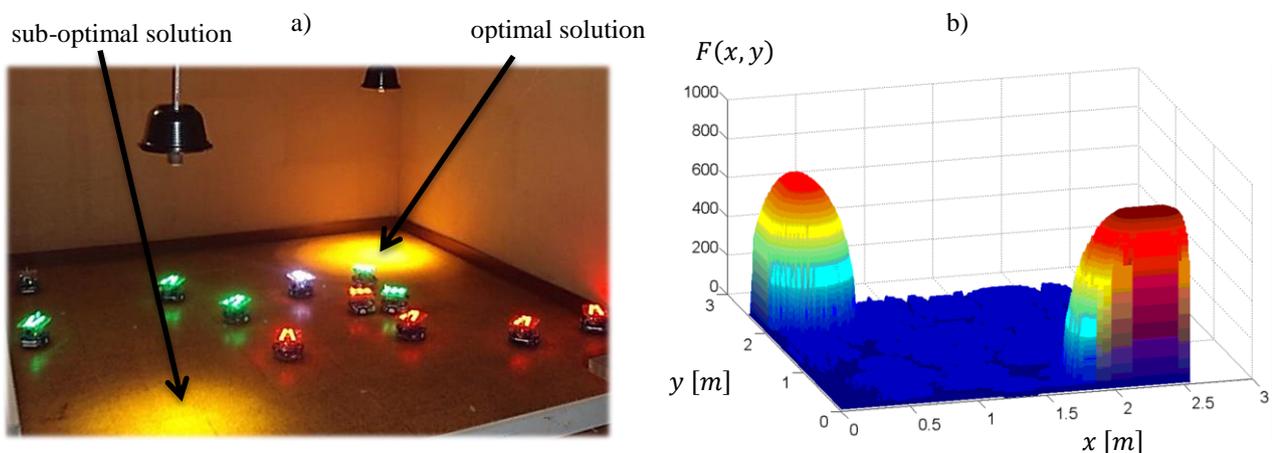


Figure 4.11. Experimental setup. a) Enclosed arena with 2 subgroups (different colours); b) Virtual representation of the target distribution.

Inter-robot communication to share positions and local solutions was carried out using *ZigBee* 802.15.4 wireless protocol. Since robots were equipped with *XBee* modules allowing a maximum communication range larger than the whole scenario, robots were provided with a list of their teammates' address in order to simulate the *MANET* communication with limited range. The maximum communication distance between robots d_{max} was varied between 0.5 meters and 1.5 meters. At each trial, robots were manually deployed on the scenario while preserving the maximum communication distance d_{max} .

Table 4.6 summarizes the whole *RDPSO* configuration. Note that, once again, we still do not hold any sort of knowledge regarding the *RDPSO* parameters and, as such, the parameters presented on Table 4.6 were the same previously retrieved by trial-and-error (as in previous section).

Table 4.6. *RDPSO* parameters obtained by trial-and-error and used in its first evaluation with physical robots.

<i>RDPSO</i> Parameter	Value
Number of trials	20
Time per trial [seconds]	180
N_T	{4,8,12}
N_s^{min}	0
N_s^I	2
N_s^{max}	3
SC_{max}	30
d_{max} [m]	{0.5,1.5}
Δx_{max} [m]	0.1
α	0.5
ρ_1	0.2
ρ_2	0.4
ρ_3	0.8
ρ_4	0.8

The previously described conditions give a total of 120 experimental trials, thus leading to a runtime of 6 hours. The sequence of frames in Figure 4.12 presents a trial of the team's performance with $N = 12$ and $d_{max} = 1.50$ meters.

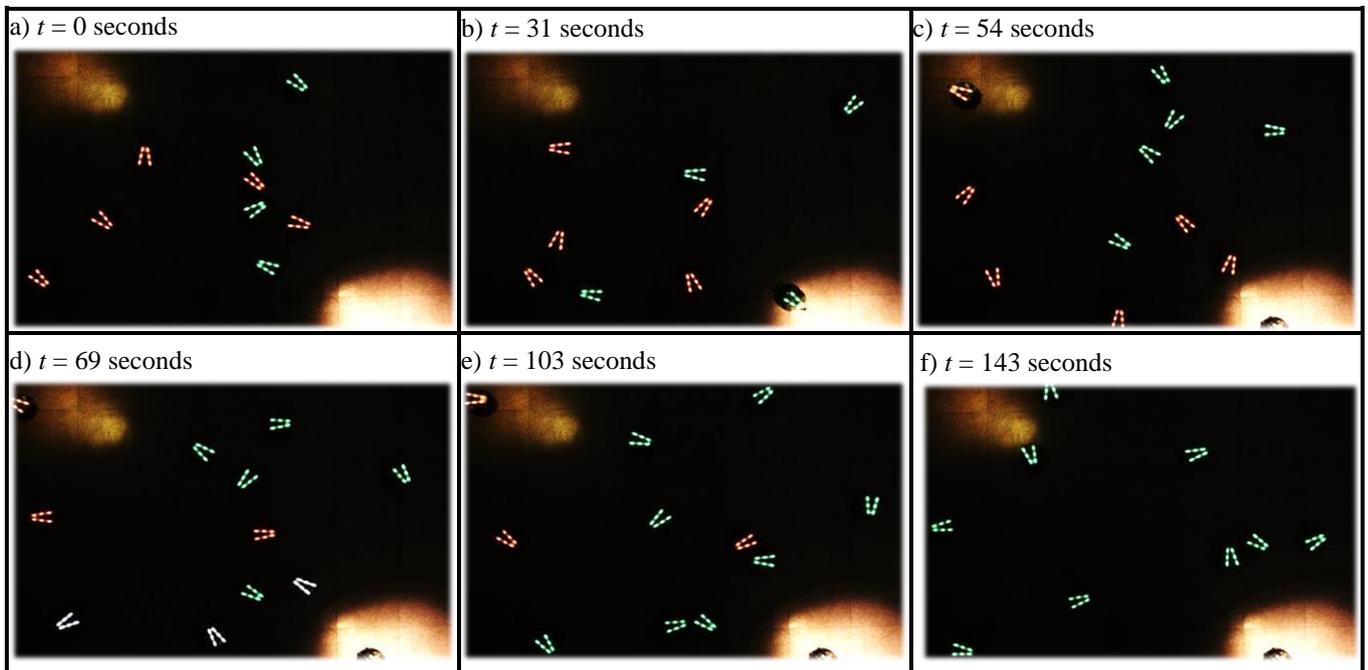


Figure 4.12. Frame sequence showing the *RDPSO* performance on a population of 12 robots (some robots may be outside camera's range). a) The population is initially divided into two swarms – green and red – deployed in a spiral manner; b) The swarms independently search for the brighter site taking into account a maximum communication distance of 1.5 meters between robots of the same swarm; c) One robot from the red and green swarm finds the sub-optimal and optimal solution, respectively; d) As the red swarm does not improve, some robots are excluded, thus being added to the socially excluded subgroup (white swarm); e) Since the green swarm has improved, it is able to call new members from the socially excluded subgroup; f) Finally, the green swarm proliferates calling all the previously excluded robots that were unable to improve their solution. Note that robots do not all converge the optimal solution as they try to maintain a distance of d_{max} between them.

Once again, let us evaluate both the completeness of the mission and the time needed to complete it. Figure 4.13 depicts the convergence of the *RDPSO* for the several proposed conditions. The median of the best solution in the 20 experiments was taken as the final output in the set $N = \{4, 8, 12\}$ for each d_{max} .

Analysing Figure 4.13, it is clear that the proposed mission could be accomplished by any number of robots between 4 and 12. In fact, independently on the number of robots, the swarm converged to the solution in approximately 90% of the experiments. The charts also show that increasing the number of robots slightly increased the runtime needed to accomplish the mission. A population of 4, 8 and 12 robots took an average of 77, 106 and 112 seconds to converge to the optimal solution, respectively. This is a consequence of having more robots inside the same arena – the number of dynamic obstacles is higher. As expected, increasing the maximum communication distance generally resulted in a faster convergence to the optimal solution. However, this relationship was found to be not linear and varied depending on the number of robots in the population. For instance, for a swarm population of 12 robots, the performance slightly decreased for a communication distance of $d_{max} = 1.5$ meters. This is an expected factor since the communication interference between robots increases with the population, and, as a consequence, the existence of repulsive forces to maintain ideal inter-robot distances of 1.5 meters constrains robots' motion ($\chi_4[t]$ from equation (4.5)). Adding this to the fact that the scenario was considerably small, yielded a considerably large amount of constraints, thus jeopardizing the success of the mission.

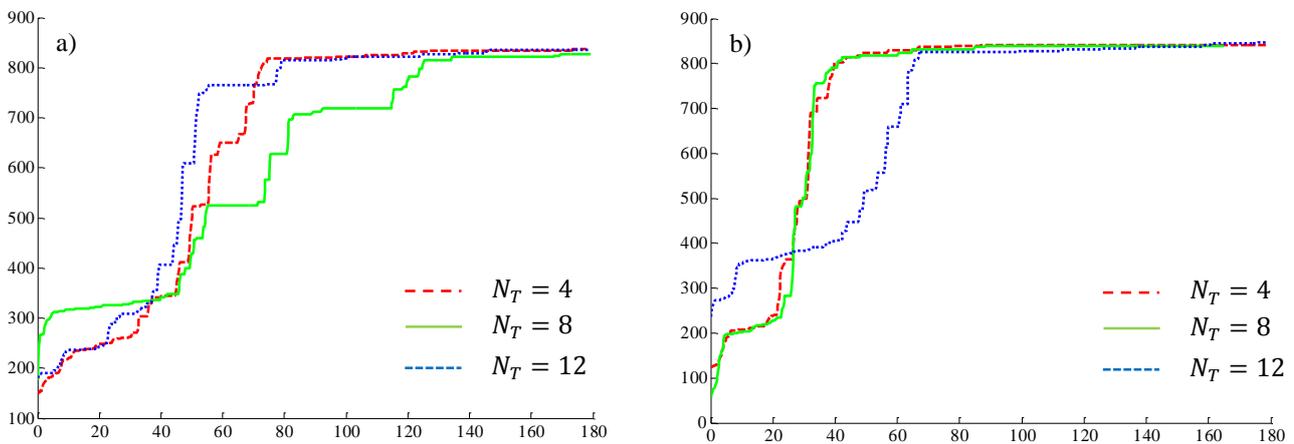


Figure 4.13. Performance of the algorithms changing the number of robots N in the population: a) $d_{max} = 0.5$ meters; b) $d_{max} = 1.5$ meters.

Another important factor is that some robots of a given swarm were unable to converge to the final solution when one robot of the same swarm found it. This issue was related with odometry

limitations of the platforms which resulted in the accumulation of positioning errors. The use of encoders such as the ones used in these robots is a classical method, being of low-cost and simple use. However, this yielded to the existence of cumulative errors inherent to their use, which makes it difficult for the robots to complete the proposed odometry objectives accurately.

A video of the real experiments is provided to better understand how real robots behave under the *RDPSO* algorithm²⁷.

4.8 Discussion

This chapter described the leading development of the *Robotic Darwinian Particle Swarm Optimization (RDPSO)* that integrates insights from the traditional optimization method *PSO*, real-world *MRS* features and a natural selection mechanism. This last feature is achieved by creating some *social exclusion and inclusion rules* to manage the whole swarm into clusters, herein denoted as subgroups. The dependency between subgroups gives rise to a *competitive evolutionary process* mimicking the animal nature as following the *Darwinian survival-of-the-fittest* principle. On the other hand, as many other biological societies involved in diverse survival conditions, the outcome of this competitive evolutionary process is reflected into social cooperation among the members from the same group. This is a highly recurrent process in nature denoted as *coopetition* (Tsai, 2002). For instance, certain birds are unable to reach parasites on some parts of their bodies, thus benefiting from preening one another. Hence, there is an entire flock of potential preeners which compete in hopes of establishing a beneficial cooperative relationship. To the similarity of the *RDPSO*, birds that try to be preened without preening others are excluded from these relationships as they do not compete.

The theoretical procedures were first thoroughly evaluated using numerical simulations for statistical significant purposes (section 4.7.1). As previously depicted in Table 4.4, Table 4.5, Figure 4.9 and Figure 4.10, the conditions of the independent variables, that is to say the swarm population N_T and the maximum communication range d_{max} , can be divided into different homogeneous subsets. For instance, since there are no statistically significant differences between teams of 3, 9 and 15 robots in the analysis of both the final global solution and the runtime, this can be considered as a subset of N_T , *i.e.*, $N_T^1 = \{3,9,15\}$ and $N_T^2 = \{21,27,33\}$. In other words, in an application where the cost of the solution needs to be considered, and since there are no significant advantages of having 15 robots instead of having just 3 or having 33 robots instead of 21, the choice would be using the minimum

²⁷ http://www2.isr.uc.pt/~michaelcouceiro/media/RDPSO_initial.mp4

number of robots of each subset of N_T . The same analysis can be conducted for the maximum communication distance. However, in this specific situation, three subsets can be considered analysing the statistically significant differences between the different values of d_{max} , *i.e.*, $d_{max}^1 = \{No\ Limit\}$, $d_{max}^2 = \{WiFi, ZigBee\}$ and $d_{max}^3 = \{Bluetooth\}$. Put differently, whenever one is unable to have a pre-existent infrastructure (*i.e.*, d_{max}^1), the choice between wireless technologies may be centred on the *WiFi* and *ZigBee* technologies.

The last section of experimental results was primarily introduced to explore the effectiveness of using the *RDPSO* algorithm on swarms of real robots of up to 12 *eSwarBots* (section 4.7.2). As such, those experiments were prone to robot dynamics, odometry errors, loss of communication packets, among other real-world phenomena. Despite those limitations, the results suggest that robots are able to converge to the optimal solution regardless on the swarm population. Nevertheless, its influence, as well as the influence of the maximum communication range is still evident, thus supporting the results retrieved by the numerical simulations.

4.9 Summary

A modified version of the *PSO* based on real-world *MRS* characteristics such as obstacles avoidance abilities and communication issues was developed and entitled as *Robotic Darwinian Particle Swarm Optimization (RDPSO)*.

The features presented in this chapter were first implemented in a numerical simulation environment in *MatLab* and afterwards further validated using real *eSwarBot* platforms. Experimental results show the performance of a *MRS* with a biologically inspired behaviour based on natural selection and social exclusion. As expected, the influence inherent to communication's limitations can be attenuated as the number of robots or the communication range/quality increases. This is a promising result for communities of swarm robots with many individuals since they can develop efficient coordination techniques, just like natural swarm agents, allowing cooperative and competitive work in large and super-large societies. However, the choice on the number of robots and the wireless technology needs to take into account the global cost of the solution depending on the statistic significant differences between the independent variables. Moreover, as the size of the swarm increases, more interference robots create between themselves.

Although robots were randomly deployed in the experiments considered in this chapter, the fact is that their initial pose, and respective sensed solution, suggested some influence on the final outcome. Therefore, a deeper study around the initial deployment of exploring agents in the search space needs to be conducted. Moreover, although we were able to assess the influence of the communication

range within the *RDPSO* performance, there is still a considerably large amount of work to do around communication. For instance, if one would consider a faulty environment, the strategy proposed so far would not be able to comply with the necessary requirements. For instance, the failure of a single robot would split the subgroup *MANET* and, as a consequence, robots would be unable to share information with their teammates. For that reason, next chapter resorts to the state-of-the-art so as to propose a strategy that may be applied considering *RDPSO* dynamics.

Deployment and Fault-Tolerance

ENDOWING teams of robots with fault-tolerant mechanisms and self-deploying capabilities have a particular interest on real-world applications. However, as previously stated in sections 2.2.3 and 2.2.4 where the most recent works around these subjects were discussed, one can conclude that there is still a considerably large amount of work to be done around both fault-tolerant *MANETs* and the development of realistic fault-tolerant initial deployment strategies.

To ensure *MANETs* connectivity and robustness is much more demanding than using infrastructured networks. As a result, to prolong the *MANET* lifetime and prevent loss of connectivity, fault-tolerant strategies are needed. As discussed in sections 2.2.3, a simple but efficient strategy is to control robots movement to allow significant node redundancy guaranteeing a multi-connectivity strategy. The k -connectivity or k -fault-tolerance, $k \in \mathbb{N}$, of a network means that each pair of robots is connected by at least k robot-disjoint paths. This means that, in the worst case, a k connected *MANET* requires k or more robot failures to become disconnected. Thus, multi-connectivity is favourable for both fault-tolerance and communication capacity. In *MRS* context, most researchers have focused on establishing bi-connectivity ($k = 2$) through robot movement control (Casteigts, Albert, Chaumette, Nayak, & Stojmenovic, 2010).

Bearing this idea in mind, this chapter is divided into three key contributions demonstrated through the *RDPSO* algorithm:

- i)* The initial deployment problem is formally introduced and an autonomous, realistic and fault-tolerant hierarchical strategy is proposed (section 5.1);
- ii)* The *MANET* connectivity problem in section 4.4 is further extended considering communication constraints in faulty environment, by resorting to a set of attractive and repulsive forces so as to ensure the k -connectivity of *MANETs* (section 5.2);
- iii)* Afterwards, the proposed methodologies are evaluated using both physical and virtual robots (section 5.3).

Sections 5.4 and 5.5 outline the discussion and main conclusions of this chapter.

5.1 Initial Deployment

One of the major concerns in the *RDPSO* approach is that all robots should have an initial deployment that preserves the communication between robots in each subgroup. Moreover, it is also known that, in classical *PSO*-based algorithms, particles need to be scattered throughout the scenario. As previously addressed in section 2.2.4, the deployment problem considers the number of needed robots for a specific situation (*e.g.*, objective, scenario, constraints) and their initial locations. Typically, this problem is addressed using large robotic platforms that carry a set of smaller exploring robots.

The herein proposed strategy tries to get benefit of a random planar deployment of robots, while eliminating the disadvantages inherent to it and taking into account the communication constraints. This is achieved by getting inspiration on a deployment strategy based on the *Spiral of Theodorus*, *aka.* square root spiral. This spiral is composed of contiguous right triangles with each cathetus, *aka.* leg, having a length equal to 1 (Hahn & Schoenberger, 2007). Triangle's hypotenuses h_i is given by the square root to a consecutive natural number, with $h_1 = \sqrt{2}$.

Since this approach benefits from the *spiral of Theodorus* to carry out the initial deployment of robots, two general adjustments are considered as this time:

- i)* The initial position of each robot is set at the further vertex of the centre of the spiral for each right triangle with a random orientation;
- ii)* The size of the cathetus is set as the maximum communication range d_{max} (instead of having the unit length 1) consequently changing the triangles' hypotenuses to the product between the maximum communication range and the square root of the consecutive natural number.

In real situations, the maximum communication distance d_{max} may be established considering the worst case situation (*e.g.*, urban environment with obstacles and other harsh phenomena). These assumptions make it possible to have an initial deployment of the robots in an area that depends on both the number of robots and the communication constraints.

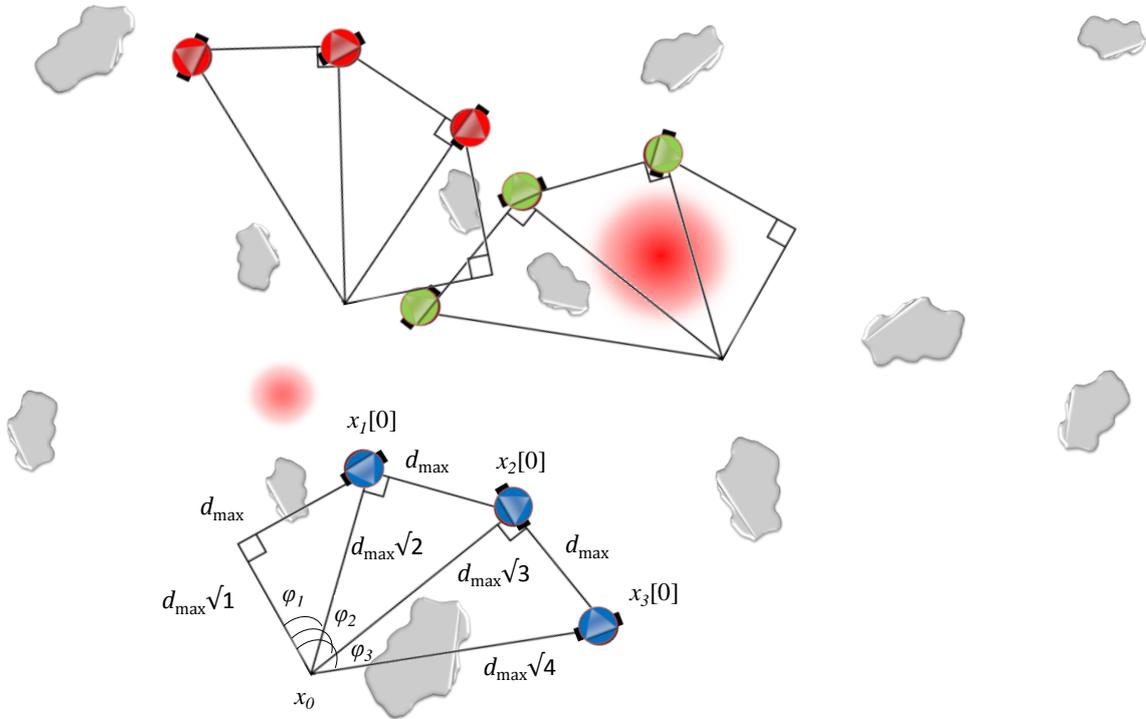


Figure 5.1. Initial deployment of the *RDPSO* algorithm of a population of robots divided in 3 subgroups of 3 robots each.

The total angle φ_n of the n^{th} robot situated in the j^{th} triangle (or spiral segment), can be calculated as the cumulative sum:

$$\varphi_n = \sigma_s \sum_{k=1}^n \arctan\left(\frac{1}{\sqrt{j}}\right), \quad (5.1)$$

in which σ_s is randomly set to ± 1 for each subgroup, thus allowing to compute the initial planar position of each robot n as if follows:

$$x_n[0] = x_0 + \begin{bmatrix} d_{max}\sqrt{n+1} \cdot \cos(\varphi_n + \varphi_0) \\ d_{max}\sqrt{n+1} \cdot \sin(\varphi_n + \varphi_0) \end{bmatrix}, \quad (5.2)$$

where x_0 and φ_0 is the centre and the orientation, respectively, of the spiral which can be randomly assigned at each trial ensuring the efficiency of the stochastic algorithms. In short, the initial deployment of each subgroup will correspond to a spiral in which the position of each robot depends on the prior deployed robot and the centre of the spiral x_0 (Figure 5.1). To allow the autonomous deployment of robots in a scenario, a pre-processing of the environment needs to be undertaken in order to prevent

robots from being deployed into areas of no interest (*e.g.*, water, obstacles, other robots). This can be accomplished with UAVs through image segmentation (Kulkarni & Venayagamoorthy, 2010).

Nevertheless, this *spiral of Theodorus* in which our first naïve approach to the initial deployment problem was based in, present three major issues (Couceiro, Rocha, & Ferreira, 2011b):

- As previously mentioned, it requires a pre-processing of the environment to avoid deploying robots into areas of no interest;
- It only considers the maximum range between robots, thus ignoring the signal quality;
- It does not allow the creation of a k -connected *MANET* (*i.e.*, fault-tolerance).

In this work, and given the *RDPSO* dynamics, the initial deployment problem can be formally defined as next section states.

5.1.1 Problem Statement

Consider a population of N_T robots, denoted as scouts, which are initially clustered into subgroups of N_S scouts, wherein each scout is both an exploring agent of the environment and a mobile node of a *MANET* within subgroup s that performs packet forwarding, according to a paradigm of multi-hop communication. The goal is to ensure that the N_S scouts from the same subgroup are initially deployed by a robot, denoted as ranger, in an unknown environment, while avoiding areas of no interest (*i.e.*, obstacles) and ensuring that the *MANET* is k -connected, $k \in \mathbb{N}$.

5.1.2 General Approach

Similarly to Rybsky's work, the initial deployment of robots will be carried out hierarchically dividing the population of robots into *rangers* and *scouts* (Rybski, et al., 2000). Each ranger handles the initial deployment of an entire swarm of scouts allowing a distributed and autonomous transportation, thus sparing the need of a pre-processing procedure (*e.g.*, topological features extraction using unmanned aerial vehicles). This hierarchical division of the population of robots into rangers and scouts was considered within the preliminaries described in sections 3.2.1 and 3.2.2 and is herein summarized:

- Rangers are large robots that need to be able to transport an entire subgroup of scout platforms and process the sensor data (*i.e.*, scouts positioning, obstacle avoidance and communication constraints);

- Scouts are small low-cost robots, with limited sensing and communication capabilities and finite memory, which need to be easily deployable and able to sense their environment communicating with each other. It is noteworthy that scouts need to be able to communicate with rangers.

Since this work focuses on unknown environments, rangers should deploy scouts while avoiding obstacles, based on the communication link between the previously deployed scout and itself. Therefore, contrarily to the *spiral of Theodorus* approach previously addressed, this *Extended Spiral of Theodorus (EST)* will not have a fixed central point x_0 defined in equation (5.2). Instead, the central point will vary over time depending on the scouts previously deployed (*i.e.*, number of deployed scouts and distance between them). It is also noteworthy that the distances between pairs of deployed robots will not be the same since there is not a linear relationship between the maximum communication distance d_{max} and the minimum signal quality q_{min} . In other words, scouts from the same subgroup are successively deployed one after another by the same ranger such that the pose of the n^{th} robot always depends on the pose of the $(n - 1)^{th}$ robot and the existence of obstacles in the path between them (Couceiro M. S., Figueiredo, Portugal, Rocha, & Ferreira, 2012).

The behaviour of a ranger transporting an entire subgroup of scouts can then be described as it follows. The ranger first moves to a random initial location while avoiding obstacles. The way the ranger avoids obstacles can be a simple wall-follower mechanism when it encounters an obstacle in its path (*cf.*, Algorithm 5.1). When the ranger reaches the desired initial position or its vicinities (due to obstacles constraints) it will unload the first scout, thus informing it that it will start with a pose related to its own pose at the time, *i.e.*, $\langle x_1[0], \theta_1[0] \rangle$. At this point, the scout will broadcast a message containing its *ID* (in this case, the *ID* identifies it as the first robot) and pose – let us call this as the beacon message. Also, it will not start the mission until it ears a reply message from the ranger. The ranger will then choose a random orientation and start moving apart from the unloaded robot while avoiding obstacles and earing its beacon message. When the signal quality of the deployed robot reaches the minimum desired value, *i.e.*, $q_1 = q_{min}$, then the ranger will unload the second robot, once again informing it that it will start with the same pose that the ranger have at the time, *i.e.*, $\langle x_2[0], \theta_2[0] \rangle$.

At this time, the ranger has all the information it needs to compute the next possible location to deploy the third robot. As it knows both the pose of robot 1 and 2 (and so the Euclidean distance between them) it is able to define a possible spiral centre using the following general equation:

$$x_0^{i,i+1} = x_i[0] - \begin{bmatrix} d_{max}^{i,i+1} \sqrt{i+1} \cdot \cos(-\sigma_s \cdot 90 + \theta_{i+1}[0]) \\ d_{max}^{i,i+1} \sqrt{i+1} \cdot \sin(-\sigma_s \cdot 90 + \theta_{i+1}[0]) \end{bmatrix}, \quad (5.3)$$

wherein σ_s is randomly set to ± 1 for each subgroup (or ranger) and, in this case, $i = 1$ such that $x_0^{1,2}$ corresponds to the possible spiral center formed by the triangle defined by robot 1 and 2 and $d_{max}^{1,2}$ is the Euclidean distance between them, *i.e.*, the triangle cathetus. Note that in particular cases in which one needs to control the deployment of robots based on the maximum communication range, $d_{max}^{i,i+1}$ is set as a constant input, *i.e.*, $d_{max}^{i,i+1} = d_{max} \forall i \in \mathbb{N}$.

The orientation of the spiral $\varphi_0^{i,i+1}$ is defined as:

$$\varphi_0^{i,i+1} = \varphi_i' - \sigma_s \sum_{j=1}^i \arctan\left(\frac{1}{\sqrt{j}}\right), \quad (5.4)$$

where φ_i' is the angle between the i^{th} robot and the center of the spiral $x_0^{i,i+1}$.

As a result, similarly to equation (5.2), a new desired position to the third robot can be defined as:

$$x_{i+2}^d[0] = x_0^{i,i+1} + \begin{bmatrix} d_{max}^{i,i+1} \sqrt{i+3} \cdot \cos(\varphi_{i+2} + \varphi_0^{i,i+1}) \\ d_{max}^{i,i+1} \sqrt{i+3} \cdot \sin(\varphi_{i+2} + \varphi_0^{i,i+1}) \end{bmatrix}, \quad (5.5)$$

in which φ_{i+2} is calculated using equation (5.1).

It is noteworthy that the ranger may be unable to reach this desired location. Either because obstacles may constrain the ranger's movements or the signal quality of the beacon message from the second scout q_2 may achieve q_{min} before the ranger reaches $x_{i+2}^d[0]$. Either way, a new unloading location $x_{i+2}[0]$ will be found and the same process will be replicated for the remaining scouts until the ranger unloads all N_s scouts. After deploying the whole team, the ranger broadcasts a message to start the mission. The message will be replicated by scouts inside its communication range, thus reaching all robots within the subgroup.

Note that the *RDPSO* dynamics handles several independent subgroups (*i.e.*, networks of scouts). Despite being possible to start with only one subgroup (as the algorithm will create others over time), which would require only one ranger, a larger number of subgroups would allow a most widely distributed approach. On the other hand, as scouts from a given subgroup are independent from others, they may asynchronously start the mission. Therefore, multiple rangers are recommended to deploy multiple subgroups throughout the environment. This allows that the *EST* algorithm completeness is guaranteed for a given ranger when all of its N_s scouts are deployed.

Note that d does not depend on the direction of the spiral σ_s since the cosine function is unaffected by the signal. Also, as the requirements of the *MANET* connectivity are softened (for smaller k), the denominator of equation (5.7) presents a more linear relation with k . This relation can be easily observed in Figure 5.3.

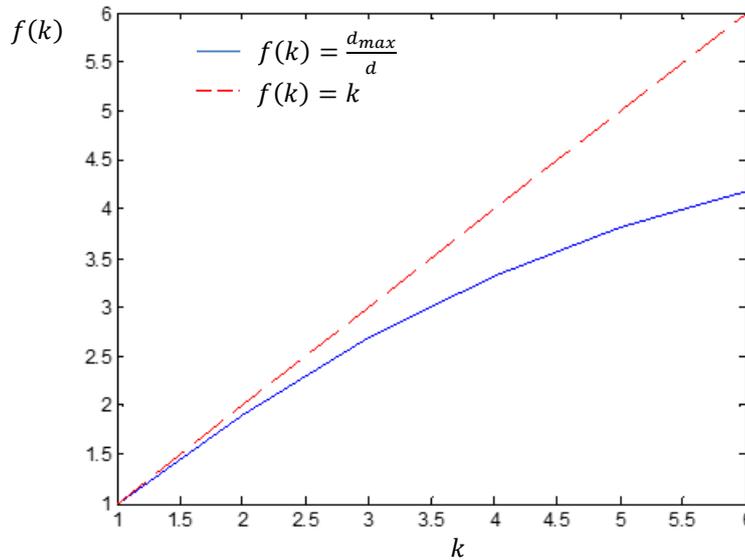


Figure 5.3. Relation between d and d_{max} varying the *MANET* connectivity k .

Hence, one could consider the following approximation since the multi-connectivity level k in *MRS*, contrarily to wireless sensor networks, is usually small (*e.g.*, below 5):

$$d = \frac{d_{max}}{k}, \quad (5.8)$$

The same analysis can be conducted in a minimum signal quality q_{min} problem. However, note that this may only work when using signal quality units that present an approximately inversely proportional relation with the distance. In other words, most of wireless equipment returns the *RSSI* signal in Watts (W) or decibels (dB) – while in most situations the signal quality in Watts is almost inversely proportional to the distance between nodes, decibels present a logarithmic relation. Taking this into account, one can estimate a desired signal quality between robot 1 and 2 as equation (5.6) shows. ■

When facing obstacles, the ranger should follow an optimized decision about whether it should rotate left or right, thus choosing the smallest rotation it needs to perform based on the configuration of the sensed obstacles. When the ranger is unable to compute the best turning decision, it randomly

chooses between rotating left or right. In this way, the growth of the radius at a certain robot i can be approximated using the following equation:

$$\Delta R_i \approx d_{max}^{i,i+1}(\sqrt{i+1} - \sqrt{i}), \quad (5.9)$$

thus allowing to understand the distribution of the robots over a scenario by calculating the approximated total radius of the *EST* strategy within subgroup s :

$$R_T \approx \sum_{i=1}^{N_s} \Delta R_{i+1}. \quad (5.10)$$

It is noteworthy that equations (5.9) and (5.10) are only approximated measures since the distance between pairs of robots may change²⁸ and the existence of obstacles may increase or decrease the growth of the *EST* radius.

Therefore, to further assert the distribution of the *EST* strategy, a set of deployment trials were numerically computed changing the number of robots within a subgroup in an environment with a large density of randomly deployed obstacles. A fixed maximum communication range was used, *i.e.*, $d_{max}^{i,i+1} = d_{max} \forall i$, since it is easier to implement in simulation. Also, fault-tolerance was not considered, *i.e.*, $k = 1$. As the random initial deployment of robots is the most common strategy in the literature (*cf.*, section 2.2.4), its distribution was compared with the *EST* deployment. In the random distribution, robots are successively randomly deployed within a circumference of radius d_{max} and centered in the previously deployed robot while avoiding the overlap with obstacles. For a more detailed description of this random initial deployment please refer to (Couceiro M. S., Figueiredo, Portugal, Rocha, & Ferreira, 2012).

The number of scouts to be deployed was set to $N_s = \{5,10,15,20,25\}$ with 100 trials each for both strategies. Figure 5.4 presents a couple of simulated examples of a team of 10 scouts deployed using both strategies. Blue scouts were deployed using the *ETS* while red scouts were deployed using the random distribution. As it is possible to note, contrarily to the *ETS* in which scouts are scattered throughout the scenario, the random deployment turns out to reveal an unbalanced distribution of scouts. Nevertheless, to further measure the dispersion of both deployment strategies, a metric based on the average distance from each scout to the centroid $x_c[0]$ was used:

²⁸ The signal depends not only on the distance, but also on the multiple paths from walls and other obstacles.

$$\sigma_s = \frac{1}{N_s} \sum_{i=1}^{N_s} \|x_i[0] - x_c[0]\|. \quad (5.11)$$

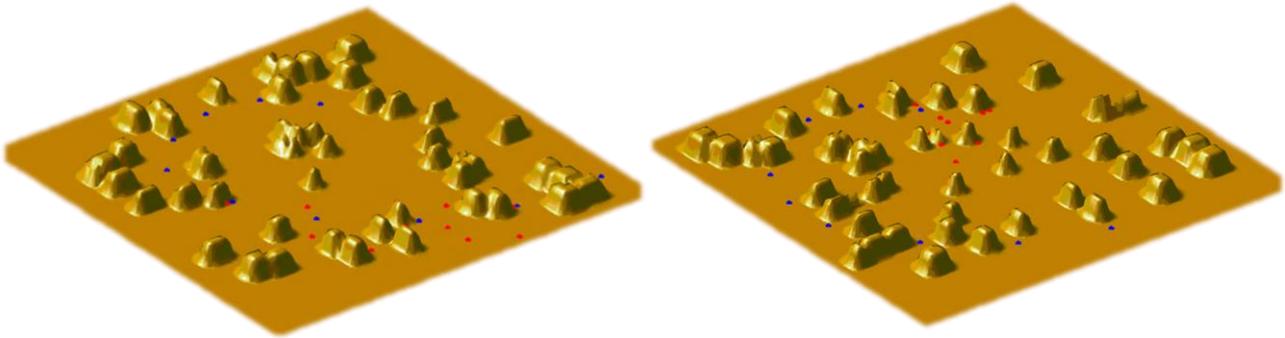


Figure 5.4. Computational outputs of the *EST* (blue) and random (red) deployment of 10 scouts over a given scenario endowed with obstacles.

Figure 5.5 depicts the dispersion of the robotic team σ_s for both strategies.

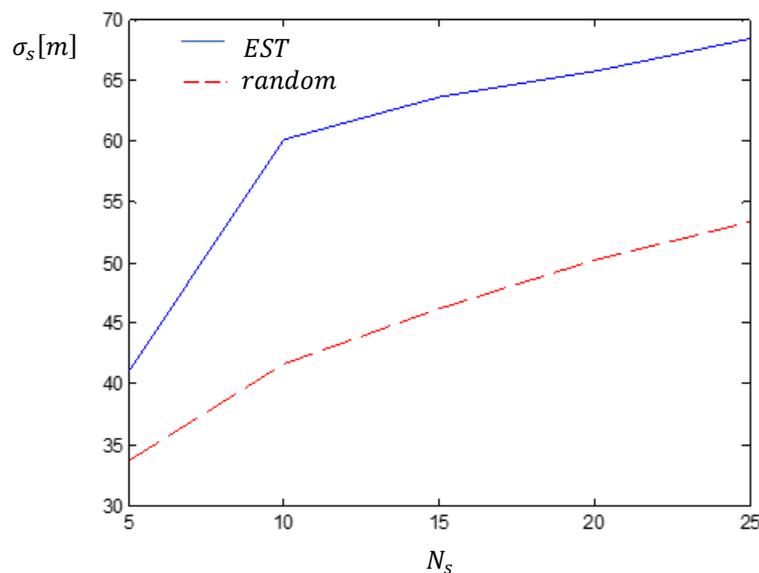


Figure 5.5. Relation between d and d_{max} varying the *MANET* connectivity k .

As one can observe, the dispersion of the team of robots using the *ETS* deployment is significantly higher than using a random distribution. In fact, the random deployment does not present a substantial increasing dispersion as the number of robots increases.

Algorithm 5.1 generalizes the ranger *EST* behaviour to deploy a whole subgroup of scouts in an unknown environment while ensuring that the *MANET* is k -connected (Couceiro M. S., Figueiredo, Rocha, & Ferreira, 2014 (In Press)).

```

 $\sigma_s = random\_direction$  // desired direction of the spiral (i.e.,  $\pm 1$ )
 $x_1^d[0] = random\_position$  // desired random initial position of scout 1
 $goto\_position(x_0[0], x_1^d[0])$  // travel from the current position to scout 1 position while avoiding obstacles (e.g., wall following algorithm)
 $send\_pose(1, x_1[0], \theta_1[0])$  // informs scout 1 of its initial pose
 $\theta_2^d[0] = random\_orientation$  // desired random orientation of scout 2
While  $d_1 < \frac{d_{max}}{k}$  OR  $q_1 > k \cdot q_{min}$  // depending if it is an application under maximum communication distance OR minimum signal quality
     $[d_1, q_1] = move(\theta_2^d[0], x_1[0])$  // change the orientation and travel forward from scout 1 position to scout 2 while avoiding obstacles
 $send\_pose(2, x_2[0], \theta_2[0])$  // informs scout 2 of its initial pose
For  $i = 1: N_S - 2$ 
     $x_0^{i,i+1} = x_i[0] - \begin{bmatrix} d_{max}^{i,i+1} \sqrt{i+1} \cdot \cos(-\sigma_s \cdot 90 + \theta_{i+1}[0]) \\ d_{max}^{i,i+1} \sqrt{i+1} \cdot \sin(-\sigma_s \cdot 90 + \theta_{i+1}[0]) \end{bmatrix}$ 
     $\varphi_0^{i,i+1} = \varphi_i' - \sigma_s \sum_{j=1}^i \arctan\left(\frac{1}{\sqrt{j}}\right)$ 
     $x_{i+2}^d[0] = x_0^{i,i+1} + \begin{bmatrix} d_{max}^{i,i+1} \sqrt{i+3} \cdot \cos(\varphi_{i+2} + \varphi_0^{i,i+1}) \\ d_{max}^{i,i+1} \sqrt{i+3} \cdot \sin(\varphi_{i+2} + \varphi_0^{i,i+1}) \end{bmatrix}$ 
     $\theta_{i+2}^d[0] = calc\_orientation(x_{i+1}^d[0], x_{i+2}^d[0])$  // desired orientation of scout  $i + 2$ 
    While  $\max(d_i, \dots, d_{i+k-1}) < d_{max}$  OR  $\min(q_i, \dots, q_{i+k-1}) > q_{min}$ 
         $[d_{i+1}, q_{i+1}] = move(\theta_{i+2}^d[0], x_{i+1}[0])$  // change the orientation and travel forward from robot  $i + 1$  position to  $i + 2$  while avoiding obstacles
     $send\_pose(i + 2, x_{i+2}[0], \theta_{i+2}[0])$  // informs scout  $i + 2$  of its initial pose
 $send\_start()$  // broadcast information to start the mission

```

Algorithm 5.1. Initial deployment *EST* algorithm for ranger n .

5.1.4 Rangers' Computational Requirements

The computational requirements of the *EST* linearly depends on the number of deployed scouts as the processing of a new pose relies on the estimation of a new spiral formed by the scouts already deployed. On the other hand, due to Proposition 1 and since $k \leq N_S$, ensuring a k -connected *MANET* does not increase the computational cost. Hence, the time complexity necessary to compute a desired pose is $\mathcal{O}(N_{Sd} + 1)$, wherein N_{Sd} corresponds to the number of scouts from swarm s already deployed at the moment, $N_{Sd} < N_S$.

Although the *EST* allows an efficient initial deployment of robots to form a k -connected *MANET* in order to maintain fault-tolerance throughout the mission, one needs to ensure that each robot is still able to communicate with k other robots after their initial deployment.

5.2 Fault-Tolerance Assessment

Most of *MRS* works handle the fault-tolerance problem through biconnectivity since it is the baseline graph theoretic metric to node failures, thus allowing to keep the network connectivity even when unexpected node failures occur (section 2.2.3). However, one cannot claim that a biconnected *MANET* will be invulnerable against all possible robot failures. Actually, robots' survivability highly depends on the application domain – while biconnectivity ($k = 2$) may be enough to ensure a desired

performance in a simple exploration scenario (*e.g.*, finding a gas leak in a building), a military application may impose more limited conditions (*e.g.*, hostile attacks), thus increasing the necessity for a more connected *MANET* ($k > 2$).

Given the *RDPSO* description from chapter 4, one can define the following problem formulation.

5.2.1 Problem Statement

Consider a subgroup of N_S robots, wherein each robot is both an exploring agent of the environment and a mobile node of a *MANET*, which performs packet forwarding according to a paradigm of multi-hop communication. The goal is to make sure that the robots explore an unknown environment while retaining the *MANET* k -connectivity throughout the mission, $k \in \mathbb{N}$ and $k \leq N_S - 1$.

Considering this problem formulation, an extension of the *MANET* connectivity algorithm previously presented in section 4.4 with k -fault-tolerance is introduced in the next section.

5.2.2 Fault-Tolerance Generalization

As previously described in section 4.4, each robot needs to choose its nearest neighbour in order to compute $\chi_4[t]$, hence forcing the *MANET* connectivity, *i.e.*, $k = 1$. To generalize to multi-connectivity, *i.e.*, $k > 1$, two features need to be addressed:

- i)* A given robot should choose its k “best” neighbours, *i.e.*, either the ones closer to it or presenting a better signal quality;
- ii)* The virtual force represented by $\chi_4[t]$ will be the vector sum of the k virtual forces.

It is noteworthy that due to the dynamical partitioning inherent to the *RDPSO* algorithm (*cf.*, section 4.5), and since the connectivity level k cannot exceed the number of robots within a group, k may change over time so as to ensure $k \leq N_S - 1$. This needs to be secured as the number of neighbours of a given robot cannot exceed the remaining number of robots within the group. Also, note that the total number of robots in a group will always be equal to or greater than the minimum number of acceptable robots to form a subgroup N_{min} , $N_{min} \in \mathbb{N}$.

The first feature can be easily addressed by computing the minimum/maximum k elements of each row of link matrix L , after excluding the diagonal elements and the previously chosen $\langle i, j \rangle$ pairs. This will return a discrete subset of \mathbb{R}^{ϖ} , denoted as $\vec{Q}_n[t]$ whose cardinality is, at most, k , *i.e.*, $\#\vec{Q}_n[t] = \min(k, N_n)$, wherein N_n represents the number of “available” robots that still did not choose robot n . Each element corresponds to either the distance or the signal quality vector between robots n and i , whose direction is defined by the vector connecting $x_n[t]$ to $x_i[t]$.

In section 4.4, $\chi_4[t]$ was represented by the position of the nearest neighbor increased by the maximum communication range d_{max} toward the robot's current position. This was achieved by considering an inversely proportional relation between the maximum communication distance d_{max} and the minimum signal quality q_{min} . However, one should note that if an obstacle is in the communication path between two robots separated by a distance of d_{max} , they may have a significantly lower *RSSI* value than two other robots that are at the same distance but without any interference in their way. Therefore, to avoid this assumption, $\chi_4[t]$ will be defined depending on whether the application considers the maximum communication distance d_{max} or minimum signal quality q_{min} for robot n through equation (5.12), wherein i represents the chosen neighbours (Couceiro, Rocha, & Ferreira, 2013b).

$$\chi_4[t] = x_n[t] + \vec{Q}_n[t], \text{ with}$$

$$\vec{Q}_n[t] = \begin{cases} \sum_i (q_{min} - L_{n,i}) \frac{x_i[t] - x_n[t]}{\|x_i[t] - x_n[t]\|}, & q_{min} \text{ problem} \\ \underbrace{\sum_i (L_{n,i} - d_{max}) \frac{x_i[t] - x_n[t]}{\|x_i[t] - x_n[t]\|}}_{\vec{Q}_{n,i}[t]}, & d_{max} \text{ problem} \end{cases} \quad (5.12)$$

Let us consider the same planar topology presented in Figure 4.4 in which it is now necessary to guarantee a biconnected network ($k = 2$). Note that values in the link matrix L in Figure 5.6 are in *dBm*. For a realistic analysis of this illustrative example, let us consider both positions of the robots and common *RSSI* values retrieved with *XBee OEM RF* modules from *Maxstream* (cf., section 3.2.3).

In this case, and as robot 1 cannot communicate with robot 4, and vice-versa, the receiver sensitivity of **-94 dBm** (in bold) was used as the maximum threshold for purposes of computation of equation (5.12). The chosen neighbours are identified by the numbers in blue in the link matrix. As it is possible to observe, robot 1 chooses robot 2 and 4 as its nearest neighbours since they are the nearest ones or the ones that present the higher signal quality. As previously mentioned, the link between robot 1 and 2 corresponds to the ideal situation wherein no attractive or repulsive force is necessary. However, robot 4 is too far away from robot 1, thus inducing a virtual attractive force toward it. Robot 2 chooses robot 3 and robot 4 as its nearest neighbours, since robot 1 has first chosen robot 2. As robot 3 is too close to robot 2, a repulsive force is generated. On the other hand, as robot 4 is too far away from robot 2, an attractive force is generated. The resulting force will then allow robot 2 to move away from robot 3, while at the same time it will also move closer to robot 4. Finally, the two nearest neighbours of robot 3 that did not choose it as their nearest neighbour are robot 1 and robot

4, which are too far away, hence being affected by attractive forces toward each other. As it is possible to observe, robot 4 is once again unable to choose any neighbour since it was the last one updating its status, *i.e.*, $\vec{Q}_4[t] = \vec{0}$.

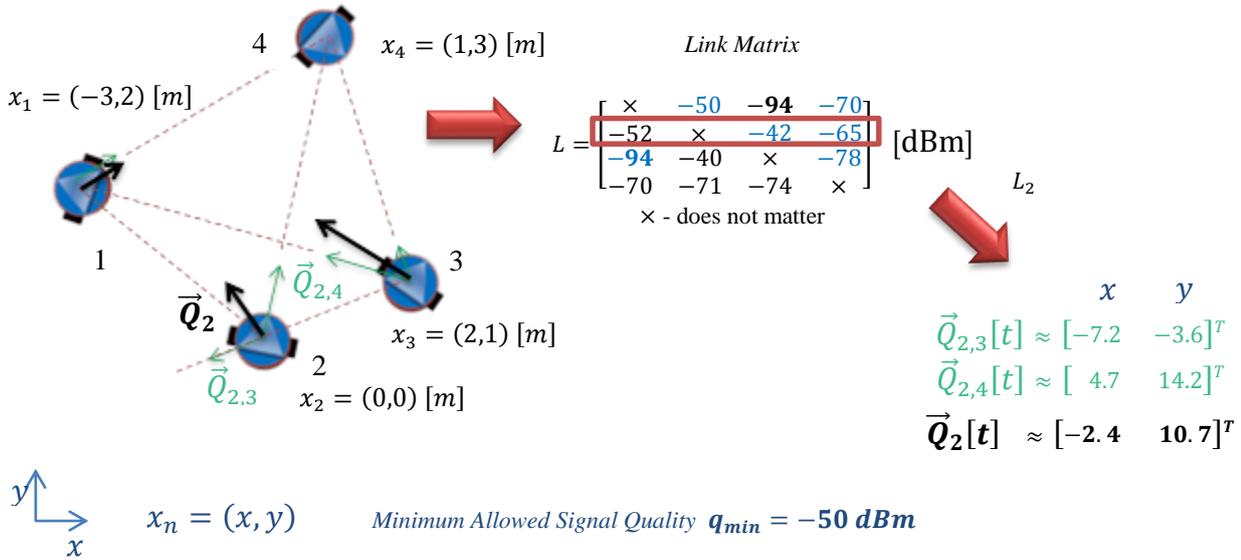


Figure 5.6. Illustration of a *MANET* topology. The dashed lines represent the vector $\vec{Q}_n[t]$ for each robot, thinner arrows represent the force vectors $\vec{Q}_{n,i}[t]$ regarding chosen neighbour i , and larger arrows represent the resulting force vectors $\chi_4[t]$.

As previously mentioned, it is noteworthy that the success of maintaining a k -connected *MANET* highly depends on the network connectedness enforcement component ρ_4 . Algorithm 5.2 formalizes the k -fault-tolerance algorithm (Couceiro M. S., Figueiredo, Rocha, & Ferreira, 2014 (In Press)).

```

If  $d_{max\_prob} = 1$  // application under maximum communication distance
   $[L_n, index_n] = sort\_ascending(L_{n,1:N_S})$  // sort the elements of line  $n$  from link matrix  $L$  in ascending order
Else-If  $q_{min\_prob} = 1$  // application under minimum signal quality
   $[L_n, index_n] = sort\_descending(L_{n,1:N_S})$  // sort the elements of line  $n$  from link matrix  $L$  in descending order
 $cnt = 0$ 
If  $k_{max} > N_S - 1$ 
   $k = N_S - 1$  // the connectivity level depends on the number of available scouts within the swarm
For  $i = 1:N_S$ 
  If  $index_n(i)$  has not yet chosen it as its nearest neighbor
     $\vec{Q}_i[t] = L_{n,i} \frac{x_i[t] - x_n[t]}{\|x_i[t] - x_n[t]\|}$  // virtual force vector from  $i$  to  $n$ 
    If  $d_{max\_prob} = 1$ 
       $\chi_4[t] = x_n[t] + \sum(d_{max} - \vec{Q}_i[t])$ 
    Else-If  $q_{min\_prob} = 1$ 
       $\chi_4[t] = x_n[t] + \sum(\vec{Q}_i[t] - q_{min})$ 
     $send\_choice(i)$  // communicate to scout  $i$  that it was chosen by scout  $n$ 
     $cnt = cnt + 1$  // neighbours counter
    If  $cnt = k$  // maximum number of possible chosen neighbours
      break from For

```

Algorithm 5.2. k -fault-tolerance algorithm for scout n .

5.2.3 Scouts' Computational Requirements

The computational requirements of the k -fault-tolerance *RDPSO* for a given robot only depend on the number of teammates since the connectivity level k (*i.e.*, the number of neighbours a robot needs to ensure connectivity) is always inferior to the number of robots within the group. Hence, the time complexity of the k -fault-tolerance *RDPSO* is $\mathcal{O}(2N_S)$ as one needs to sort the links (*i.e.*, distances or signal quality) between the robots from the same group. For a list of computationally efficient sorting algorithm, please refer to (Bhalchandra, Deshmukh, Lokhande, & Phulari, 2009). Note that swarm robots are memory limited and this statement is *per robot*. Hence, the maximum number of robots allowed within a subgroup, N_{max} , needs to be defined while considering the hardware specifications of a robot.

Given the aforementioned fault-tolerance assessment, next section presents experimental results to evaluate the *RDPSO* using real and simulated robotic platforms in a large environment.

5.3 Experimental Results

The previous sections presented the initial deployment strategy and fault-tolerance assessment within the *RDPSO* algorithm. To validate these methodologies, this section provides experimental results obtained using real and simulated robots. To that end, the whole population of robots was divided into two physically different platforms in which each of them as the necessary features to assume the role of a scout or a ranger, namely, *eSwarBots* (*cf.*, section 3.2.1) and *TraxBots* (*cf.*, section 3.2.2).

5.3.1 Real-World Experiments

Once again, due to the stochasticity of the *RDPSO*, test groups of 20 trials of 360 seconds each were considered for 15 *eSwarBots*, *i.e.*, $N_T = 15$ comparing the distributed spiral approach proposed in section 5.1 with a random distribution. On those experiments, *eSwarBots* were autonomously deployed by *TraxBots* on the scenario using the *EST* strategy (*cf.*, section 5.1) while preserving a minimum signal quality $q_{min} = -75 \text{ dBm}$, resulting in inter-robot distances, in average, of 3 meters according to Figure 3.7. This was compared to a random initial deployment that was carried out by unloading *eSwarBots* from the same subgroup within a circumference of 3 meters radius (centred on the previously deployed scout), thus ensuring that robots are able to communicate with a *RSSI* superior to -75 dBm (Couceiro M. S., Figueiredo, Portugal, Rocha, & Ferreira, 2012).

During the search task, a minimum, initial and maximum number of 0 (all robots socially excluded), 3 and 5 subgroups were considered, thus representing an initial subgroup size of $N_S = 5$

eSwarBots (i.e., the maximum load capacity of *TraxBot* platforms). The maximum travelled distance between iterations was set as 0.2 meters, i.e., $\Delta x_{max} = 0.2$. All of the experiments were carried out in a 10×20 meters ($A = 200 \text{ m}^2$) sports pavilion at the *College of Education of Coimbra* from *IPC (ESEC-IPC)*, in which obstacles were randomly deployed (Figure 5.7a). The experimental environment contained two sites represented by an illuminated spot uniquely identifiable by controlling the brightness of the light.

The main objective of robots was to find the brighter site (optimal solution). As previously carried out for the smaller scenario from section 4.7.2, the intensity values $F(x, y)$ represented in Figure 5.7b were obtained sweeping the whole scenario with a single robot using its *LDR*. The maximum intensity values obtained (hot colour in Figure 5.7b) was found between 600 and 660.

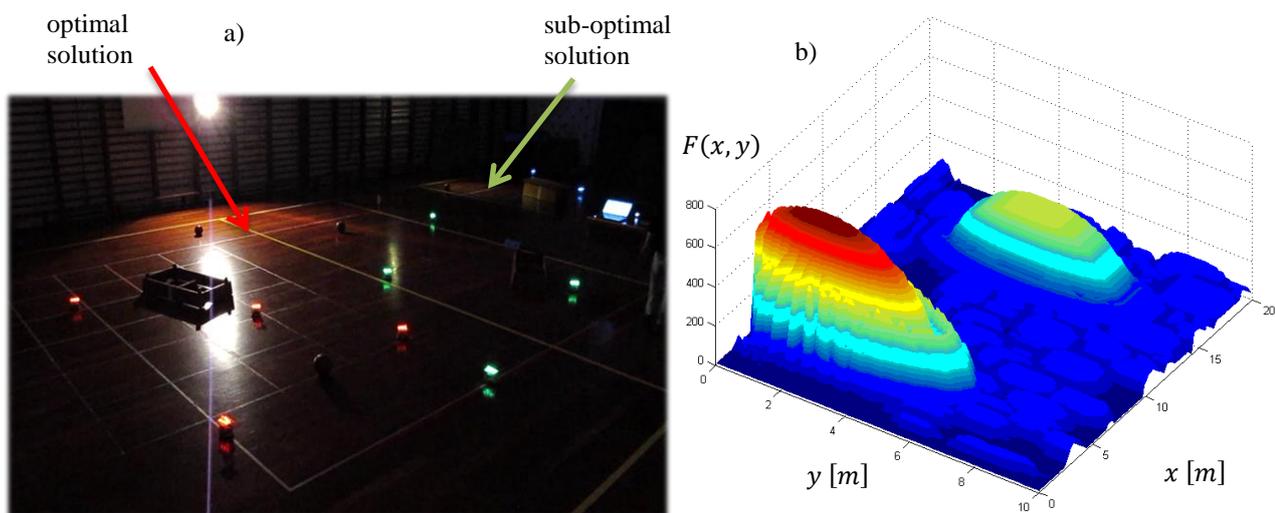


Figure 5.7 Experimental setup. a) Arena with 3 subgroups (different colours) of 5 *eSwarBots* each deployed by 3 *TraxBot* (one for each subgroup); b) Virtual representation of the target distribution.

Table 5.1 summarizes the whole *RDPSO* configuration. Once again, the parameters presented on Table 5.1 were retrieved by trial-and-error to ensure the convergence of robots towards the solution. Note, once again, that the number of trials is for each different configuration.

Figure 5.8 presents a sequence of frames illustrating the *EST* deployment strategy of the whole population of 15 scouts using three rangers. As it is possible to observe, subgroups started the mission asynchronously from each other since scouts inside a given subgroup acts independently of scouts of other subgroups. The time needed to fulfil the *EST* deployment of each subgroup in the 20 experiments was 126.1 ± 11.2 seconds against the 117.2 ± 19.4 seconds of the random deployment strategy.

Table 5.1. *RDPSO* parameters obtained by trial-and-error and used in a larger experiment with physical robots.

<i>RDPSO</i> Parameter	Value
Number of trials	20
Time per trial [seconds]	360
N_T	15
N_S^{min}	0
N_S^i	3
N_S^{max}	5
SC_{max}	30
q_{min} [dBm]	-75
Δx_{max} [m]	0.2
α	0.5
ρ_1	0.2
ρ_2	0.4
ρ_3	0.8
ρ_4	0.8

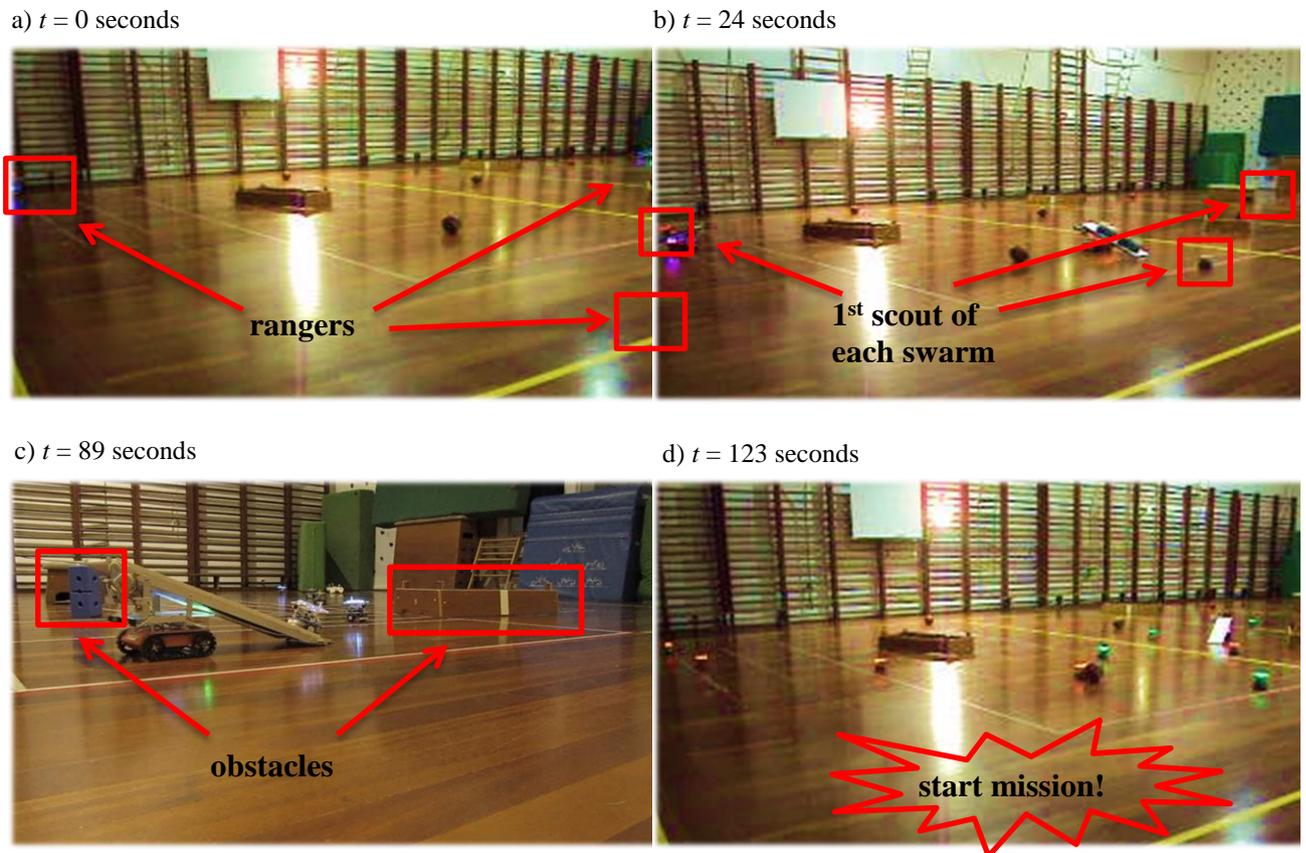


Figure 5.8. Frame sequence showing the *EST* deployment strategy on a population of 15 scouts and 3 rangers. a) The population of scouts is initially divided into three subgroups – red, green and blue – each subgroup loaded on top of a different ranger (one of the rangers is outside camera’s field-of-view); b) Each ranger randomly chooses the first position to deploy the first scout of each subgroup; c) The rangers will deploy the other successive scouts considering the previously deployed ones while avoiding obstacles; d) After deploying all the scouts from one subgroup, the ranger in charge of such deployment broadcasts a message to start the mission.

Figure 5.9 depicts the performance of the *RDPSO* algorithm, for the two different initial deployment strategies. The coloured zones between the solid lines represent the interquartile range (*i.e.*, midspread) of the best solution in the 20 trials that was taken as the final output for each different condition. In other words, the lower line corresponds to the first quartile (*i.e.*, splits lowest 25% of data), the middle one to the second quartile (*i.e.*, median value) and the upper line to the third quartile (*i.e.*, splits highest 25% or lowest 75% of data).

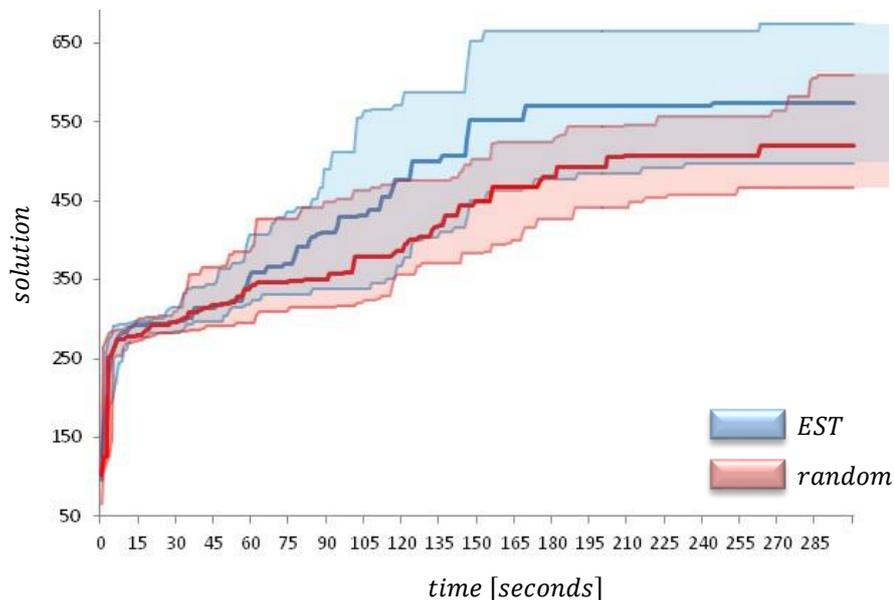


Figure 5.9 Performance of the *RDPSO* under two different deployment strategies. Coloured zones correspond to the interquartile range of the best solution of the 20 trials for each different deployment.

As one may observe, the *EST* deployment allows a faster convergence of scouts toward the optimal solution. This may be due to the larger distribution obtained with the *EST* approach that grants a larger diversity of initial solutions, thus yielding to better results. On the other hand, such diversity is also responsible for having a larger interquartile range than the random deployment. Boxplot charts were used to graphically represent the difference between the diversity of solutions obtained with both deployment strategies at the instant scout starts their mission (Figure 5.10). The ends of the blue boxes and the horizontal red line in between correspond to the first and third quartiles and the median values, respectively.

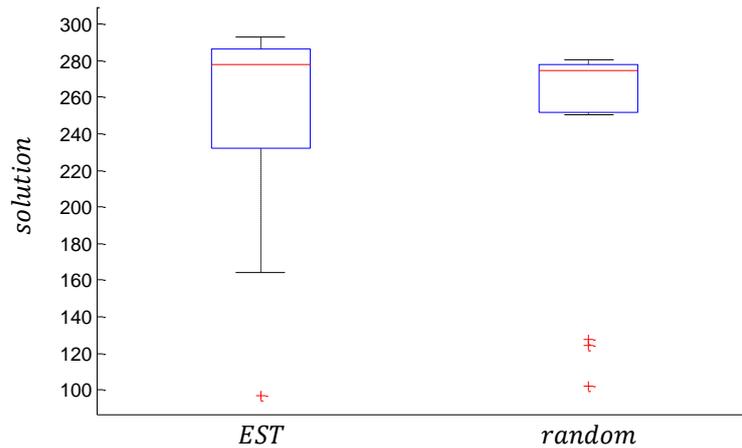


Figure 5.10 Diversity of solutions at the instant scouts start their mission.

One may also observe in Figure 5.10 that the 25th quartile of the *EST* approach is similar to the median value of the random deployment. In other words, 75% of the experiments under the *EST* approach turn out to have a performance equal or superior to 50% of the experiments under the random deployment strategy. Nevertheless, independently on the initial deployment distribution, the median value of the solution was near 500, thus corresponding to the vicinities defined by a 2 meters ellipse around the optimal solution (*cf.*, Figure 5.7b).

However, the use of low-cost platforms brings a limitation inherent to power consumption. As presented in section 3.2.1, *eSwarBots* were built to present an energy autonomy of approximately 3 hours. Nevertheless, each set of experiments was performed in different occasions in which robots needed to be active for 2.5 hours to 3.5 hours (deployment time plus mission execution). At some point, *i.e.*, approximately after the 14th trial of each different condition, some robots (around 1 to 3 robots) were unable to fulfil the whole trial (with 5 minutes duration) due to energy depletion. In both *EST* and random approach, without the herein proposed fault-tolerance strategy, this turned out to cause some minor failures in the *RDPSO* performance since it resulted in subgroups partitioning. Figure 5.11 depicts the trajectory performed by scouts, under the *EST* approach, in which one of such situations happened in the 16th trial. As Figure 5.11 depicts, the partitioning of subgroups may not greatly jeopardize the performance of the algorithm, as the *RDPSO* is endowed with evolutionary behaviour that allows the progression of scouts based on their performance. Nevertheless, the failure of robots may be more critical in larger scenarios or when a high number of robots is affected by failures.

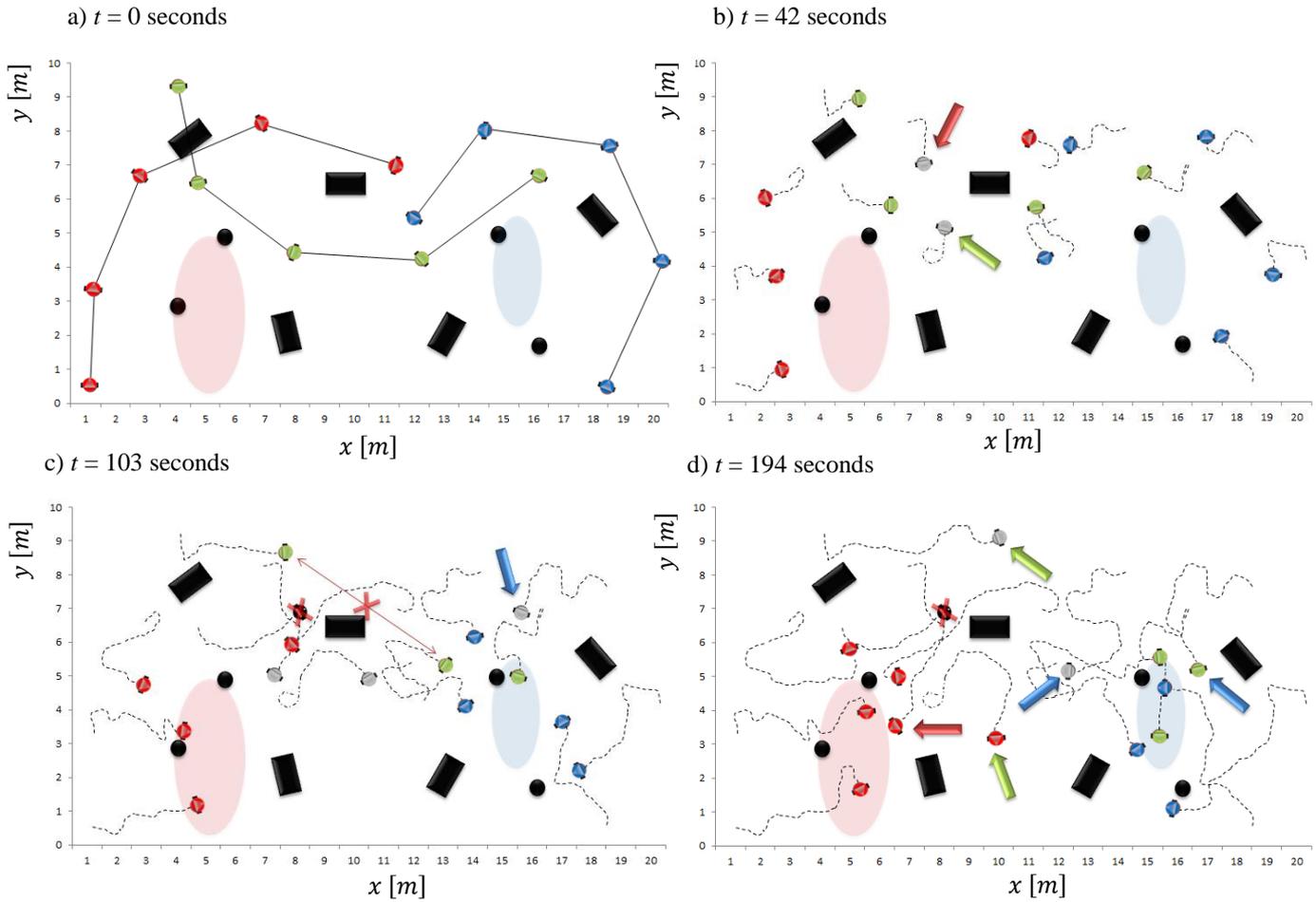


Figure 5.11 Trajectory of scouts under the *EST* approach in the 16th trial (dark objects corresponds to obstacles). a) The configuration evolves starting with a spiral-like deployment; b) The scouts try to maintain a signal quality of -50dBm between their nearest neighbour while subgroups evolve (grey scouts correspond to the socially excluded subgroup); c) One scout from the green subgroup had an energy depletion due to the exhaustive experiments, thus partitioning its subgroup; d) As the top green scout thinks its subgroup is not improving, it got socially excluded.

To improve the system robustness to such possible failures, 20 new trials under the same conditions with *EST* deployment were performed while considering a biconnected *MANET*, *i.e.*, $k = 2$. Hence, rangers deployed scouts while maintaining a minimum signal quality between the previously deployed scout of -71.99dBm, obtained by means of equation (5.13) as:

$$q_{1dBm} = 10 \times \log_{10} \left(2 \times 10^{-75}/10 \right). \tag{5.13}$$

Once again, Figure 5.12 depicts the performance of the *RDPSO* algorithm, by changing the k -fault-tolerance from 1 to 2. As one may observe, the single connected *MANET* (*i.e.*, $k = 1$) allows a faster convergence of scouts toward the optimal solution. Such behaviour was expected since besides

presenting a larger distribution that grants a larger diversity of solutions, robots are less constrained by the motions of their neighbours. Nevertheless, one should observe that the biconnected *MANET* (*i.e.*, $k = 2$) presents a similar behavior than when the *RDPSO* is under the random deployment strategy without any fault-tolerance mechanism (*cf.*, Figure 5.9). This phenomenon leads us to conclude that the *RDPSO* presents sensitivity to initial conditions, *i.e.*, initial pose of robots, which is a pervasive feature of chaotic and stochastic systems.

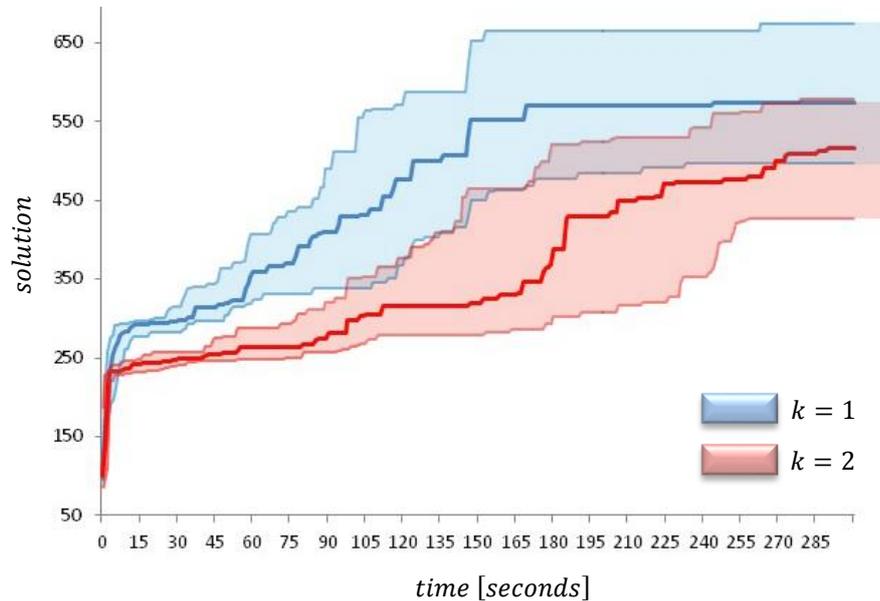


Figure 5.12 Performance of the *RDPSO* under the *EST* approach with and without fault tolerance. Coloured zones correspond to the interquartile range of the best solution of the 20 trials for each different deployment.

To further assess how fault-tolerance affects the global system performance and to improve the comparison between the single connected *MANET* with *EST* and random deployments, Table 5.2 summarizes some relevant communication data obtained throughout the 60 experiments, 20 for each different condition.

Table 5.2. Average and standard deviation of communication data.

AVG \pm STD	EST		Random
	$k = 1$	$k = 2$	$k = 1$
Signal quality [dBm]	-71.5 \pm 3.6	-69.2 \pm 2.9	-71.1 \pm 5.1
Number of hops	1.6 \pm 0.4	1.0 \pm 0.5	1.4 \pm 0.5
Fiedler value λ_2	2.6 \pm 0.7	4.4 \pm 0.5	2.7 \pm 1.0

The signal quality was measured between all one-hop communications, *i.e.*, between neighbour scouts. The number of hops was the number of scouts necessary to forward the message between non-neighbour scouts. Finally, the *Fiedler value* was calculated based on the topology of the *MANET* at

each iteration. As presented in Nathan *et al.* work (Michael, Zavlanos, Kumar, & Pappas, 2009), the connectivity of a network of N_s robots can be represented by the *second smallest eigenvalue*, also known as *Fiedler value*, λ_2 of the *Laplacian matrix* $\mathcal{L} = [\ell_{ij}] \in \mathbb{R}^{N_s \times N_s}$ defined by:

$$\mathcal{L} = \Delta - A_c, \quad (5.14)$$

where A_c is the adjacency matrix that directly depends of the link matrix L (*cf.*, section 4.4) and $\Delta = [\Delta_{ij}] \in \mathbb{R}^{N_s \times N_s}$ is the valency matrix (*i.e.*, diagonal matrix) of A_c described as:

$$\Delta = \text{diag}(\sum_{j=1}^n a_{ij}). \quad (5.15)$$

For a more detailed description about the adjacency and link matrices in the *RDPSO* context, refer to section 4.4. It is noteworthy that the graph is connected when the Fiedler eigenvalue is greater than zero, *i.e.*, $\lambda_2 > 0$.

As it is possible to observe in Table 5.2, and as expected, both *EST* and random deployment without any fault-tolerance strategy (*i.e.*, $k = 1$) present similar results regarding the signal quality between direct transmission and the number of scouts necessary to forward a message and the connectivity of the *MANET*. The main difference between both lies in the variability of the results as the standard deviation is larger in the random deployment strategy than in the *EST*. This may be explained due to the initial configuration under the random deployment does not guarantee the adequate topology to ensure the minimum *MANET* connectivity, *i.e.*, each scout may directly communicate with 2 or more scouts. Although this may be necessary in some situations, it also reduces the diversity of the algorithm in the first iterations. The strategy that clearly stands out in Table 5.2 is the *RDPSO* under the *EST* approach with biconnected *MANET* (*i.e.*, $k = 2$). The average signal quality between neighbors is considerably larger than in the other strategies (note that values are in *dBm*), thus resulting in a signal level, in watts, 50% larger than in the other strategies without fault-tolerance mechanisms. Taking this into account brings us to a smaller number of scouts necessary to forward the message to other non-neighbour scouts. In fact, the data shows that most of the exchanged data between non-neighbour scouts only needs approximately one scout to forward the message.

Let us now evaluate the scalability of the *EST* approach using larger populations of robots in a simulated environment.

5.3.2 Scalability Evaluation through Simulations

In this section, the effectiveness of the deployment strategies is further explored. To that end, the *RDPSO* is used after the initial deployment of large swarms of simulated robots, to perform a collective foraging task in a simulated scenario, modelled after the sports pavilion used in the experiments with physical robots (*cf.*, Figure 5.7).

MRSim was used to evaluate and compare the approaches (*cf.*, section 3.2.4). Due to the lack of a pre-existent model of *WiFi* propagation (radio frequency at 2.4 GHz) in *MRSim* simulator at the time, this work contemplated its implementation based on Luca *et al.* work (Luca, Mazzenga, Monti, & Vari, 2006), which used the well-known multi-wall radio frequency (*RF*) propagation model. The attenuation over the transmitter-receiver distance d [m] was calculated as:

$$L = l_c + 10\gamma_{loss} \log d + \sum_W l_W, \quad (5.16)$$

where W represents the number of walls with attenuation l_W between the transmitter and the receiver. The constant factor l_c corresponds to the reference loss value at 1 m. This was defined as $l_c = 47.4$ dB and experimentally validated in indoor scenarios by Luca *et al.* (Luca, Mazzenga, Monti, & Vari, 2006). The path loss exponent γ_{loss} is usually defined between 2 and 4, wherein values near 2 correspond to propagation in free space and values near 4 represent lossy environments. The parameter γ_{loss} was uniformly distributed over the interval 3 and 4, thus providing a stochastic effect on the communication propagation (Sklar, 1997).

Figure 5.13a clarifies how the *WiFi* propagation in such scenario is modelled and illustrates the -75 dBm threshold previously defined as the minimum signal quality considered to carry out the initial deployment (*cf.*, Section 3.2.3). As it is possible to observe, a robot may be unable to communicate with its teammates in some zones due to occlusion by obstacles and distance.

a)

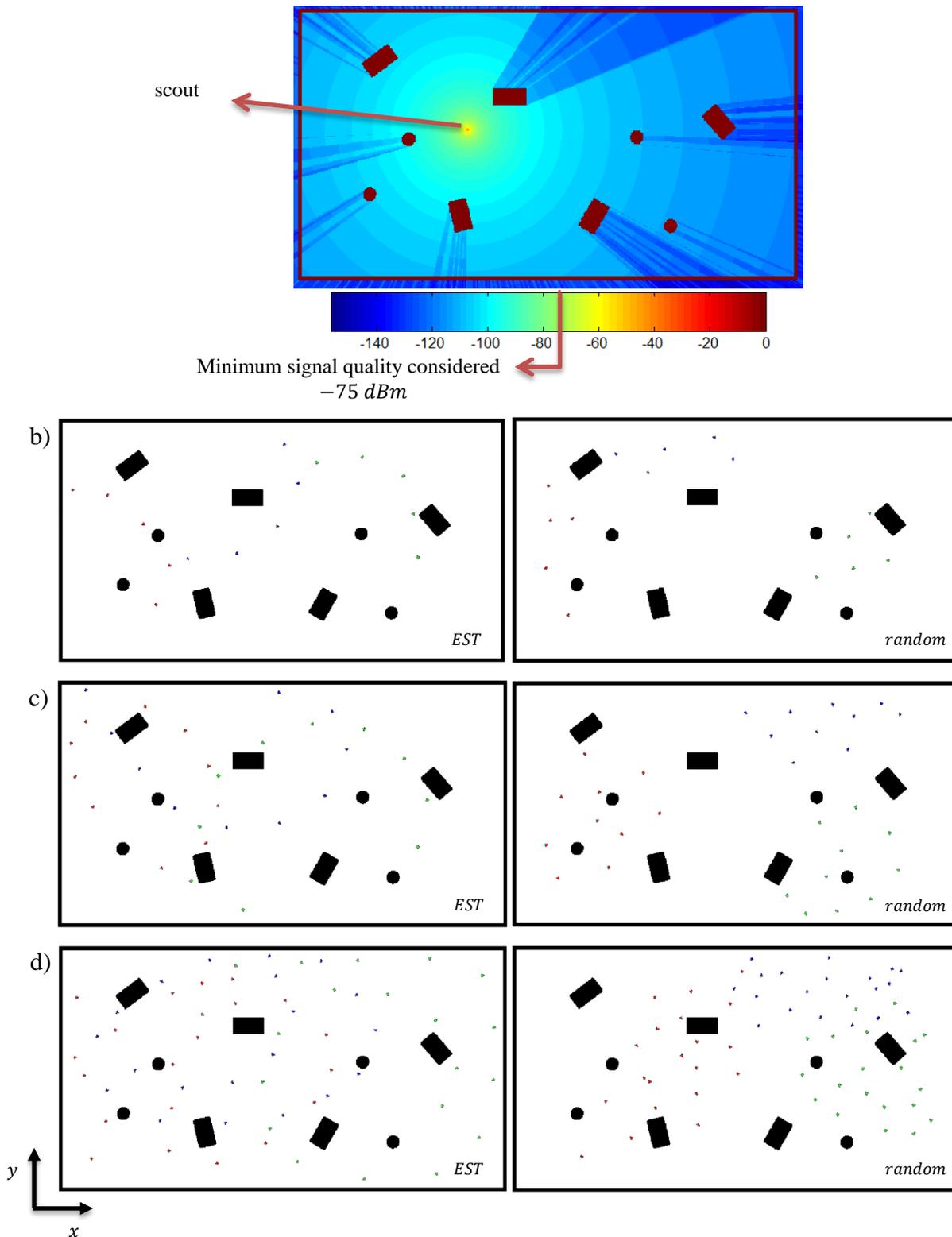


Figure 5.13 Simulation experiments in a 20×10 meters indoor scenario (sports pavilion): a) *WiFi* communication propagation b) $N_T = 15$; c) $N_T = 30$; d) $N_T = 60$.

As seen in Figure 5.13, the *EST* and the random deployment lead to distinct dispersion of scouts in the environment. As expected, the teams are able to cover the area in a wider way, with the increase of the number of robots N_T , for both approaches. Nevertheless, the figure shows that, generally, the

dispersion obtained using the *EST* approach is superior to that shown by the random deployment. In addition, we consider three subgroups deployed independently, which are identified by a unique colour. Results show that the random deployment tends to deploy each subgroup in a specific region of the environment, while *EST* promotes the dispersion of several members belonging to the same subgroup throughout the area, and close by to members of other subgroups. This is particularly visible in Figure 5.13d. Spreading the scouts that belong to the same subgroup in a large area has the benefit of collecting more information about the search space and eventually promoting more variability in the observations within the same subgroup of scouts. As a consequence, this leads to the reduction of the convergence time of the foraging approach used after the initial deployment.

The sensed light from the real scenario in a given position (x, y) was represented by a matrix $F(x, y)$ with the intensity values previously obtained by sweeping the whole scenario with a single *eSwarBot* (Figure 5.7b). One test group for each deployment configuration and number of scouts was evaluated within 20 trials. In other words, a population of $N_T = \{15, 30, 60\}$ scouts was tested for both *EST* and random deployment. Hence, Figure 5.14 depicts the performance of the *RDPSO*, by changing the initial deployment strategy and the total number of scouts $N_T = \{15, 30, 60\}$. Figure 5.14a also comprises the output from Figure 5.9 for the purposes of comparison with the previous results obtained using 15 real *eSwarBots* (pattern regions). The remaining parameters were the same as from Table 5.1.

It should be noted that simulation results are consistent with the real experiments previously carried out, especially as the mission develops further in time. Despite some discrepancies, the similarities between the real experiments with 15 *eSwarBots* and 15 virtual scouts are worth mentioning. This suggests that the phenomena implemented within *MRSim*, in particular the *WiFi* propagation depicted on Figure 5.13a, are in accordance with reality. Moreover, the amplitudes of the results also suggest that the virtual representation experimentally retrieved in Figure 5.7b is a decent approximation of the light intensity.

In general, as one may observe once again, results using both deployment approaches with different population sizes show that, as a rule, *EST* deployment yields a larger distribution, thus resulting in a faster convergence in the exploration phase towards the optimal solution when compared to the random deployment. The difference in the performance is more noticeable with smaller populations, as *EST* leverages from superior space distribution, while in larger populations this situation is mitigated. In fact, this is clear by findings obtained with $N_T = 30$ and $N_T = 60$. The physical restrictions of the space cause the performance to be similar under those configurations, as shown by Figure 5.14b and Figure 5.14c.

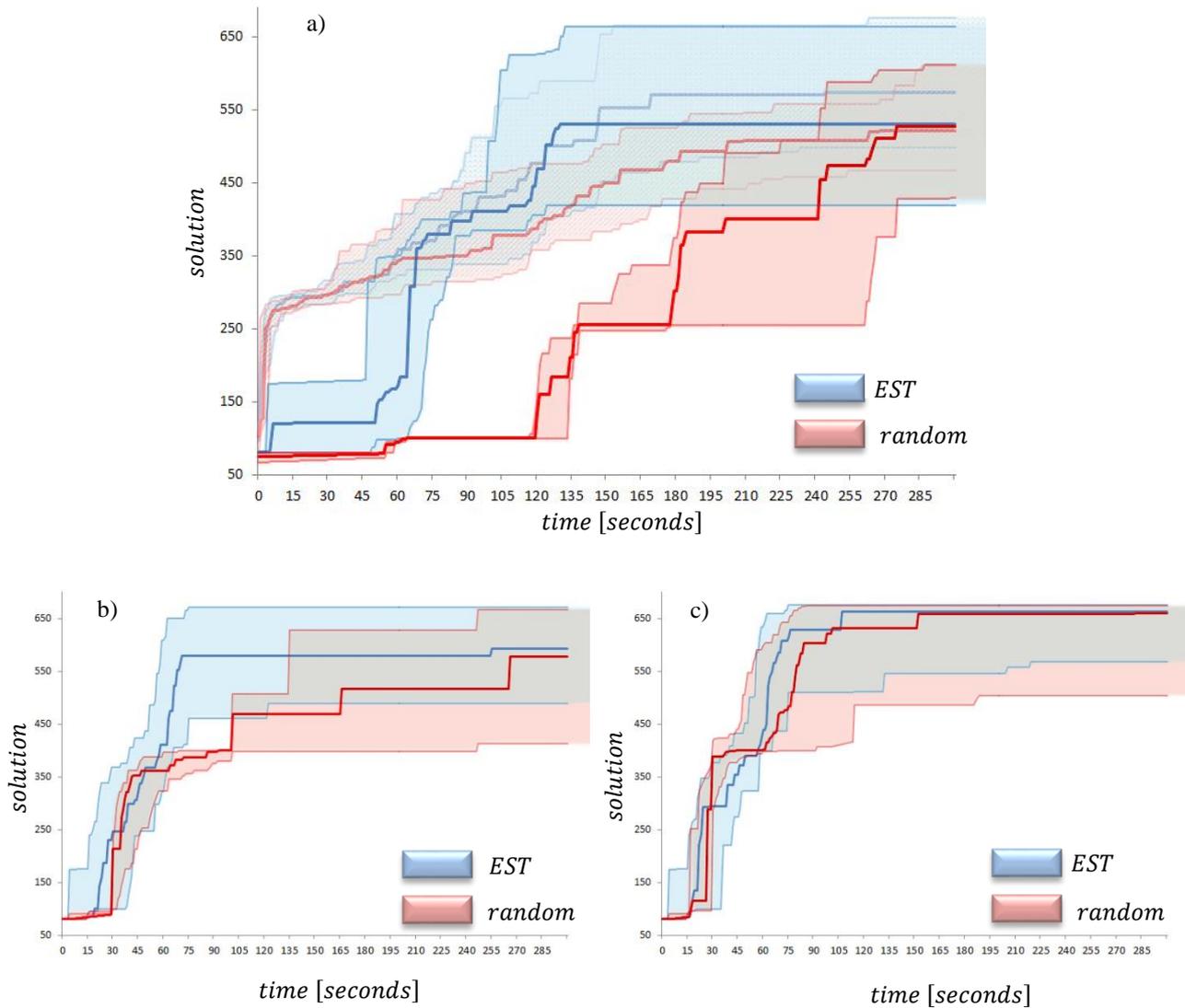


Figure 5.14 Performance of the *RDPSO* under different deployment strategies and number of scouts. Coloured zones correspond to the interquartile range of the best solution of the 20 trials for each different deployment. a) $N_T = 15$ (the pattern regions are the representation of the real experiments from Figure 5.9); b) $N_T = 30$; c) $N_T = 60$.

To go further into comparing both deployment strategies, the area covered by the scouts immediately after being deployed was studied (Mei, Lu, Hu, & Lee, 2005). To do so, let us consider that each scout is able to sense an area of 1 meter radius around itself with its light sensors. Figure 5.15 depicts the area covered by 3 teams of 5 scouts each over the scenario. The total area covered by all the scouts is retrieved using the union operator from set theory.

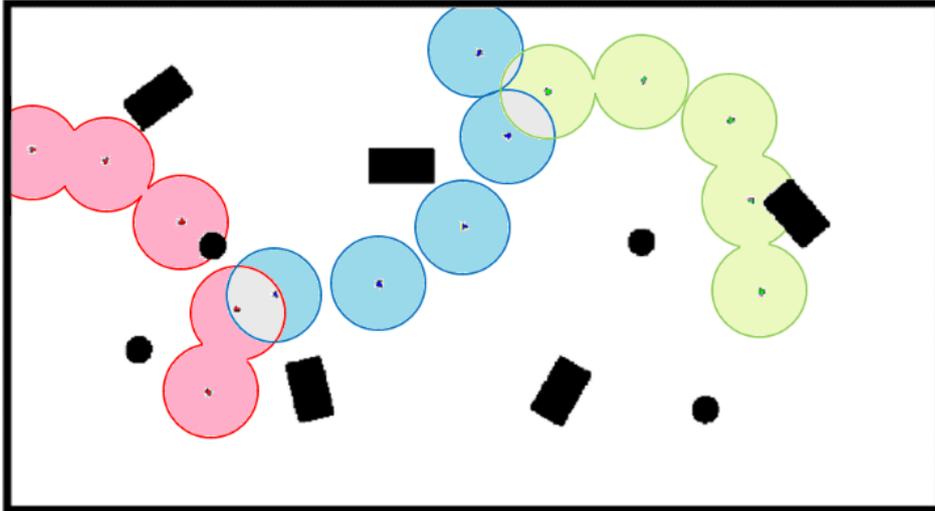


Figure 5.15 Illustration of the area covered by 3 subgroups of 5 scouts each using the *EST* approach. Each different coloured region represents the area covered by a subgroup. Grey regions represent the intersection between areas covered by different subgroups.

Considering the scenario dimensions (200 m^2 without obstacles), a single team of less than 64 scouts uniformly distributed throughout the scenario would be able to fully cover it without even moving. However, such deployment would only be possible if: *i*) the robots would be aware of the scenario dimensions and obstacles location; and *ii*) all scouts would belong to the same subgroup or they would be able to share information with scouts from different subgroups. As both assumptions cannot be held under real conditions from which this work is sustained, such optimal assignment cannot be guaranteed. Therefore, the ratio between the area covered by the scouts with each configuration and the total area of the scenario is used to compare both deployment strategies.

Figure 5.16 depicts the ratio of the covered area for each different configuration, *i.e.*, different team size. The vertical lines within the charts represent the inter-quartile range retrieved from the 20 trials of each configuration. The chart shows that *EST* provides a larger coverage immediately after the initial deployment. Furthermore, the differences in the covered area of both strategies are more apparent with larger populations, because of the high number of intersection in the sensed areas of different scouts, when these are deployed using a random deployment.

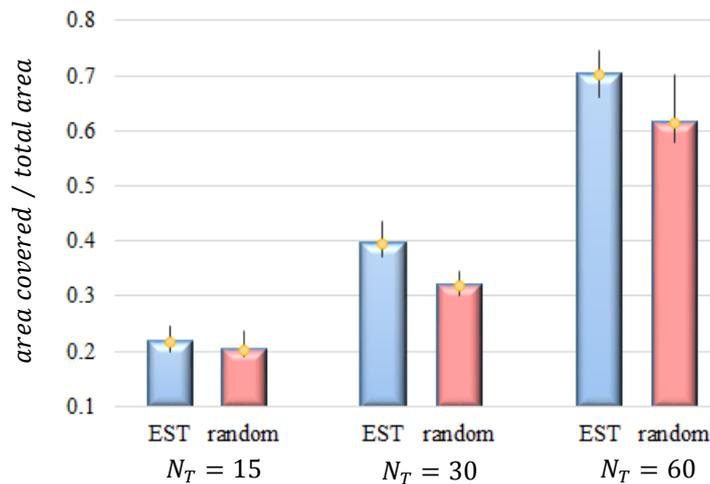


Figure 5.16 Ratio of covered area for a population of 15, 30 and 60 scouts grouped into three subgroups.

A video of the experiments is provided to better understand how the deployment influences the *RDPSO* performance²⁹.

5.4 Discussion

Once again, nature offers a fascinating strategy from which researchers can get inspiration for the design of initial deployment solutions: an infraclass of mammals known as *marsupial*. Although the idea of having robots transporting robots has not been retailed as it should, some preliminary works highlight its relevance on real-world situations (Rybski, et al., 2000). In brief, when a transporting robot (ranger) can no longer move in a given environment due to its design, other robots (scouts) with different capabilities may be able to succeed and use the progress made by the previous robots to its advantage. The main motivation of using *marsupial* robot systems is precisely the inability of reaching remote locations using solely the carrier robot, taking advantage of different strengths and weaknesses of a heterogeneous multi-robot system. Moreover, and as already stated in this Thesis, a network infrastructure is not usually present in real-world scenarios. Therefore, cooperative robots have to fulfil their mission while maintaining connectivity among teammates. In this Ph.D. work, this is considered a hard assumption and, as such, the *marsupial* strategy presented in this chapter, denoted as *Extended Spiral of Theodorus (EST)*, guarantees that the *MANET* stays k -connected during the initial deployment of *scout* robots, wherein k depends on the desired level of connectivity (section

²⁹ http://www2.isr.uc.pt/~michaelcouceiro/media/Deployment_Robotica_new.mp4

5.1). This was followed by the extension of the *RDPSO* with a fault-tolerant distributed search to prevent communication network splits.

Experimental results were first conducted in a large indoor scenario (sports pavilion) endowed with obstacles. For the purpose of evaluating the deployment strategy and the relevance of having k -connected *MANETs* on the *RDPSO* performance, 3 *Traxbots* rangers (*cf.*, section 3.2.2) were used to transport 5 *eSwarBots* scouts each (*cf.*, section 3.2.1). The superiority of the *EST* approach over a random deployment is evident on the initial distribution of scouts throughout the environment and, as a consequence, it speeds up scouts' convergence towards the solution (*e.g.*, Figure 5.9). On the other hand, increasing the level of connectivity k significantly slows down scouts' convergence. This is only natural as the relevance of communication constraint represented by component $\chi_4[t]$ previously introduced on section 4.4 increases with k . Despite this, the connectivity of subgroups under the biconnected mechanism ($k = 2$) herein proposed presents a significant increase when compared to the other strategies (Table 5.2). Such connectivity increase is, in a broad sense, the difference between the connectivity of a linear tandem network, *i.e.*, in which each scout directly communicates with, at most, two neighbours, and a "star-mesh" network, *i.e.*, scouts in a star network are cross-connected as well as radially connected to ensure biconnectivity (*e.g.*, Figure 5.2).

To improve the evaluation of the *EST* deployment strategy, simulation experiments were conducted with a larger number of simulated robots on *MRSim* environment (section 5.3.2). Besides depicting the close relationship from the previous results retrieved with real platforms, the simulation experiments widen the comparison between the *EST* approach and a random deployment by presenting the ratio of the area covered by a population of up to 60 scouts grouped into three subgroups (3 rangers). For instance, while a random distribution of the scouts over the scenario results in an average covered area of 60%, the *EST* is able to rise the covered area to 70% (Figure 5.16).

5.5 Summary

This chapter proposed a realistic deployment strategy inspired on the *Spiral of Theodorus* by benefiting from marsupial systems in distributed *MRS*. The scientific contribution of the chapter was further enriched by considering faulty environment and, as such, propose fault-tolerant communication strategies to avoid *MANET* ruptures.

A population of 15, 30 and 60 scouts and 3 rangers was used to evaluate both marsupial approaches within the *RDPSO* approach introduced on chapter 4. Each ranger handled the initial deployment of an entire subgroup of scouts allowing a distributed and autonomous transportation, thus sparing the need of a pre-processing procedure (*e.g.*, topological features extraction using unmanned

aerial vehicles). Experimental results, obtained in simulations as well as with physical teams of mobile robots, show that the exploration strategy converges sooner when using the so-called *Extended Spiral of Theodorus (EST)* deployment approach, demonstrating the importance of an informed choice of an initial deployment strategy in exploration tasks in unknown scenarios. Moreover, using fault-tolerance strategies allows overcoming robot failures such as energy depletion.

Through this chapter, one can conclude that the performance of the *RDPSO* algorithm is susceptible to the initial pose of robots. This sensitivity to initial conditions is a prime source of the unpredictability inherent to chaotic systems. Although this is an intrinsic feature of stochastic swarm algorithms, such unpredictability can be minimized by efficiently sharing the necessary amount of information between teammates. Nevertheless, increasing the communication cost also brings many disadvantages. Therefore, next chapter presents a thorough study around the communication complexity of the *RDPSO* and proposes a rationale so as to minimize the communication overhead, while maintaining the team aware of local actions.

Communication Optimization

COMMUNICATION is an essential resource to foster effective cooperation among robots (Parker L. E., 2008b). In brief, and as described in section 2.1.3, the way robots communicate can be divided into basically three types: *i*) implicit communication (*i.e.*, *stigmergy*); *ii*) passive action recognition; and *iii*) explicit communication. Within the three techniques, the use of explicit communication is the most appealing method and the one adopted in this work, because of its directness and ease with which robots can become aware of the actions and goals of their teammates. However, and as previously addressed in this thesis, the development of robot teams for unstructured scenarios, such as *SaR*, requires robots to maintain communication among them without the aid of a communication infrastructure. In other words, and as described in section 4.4, robots need to be able to deploy and maintain a *MANET* in order to explicitly exchange information within multi-hop network paths without unnecessarily restricting the team's range. Nevertheless, as previously stated, such networks typically consist of a large number of distributed simplistic nodes (*i.e.*, swarm robots) that organize themselves into multi-hop wireless networks. Adding this to the fact that robots are deployed in either hostile or inaccessible environments, then the *MANET* connectedness may significantly drop, thus jeopardizing the success of the mission.

In section 5.2, it was proposed a fault-tolerance strategy to guarantee k -connected *MANETs* within subgroups of N_S robots, $k \in \mathbb{N}$ and $k \leq N_S - 1$. Hence, a given robot would choose its k -nearest neighbours and the virtual force to maintain the *MANET* connectivity was represented by the vector sum of k -virtual forces. This follows the same principles previously addressed in other works such as (Sabattini, Chopra, & Secchi, 2011a) and (Casteigts, Albert, Chaumette, Nayak, & Stojmenovic, 2010) regarding the need to maintain a pervasive *MANET*. Nevertheless, only by securing that each robot may communicate with its teammates does not ensure an efficient group communication. The more robots needed to explicitly exchange messages between themselves, the most the swarm is susceptible to flaws. Therefore, besides studying the necessary information to be exchanged between teammates, routing protocols should be designed based on the mission-related contextual information, *i.e.*, based on the behaviour that one should expect from the *MRS*.

Bearing in mind such assumptions, many works on *MRS* has been focused on efficiently sharing information between teammates (Rocha R. P., 2006b; Hereford & Siebold, 2008; Shah & Meng, 2007). Rocha addressed the problem of building volumetric maps efficiently sharing the necessary

information based on mutual information minimization (Rocha R. P., 2006b). To that end, the author presented a distributed architecture model with efficient information sharing, wherein entropy was used to define a formal information-theoretic background to reason about the mapping and exploration process. This allows to share only information that may be relevant for the team. It was with that same principle that Hereford and Siebold proposed a swarm exploration strategy wherein robots only shared their position if their own solution was the best solution in the whole subgroup (Hereford & Siebold, 2008). Although this is an interesting strategy, robots still need to share information concerning their own solution and a global assessment of the collective performance needs to be carried out. Similarly, Shah and Meng proposed a communication-efficient dynamic task scheduling algorithm for *MRS* (Shah & Meng, 2007). This algorithm avoided unnecessary communication by broadcasting global information only to the robots who were interested in it, thus reducing the communication overhead. Simulation experiments showed that the proposed strategy was able to reduce the communication cost to almost half when compared to a common broadcast approach.

Besides exploiting the necessary information that robots should share, routing protocols, such as the well-known *Ad hoc On-demand Distance Vector (AODV)*, have been successively extended based on the mobile network requirements (Abedi, Fathy, & Taghiloo, 2008; Asenov & Hnatyshin, 2009; Ayash, Mikki, & Kangbin, 2012). For instance, the authors of Abedi *et al.* extended the *AODV* routing protocol based on the *Manhattan mobility model*, thus making it more fitted for *Vehicular Ad hoc Network* applications (Abedi, Fathy, & Taghiloo, 2008). Such strategy allowed establishing more stable routes, especially in applications demanding a high mobility of nodes, thus reducing the communication overhead of the network. More generally, Asenov and Hnatyshin extended the *AODV* based on the geographical position of nodes retrieved with *GPS* (Asenov & Hnatyshin, 2009). This improves the performance of the *route discovery* process in *AODV* routing (*cf.*, section 6.2.1 for a description about this mechanism). Nevertheless, such strategy assumes that each robot in the network is aware of all teammates' position, thus increasing the communication complexity. Similarly, the work presented by Ayash *et al.* proposed two *GPS*-based strategies, namely the *AODV Location Aided Routing* protocol and the *AODV Line* protocol, to minimize the control overhead of the *AODV* protocol, thus limiting the flooding area of *AODV* (Ayash, Mikki, & Kangbin, 2012). While the first protocol limits the *route discovery* to a small area of the network, the second protocol uses node location information to restrict route search area to be only near the line connecting source and destination nodes. However, both strategies still present the same disadvantage as the work of Asenov and Hnatyshin (Asenov & Hnatyshin, 2009), *i.e.*, the knowledge about the current position of all robots.

Following those same principles, this chapter presents three main contributions:

- i)* The data exchanged between robots of the same subgroup, *i.e.*, network, is studied in depth and a rationale is presented for each different situation within the *RDPSO* context so as to minimize the communication overhead (section 6.1);
- ii)* The traditional *AODV* reactive routing protocol is extended based on the *RDPSO* dynamics to minimize the number of updates regarding the routes connecting pairs of robots, thus avoiding unnecessary flooding (section 6.2);
- iii)* Based on the proposed approaches, the communication complexity of the *RDPSO* is evaluated using both physical and virtual robots in a large indoor environment (section 6.3).

Sections 6.4 and 6.5 outline the discussion and main conclusions of this chapter.

6.1 Sharing Information in the *RDPSO*

As previously described, the *RDPSO* ensures the connectivity of the network (*cf.*, $\chi_4[t]$ term in described in section 4.4). Nevertheless, how this is carried out in practice without overloading the communication channel needs to be addressed. Moreover, the communication packet structure shared between robots needs to be specified and a rationale behind it should be introduced. Generally, the packet data structure may be illustrated as presented in Figure 6.1.

Header bit [0,1]	Data byte(s)
0	Local Broadcast to neighbours
1	Broadcast to whole subgroup

Figure 6.1. General communication packet structure for a subgroup of N_s robots.

It is noteworthy that the broadcast to the whole subgroup should be avoided as it represents a high communication complexity. In brief, in order to broadcast information to the whole subgroup by multi-hop communication, the message needs to be addressed to each Robot *ID*. The number of bytes necessary for the main message, *i.e.*, Data byte(s), will depend on the message itself. For instance, if a robot wants to share its position and considering a planar scenario, then two bytes may be enough to represent the coordinate on each axis.

6.1.1 Ensuring Connectivity

In section 5.1, an initial deployment strategy denoted as *EST* was presented. The *EST* was introduced as an autonomous, realistic and fault-tolerant initial deployment strategy based on the *RSSI* signal quality signal. The initial deployment was able to ensure that each exploring robot would be able to

communicate with k neighbours from the same subgroup, $k \in \mathbb{N}$, thus ensuring that the *MANET* is k -connected. After the initial deployment process is concluded, robots explore the environment while ensuring the same k -connectivity of the subgroup by defining $\chi_4[t]$ as a set of attractive and repulsive forces (section 5.2).

Based on the strategy presented in section 5.2, it is possible to ensure the k -connectivity of the network by simply sharing the position to the k neighbours. Therefore, only taking into consideration the information of the N_b robots within the one-hop path (*i.e.*, neighbours) would allow ensuring the connectivity of the whole subgroup. Figure 6.2 presents the packet structure of communication for this particular situation.

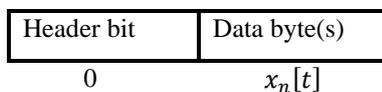


Figure 6.2. Communication packet structure that allows robots in maintaining the *MANET* k -connectivity within their subgroup of N_s robots.

For instance, as previously stated, an alternative to broadcasting the position to the N_b neighbors would be the use of strategies to find the teammates position under their visual range, *e.g.*, robots equipped with *LRFs* can use retro-reflective markers for recognition (Kulkarni & Venayagamoorthy, 2010). To that end, one should ensure that the sensing radius R_w is equal or superior to the maximum distance of neighbours d_{max} , which depends on the minimum inter-robot signal quality *RSSI*.

6.1.2 Converging to the Optimal Solution

As previously presented in section 4.1, $\chi_2[t]$ represents the best positions of the social component. Therefore, robots from the same active subgroup, *i.e.*, not in the socially excluded subgroup, need to share their best cognitive solution $f_n[t]$ and current position $x_n[t]$, so as to compute the position of the robot that has the best social solution. For instance, if one wishes to find a gas leak, the best performing robot will be the one with the highest solution, *i.e.*, $\max_{n \in N_s} f_n[t]$. Nevertheless, efficiently sharing this information may allow to drastically reduce the communication complexity of the *RDPSO*. For instance, if a robot from the active subgroup was unable to improve, then the information about its position and solution is irrelevant to the group, *i.e.*, the collective behaviour will not change. Therefore, and as a rule of thumb, a robot only needs to share its current solution and position if it is able to improve its best cognitive solution, *i.e.*, $f_n[t + j] > f_n[t]$, $j \in \mathbb{N}$. Otherwise, and as robots are able to memorize the best solution of the subgroup and corresponding position so far, without signif-

ificantly increase the memory complexity, robots will simply continue computing their algorithm without communicating. Figure 6.3 represents the packet structure sent from a robot that was able to improve its solution.

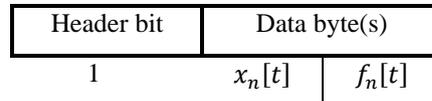


Figure 6.3. Communication packet structure that allows robots from active subgroups to cooperatively converge to the solution. This packet is only sent if a robot improves its best cognitive solution.

Note that this significantly reduces the communication complexity as this data needs to be exchanged between all teammates, *i.e.*, broadcasted to the whole subgroup by means of multi-hop communication. For instance, in the experiments from section 4.7.2, a setup of 4, 8 and 12 *eSwarBots* on a small scenario with one optimal and one sub-optimal solution was presented (*cf.*, Figure 4.11). As Figure 6.4 depicts, using 12 robots represent the most critical situation tested regarding the chances that the subgroup has to improve. Even so, in a population of 12 robots under the 80 trials of 180 seconds each, it was possible to observe that a robot was only able to improve in approximately 15% of the iterations, *i.e.*, only approximately 15% of the information shared is useful to the collective performance. As the number of robot decreases for the same scenario, the probability that a robot has to improve also slightly decreases, thus slightly decreasing the amount of useful information (Figure 6.4).

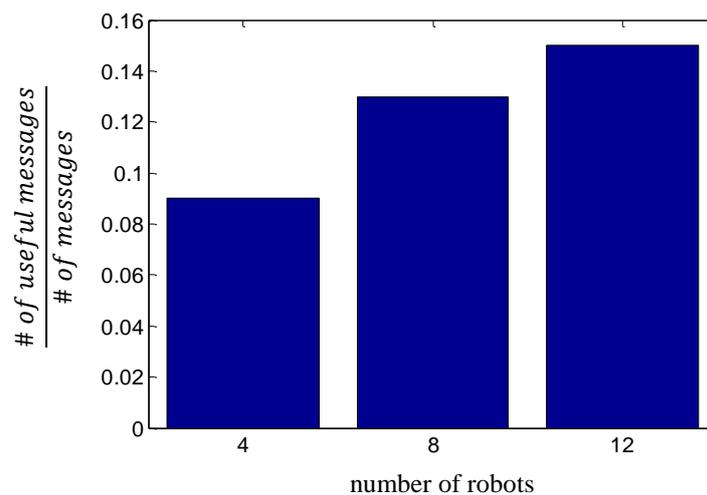


Figure 6.4. Ratio between the number of useful messages and the total number of messages retrieved from the experimental evaluation on section 4.7.2.

It is noteworthy that the amount of useful information will vary depending on several conditions (*e.g.*, number of robots, scenario, mission objectives, among others). Nevertheless, efficiently sharing information based on the herein proposed strategy will always significantly reduce the communication complexity of the algorithm as robots will not always improve at each iteration. After this analysis on the data exchanged between robots from active subgroups, next section shows an efficient way to share information between robots within the socially excluded subgroup.

6.1.3 Avoiding Sub-Optimality

As previously presented in section 4.5, the way the *RDPSO* handles sub-optimal avoidance is by socially excluding robots that have nothing to offer to the group, *i.e.*, that are unable to improve for a certain stagnancy threshold. Note that, as previously stated in section 4.5, although excluded robots do not help the swarm searching for the optimal solution, they are always aware of their best cognitive solution. That being said, the only regular information excluded robots need to share is their current position to their neighbours so as to maintain the *MANET* connectivity (*cf.*, Section 6.1.1).

However, if an active subgroup continues to improve for a certain amount of time, there will be a probability to be rewarded with the best performing robot from the socially excluded group. Moreover, the socially active subgroup will also have a small probability of creating a new subgroup from the best performing robots from the socially excluded subgroup. Therefore, when excluded robots receive a calling from an active subgroup, they will broadcast their best cognitive solutions and respective positions to the whole socially excluded subgroup by means of multi-hop communication (*cf.*, Section 6.1.2). Thereby, they will be able to assess the best performing excluded robots so far and evaluate which ones would be a part of an active subgroup.

Although one wishes to avoid broadcasting to the whole multi-hop network, this event will only occur from time to time since it depends on the constant improvement of subgroups and a probability of successful calling. Furthermore, an adequate choice on the routing protocol may allow overcoming or, at least, minimizing the broadcast overhead.

6.2 Routing Protocol

In *MANETs*, the communication between source and destination nodes may require traversal of multiple hops. Since the introduction of such networks, a community of researchers has proposed a variety of routing algorithms, mainly divided into two classes: *i) proactive*; and *ii) reactive*. In the first class, every node maintains a list of destinations and their routes by processing periodic topology broadcasts originated by each node in the network. In reactive routing protocols, nodes maintain their

routing tables on a need-to-use basis. For more information about those two classes please refer to (Natesapillai, Palanisamy, & Duraiswamy, 2009).

Although many works have been comparing such routing protocols, those have been mostly carried out in simulation and outside the scope of swarm robotic applications, wherein a large quantity of highly dynamic nodes need to be considered (Lee, Gerla, & Toh, 1999) (Bertocchi, Bergamo, Mazzini, & Zorzi, 2003) (Wu, Xu, Sadjadpour, & Garcia-Luna-Aceves, 2008). Within such assumptions, the class of proactive routing protocols utterly falls apart. Besides being unsuitable to use in highly mobile nodes, proactive routing requires a high communication cost to constantly maintain all topological information.

Therefore, and as swarm robotics aims for scalability under an increasing numbers of robots and mobility rate within the network, this work will focus on reactive routing protocols. One of the most well-known reactive protocols is the *Ad hoc On-demand Distance Vector* routing protocol (Perkins, Royer, & Das, 1999).

6.2.1 Ad hoc On-demand Distance Vector (AODV)

The Ad hoc On-demand Distance Vector (AODV) routing protocol (Perkins, Royer, & Das, 1999) is one of the most adopted reactive *MANET* routing protocols. This protocol exhibits a good performance on *MANETs*, thus accomplishing its goal of eliminating source routing overhead. Nevertheless, at considerably high rates of node mobility, it requires the transmission of many routing overhead packets. Despite this limitation, the AODV has been extensively applied in most wireless equipment, such as the one used on the robotic platforms *eSwarBots*; the Original Equipment Manufacturers *RF (OEM-RF) XBee Series 2* from *Digi International* (*cf.*, section 3.2.3).

Under the AODV protocol, when a robot A needs to communicate to robot B, it broadcasts a *route discovery* message to its neighbours (*i.e.*, local broadcast), including the last known sequence number for that destination (Broch, Maltz, Johnson, Hu, & Jetcheva, 1998). The *route discovery* is flooded through the network until it reaches a robot that has a route to the destination. Each robot that forwards the *route discovery* creates a reverse route for itself back to robot A. When the *route discovery* reaches a robot with a route to robot B, that robot generates a *route reply* that contains the number of hops necessary to reach robot B and the sequence number for robot B most recently seen by the robot generating the *route reply*. Each robot that participates in forwarding this *route reply* back toward robot A creates a forward route to robot B. Hence, each robot remembers only the next hop and not the entire route.

In order to maintain routes, AODV normally requires each robot to periodically transmit a *hello message*. Within the *RDPSO* algorithm, this may be accomplished at each step of the algorithm, *i.e.*,

after reaching a desired position $x_n[t + 1]$, thus benefiting from the need to share its current position in order to ensure *MANET* connectivity (Section 6.1.1). A previously defined link may be considered to be broken if a robot does not receive three consecutive *hello messages* from a neighbour. Under that condition, any upstream robot that has recently forwarded packets to a destination using that link is notified via an *unsolicited route reply* containing an infinite metric for that destination. Upon receipt of such a *route reply*, a robot must acquire a new route to the destination using the *route discovery* once again.

6.2.2 RDPSO based AODV

Although the mechanics of the *AODV* are quite transparent for users in most wireless technology (e.g., *OEM-RF XBee Series 2*), one may need to extend the original *AODV* features so as to further adapt it to the application itself (Abedi, Fathy, & Taghiloo, 2008). In this work, the *AODV* is extended based on two key elements:

- As the teams of robots begin connected by mean of the *EST* initial deployment (cf., section 5.1), a *node discovery* functionality was introduced;
- The mobility of robots within the *RDPSO* behaviour is taken into account so as to establish more stable routes.

The *node discovery* basically allows discovering the *IDs* of all robots that have joined the network. Each robot will then broadcast a *node discovery* command throughout the network. All robots that receive the command will send a response that includes their own address. A timeout is defined by the *node discovery* sender, thus allowing specifying an amount of time a robot will spend in discovering its teammates. In other words, the *node discovery* functionality is highly suitable as the *RDPSO* handles multiple subgroups and it is unfeasible to predefine a static population of specific robots to form a subgroup in advance. Moreover, such strategy avoids the need to configure the address of each robot independently as each robot will acquire the default *ID* of its teammates in the beginning of the mission. Therefore, after each subgroup is deployed within the scenario, the very first action robots must perform is the *node discovery* command. Afterwards, the *route discovery* will be carried out and the mission will start.

Subsequently, it is possible to improve the *AODV* based on the mobility of robots, by first understanding how robots may generally behave within the *RDPSO* algorithm. As previously presented in chapter 4, the *RDPSO* model depends on the sensed information, both cognitive and social, and the inertial coefficient based on the approximate fractional difference of order α . That being said, a robot may predict where a neighbour, i.e., a one-hop robot, will be in the next iteration by knowing its previous positions, its best position so far and the social solution of the group. The later situation

is the simplest one as each robot is always aware of the best solution of the whole group so far (section 6.1.2). Hence, this requirement does not increase the memory complexity of the algorithm at all.

Similarly, a robot may know the best position of its neighbours as it is intrinsic to the communication packet structure shared when robots improve their individual solution (section 6.1.2). For this situation, each robot will need to keep the position received by robots when they are able to improve *i.e.*, $f_n[t + j] > f_n[t]$, $j \in \mathbb{N}$. Nevertheless, the position of non-neighbour robots may be discarded as this is a distributed strategy that only considers information from one-hop robots. Therefore, this results in an addition of the memory complexity *per* robot equal to the number of neighbour robots, *i.e.*, $\mathcal{O}(N_b)$. Note, however, that this only represents memorizing twice N_b bytes necessary to represent the planar best position of each neighbour robot.

The most memory demanding situation will be inevitably memorizing the position of neighbours over time. Based on equation (4.11), one may compute the motion of robots with the information of the four last steps, *i.e.*, $v_n[t - j]$, $j = 0, \dots, 3$. As neighbour robots share their current position $x_b[t]$, a robot needs to memorize the two consecutive positions $x_b[t]$ and $x_b[t - 1]$ of all its neighbours so as to calculate their current velocity $v_b[t]$. In other words, a robot will need to keep track of the position of all its N_b neighbour robots for the last 5 steps to estimate their position, *i.e.*, $\mathcal{O}(5N_b)$.

In sum, to extend the *AODV* based on the *RDPSO* behaviour, one needs to increase the memory complexity of robots by $\mathcal{O}(6N_b)$. Note that this is a small increment to the memory complexity of each robot when compared to the benefit that this novel mechanism may provide in reducing the communication complexity of the whole swarm.

Having the information described above, each robot may be able to predict all neighbours' next position $x_b[t + 1]$ by means of equation (4.5), (4.6) and (4.11), while considering solely their cognitive and social components. Nevertheless, as the *RDPSO* is endowed with a stochastic effect, *i.e.*, r_i , $i = 1, 2, 3, 4$, it is almost impossible for a robot to estimate its neighbours' exact next position accurately. However, one may improve the precision of such estimate by considering the expected value of the uniform random parameters. In other words, for the position estimate of neighbours, a deterministic simplified version of the *RDPSO* is considered. The deterministic simplified *RDPSO* is obtained by setting the random numbers to their expected values:

$$E(r_i) = \frac{1}{2}, i = 1, 2. \quad (6.1)$$

Thus, for the deterministic simplified *RDPSO*, replacing the random factors r_i by $\frac{1}{2}$, equations 1, 2 and 3 may be rewritten in a single equation as:

$$\begin{aligned}
x_{n,b}^e[t+1] = & \left(-1 - \alpha + \frac{1}{2}\sum_{i=1}^4 \rho_i\right) x_{n,b}[t] + \left(\frac{1}{2}\alpha\right) x_{n,b}[t-1] + \left(\frac{1}{3}\alpha + \right. \\
& \left. \frac{1}{6}\alpha^2\right) x_{n,b}[t-2] + \left(-\frac{1}{24}\alpha^3 - \frac{1}{24}\alpha^2 + \frac{1}{12}\alpha\right) x_{n,b}[t-3] + \\
& \left(\frac{1}{24}\alpha^3 - \frac{1}{8}\alpha^2 + \frac{1}{12}\alpha\right) x_{n,b}[t-4] + \frac{1}{2}\rho_1\chi_{1n,b}[t] + \frac{1}{2}\rho_2\chi_{2n,b}[t],
\end{aligned} \tag{6.2}$$

in such a way that $x_{n,b}^e[t+1]$ represents the position of robot b estimated by its neighbour n . Note that the remaining parameters in equation (6.2) are explained on section 4.1. Although the estimated position is unlikely to be exactly the same as the real position, *i.e.*, $x_{n,b}^e[t+1] \neq x_b[t+1]$, a good approximation may be enough to select if robot b may be a candidate to be the intermediate in route between source and destination robots.

Therefore, to improve the *AODV* routing protocol, when a source robot wants to send a packet to a destination robot, it will first estimate the next position of neighbour robots. Then, it will recognize the intermediate robot that can participate in the routing of the message. The robot can be selected as the next hop if its estimated position is the closest to the destination robot, *i.e.*, the one with the smallest Euclidean distance.

$$ID_b = \underset{b \in N_b}{\operatorname{argmin}} d(x_{n,b}^e[t+1], x_f^e[t+1]), \tag{6.3}$$

wherein ID_b will represent the *ID* between all robot n 's neighbours that has the smallest distance to the destination robot and $x_f^e[t+1]$ the estimated position of the destination robot. After the message reaches the selected robot, the same process is carried out in order to assess the neighbour robot that would yield the next most fitted hop.

Henceforth, source robot, destination robot and candidate robot for next hop are the inputs of the herein proposed strategy for each robot. It is noteworthy that the information that will be used from the destination robot will be the last known information obtained from the broadcast to the whole subgroup (*cf.*, Section 6.1.2). Although the destination robot is likely to have changed its position in the meanwhile, the idea is to have an estimate on the region where to send the message to and choose the most adequate path.

Routes established within such strategy are more stable and have less overhead than the original *AODV* routing method. Nevertheless, this is a greedy distributed strategy and it may happen that a robot cannot find any intermediate node as next best hop. For instance, the source robot may choose the incorrect neighbour robot based on its location without knowing that it may not have any other

neighbours at all besides itself. In this situation, *i.e.*, when a message returns to a robot that already forwarded it or to the source robot, then the common *AODV* mechanism of *route discovery* is used between that robot and the destination one (*cf.*, Section 6.2.1).

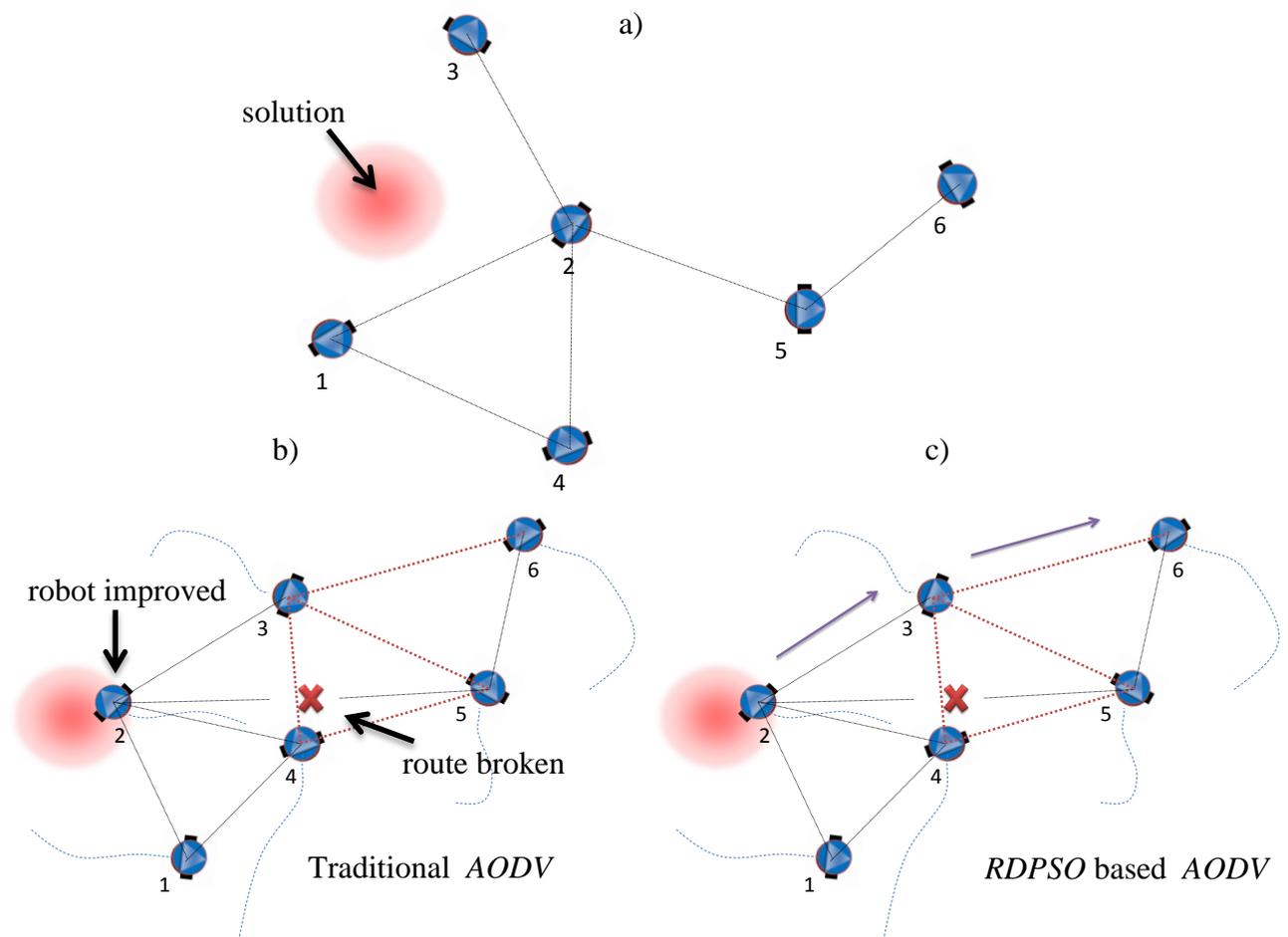


Figure 6.5. *RDPSO* based *AODV* routing protocol. Red bolder lines between robots represent that there exists a possible link between them but that the *AODV* protocol is unaware of. a) The robots start connected by means of the *EST* initial deployment strategy, thus enforcing the *MANET* connectivity of the whole subgroup (section 5.1). The node discovery and route discovery allows to retrieve the *ID* of all robots and build the routes between them (blue thin lines). b) After a while, robot 2 improves and tries to broadcast its new solution and position to the whole network. However, as robot 2 is unable to communicate with robot 6 by means of the route previously built using *AODV*, a new route discovery needs to be sent (red thick lines). c) Using the *RDPSO* based *AODV* will allow robot 2 to choose the neighbour that is near robot 6, *i.e.*, robot 3, that will forward the message to its destination, *i.e.*, robot 6.

To easily understand the herein proposed strategy, Figure 6.5 presents an illustrative example of a subgroup under the *RDPSO* algorithm. In the beginning (Figure 6.5a), and due to *RDPSO* main mechanisms (chapter 4), robots are able to communicate between themselves, thus guaranteeing the *MANET* connectivity. Since the *AODV* routing protocol is the one adopted in this work, its main

mechanism to retrieve all routes between robots is fulfilled, *i.e.*, *route discovery*, as presented in Section 6.2.1. The routes between robots are represented by the blue thin lines that connect them. Due to the particularities of the *RDPSO*, the *node discovery* is carried out so as to retrieve the *IDs* of all robots within the same subgroup. While any robot improves, they will continue exploring the scenario informing its neighbours about its position to maintain the *MANET* connectivity (Section 6.1.1). After a while (Figure 6.5b), robot 2 is able to improve its cognitive solution, thus informing all other robots within the subgroup (Section 6.1.2). Since robot 2 cannot communicate with robot 6, and considering the traditional *AODV*, a new route needs to be found, *i.e.*, the *route discovery* needs to be fulfilled once again. Those new routes are represented by the red thick lines that connect the robots. Nevertheless, the *route discovery* mechanism requires successive local broadcasts that may overload the communication channel. Figure 6.5c depicts the mechanism inherent to the *RDPSO* based *AODV*. Within such strategy, robot 2 will choose the nearest neighbour that presents the smallest distance to robot 6, *i.e.*, robot 3. As robot 3 is able to directly communicate to robot 6, it will forward the message to it.

The whole *RDPSO* communication procedure for a robot n may be briefly summarized as presented by Algorithm 6.1 (Couceiro, Fernandes, Rocha, & Ferreira, 2013 (In Press)). Note that Algorithm 6.1 only focus on the shared information between robots and the routing protocol. For a detailed description of the *RDPSO* main behaviour please refer to Algorithm 4.4.

Next section evaluates the communication complexity of the *RDPSO* with and without the herein proposed strategies.

6.3 Experimental Results

This section is divided into three sub-sections exploring and comparing the properties of the “regular” version of the *RDPSO* (following the concepts previously presented before this chapter) to its counterpart version proposed in this chapter – the “optimized” *RDPSO* – which aims at reducing the communication overhead.

```

starting() // wait for information about initial position  $x_n[0]$  and swarmID
fullIDs = node_discovery(swarmID) // full list of robot IDs from the same subgroup swarmID
routesIDs = route_discovery(fullIDs) // list of routes within the same subgroup swarmID
Main_Loop
  If swarmID  $\neq$  0 // it is not an excluded robot
    send(0,  $x_n[t]$ ) // local broadcast may be avoided applying recognition techniques in visual range (section 6.1.1)
     $f_n[t] = \text{sense}()$  // evaluate individual solution  $f_n[t]$ 
    If robot_improved( $f_n[t-1], f_n[t]$ ) // robot n will globally broadcast its current solution and position (section 6.1.2)
      listIDs = send(1,  $x_n[t], f_n[t]$ ) // listIDs is an array of robot IDs that did not received the message
      resend(listIDs) // use the RDPSO based AODV
      If call_robot() // robot n may call a new robot from the excluded group to its subgroup
        send(0, swarmID_call) // broadcast the possibility to receive a new robot
      If create_swarm() // robot n may create a new subgroup from the excluded group
        send(0, swarmID_new) // broadcast the possibility to create a new subgroup
    If received_fwdmsg( $ID_f, x_n[t], f_n[t], \text{routesIDs}$ ) //  $ID_f$  represents the ID of the destination robot
      resend( $ID_f$ ) // use the RDPSO based AODV
    Else // it is an excluded robot
      wander() // section 6.1.3
      send(0,  $x_n[t]$ ) // local broadcast may be avoided applying recognition techniques in visual range (section 6.1.1)
       $f_n[t] = \text{sense}()$  // evaluate individual solution  $f_n[t]$ 
      If received(swarmID_call) Or received(swarmID_new) // call for a new robot or subgroup received
        listIDs = send(1,  $x_n[t], f_n[t]$ ) // listIDs is an array of robot IDs that did not received the message
        resend(listIDs) // use the RDPSO based AODV
  End // until stopping criteria (e.g., convergence, time)
resend(listIDs) // RDPSO based AODV function
  For  $i = 1$  to len(listIDs) // check unreached robots one by one from listIDs
    For  $j = 1$  to  $N_b$  // estimate position of its  $N_b$  neighbors (equation 5)
       $b = \text{fullIDs}(j)$ 
       $x_{n,b}^e[t+1] = \text{estimate\_pos}(x_{n,b}[t], \dots, x_{n,b}[t-4], \chi_{1,2n,b}[t])$ 
       $ID_b = \min_{b \in N_b} d(x_{n,b}^e[t+1], x_{i,\text{listIDs}(i)}^e[t+1])$  // find closest neighbour to robot listIDs(i)
      If  $ID_b = \text{listIDs}(i)$  // the unreached robot listIDs(i) is a neighbor
        send( $ID_b, x_n[t], f_n[t]$ ) // send message directly to robot  $ID_b$ 
      Else
        If find( $ID_b, \text{routesIDs}$ ) // the robot  $ID_b$  already exists in the route necessary to reach listIDs(i)
          routesIDs = route_discovery(fullIDs) // necessary as it is unable to reach the destination robot
          send(listIDs(i),  $x_n[t], f_n[t]$ ) // send message to robot listIDs(i)
        Else
          routesIDs = update_route( $n, ID_b, \text{listIDs}(i)$ ) // update the route necessary to reach listIDs(i)
          send( $ID_b[\text{listIDs}(i)], x_n[t], f_n[t], \text{routesIDs}$ ) // send message to robot  $ID_b$  so as to reach listIDs(i)
    End
  End
End

```

Algorithm 6.1. Sharing information within the RDPSO.

6.3.1 Real-World Experiments

In this section, it is explored the effectiveness of the proposed communication methodology on a group of 15 *eSwarBots*, *i.e.*, $N_T = 15$, performing a distributed exploration task under the RDPSO behaviour. The same 200 m^2 indoor scenario from section 5.3.1 was considered (the sports pavilion illustrated in Figure 5.7).

As this chapter emphasises on the analysis the communication complexity of the RDPSO, the convergence of the algorithm itself was neglected. This may only be considered as the herein proposed communication methodology does not affect the decision-making of robots since the same useful information is shared between teammates regardless on the protocol used. Therefore, as *eSwarBots* are equipped with LDR light sensors (*cf.*, section 3.2.1), their solution was affected by the current

room lighting conditions, either natural or not. Just for the purpose of illustrating the variability of light over time, Figure 6.6 represents the intensity values of light $F(x, y)$ over a day of experiments.

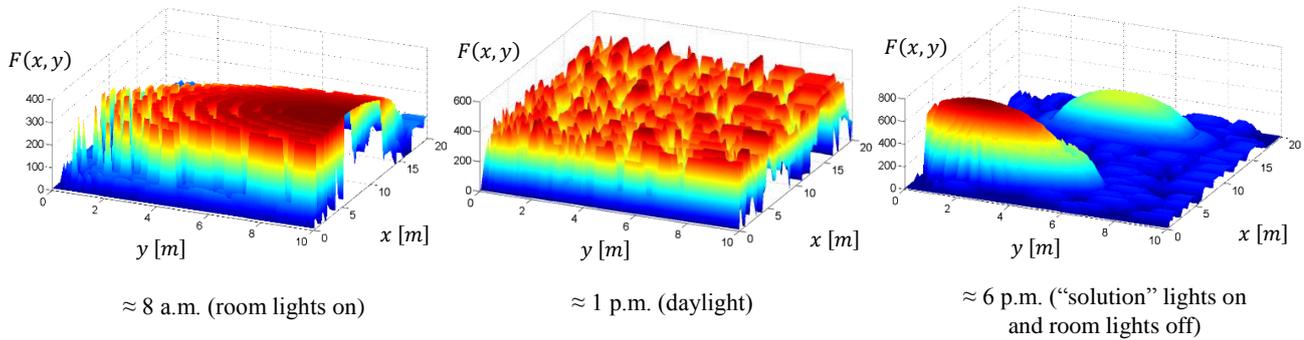


Figure 6.6. Virtual representation of the target distribution over day under different lighting conditions for the scenario depicted in Figure 5.7.

Exactly the same *RDPSO* parameterization from Table 5.1 was considered, in which one should highlight the number of 20 trials for each set of experiments. In other words, the “regular” *RDPSO* (first set of trials) was compared with the “optimized” *RDPSO* (second set of trials), *i.e.*, the extension of the *RDPSO* based on the strategies presented in this chapter.

As previously stated, by employing the optimized communication strategies presented in this chapter, it is expected to significantly reduce the communication cost of the *RDPSO* algorithm. One of the methods to evaluate the communication cost consists in counting the average number of packets sent and the processing time to handle the communication procedure, *i.e.*, pause time, for each robot over the 360 seconds of each trial. The number of packets sent was easy to retrieve since a robot under the “regular” *RDPSO* communicates after each iteration step to its own subgroup, *i.e.*, if it is a subgroup of 5 robots then the robot will send 4 packets, whereas in the “optimized” one the robot follows the rules presented in section 6.1. Regarding the pause time inherent to the whole communication procedure, a timer was used to count the time before entering the function that allows for a robot to send and receive the data packets from its own subgroup. It is noteworthy that during that time the robot is unable to perform any other action. Table 6.1 compares the average (*AVG*) and standard deviation (*STD*) communication cost of the *RDPSO* with and without the proposed strategy.

Table 6.1. Communication cost.

	AVG \pm STD Number of packets	AVG \pm STD Pause time [seconds]
“Regular” <i>RDPSO</i>	742 \pm 24	126 \pm 4
“Optimized” <i>RDPSO</i>	415 \pm 37	39 \pm 7

As it is possible to observe, the number of messages significantly decreases using the proposed methodology. This is highly valuable as the number of exchanged messages has a high influence on the power consumption of each robot. On the other hand, reducing the number of times each robot needs to share its information allows reducing the time allocated for such task. Note that this is not proportional since that, in the “optimized” *RDPSO*, robots still communicate at each iteration step to their neighbours since *eSwarBots* are not equipped with sensing capabilities that allows retrieving teammates position. Communication to the whole subgroup is constrained by how each robot improves over time. In other words, while each robot allocates approximately 35% of the mission time to exchange information within the “regular” *RDPSO*, this novel approach allows reducing this value to approximately 10%, thus increasing robots’ mobility. This is due to both requiring less data to be exchanged (section 6.1) but also the minimization of *route discovery* messages inherent to the *RDPSO* based *AODV* (section 6.2). Therefore, the herein proposed approach would be more energy efficient and allow each robot to spend less time without moving than the “regular” one.

Nevertheless, the efficiency of a communication paradigm cannot be measured by only comparing the total number of exchanged packets. One of the most well-known performance metrics to evaluate the network throughput is the packet delivery ratio (Natesapillai, Palanisamy, & Duraiswamy, 2009). The packet delivery ratio can be calculated by dividing the number of packets received by a robot by the number of packets sent to it. This allows specifying the packet loss rate, which limits the maximum throughput of the network. Therefore, the average packet delivery ratio was evaluated based on the number of robots within the same subgroup (either active subgroup or the socially excluded subgroup). As previously mentioned in section 4.5, the *RDPSO* uses a “punish”-“reward” mechanism to avoid sub-optimality by socially excluding and including robots within active subgroups. In other words, at some point over the 360 seconds of each trial, *i.e.*, 7200 seconds for each set, a subgroup may be formed by only two robots or even by the 15 robots from the population. Figure 6.7 depicts the average packet delivery ratio when subgroups are formed by a specific number of robots, even if some of those cases, namely subgroups formed by less than 3 robots or by more than 10 robots, only occur in some few occasions (around 5% of the whole time).

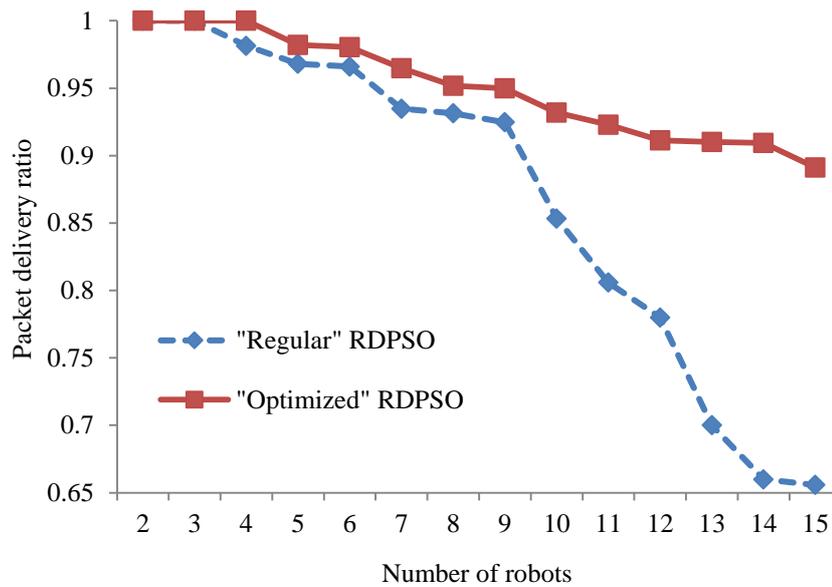


Figure 6.7. Packet delivery ratio within robots from the same subgroup as a function of the subgroup size.

As one may observe, there is a sharp decrease on the packet delivery ratio for the “regular” *RDPSO* when a subgroup is formed by more than 10 robots, dropping down to approximately 65% for a maximum network load of 15 robots. This is explained by the high number of exchanged messages that, for a network load above 10 robots, does not satisfy the capacity of the buffer or the packet buffering time exceeds the time limit. As the “optimized” *RDPSO* significantly decreases the number of exchanged messages (*cf.*, Table 6.1), robots are still capable of receiving more than 90% of the data even within a subgroup of 15 robots.

The first key contribution of this chapter, *i.e.*, the efficient way to share information within the *RDPSO* algorithm (section 6.1), is the major reason for such significant reduction in both communication cost (Table 6.1) and number of dropped packets (Figure 6.7). Although the adapted *AODV* improves the communication efficiency of the *RDPSO* algorithm, it is still not clear how much advantageous this specific extension may be so far.

The routing overhead has been frequently used in the literature to evaluate routing algorithms, being commonly represented by the ratio between the number of *route discovery* messages and the number of data packets (Natesapillai, Palanisamy, & Duraiswamy, 2009). Let us then compare the routing overhead of the “regular” *RDPSO* with the “optimized” *RDPSO*, for each different team size from 2 to 15 robots under the 7200 seconds of each set of trials (Figure 6.8).

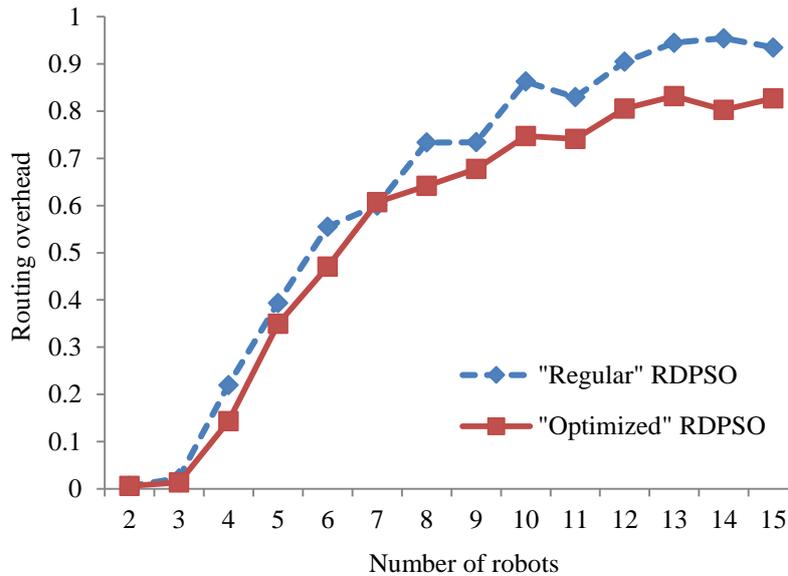


Figure 6.8. Routing overhead within robots from the same subgroup.

Once again, the “optimized” *RDPSO* clearly overcomes the “regular” one for larger population of robots. Even though the number of data packets is reduced due to the efficient way to share information between robots (section 6.1), the number of *route discovery* messages decreases more significantly (section 6.2), thus resulting in a smaller routing overhead for a larger number of robots. It would be expected to have a worse routing overhead ratio when robots communicate less while they are moving since the routes would be completely outdated. Nevertheless, the *RDPSO* based *AODV* is able to reduce the number of *route discovery* messages in such a way that it allows overcoming that issue. This is due to the proposed geographically-based *AODV* that takes into account the dynamics of the *RDPSO*, thus creating on-the-fly routes (Figure 6.8). However, one needs to analyse the number of hops forming such routes to understand how better those are when compared to the alternatives returned by the traditional *AODV*.

The average hop count may be represented by the sum of the number of hops necessary to deliver the packets from their sources to destination divided by the total number of successful delivered packets. The average hop count is measured in number of hops.

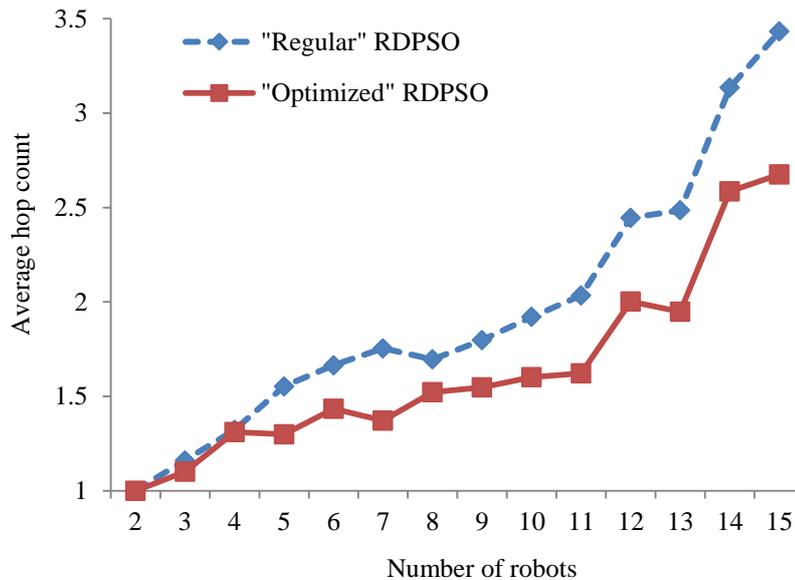


Figure 6.9. Average hop count within robots from the same subgroup.

As Figure 6.9 depicts, the applicability of the novel *AODV* routing protocol may be observed for a subgroup of, at least, 5 robots. For smaller subgroups, the improvement of the *RDPSO* based *AODV* is meaningless which, on the other hand, turns out to be a worse alternative to the traditional *AODV* since it slightly increases the memory complexity of the algorithm (*cf.*, section 6.2). However, as analysing swarm algorithms within small populations may not represent the required collective performance (*cf.*, (Beni, 2004)), let us focus on larger subgroups, *i.e.*, above 5 robots. As it is possible to observe, in some situations, the *RDPSO* based *AODV* reduces around 20% the number of required hops to deliver a packet. Although this may not seem relevant, this contributes to a smaller pause time and, consequently, a higher mobility of the robots. Moreover, reducing the number of hops necessary to deliver the packets also reduces the energy consumption of each robot, thus increasing the autonomy of the whole swarm.

It is noteworthy that the two key contributions of this chapter, *i.e.*, the efficient way to share information within the *RDPSO* algorithm and the adapted *AODV* routing protocol, result in significant differences compared to its “regular” counterpart. Moreover, such differences increase with the number of robots, thus improving the scalability of the *RDPSO* algorithm. Yet, in order to further explore inter-robot communication dynamics under the “optimized” *RDPSO*, let us observe how such information is shared within different social statuses, *i.e.*, within socially active and excluded subgroups.

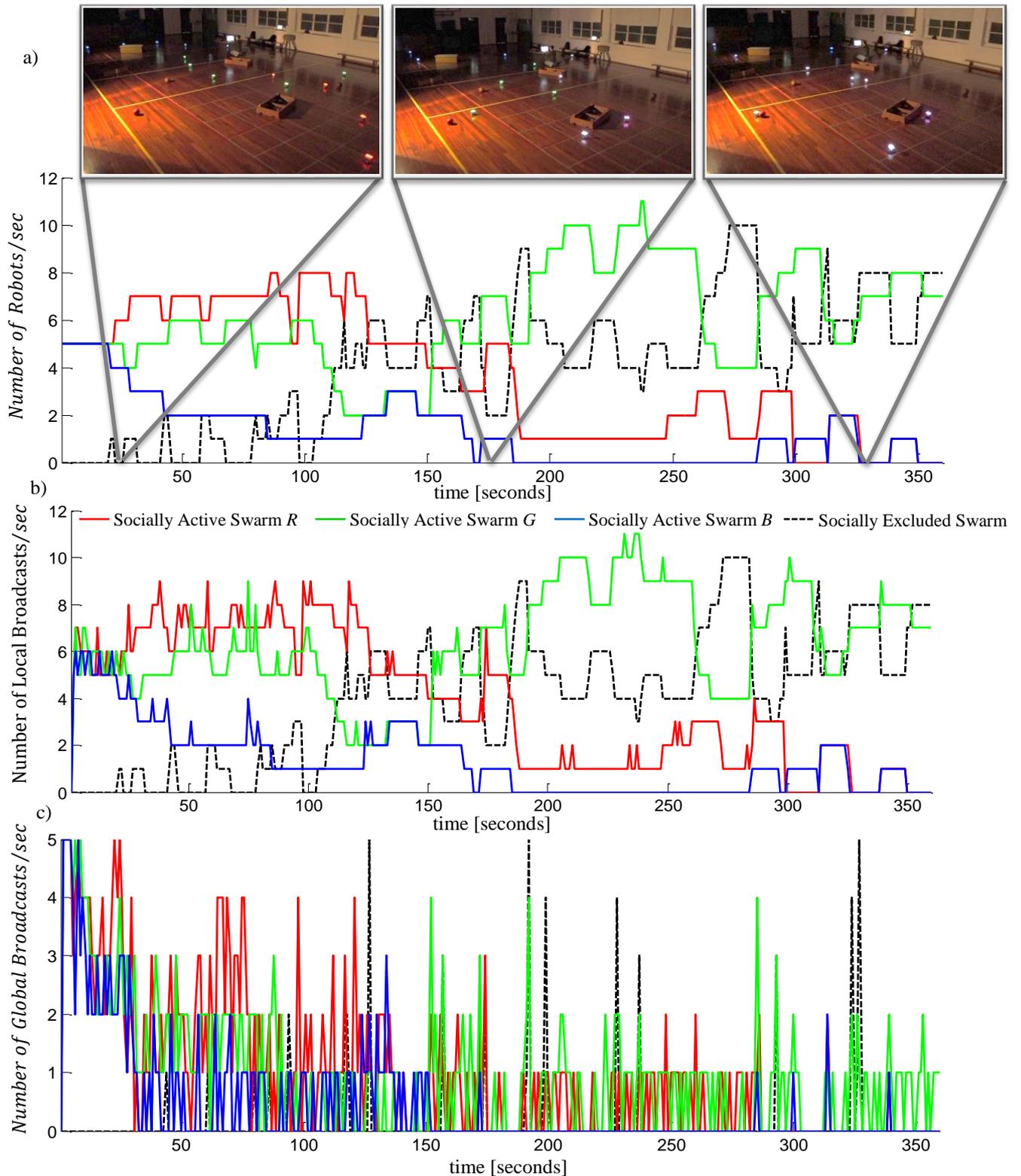


Figure 6.10. Evolution of robotic subgroups over a trial of 360 seconds. a) Population size; b) Number of local broadcasts; c) Number of global broadcasts.

In order to achieve this, the number of local and global broadcasts within each subgroup was analysed. For a better understanding of how robots within the *RDPSO* evolve, let us take a look at one of the 20 trials in which the “optimized” *RDPSO* was evaluated, *i.e.*, a single trial of 360 seconds.

Figure 6.10 depicts the distribution of robots (Figure 6.10a) and highlights the respective total number of local (Figure 6.10b) and global broadcasts (Figure 6.10c) within each subgroup over time. While the coloured lines correspond to each socially active subgroup, respectively R (red), G (green) and B (blue) subgroups, the dark dashed line corresponds to the socially excluded subgroup. The mission starts with 5 robots within each active subgroup as previously stated. As one may observe, the number of workers in active subgroups tends to decrease over time. This is an expected phenomenon as the resources begin to dwindle over time, *i.e.*, in this specific case study robots become unable to find ever improving light intensities. At some point it is even possible to observe that subgroups B and R extinguish while subgroup G proliferates, thus reaching a population of up to 11 robots. This happens right before the population in subgroup G decreases to approximately 7 robots. Consequently, this leads to an increase of socially excluded robots with a maximum of 10 robots after the 4th minute. Regarding the local broadcasts, such temporal variations would be expected by considering the rules previously stated in section 4.5. The local broadcasts necessary to maintain the network connectivity remain at each step of the algorithm, thus presenting a proportional amount to the number of robots within each subgroup. Such proportionality is only broken when a socially active subgroup claims a new robot or tries to create a new subgroup (small peaks observed in the coloured lines). A rationale behind the global broadcasts is harder to achieve. As one may observe, in general, socially active robots present a higher amount of messages flooded through the whole subgroup. This is interesting to observe as such global broadcast is related to subgroups' improvement that requires the global consent of the population. As a result, such global broadcasts diminish over time. This kind of global message seems to be significantly less recurrent in socially excluded subgroups.

As one may observe, the time a certain amount of robots is socially excluded may not correspond to the time that the same amount is socially active. Therefore, to further compare the information shared within the different social statuses over the 7200 seconds of the whole set of experiments, a simple normalization of the data over time was adopted. Figure 6.11 depicts the average number of local and global broadcasts within each subgroup configuration. As a rule of thumb, the local broadcasts increase almost proportionally to the population of robots. This may be observed in both socially excluded and active subgroups with a minor difference between both. The main difference between robots belonging to different social statuses may be seen in the number of global broadcasts. Socially excluded robots barely communicate to the whole swarm. In fact, such communication only depends on the improvement of socially active subgroups. Hence, as the overall amount of socially active robots decreases, the number of socially excluded robots increases and the probability of success (*i.e.*, improving the current solution) also decreases. Consequently, this reduces the required number of global broadcasts from excluded subgroups.

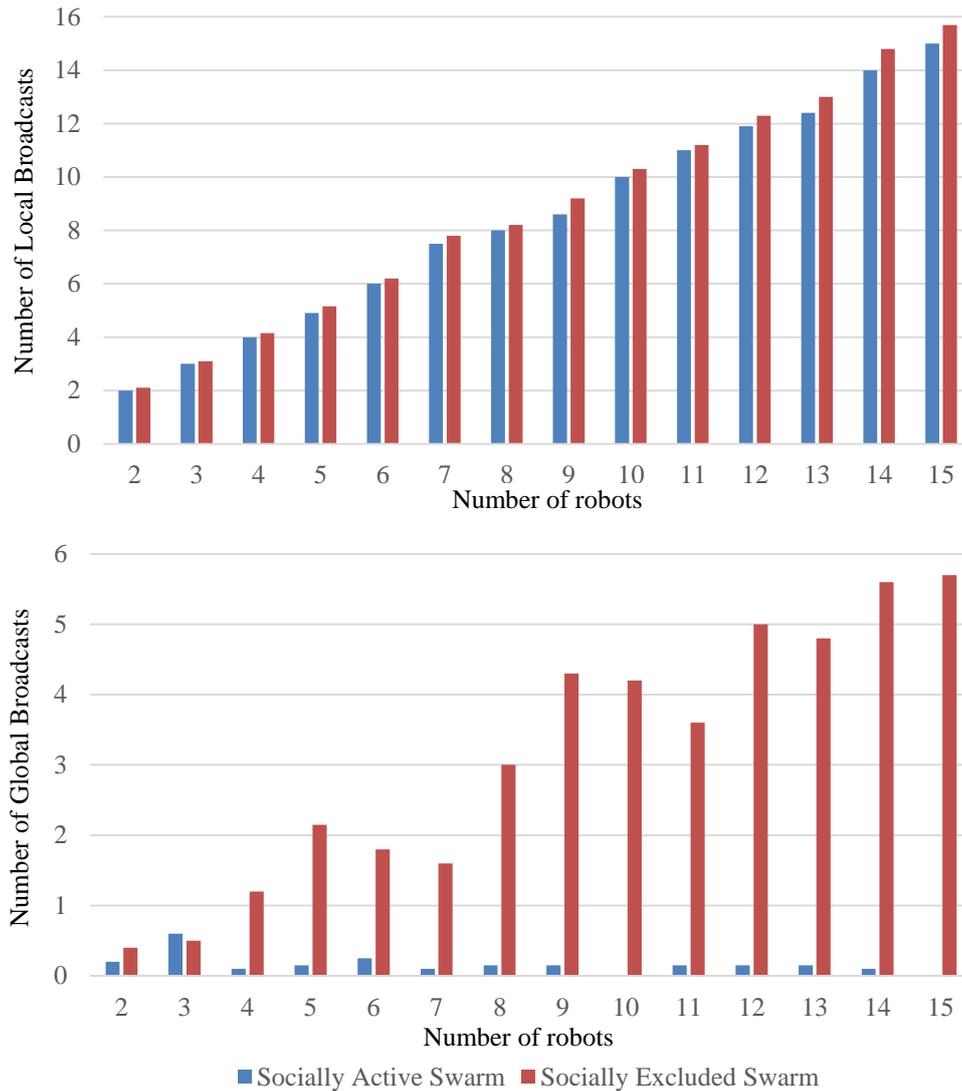


Figure 6.11. Normalized temporal average number of local and global broadcasts.

As the experiments presented so far are limited to a maximum number of 15 physical robots within the same subgroup, it was necessary to perform simulation experiments to evaluate the scalability of the “optimized” *RDPSO*.

6.3.2 Scalability Evaluation through Simulation

The *MRSim* (see section 3.2.4) was used to evaluate the herein proposed “optimized” *RDPSO*. As a means of simplification, and in line with the previous real experiments, the same 20×10 meters indoor scenario (sports pavilion from Figure 5.7) was considered on *MRSim* (*cf.*, Figure 5.13). Due to the computational cost of the simulator, which increases with the number of robots, only experiments of up to 60 robots were possible to carry out. Similarly of what was made in section 5.3.2, *WiFi* communication propagation was also considered (*cf.*, Figure 5.13a).

As *MRSim* is a step-based simulator (without real time iterations), the ratio between the number of packets exchanged within the “optimized” and the “regular” *RDPSO* was analysed. Note that this depends on the type of communication (*i.e.*, local or global broadcast). For instance, in a subgroup of 10 robots a global broadcast from a single robot corresponds to 9 packets exchanged, *i.e.*, one for each teammate. However, if that same robot has only 4 neighbours (one-hop robots) then a local broadcast will correspond to only 4 packets exchanged. Due to the stochastic nature of the *RDPSO*, boxplot charts were used to represent the ratio between the number of packets exchanged within the “optimized” and the “regular” *RDPSO* over the 30 trials with a maximum of 5000 steps each (Figure 6.12). To easily observe the differences, the ratio was averaged at each 100 steps. Once again, note that the number of robots within the same subgroup may vary from 2 robots to the total number of robots within the population (60 robots).

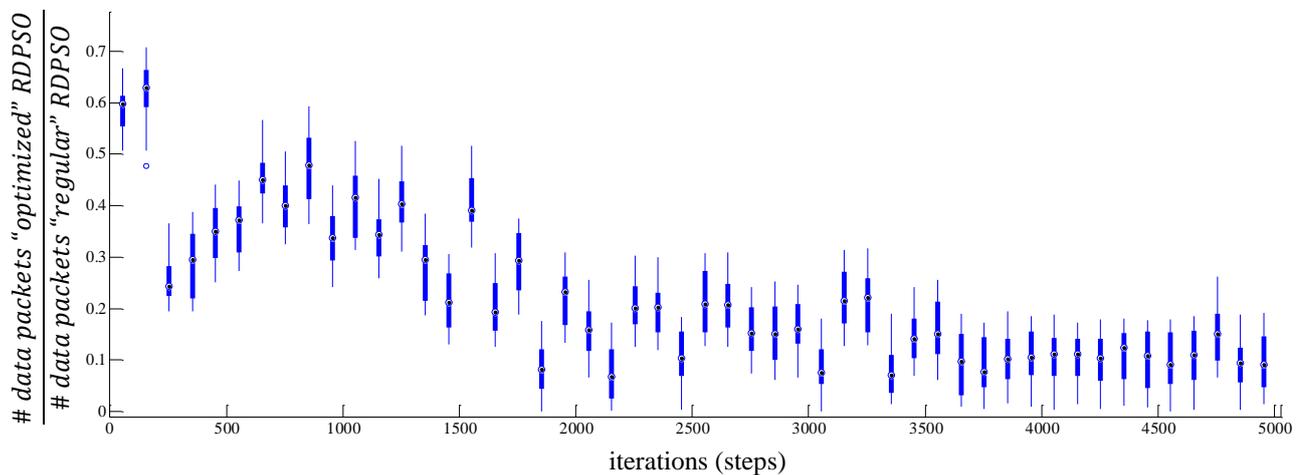


Figure 6.12. Ratio between the number of packets exchanged using the “optimized” *RDPSO* and the “regular” *RDPSO* over the number of iterations in a population of 60 robots.

As one may observe, the difference between the “optimized” and the “regular” *RDPSO* grows with time. The decreasing tendency observed in Figure 6.12 is an expected phenomenon. As a swarm progresses in the exploration using the “optimized” *RDPSO*, the number of global broadcasts necessary to converge to the optimal solution decreases (section 6.1.2). After a certain amount of time (half the mission time), the “optimized” *RDPSO* is able to decrease the number of exchanged data packets to approximately 20% of the number of data packets exchanged under the “regular” *RDPSO*. In terms of communication cost this may be considered as a significant improvement. As an example, the *eSwarBots* platforms usually present a battery autonomy of up to 4 hours without using the *XBee Series 2* modules. However, such autonomy drops to approximately 2 hours with constant data transmission (Couceiro M. S., Figueiredo, Luz, Ferreira, & Rocha, 2011). Another example such as the

well-known *e-puck* robot is even more significant (Mondada, et al., 2009). The *e-puck*'s battery autonomy can drop from 3 hours to approximately 1 hour using the *WiFi* communication from the *Gumstix Overo COM* (section 3.2.3).

6.4 Discussion

The motivation behind this chapter was to explore a strategy for improving the scalability of the *RDPSO* by optimizing its communication complexity. This was achieved by analysing judiciously the information to be explicitly exchanged between robots and proposing a way to efficiently share it without decreasing the collective performance of the algorithm. Afterwards, the well-known Ad hoc On-demand Distance Vector (*AODV*) was adapted based on *RDPSO* dynamics.

Real and simulation experiments were conducted to observe the effect of the proposed optimized strategy. The mission consisted of collectively exploring a 20×10 meters scenario in which robots' cognitive solution was affected by the light sensed at their current position. The superiority of the "optimized" *RDPSO* over the "regular" one was especially visible in the number of packets exchanged between robots and the packet delivery ratio. Although the differences between the routing overhead and the required number of hops to deliver a packet were not significant for small groups of robots, the "optimized" *RDPSO* was still able to reduce both to approximately 20% less for swarms of 15 robots. Those differences were even more visible in the simulations with a swarm of 60 robots examining the ratio between the total number of packets exchanged within the "optimized" and the "regular" *RDPSO*. Although in the beginning of the mission the "optimized" *RDPSO* presented a rather modest reduction of approximately 50% of the number of packets exchanged, as robots continuously explored the scenario such differences increased to approximately 85%. To improve the analysis of the communication architecture within the *RDPSO*, the differences between the two social statuses were also represented, thus revealing that the principle of cooperation undergoes several phases that depend on more than just mission-related contextual information (*e.g.*, sensed solution).

Those results pave the way towards an insightful reassessment and revolution of the *RDPSO* algorithm. Considering the recent advances in the control of aggregation behaviours without communication (Kernbach, et al., 2013), the most expected improvement would be the development of a *stigmergetic RDPSO* without significantly reducing the collective performance of the subgroups. In this case, the macroscopic capabilities of the *RDPSO* should be defined by spatial or dynamical conditions in the environment. In other words, the system and environment itself build a closed macroscopic feedback loop, which works in a collective way as a distributed control mechanism (Kernbach, et al., 2013). In this case, robots interact kinetically or through *stigmergy* effects (Bonabeau, Dorigo,

& Theraulaz, 1999). For instance, emulating Darwin's survival-of-the-fittest without explicit communication would not only require robots to possess the capability of discerning collisions with obstacles and with other robots, but also between robots from different subgroups. Such could be attained by endowing robots with simple low-cost vision capabilities such as the *ArduEye* vision sensor³⁰.

All that being said, one may state that it is still difficult at this point to go from an algorithm sustained by explicit communication to a *stigmergetic* one. However, the authors argue that this chapter provides an exhaustive rationale on the necessary explicit communication within the *RDPSO* that gives the first step in that direction.

6.5 Summary

An optimization of the communication procedure between robots under the *RDPSO* behaviour was presented in this chapter. Moreover, the traditional Ad hoc On-demand Distance Vector (*AODV*) was improved considering robots' motion and behaviours inherent to the *RDPSO*. Such improvements were motivated by the need to use large teams of robots without significantly increasing the communication overhead.

Several experimental results with up to 15 real robots and up to 60 virtual robots in a 20×10 meters scenario clearly show the advantages of such an optimized strategy regarding the scalability of the algorithm, thus suggesting the use of the *RDPSO* on future swarm applications with hundreds or thousands of robots.

During the experiments and, more specifically, the temporal analysis represented in Figure 6.10, it was possible to observe an interesting, and yet so far concealed, phenomenon in which robots get more and more socially excluded as times passes by. This is somehow expected since that, according to the *RDPSO* "punish-reward" mechanism (section 4.5), robots are excluded based on their contribution to their subgroup. As such, if all robots stagnate and are unable to further improve due to the lack of ever-improving solutions (*e.g.*, find a brighter intensity of light), a massive social exclusion may occur. This massive social exclusion phenomena can also be observed in nature, especially in social ostracism depicted by many fish species (*cf.*, section 4.5). As a consequence, and as everything else in nature, those species also present the key to minimize, or at least delay, such massive social exclusion through adaptation mechanisms. Nevertheless, one of the main limitations of the *RDPSO*

³⁰ <http://ardueye.com/>

algorithm resides in having several parameters, such as the fractional order α , influencing the performance of the robotic team. Therefore, next chapter starts by analytically study the *RDPSO* in order to find a relationship between parameters, thus optimizing the algorithm with regard to the main objective, robot dynamics, obstacles susceptibility and *MANET* connectivity. Afterwards, an adaptive mechanisms applied to such parameters is introduced into the *RDPSO* algorithm.

Parameterization and Adaptability

THE literature states that all parameterized algorithms present some drawbacks when facing dynamic and complex problems, *i.e.*, problems with many sub-optimal solutions changing over time. Regardless on *PSO* main variants (Abd-El-Wahed, Mousa, & El-Shorbagy, 2011; Rapaic, Kanovic, & Jelcic, 2009), including the *RDPSO* herein proposed, the difficulties in setting and adjusting the parameters, as well as in maintaining and improving the search capabilities for higher dimensional or constrained problems, are still recent ongoing works, *e.g.*, (Clerc & Kennedy, 2002) (Kadiramanathan, Selvarajah, & Fleming, 2006) The lack of a set of rules to define the most fitted parameters and the adaptability to contextual information usually observed in nature, turns out to result in sub-optimal solutions that are usually overwhelmed by using exhaustive methods (*e.g.*, sweeping the whole scenario with robots) (Suarez & Murphy, 2011). For instance, robots in *SaR* applications must be efficient in persistently searching for victims while there remains a chance of rescuing them. Although the *RDPSO* previously presented is endowed with “punish-reward” rules inspired on natural selection to avoid stagnation (see section 4.5), robots may still take too long to realize that they are stuck in a sub-optimal solution or that the solution is changing over time. A good example of that may be found on olfactory-based swarming, wherein a plume is subject to diffusion and airflow, thus being hard to find its source (Marques, Nunes, & Almeida, 2006; Jatmiko, Sekiyama, & Fukuda, 2006). Although this is a well-known problem, as its complexity significantly increases as a function of the number of parameters, solving it still remains a challenge. Moreover, if the input parameters are not fixed (*i.e.*, they adapt to the contextual information), then all known solving algorithms for *NP*-hard problems require time that is exponential in the total size of the number of parameters (Downey & Fellows, 1999).

One of the most common strategies presented in the literature to solve issues in setting and adjusting *PSO* parameters is based on the stability analysis of the algorithm. Clerc and Kennedy analysed the individual particle’s trajectory leading to a generalized model, which contains a set of coefficients to control system’s convergence (Clerc & Kennedy, 2002). The resulting system is linear of second-order with stability and parameters depending on the poles, or on the eigenvalues of the state matrix. Kadiramanathan *et al.* proposed a stability analysis of a stochastic particle dynamics by representing it as a nonlinear feedback controlled system (Kadiramanathan, Selvarajah, & Fleming, 2006). The Lyapunov stability method was applied to the particle dynamics in determining

sufficient and conservative conditions for asymptotic stability. However, the analysis provided by the authors addressed only the issue of absolute stability, thus ignoring the optimization toward the optimal solution (Kadiramanathan, Selvarajah, & Fleming, 2006). More recently, Yasuda *et al.* presented an activity-based numerical stability analysis method, involved the feedback of swarm activity to control diversification and intensification during the search (Yasuda, Iwasaki, Ueno, & Aiyoshi, 2008). The authors showed that the swarm activity can be controlled by employing the stable and unstable regions of *PSO*.

Contrarily to the herein proposed multi-robot foraging approach, all the above works only consider the *PSO* and its main variants applied to optimization problems. As previously stated, robots are designed to act in the real-world in which both their dynamics and obstacles need to be taken into account. Furthermore, since in certain environments the communication infrastructure may be damaged or missing (*e.g.*, *SaR*), the self-spreading of autonomous mobile nodes of a *MANET* over a geographical area needs to be considered, thus significantly increasing the complexity of the problem. As such, and despite the number of works around swarm robotics for search applications, still none of them presented a formal convergence analysis to find the best set of parameters, nor a strategy to systematically adapt such parameters while considering *MRS* characteristics. This last point supports the idea that the behaviour of robots needs to change accordingly to contextual information about the surroundings. This concept of *contextual knowledge* needs to be taken into account to adapt swarms and robots' behaviour while considering agent-based, mission-related and environmental context (Turner, 1998). For example, Calisi's *et al.* work presented a context-based architecture to enhance the performance of a robotic system in search and rescue missions using a rule system based on first-order Horn clauses (Calisi, Iocchi, Nardi, Randelli, & Ziparo, 2009). The set of metrics used as inputs was obtained considering an *a priori* map about the difficulty levels concerning mobility and victim detection. Nevertheless, in real applications this would mean a previous knowledge about the scenario which is not always possible and can be difficult to predict.

This chapter tries to go a step forward by providing a methodology to analyse the stability of the *RDPSO* algorithm and, as a result, retrieve the ideal range of values wherein its parameters should be defined so as to ensure the convergence of robots toward the solution. Furthermore, it introduces an adaptive architecture inspired on the concepts of fuzzy logic (section 3.1.3) to vary the *RDPSO* parameters within the previously defined range. More unambiguously, this chapter presents four key contributions (Couceiro, Machado, Rocha, & Ferreira, 2012; Couceiro M. S., Martins, Rocha, & Ferreira, 2013a (Under Review)):

- i)* First and foremost, a formal analysis of *RDPSO* in order to better understand the relationship between the algorithm's parameters and its convergence is presented (section 7.1);

- ii) A set of context-based evaluation metrics, at both microscopic and macroscopic levels, are proposed to assess the *RDPSO* behaviour, by studying the several concepts inherent to swarm techniques (*e.g.*, exploration vs exploitation) with a phase space analysis of robots' motion (*e.g.*, chaoticity) (section 7.2);
- iii) Afterwards, this knowledge is used as input of a fuzzy system so as to systematically adapt the *RDPSO* parameters (*i.e.*, outputs of the fuzzy system), thus improving its convergence rate, susceptibility to obstacles and communication constraints (section 7.3);
- iv) Experimental results obtained using physical and simulated robots, wherein the adaptive version of the *RDPSO* was evaluated and compared to the non-adaptive one, are also presented (section 7.4).

Sections 7.5 and 7.6 outline the discussion and main conclusions of this chapter.

7.1 Convergence Analysis

Due to the limitations in the state-of-the-art, let us first formalize the problem we will be working on in the first part of this chapter.

7.1.1 Problem Statement

The above presented *RDPSO* is a stochastic procedure in which the *DE* system comprising on equations (4.5), (4.6) and (4.8) describe the discrete-time motion of a robot with four external inputs $\chi_i[t]$, $i = 1,2,3,4$. The main problem when analyzing this kind of algorithms lies in the fact that external inputs vary in time. However, one can consider that each robot converges to an *equilibrium point* defined by the limit values of the attractor points χ_i . Therefore, assuming that the algorithm converges, this section presents the stability analysis of the *RDPSO*.

Consider a swarm of N_T robots wherein each robot needs to cooperatively find the optimal solution of a given mission within its subgroup. The goal is to find the attraction domain \mathcal{A} such that, if coefficients $\alpha, \rho_i \in \mathcal{A}$, $i = 1,2,3,4$, the global asymptotic stability of the *DE* system (4.5)-(4.8) is guaranteed. In other words, the attraction domain \mathcal{A} represents the region wherein *RDPSO* parameters may be defined in such a way that robots can find the optimal solution while avoiding obstacles and ensuring *MANET* connectivity.

7.1.2 General Approach

Knowing that $v_n[t - k] = x_n[t - k] - x_n[t - (k + 1)]$ with $k \in \mathbb{N}_0$, and considering equation (4.6) and the particular finite case in (4.11), one can rewrite the *DE* system as a nonhomogeneous five-order difference equation:

$$\begin{aligned} x_n[t + 1] + (-1 - \alpha + \sum_{i=1}^4 \rho_i r_i) x_n[t] + \left(\frac{1}{2} \alpha\right) x_n[t - 1] + \left(\frac{1}{3} \alpha + \frac{1}{6} \alpha^2\right) x_n[t - 2] \\ + \left(-\frac{1}{24} \alpha^3 - \frac{1}{24} \alpha^2 + \frac{1}{12} \alpha\right) x_n[t - 3] + \left(\frac{1}{24} \alpha^3 - \frac{1}{8} \alpha^2 + \frac{1}{12} \alpha\right) x_n[t - 4] = \\ \sum_{i=1}^4 \rho_i r_i \chi_i[t]. \end{aligned} \quad (7.1)$$

The *equilibrium point* x_n^* can be defined as a *constant position solution* of (7.1) such that, when each robot reaches x_n^* , the velocity $v_n[t + k]$ is zero, *i.e.*, robots will stop at the equilibrium point x_n^* . Supposing that χ_i are constants, *i.e.*, the algorithm does converge, the particular solution x_n^* of each robot can be obtained replacing $x_n[t + 1 - k]$ in equation (7.1) by x_n^* :

$$\begin{aligned} x_n^* + (-1 - \alpha + \sum_{i=1}^4 \rho_i r_i) x_n^* + \left(\frac{1}{2} \alpha\right) x_n^* + \left(\frac{1}{3} \alpha + \frac{1}{6} \alpha^2\right) x_n^* + \left(-\frac{1}{24} \alpha^3 - \frac{1}{24} \alpha^2 + \frac{1}{12} \alpha\right) x_n^* \\ + \left(\frac{1}{24} \alpha^3 - \frac{1}{8} \alpha^2 + \frac{1}{12} \alpha\right) x_n^* = \sum_{i=1}^4 \rho_i r_i \chi_i \Leftrightarrow x_n^* = \frac{\sum_{i=1}^4 \rho_i r_i \chi_i}{\sum_{i=1}^4 \rho_i r_i}. \end{aligned} \quad (7.2)$$

In other words, each robot will converge to the particular solution x_n^* , based on the following theorems (Elaydi, 2005):

Theorem 7.1. (Elaydi, 2005) *All solutions of (7.1) converge to x_n^* as $t \rightarrow \infty$, if and only if the homogeneous difference equation of (7.2) is asymptotically stable.*

Theorem 7.2. (Elaydi, 2005) *The homogeneous difference equation of (7.1) is asymptotically stable if and only if all roots of the corresponding characteristics equation of (7.3) have modulus less than one.*

Therefore, in order to study the homogeneous *DE* (7.1) stability, let us consider the following characteristic equation:

$$p(\lambda) \equiv \lambda^5 + (-1 - \alpha + \sum_{i=1}^4 \rho_i r_i) \lambda^4 + \left(\frac{1}{2} \alpha\right) \lambda^3 + \left(\frac{1}{3} \alpha + \frac{1}{6} \alpha^2\right) \lambda^2 + \left(-\frac{1}{24} \alpha^3 - \frac{1}{24} \alpha^2 + \frac{1}{12} \alpha\right) \lambda + \left(\frac{1}{24} \alpha^3 - \frac{1}{8} \alpha^2 + \frac{1}{12} \alpha\right) = 0. \quad (7.3)$$

Due to the complexity in obtaining the roots of the characteristics equation of homogeneous difference equation (7.3), it is established a result based on Jury-Marden's Theorem (Barnett, 1983) that ensures that all roots of the real polynomial $p(\lambda)$ have modulus less than one (*cf.*, Theorem 3.1 from section 3.1.2). For that reason, let us present the following result.

Proposition 7.1. *All roots of $p(\lambda)$ have modulus less than one if and only if the following conditions are met.*

$$\begin{cases} 0 < \sum_{i=1}^4 \rho_i r_i \leq \alpha + 2 & , 0 < \alpha \leq 0.6 \\ \frac{15}{4} \alpha - \frac{9}{4} \leq \sum_{i=1}^4 \rho_i r_i \leq \alpha + 2 & , 0.6 < \alpha \leq 1 \end{cases} \quad (7.4)$$

Proof: The real polynomial $p(\lambda)$ described in equation (7.3) can be rewritten as:

$$a_0 \lambda^5 + a_1 \lambda^4 + a_2 \lambda^3 + a_3 \lambda^2 + a_4 \lambda + a_5 = 0, \quad (7.5)$$

Furthermore, one can construct an array having initial rows defined as:

$$\begin{aligned} \{c_{11}, c_{12}, \dots, c_{1,6}\} &= \{a_0, a_1, \dots, a_5\}, \\ \{d_{11}, d_{12}, \dots, d_{1,6}\} &= \{a_5, a_4, \dots, a_0\}, \end{aligned} \quad (7.6)$$

and subsequent rows defined by:

$$c_{\beta\gamma} = \begin{vmatrix} c_{\beta-1,1} & c_{\beta-1,\gamma+1} \\ d_{\beta-1,1} & d_{\beta-1,\gamma+1} \end{vmatrix}, \quad (7.7)$$

$$d_{\beta\gamma} = c_{\beta,8-\gamma-\beta}, \quad (7.8)$$

where $\beta = 2,3,4,5,6$ and $\gamma = 0,1,2,3$.

By Theorem 3.1, we consider that all roots of polynomial $p(\lambda)$ have modulus less than one if and only if $d_{21} > 0, d_{\tau 1} < 0$, for $\tau = 3,4,5,6$.

Hence,

$$\begin{cases} d_{21} > 0 \\ d_{31} < 0 \\ d_{41} < 0 \\ d_{51} < 0 \\ d_{61} < 0 \end{cases} \Leftrightarrow \begin{cases} 1 - a_5^2 > 0 \\ (1 - a_5 a_1)^2 - (d_{21})^2 < 0 \\ ((1 - a_5 a_1)(a_1 - a_5 a_4) - d_{21}(a_3 - a_5 a_2))^2 - (d_{31})^2 < 0 \\ (c_{41})^2 - (d_{41})^2 < 0 \\ (c_{51})^2 - (d_{51})^2 < 0 \end{cases}. \quad (7.9)$$

By solving (7.9), we obtain the conditions on (7.4). ■

Consequently, by Proposition 7.1, Theorem 7.1 and Theorem 7.2, the conditions in (7.4) are obtained so that all solutions of the *DE* system (4.5)-(4.8) converge to x_n^* resulting in a *global attraction domain* \mathcal{A} (please see Figure 7.1). Although it was possible to define a relatively small attraction domain, next section further explores particular conditions of the algorithm, by redefining and adjusting parameters values.

7.1.3 Robot Constraints

One way to improve the convergence analysis of the algorithm consists on adjusting the parameters based on physical mobile robots constraints, such as acceleration and deceleration states inherent to their dynamical characteristics. These states are usually unaddressed in the literature while analysing the traditional *PSO* and its main variants, since virtual agents (*i.e.*, particles) are not constrained by such behaviours. Let us then suppose that a robot is traveling at a constant velocity such that $v_n[t - k] = v$ with $k \in \mathbb{N}_0$ and it is able to find its *equilibrium point* in such a way that $x_n[t] = \chi_i$, $i = 1, 2, 3, 4$. In other words, the best position of the cognitive, social, obstacle and *MANET* components are the same. As a result, the robot needs to decelerate until it stops, *i.e.*, $v > v_n[t + 1] \geq \dots \geq v_n[t + k] \geq \dots \geq 0$.

Consequently, considering once again the particular finite case from equation (4.11), one can write the following condition:

$$0 \leq v \left(\alpha + \frac{1}{2} \alpha + \frac{1}{6} \alpha (1 - \alpha) + \frac{1}{24} \alpha (1 - \alpha) (2 - \alpha) \right) < v, \quad (7.10)$$

thus resulting in

$$0 < \alpha \leq 0.632. \quad (7.11)$$

Let us now consider the opposite scenario, *i.e.*, a robot that was stopped $v_n[t - k] = 0$ with $k \in \mathbb{N}_0$ starts to move since $x_n[t] \neq \chi_i$, $i = 1, 2, 3, 4$. The robot needs to accelerate until it reaches the maximum velocity defined by equation (4.11).

Similarly to the procedure presented in the previous section, but considering the described conditions from above, the following nonhomogeneous first-order difference equation results:

$$x_n[t + 1] + (\sum_{i=1}^4 \rho_i r_i - 1)x_n[t] = \sum_{i=1}^4 \rho_i r_i \chi_i[t]. \quad (7.12)$$

Hence, the characteristic equation associated to (7.12) is:

$$p_1(\lambda) \equiv \lambda + (\sum_{i=1}^4 \rho_i r_i - 1) = 0. \quad (7.13)$$

Proposition 7.2. *The homogeneous difference equation of (7.13) is asymptotically stable iff:*

$$0 < \sum_{i=1}^4 \rho_i r_i < 2. \quad (7.14)$$

Proof: Based on Theorem 7.2 one can consider that the homogeneous difference equation (7.13) is asymptotically stable if and only if the root of $p_1(\lambda)$ have modulus less than one. Therefore,

$$p_1(\lambda) = 0 \Leftrightarrow \lambda + (\sum_{i=1}^4 \rho_i r_i - 1) = 0 \Leftrightarrow \lambda = -(\sum_{i=1}^4 \rho_i r_i - 1). \quad (7.15)$$

Then,

$$|\lambda| < 1 \Leftrightarrow |-(\sum_{i=1}^4 \rho_i r_i - 1)| < 1 \Leftrightarrow 0 < \sum_{i=1}^4 \rho_i r_i < 2. \quad (7.16)$$

■

Consequently, by Proposition 7.2, Theorem 7.1 and Theorem 7.2, the conditions in (7.14) are obtained so that all solutions of (7.12) converge to x_n^* resulting in a particular attraction domain \mathcal{A}_p .

However, since r_i randomly varies between 0 and 1, such that $\max r_i = 1$, $i = 1, 2, 3, 4$, condition (7.14) can be rewritten as:

$$0 < \sum_{i=1}^4 \rho_i < 2. \quad (7.17)$$

7.1.4 Outline

Considering the general conditions presented in (7.4) and the particular case in (7.17), Figure 7.1 depicts the global attraction domain \mathcal{A} and particular attraction domain \mathcal{A}_p , respectively. In this work, we adopt the particular attraction domain \mathcal{A}_p as the accurate representation of the parameter region describing the asymptotic stability of the *RDPSO* algorithm.

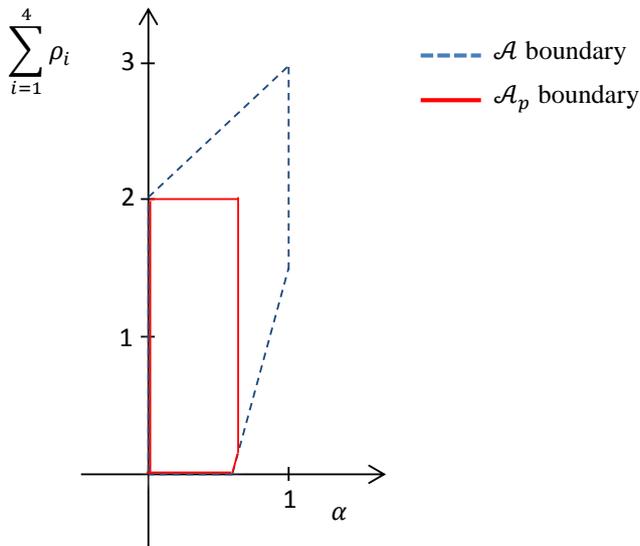


Figure 7.1. Global and particular attraction domain of the asymptotic stability of the *RDPSO*.

As a result of the above analysis, the *RDPSO* can be conceived in such a way that the system's convergence can be controlled by taking into account obstacle avoidance and *MANET* connectivity, without resorting to the definition of any arbitrary or problem-specific parameters. However, the influence of each individual parameter ρ_i and fractional coefficient α in the performance of the algorithm needs to be further explored to understand their on-the-fly influence during the mission.

7.2 Context-Based Evaluation Metrics

To endow the *RDPSO* with adaptive behaviour, a set of evaluation metrics, at both macroscopic (*i.e.*, swarm) and microscopic (*i.e.*, individual robot) levels needs to be defined. These metrics measure the performance of robots' collective movement and will be used to systematically adjust the parameters of the algorithm, thus improving its convergence rate, susceptibility to obstacles and communication

constraints. Hence, the set of evaluation indices herein proposed are computed, at each iteration, considering environmental and behavioural context. Those measures will then be used as inputs of the fuzzy system in order to control the *RDPSO* parameters (*i.e.*, outputs of the fuzzy system).

To evaluate the following proposed metrics within the *RDPSO* algorithm, a subgroup of two physical robots is adopted in the next set of experiments. Robots consisted on differential ground *eSwarBots* (section 3.2.1). Solutions were defined by illuminated spots on a 2.55×2.45 meters scenario sensed using the overhead *LDRs* of *eSwarBots*. Refer to section 4.7.2 for a more detailed description on the experimental setup. Using only two robots allows to easily retrieve the evolution of each evaluation metric when facing specific extreme situations. For instance, the use of a larger subgroup would not drastically affect how one robot behaves when detecting an obstacle within its sensing range. Also, as the scenario has a limited size and number of solutions, a smaller population results in a smaller stochastic effect, thus resulting in negligible differences between different trials, despite the existence of the random coefficients r_i , $i = 1,2,3,4$.

7.2.1 Exploitation vs Exploration

As described in the works of Yasuda *et al.* (Yasuda, Iwasaki, Ueno, & Aiyoshi, 2008) and Wakasa *et al.* (Wakasa, Tanaka, & Nishimura, 2010), a swarm behaviour can be divided into two activities: *i*) exploitation; and *ii*) exploration. The first one is related with the convergence of the algorithm, thus allowing a good short-term performance. However, if the exploitation level is too high, then the algorithm may be stuck on sub-optimal solutions. The second one is related with the diversification of the solution provided by the algorithm, which allows exploring new solutions, thus improving the long-term performance. However, if the exploration level is too high, then the algorithm may take a long time to find the optimal solution (long time to converge). As first presented by Shi and Eberhart, the trade-off between exploitation and exploration in the classical *PSO* has been commonly handled by systematically adjusting the inertia weight (Shi & Eberhart, 2001). A large inertia weight improves exploration activity while exploitation is improved using a small inertia weight.

Since the *RDPSO* presents a fractional calculus strategy to control the convergence of the robotic team, it is only natural that the coefficient α is the one that needs to be systematically adjusted in order to provide a high level of exploration while ensuring the optimal solution of the mission. In order to understand the relation between the fractional coefficient α and the *RDPSO* exploitation/exploration capabilities, the centre-of-mass trajectory in phase space of a subgroup of two physical robots, for various values of α , while fixing $\rho_i = 0.5$, will be analyzed. Both robots were randomly placed in the vicinity of the solution in (0,0) with a fixed distance of 0.5 meters between them (Figure 7.2).

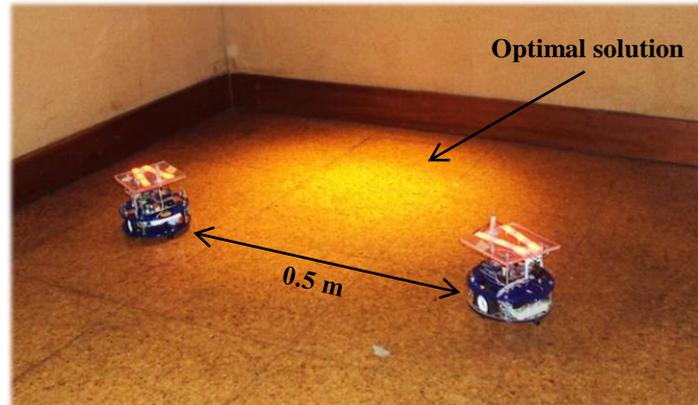


Figure 7.2. Experimental setup to evaluate the exploration/exploitation capabilities of a subgroup of two robots.

As it may be perceived on Figure 7.3, the swarm behaviour is susceptible to variations in the value of α . Figure 7.3 shows that when α is too small, *i.e.*, $\alpha = 0.01$, the exploitation level is too high, being very likely to get stuck in a sub-optimal solution. However, the intensification of the algorithm convergence is improved – it presents a quick, almost linear, convergence. When α is at the boundary of the attraction domain, *i.e.*, $\alpha = 0.632$, the trajectory of the subgroup is cyclical and presents a good balance between exploitation and exploration. In this case, robots exhibit a level of diversification adequate to avoid sub-optimal solutions and a considerable level of intensification to converge to the optimal solution, *i.e.*, it presents a spiral convergence toward a nontrivial attractor. When α is too high and outside the attraction domain, *i.e.*, $\alpha = 0.99$, despite the cyclical trajectory of the subgroup toward the optimal solution, the subgroup presents an oscillatory behaviour. This results in a high exploration level being more unstable and sometimes unable to converge, *i.e.*, it presents a difficult convergence.

We observe that α needs to be adjusted depending on the contextual knowledge for behavior specialization. Hence, the introspective knowledge about the swarm activity is used to obtain smooth transitions between behaviours. However, a method to evaluate the current swarm activity needs to be considered.

In the work of Yasuda *et al.* the swarm activity is controlled by switching between the stable and unstable regions of the *PSO* (Yasuda, Iwasaki, Ueno, & Aiyoshi, 2008). In our situation, the stable region is defined by the attraction domain presented in the previous section (Figure 7.1), wherein the swarm activity is predominantly of exploitation type. Since the equilibrium between exploitation and exploration is at the boundary of the attraction domain ($\alpha = 0.632$), α should always be set to a value in its neighbourhood.

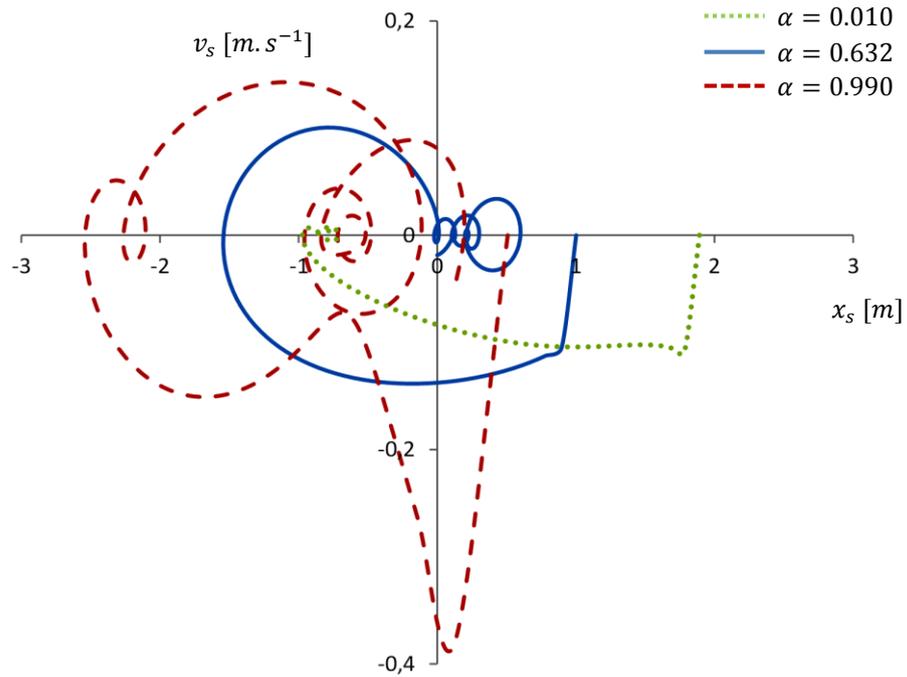


Figure 7.3. Centre-of-Mass trajectories in phase space of a subgroup of 2 robots.

Contrarily to the work of Yasuda *et al.* in which the activity is defined as the root mean square velocity of particles (Yasuda, Iwasaki, Ueno, & Aiyoshi, 2008), let us define the *swarm activity* of subgroup s as the norm of the velocity of its centre-of-mass $v_s[t]$ at each iteration t , *i.e.*, group velocity:

$$A_s[t] = \frac{\|v_s[t]\|}{\Delta x_{max}}, \quad (7.18)$$

wherein threshold Δx_{max} corresponds to the maximum step between iterations. The redefinition of swarm activity was considered in order to underline the collective activity (at the macro level) instead of the sum of activities performed by each robot. Considering the definition presented in Yasuda *et al.* (Yasuda, Iwasaki, Ueno, & Aiyoshi, 2008), robots may present a high activity but the swarm as a whole may present a small activity, *i.e.*, $v_s[t] \approx \vec{0}$. Therefore, a swarm activity of $A_s[t] = 0$ means no swarm activity at all and α should increase, while $A_s[t] = 1$ corresponds to a highly chaotic behaviour and α should decrease.

It should be noted that this adapted behaviour occurs at the collective level. However, the individual behaviour of each robot also needs to be considered. In other words, the same swarm may have both exploring and exploiting robots and that state will depend on their cognition and socialization level.

7.2.2 Cognition vs Socialization

Despite the relation between the fractional coefficient α and the swarm collective behaviour, it is the combination of all *RDPSO* parameters that determines its convergence properties. The values of both cognitive and social factors ρ_1 and ρ_2 are not critical for the algorithm, but selection of proper values may result in better performance, both in terms of speed of convergence and alleviation of sub-optimal solutions. Furthermore, their values have to be taken into account when choosing the fractional coefficient α .

As previously described by other words, the cognitive component ρ_1 represents the personal “*thinking*” of each robot, thus encouraging robots to move toward their own best positions found so far. The social component ρ_2 represents the collaborative effect of the subgroup in finding the optimal solution, thus summoning robots toward the global best position found so far. Venter’s work presented experimental results in which a small cognitive coefficient ρ_1 and large social coefficient ρ_2 could significantly improve the performance of the algorithm (Venter, 2002). However, it should be highlighted that, for problems with multiple sub-optimal solutions, a larger social coefficient ρ_2 may prematurely mislead all robots toward a sub-optimal solution in which they will be unable to avoid since they are “*blind*” followers. On the other hand, a larger cognitive coefficient ρ_1 may cause each robot to be attracted to its own personal best position to a very high extent, resulting in excessive wandering.

To further understand the cognitive and social components of the *RDPSO*, let us once again consider an experimental setup of a subgroup of two robots. Each robot is initially placed near the sub-optimal and optimal solutions uniquely identifiable by controlling the brightness of the light. The brighter site (optimal solution) is considered better than the dimmer one (sub-optimal solution), and so the goal of the subgroup is to collectively choose the brighter site (Figure 7.4). It is noteworthy that using a large swarm within such scenario would not yield much different results as each subgroup global best would be collectively chosen as the same than using only two robots. In other words, increasing the number of robots would not only increase the variability of the behaviour before the collective consensus on the global best solution, as it would significantly increase the complexity on analysing the evolution of the group.

At the beginning, robots are at a distance of 1.6 meters from each other. Also, the fractional coefficient α is now fixed at 0.632 (threshold stability) and $\rho_3 = \rho_4 = 0.1$ for multiple (ρ_1, ρ_2) combinations while keeping the same absolute value $\rho_T = 1$ with $\rho_T = \rho_1 + \rho_2$.

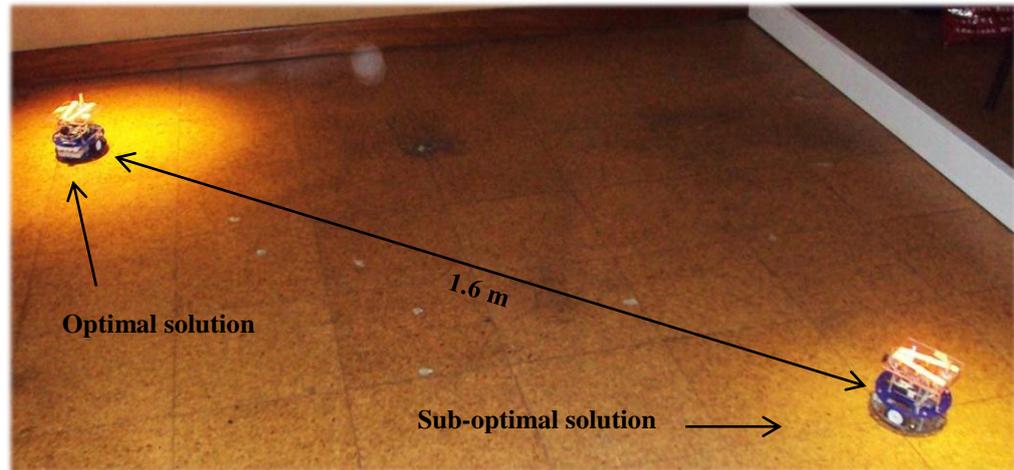


Figure 7.4. Experimental setup to evaluate the cognition/socialization between two robots of the same subgroup.

Figure 7.6 presents the Euclidean distance in phase space between the two robots, thus depicting the evolution and convergence of the distance between them. Note that the inter-robot distance turns out to represent the distance between the robot located in the sub-optimal solution and the location of the optimal solution itself. This phenomenon can be explained by how the other parameters are defined (more specifically the smaller values of ρ_3 and ρ_4) and the inexistence of any other better solution within the subgroup. Hence, the decision of the robot located in the optimal solution is not disturbed, thus staying still until a better solution is found (which never happens in this example).

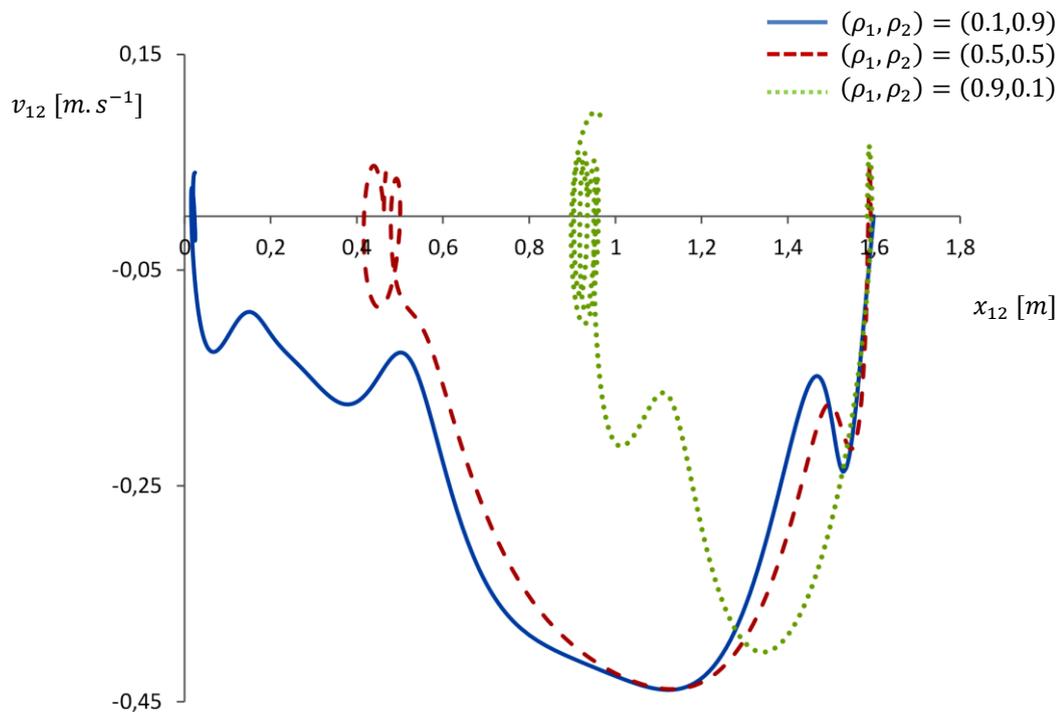


Figure 7.5. Distance between robots in phase space to evaluate the relation between ρ_1 and ρ_2 .

As expected, increasing the social weight ρ_2 decreases the Euclidean distance between robots, *i.e.*, the distance between robots tended to only a few centimetres when using $(\rho_1, \rho_2) = (0.1, 0.9)$

and near 1 meter using $(\rho_1, \rho_2) = (0.9, 0.1)$. However, the relation between the final inter-robot distance and (ρ_1, ρ_2) weights is not linear. It can also be observed that, increasing the social weight ρ_2 , the robot initially located at the sub-optimal solution converges in a more intensive way, that is, the radius of the spiral at robot's convergence position is smaller for higher ρ_2 values. Hence, the exploitation behavior increases as the distance between robots decreases, thus compromising the performance of the swarm. Moreover, robots' velocity does not directly depend on the relation between ρ_1 and ρ_2 , since the relative velocity between robots reaches a maximum velocity of approximately 0.45 m.s^{-1} in the three (ρ_1, ρ_2) combinations.

A balance between cognitive and social weights needs to be established and adapted throughout the mission depending on contextual mission-related knowledge of the cognitive or social levels of robots, thus resulting in a different social weight ρ_2 (and, therefore, cognitive weight ρ_1), for each robot.

Suresh *et al.* work presented an inertia adaptive *PSO* in which the modification involved the modulation of the inertia factor according to distance of particles of a particular generation from the global best (Suresh, et al., 2008). Similarly, a microscopic level metric, defined as *robot socialization*, can then be defined as the current Euclidean distance of robot n from its swarm global best:

$$S_n[t] = 1 - \frac{\|x_2[t] - x_n[t]\|}{\left\| x_2[t] - x_{\text{argmax}(\sum_i^{N_S} \|x_2[t] - x_i[t]\|)} \right\|}. \quad (7.19)$$

The social level of a given robot will then be the relation between its distance to the global best and the distance of the farthest robot of the swarm to the global best. Therefore, a robot with a social level of $S_n[t] = 0$ means that the best robot is the farthest robot of the swarm and ρ_2 should increase, thus decreasing ρ_1 . On the other hand, as robot social level increases, *i.e.*, the distance of the robot to the optimal solution decreases, ρ_2 should decrease (increasing ρ_1). This modulation ensures that in case of robots that have moved away from the global best, the effect of attraction towards the global best will predominate.

Depending on the social level, the fractional coefficient α should vary. As $S_n[t]$ decreases, α should also decrease so that whenever a robot moves far away from the globally best position found so far by the swarm, the effect of its inertial velocity will be minimal. The opposite situation can also be considered. As $S_n[t]$ increases, *i.e.*, the robot gets closer to the global best position, α should increase to present a higher diversification level, thus increasing the possibility to find an improved or alternative solution. Consequently, there may also be a different α for each robot depending on its social level.

7.2.3 Obstacles Susceptibility

Using multiple mobile robots for *SaR* applications requires an efficient way of avoiding obstacles while completing their main mission. The presence or absence of obstacles can affect the efficiency of the *RDPSO* since one set of parameters may result in fast convergence but may still fail in the presence of obstacles or it may increase obstacles susceptibility but swarms may be more resilient.

As previously explained, a robot is able to avoid obstacles due to a repulsive force based on a monotonic and positive *sensing function* $g(x_n[t])$ that depends on the distance between the robot and the obstacle (*cf.*, section 4.3). Its susceptibility is defined through the obstacle susceptibility weight ρ_3 . Since the characteristics of the environment are generally not known in advance, the robot itself should be able to intelligently change its own obstacle susceptibility ρ_3 based on the contextual information about the environment.

By means of equation (4.5) one can perceive that, when a robot does not sense any obstacle within its sensing radius R_w , the position that optimizes the monotonically decreasing or increasing sensing function $g(x_n[t])$ is the same as robot's current position, *i.e.*, $x_n[t] = \chi_3[t]$. This yields the following expression:

$$\rho_3 r_3 (\chi_3[t] - x_n[t]) = 0, \quad (7.20)$$

One may consider that, when a robot does not sense any obstacle within R_w , then the obstacle coefficient should be ignored, *i.e.*, $\rho_3 = 0$. Also, it is easy to remark that its obstacle susceptibility weight should increase as the distance to the obstacle decreases. However, as previously highlighted, it is not the absolute value of a coefficient that matters but the relation between all coefficients. Therefore, for better understanding the relation between ρ_3 and the rest of the *RDPSO* parameters, let us consider a new experimental setup of a subgroup of two robots. One of the robots is placed in the optimal solution (*i.e.*, the brighter site), and will summon the other robot towards it. The other robot is placed 1 meter away from the best performing robot and an obstacle is placed halfway the path between both robots, *i.e.*, 0.5 meters in front of the robot that is being summoned (Figure 7.6). Also, robots are programmed to detect obstacles at 0.5 meters from them, *i.e.*, $R_w = 0.5$.

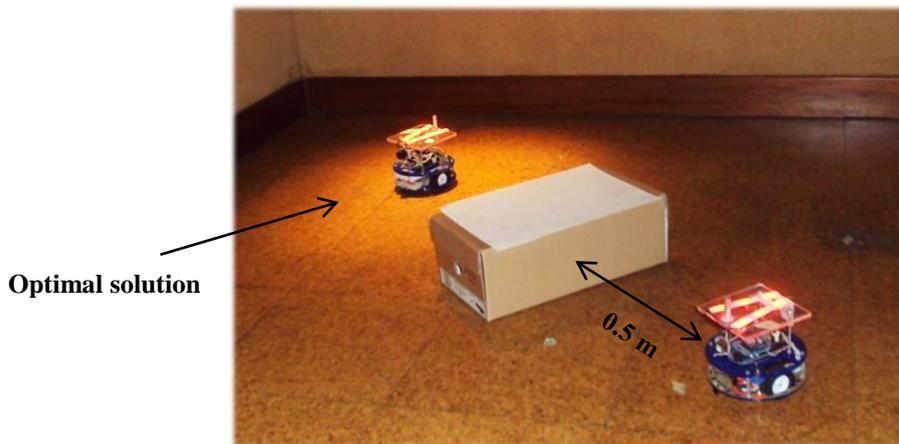


Figure 7.6. Experimental setup to evaluate the obstacle susceptibility of a robot.

To allow the manipulation of ρ_3 within a considerable range, while respecting the attraction domain represented by Figure 7.1, let us suppose the following set of parameters: $\rho_4 = 0.1$ and $\rho_T = 0.7$, with $(\rho_1, \rho_2) = (0.2, 0.5)$. In other words, the social component influences more than the cognitive one, thus allowing for the robot placed in the optimal solution to promptly lure the other one. As Figure 7.7 depicts, the obstacle susceptibility of the robot was evaluated using the following parameters $\alpha = 0.632$ and $\rho_2 = \{0.4, 0.8, 1.2\}$.

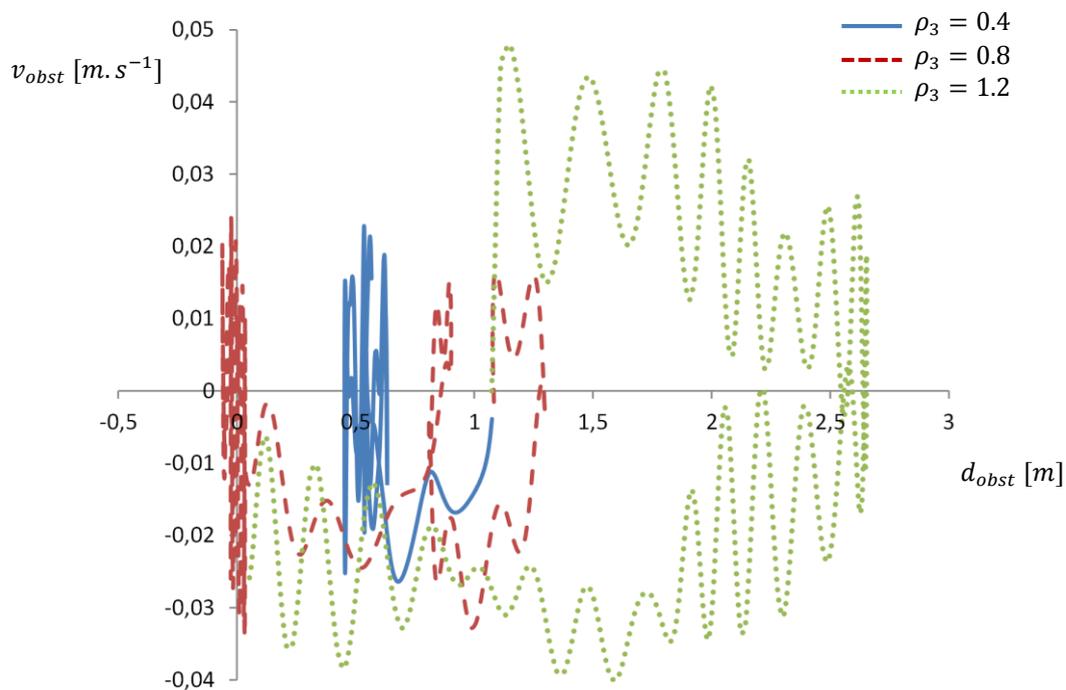


Figure 7.7. Distance from the worst performing robot to the obstacle in phase space.

Observing Figure 7.7, we conclude that the worst performing robot gets stuck in the obstacle vicinities, and sometimes collides with it for an obstacle susceptibility weight of $\rho_3 = 0.4$. For any of the other two situations ($\rho_3 = 0.8$ and $\rho_3 = 1.2$), the robot is able to circumvent the obstacle, thus reaching the optimal solution. However, notwithstanding the same final result for both $\rho_3 = 0.8$ and

$\rho_3 = 1.2$, as ρ_3 increases the robot presents a more chaotic behaviour, *i.e.*, more oscillatory. For $\rho_3 = 1.2$ the robot first moves 1 meter and a half away from its current location avoiding the obstacle in an inadequate way.

In fact, as a robot avoids an obstacle, ρ_3 should decrease allowing a wider range of possibilities for the other coefficients, such as ρ_1 and ρ_2 . For that reason, the following environmental contextual information about *robot avoidance* was defined:

$$O_n[t] = \frac{R_w - g(x_n[t])}{R_w}, \quad (7.21)$$

wherein the monotonic and positive *sensing function* $g(x_n[t])$ returns R_w when the robot does not sense any obstacle within its sensing radius. As equation (7.21) shows, as an obstacle enters a robot's sensing radius, $O_n[t]$ tends to 1, thus presenting the proximity to the obstacle. On the other hand, when $O_n[t] = 0$, *i.e.*, the robot is in an obstacle-free path, then the obstacle susceptibility weight can be neglected, *i.e.*, $\rho_3 = 0$. Naturally, as $O_n[t]$ increases, the obstacle susceptibility weight ρ_3 should also increase, thus decreasing ρ_T in order to respect the attraction domain defined in Figure 7.1.

7.2.4 Connectivity Susceptibility

As already stated in this Thesis, wireless networks play a crucial role in *MRS* since robots need to share information to infer their individual locations and solutions and control their positions to maintain the *MANET* connectivity. The requirement to ensure network connectivity often fails when robots move apart from their teammates. To improve the convergence rate of the *RDPSO* robots within the same subgroup, robots should spread out as most as possible. However, they must keep a maximum communication distance or minimum signal quality between them. In this perspective, one needs to find a good trade-off between the enforcing communication component ρ_4 and the mission parameters (*i.e.*, ρ_1 and ρ_2) since each robot has to plan its moves while maintaining the *MANET* connectivity.

The *RDPSO* takes use of the adjacency matrix A_c that directly depends of the link matrix L to identify the minimum/maximum distance/signal quality of each line, thus returning the position of the nearest neighbour in which a robot needs to ensure connectivity (*cf.*, section 4.4 and section 5.2). Similarly to the above methodology, let us now consider an experimental setup of a subgroup of two robots. Once again, one of the robots is placed in the optimal solution while the other robot is located 0.5 meters away from it (Figure 7.8).

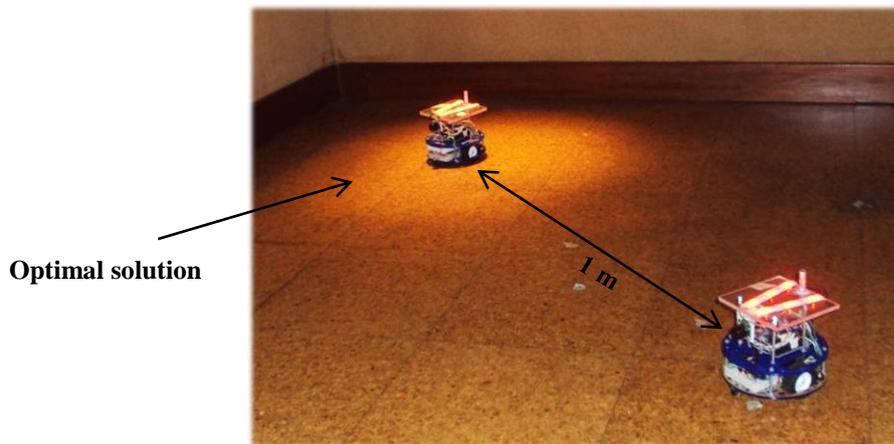


Figure 7.8. Experimental setup to evaluate the connectivity between two robots from the same subgroup.

The distance between robots x_{12} will be evaluated manipulating ρ_4 within a larger range while respecting the attraction domain represented by Figure 7.1, $\rho_3 = 0.1$ and $\rho_T = 0.7$ with $(\rho_1, \rho_2) = (0.2, 0.5)$, and $\alpha = 0.632$. The enforcing communication component ρ_4 will be set as $\rho_4 = \{0.4, 0.8, 1.2\}$ and robots will try to maintain a distance of 1 meter between them, *i.e.*, $d_{max} = 1$ meters (Figure 7.9).

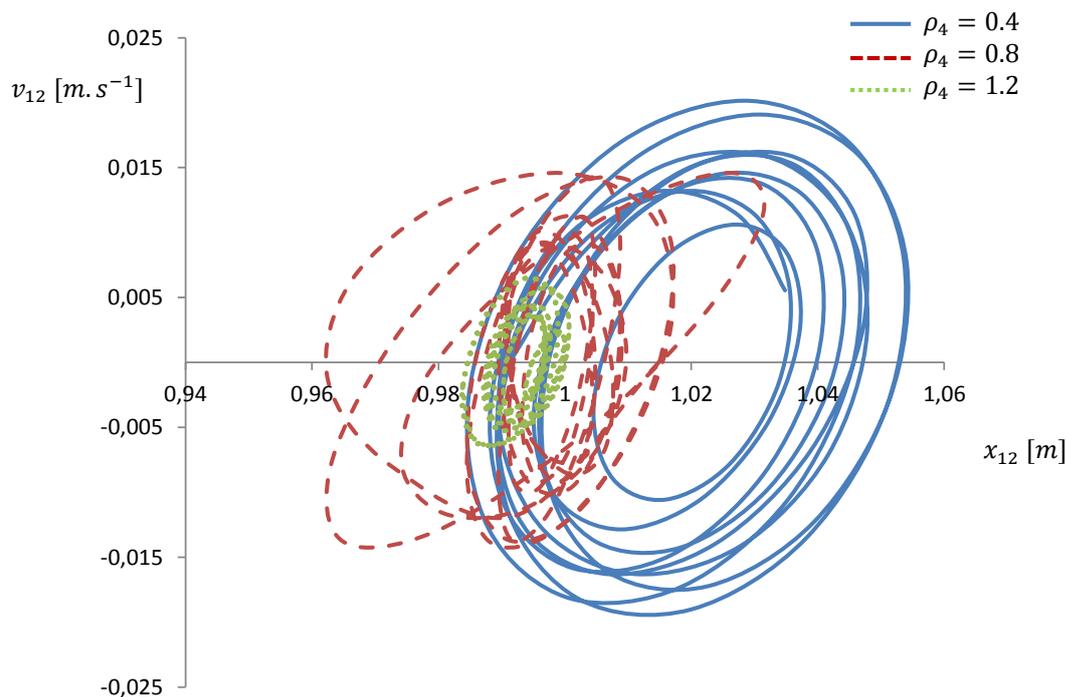


Figure 7.9. Distance between robots in phase space.

It may be observed in Figure 7.9 that, for any value of ρ_4 , the robot presents a spiral convergence in d_{max} vicinities. However, as ρ_4 increases, the convergence of the robot towards d_{max} also increases (the centre of the spiral approximates d_{max}). For $\rho_4 = 0.4$, the robot converges towards a distance superior to d_{max} with a larger spiral radius since it tries to get closer to the solution. For $\rho_4 = 1.2$, the robot ignores the solution and hardly moves from its initial position.

Considering the previous results, the easiest way to ensure the *MANET* connectivity is to increase the enforcing communication component ρ_4 when the distance between robots approximates the threshold value (*i.e.*, maximum distance or minimum signal quality). Therefore, exploiting introspective knowledge allows defining an agent-based contextual metric denoted as *robot proximity*. Nevertheless, this metric will depend on either ensuring a maximum communication distance between robots, d_{max} , or getting a minimum signal quality, q_{min} . Considering a d_{max} problem, one can define the *robot proximity* as it follows:

$$P_n[t] = \begin{cases} 1 - \frac{d_{nm}[t]}{d_{max}}, & d_{nm}[t] \leq d_{max} \\ 0 & , d_{nm}[t] > d_{max} \end{cases}, \quad (7.22)$$

where $d_{nm}[t]$ is the distance between robot n and its nearest neighbour m . Similarly, considering a q_{min} problem, the metric can be defined as:

$$P_n[t] = \begin{cases} 1 - \frac{q_{min}}{q_{nm}[t]}, & q_{nm}[t] \geq q_{min} \\ 0 & , q_{nm}[t] < q_{min} \end{cases}, \quad (7.23)$$

where $q_{nm}[t]$ is the minimum signal quality between robot n and its nearest neighbour m .

By exclusively using inter-robot relations one can only ensure the *MANET* connectivity locally. Therefore, besides the proposed microscopic level metric, a macroscopic level metric needs to be defined to globally improve the *MANET* fault-tolerance within each subgroup. As previously described in section 5.3, the connectivity of the network can be represented by the second smallest eigenvalue, also known as *Fiedler value*, λ_2 of the *Laplacian matrix* L (5.14). The value of λ_2 , depending on the number of robots within a subgroup, allows evaluating its connectivity. Therefore, a new macroscopic agent-based contextual metric that takes into account the *swarm connectivity* can be defined as:

$$C_s[t] = \begin{cases} \frac{\lambda_2}{N_s}, & \lambda_2 \geq 0 \\ 0 & , \lambda_2 < 0 \end{cases}, \quad (7.24)$$

When all robots within a subgroup can directly communicate (*i.e.*, one hop) with all their teammates, then $\lambda_2 = N_s$, thus resulting in $C_s[t] = 1$ which is representative of a fully connected subgroup. Therefore, as $C_s[t]$ tends to 0, ρ_4 should increase in order to ensure a more connected *MANET*.

7.2.5 Outline

The above presented context-based metrics can be used as benchmark to evaluate the *RDPSO* in terms of group behaviour. However, the fact that there are multiple evaluation metrics to determine the algorithm's performance makes their selection process complex.

Due to the *RDPSO* dynamics, it may not be sufficient to consider each evaluation metric independently. It is thus extremely important to find a way to evaluate its performance and ponder, simultaneously, the full set of metrics. Consequently, by systematically adjusting the parameters within the defined attraction domain in Figure 7.1, the *RDPSO* can be extended in order to control the swarm susceptibility to the main mission, obstacle avoidance and communication constraint. In this line of thought, it is based on the fuzzy approach (Couceiro, Machado, Rocha, & Ferreira, 2012), introduced in next section, that we will evaluate the performance and adaptively adjust the parameters of the *RDPSO*.

7.3 Fuzzified Systematic Parameter Adjustment

Robots' perception can significantly benefit from the use of contextual knowledge. The previous section presented the acquisition of environmental knowledge based on the sensing capabilities and shared information between teammates. Fuzzy logic (see section 3.1.3) will now be incorporated into the *RDPSO* algorithm to handle contextual information represented by the previously defined metrics. Other proposals with different formalism to represent contextual knowledge and reason, such as Bayesian decision analysis could be adopted as well (Turner, 1998) (Portugal & Rocha, 2012). Nevertheless, fuzzy logic addresses such applications perfectly as it resembles human decision-making with an ability to generate precise solutions from certain or approximate information (Liu, Abraham, & Zhang, 2007). The successful development of a fuzzy model is a complex multi-step process, in which the designer is faced with a large number of alternative implementation strategies and attributes (Garibaldi & Ifeachor, 1999). In sum, based on the information extracted from the inputs represented by the previously defined metrics, the fuzzy logic system can infer contextual knowledge which can be used to control the *RDPSO* behaviour by adapting its parameters (Figure 7.10).

This control architecture is executed at each iteration t , thus returning the fractional, social, obstacle and connectivity coefficients, α and ρ_i , $i = 2,3,4$. Subsequently, the cognitive coefficient ρ_1 is then defined in order to respect condition the attraction domain in Figure 7.1, *i.e.*, $0 < \sum_{i=1}^4 \rho_i < 2$, $i = 1,2,3,4$.

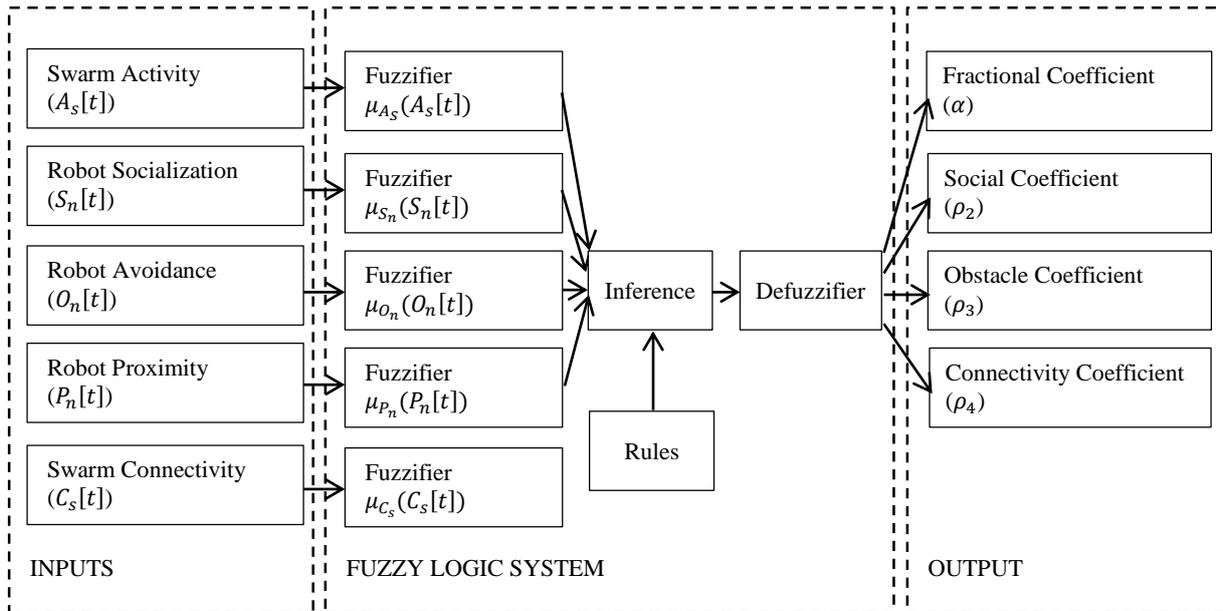


Figure 7.10. Fuzzy Logic System to dynamically adapt the parameters of the *RDPSO* algorithm.

As Figure 7.10 depicts, the overall organization of this architecture resembles the commonly used feedback controllers wherein contextual knowledge is extracted from data followed by a reasoning phase to control the robot. Hence, based on the metrics previously presented and their definition, one can assess the relation between the inputs and outputs of the fuzzy system.

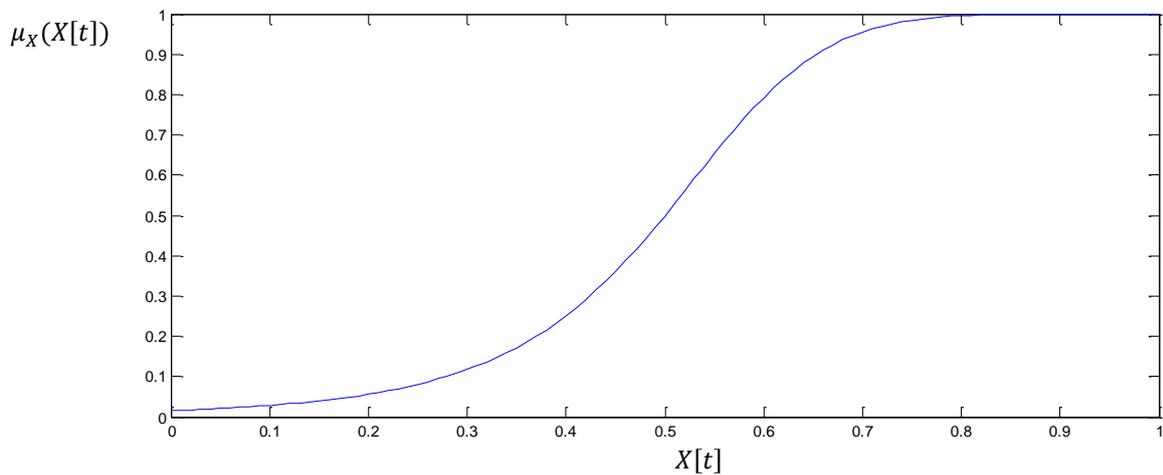


Figure 7.11. General membership function for each input.

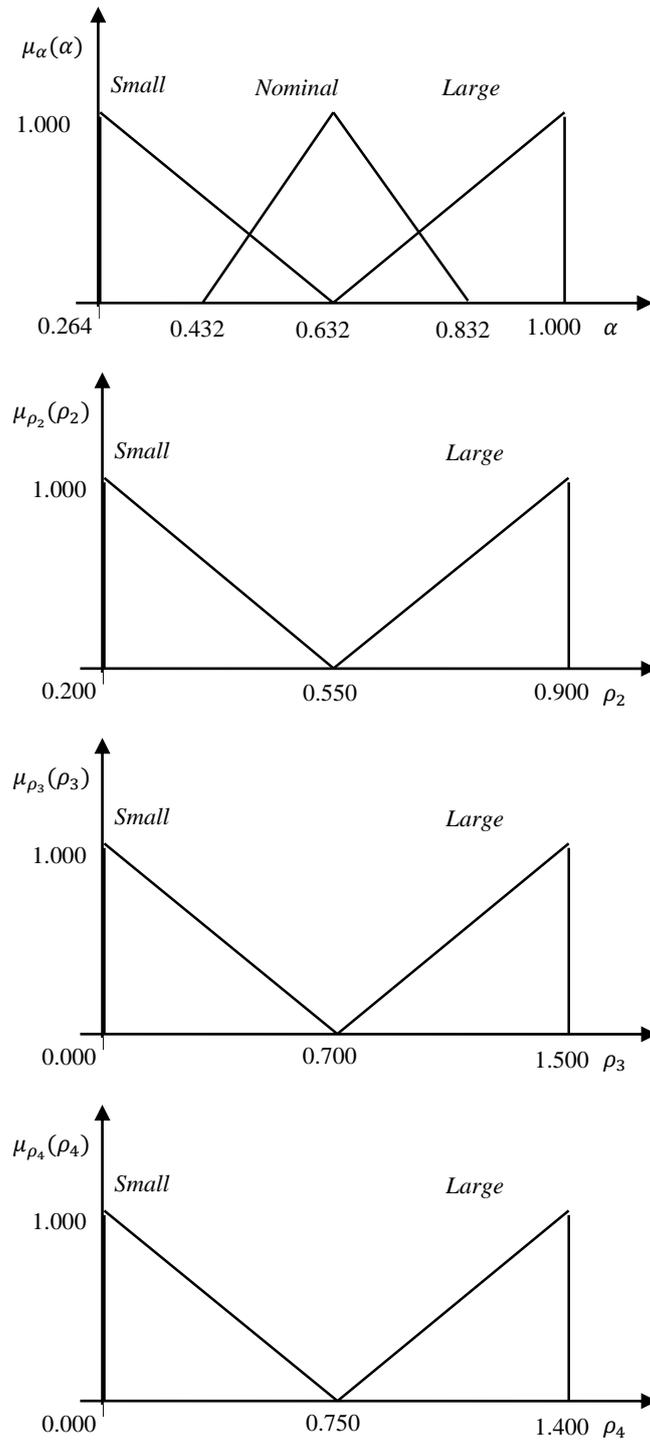


Figure 7.12. Membership functions to quantify the consequents for each coefficient.

To soften the decision-making system, the membership functions will be defined by *generalized bell-shaped* functions. The *generalized bell-shaped* function has one more parameter than the typical *Gaussian* function used in membership functions being defined as:

$$\mu_X(X[t]) = \frac{1}{\left[1 + \frac{(X[t]-c)^2}{a}\right]^{2b}}, \tag{7.25}$$

wherein parameters a , b and c correspond to the width, the slope and the center of the curve, respectively. Since metrics are all defined between 0 and 1, only half a curve is required to represent the status of the swarm and robots, *i.e.*, $c = 1$. On the other hand, for a soften response, the width and slope may be defined as $a = 0.5$ and $b = 3$ (Figure 7.11).

The *swarm activity* membership function $\mu_{A_s}(A_s[t])$ represents how *Active* the swarm is. As for the *robot socialization* $\mu_{S_n}(S_n[t])$, it represents how *Social* a robot is. The same analysis can be made for the *obstacle avoidance* membership function $\mu_{O_n}(O_n[t])$, thus representing how *Close* a given robot is to obstacles. The *robot proximity* membership function $\mu_{P_n}(P_n[t])$ represents how *Far* a robot is from its neighbour. Finally, as for the *swarm connectivity* membership function $\mu_{C_s}(C_s[t])$, it was defined to represent how *Connected* the swarm is.

As for the consequent functions, based on the preliminary experimental assessments presented in the previous section, one can define the following triangular membership relations represented in Figure 7.12. These functions not only allow softening and verbalizing the outputs but also, and more importantly, normalizing them within the attraction domain. It is noteworthy that, as previously mentioned in section 7.1, the cognitive parameter can then be defined as $\rho_1 = 2 - \rho_2 - \rho_3 - \rho_4$.

The *Mamdani-Minimum* was used to quantify the premise and implication. The *defuzzification* was performed using the centre-of-gravity (*CoG*) method (see section 3.1.3). By using the contextual knowledge represented in Figure 7.12, one can define the contextual rules that affect the behaviour of the system depending on the situation at hand. Therefore, the following diffuse **IF-THEN-ELSE** rules (Ruan & Kerre, 2002) are considered:

IF $A_s[t]$ **IS** *Active* **THEN** α **IS** *Small*
ELSE α **IS** *Large*
IF $O_n[t]$ **IS** *Close* **OR** $P_n[t]$ **IS** *Far* **OR** $C_s[t]$ **IS** **NOT** *Connected* **THEN** α **IS** *Nominal*
IF $S_n[t]$ **IS** *Social* **OR** $O_n[t]$ **IS** *Close* **OR** $P_n[t]$ **IS** *Far* **OR** $C_s[t]$ **IS** **NOT** *Connected*
THEN ρ_2 **IS** *Small*
ELSE ρ_2 **IS** *Large*
IF $O_n[t]$ **IS** *Close* **THEN** ρ_3 **IS** *Large*
ELSE-IF $P_n[t]$ **IS** *Far* **OR** $C_s[t]$ **IS** **NOT** *Connected* **THEN** ρ_3 **IS** *Small*
IF $P_n[t]$ **IS** *Far* **OR** $C_s[t]$ **IS** **NOT** *Connected* **THEN** ρ_4 **IS** *Large*
ELSE-IF $O_n[t]$ **IS** *Close* **THEN** ρ_4 **IS** *Small*

Figure 7.13. Set of *IF-THEN-ELSE* fuzzy rules do control robots' behaviour based on contextual information.

The rules from Figure 7.13 turn out to prioritize some *RDPSO* parameters over others, in which ρ_3 (*i.e.*, avoiding obstacles) and ρ_4 (*i.e.*, maintaining *MANET* connectivity) are the most pertinent

parameters. Although minor collisions are acceptable in swarm robotics, as this work focus on realistic applications such as *SaR*, it may be debatable to prioritize the mission over obstacle avoidance. The loss of multiple robots may jeopardize the mission objective (*e.g.*, finding victims). On the other hand, it is noteworthy that the obstacle avoidance parameter ρ_3 only affects the behavior of a specific robot when an obstacle is in its sensing range.

In brief, the fuzzy system herein proposed systematically adjusts the behaviour of the *RDPSO* in such a way that one can easily understand the contextual information regarding the robot and the subgroup by simply observing the parameters' evolution. Hence, the use of contextual knowledge improves robots' perception by allowing fast detection of environmental or mission changes (*e.g.*, detecting an obstacle) exploiting the information about the dynamics of real-world features. For example, Figure 7.14 depicts the evolution of ρ_i , $i = 1,2,3,4$, for a given robot facing the following situations:

- The robot is traveling until it first detects an obstacle (*i.e.*, ρ_3 increases);
- While still facing the obstacle, the robot moves too far away from its closest neighbour (*i.e.*, ρ_4 increases);
- The robot is able to circumvent the obstacle being still far from its closest neighbour (*i.e.*, ρ_3 decreases and ρ_4 increases);
- The robot is finally able to reduce the distance to its closest neighbour (*i.e.*, ρ_4 decreases).

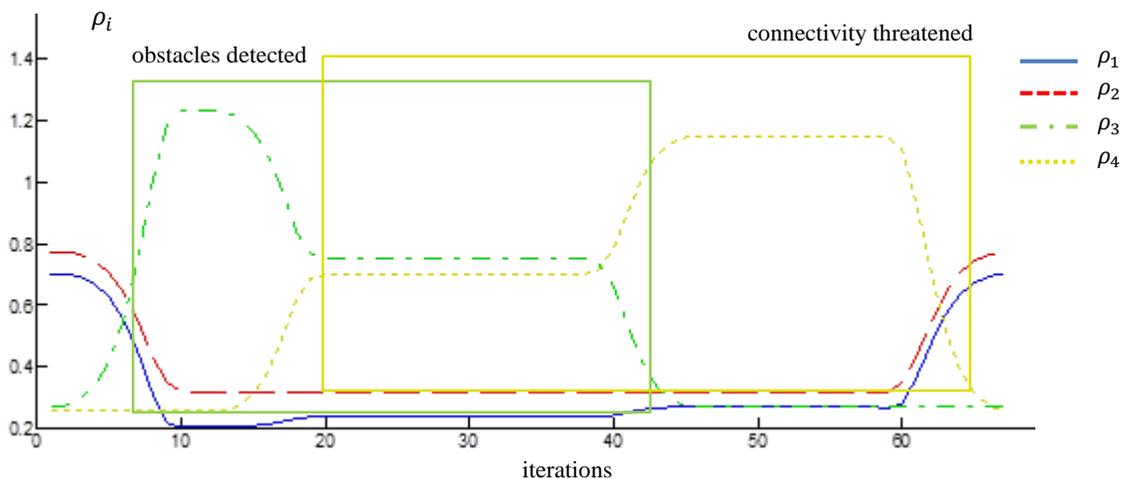


Figure 7.14. Parameters' evolution under a hypothetical situation.

Next section presents experimental results obtained using physical and simulated robots wherein the adaptive version of the *RDPSO* was evaluated and compared to the non-adaptive one.

7.4 Experimental Results

To demonstrate the relevance on correctly defining the *RDPSO* parameters and the utility of the proposed distributed adaptive strategy, a set of experimental results with multiple simulated and real robots is presented.

7.4.1 Numerical Simulations

Before evaluating the adaptive version of the *RDPSO*, a set of preliminary experiments were conducted on *MatLab* to study the parameterization effect. In this section, the use of virtual agents instead of realistic robots (*i.e.*, without considering robots dynamics and radio frequency propagation) was necessary to evaluate the *RDPSO* with arbitrarily large populations of robots and further understand parameters' influence. Hence, we first decided to evaluate the *RDPSO* under two different static configurations (Table 7.1), while respecting the particular attraction domain \mathcal{A}_p from Figure 7.1.

Table 7.1. Two sets of *RDPSO* parameters.

Parameters	α	ρ_1	ρ_2	ρ_3	ρ_4
S ₁	0.632	0.100	0.300	0.790	0.790
S ₂	0.632	0.200	0.400	0.690	0.690

Each set of parameters, S₁ and S₂, is defined by the tuple $\{\alpha, \rho_1, \rho_2, \rho_3, \rho_4\}$. The only difference between both sets S₁ and S₂ is how *RDPSO* parameters are defined within the particular attraction domain \mathcal{A}_p previously represented in Figure 7.1. The first set (S₁) is more conservative with higher ρ_3 and ρ_4 than S₂, thus allowing robots to preserve the *MANET* connectivity and avoid obstacles collision at any cost. The second set (S₂) is greedier with higher ρ_1 and ρ_2 than S₁, wherein robots' primary concern is to find the optimal solution (even if some collisions or *MANET* ruptures occur).

All of the experiments were carried out in a simulated scenario of 300×300 meters with obstacles randomly deployed at each trial, in which three two-dimensional benchmark cost functions $F_\varepsilon(x, y)$, $\varepsilon = 1, 2, 3$, were defined where x and y -axis represent the planar coordinates in meters (see Figure 7.15): *i*) *Gaussian* ($\varepsilon = 1$); *ii*) *Rastrigin* ($\varepsilon = 2$); and *iii*) *Rosenbrock* ($\varepsilon = 3$) (Molga & Smutnicki, 2005).

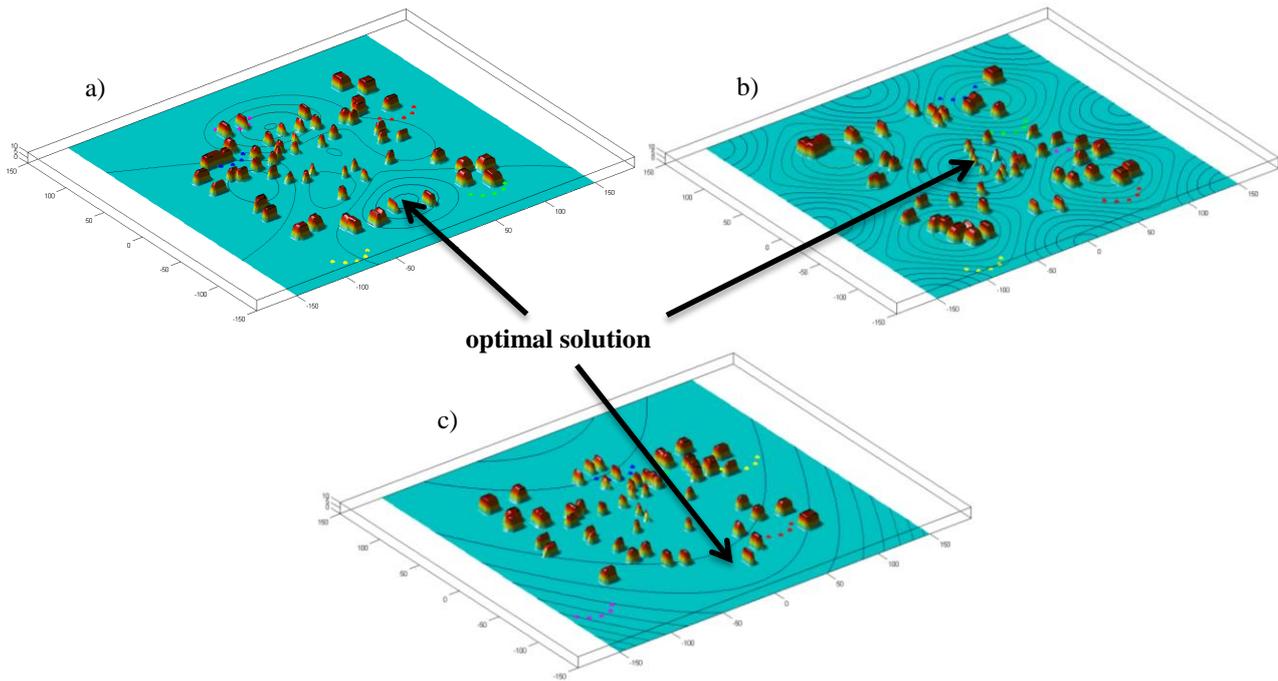


Figure 7.15. Virtual scenario with obstacles and 25 robots divided into 5 swarms. a) $f_1(x, y)$; b) $f_2(x, y)$; and c) $f_3(x, y)$.

In order to improve the interpretation of the algorithm performance, results were once again normalized as in section 4.7.1. Test groups of 100 trials of 500 iterations each were considered for $N_T = \{25, 50, 100\}$ robots. The maximum travelled distance between iterations was set to 0.5 meters, *i.e.*, $\Delta x_{max} = 0.5$ while the maximum communication distance between robots was set to $d_{max} = 30$ meters. The maximum range was considered since it is equivalent to the maximum communication distance in urban environment (*i.e.*, with obstacles) of *XBee* modules used *eSwarBots* platforms (section 3.2.3).

Let us then summarize the remaining *RDPSO* parameters (Table 7.2):

Table 7.2. *RDPSO* parameters.

<i>RDPSO</i> Parameter	Value
Number of trials	100
Time per trial [iterations]	500
N_T	{25,50,100}
N_s^{min}	0
N_s^l	5
N_s^{max}	8
SC_{max}	30
d_{max} [m]	30
Δx_{max} [m]	0.5
α	
ρ_1	
ρ_2	Table 7.1
ρ_3	
ρ_4	

Figure 7.16 depicts the performance of the algorithm by changing the total number of robots N_T , the objective function $f_\varepsilon(x, y)$ and the set of parameters S_1 and S_2 . The median of the best solution in the 100 simulation was taken as a final output for each different condition. In the *Gaussian* function $f_1(x, y)$, robots seem to perform well (Figure 7.16a-b). The reason may be that $f_1(x, y)$ only presents 2 sub-optimal regions that are far apart from the optimal solution (Figure 7.15a). In the *Rastrigin* function $f_2(x, y)$ robots also seem to have a good performance (Figure 7.16c-d). It is noteworthy that this function presents a difficult problem due to the relation between the size of the search space and the number of sub-optimal solutions (Figure 7.15b). In fact, when attempting to solve the *Rastrigin* function, most optimization or foraging algorithms easily fall into sub-optimal solutions. However, as the *RDPSO* is capable of maintaining a large diversity, it returns better results than expected. Finally, in the *Rosenbrock* function $f_3(x, y)$, the optimal solution is inside a long, narrow, parabolic shaped flat valley (Figure 7.15c). Robots are able to easily discover the valley (values above 0.9) but they seem to have some minor problems in finding the optimal solution (Figure 7.16e-f). Contrary to the previous functions (*i.e.*, $f_1(x, y)$ and $f_2(x, y)$) in which the target distribution may resemble foraging tasks in which robots need to find confined target locations (*e.g.*, toxic waste clean-up), the *Rosenbrock* function $f_3(x, y)$ is more like an olfactory-based swarming. In olfactory-based swarming a plume is a subject to diffusion and airflow, which increases the difficulty in finding the initial source (*e.g.*, detection of hazardous gases) (Marques, Nunes, & Almeida, 2006; Jatmiko, Sekiyama, & Fukuda, 2006). Nevertheless, independent of the target distribution, the median value of the solution was always greater than 0.95 (*i.e.*, optimal solution vicinities) regardless of the number of robots.

It should also be noted that the second set (S_2) presents worse results than the first one (S_1) in most situations (*e.g.*, Figure 7.16e-f). In other words, a greedy behaviour wherein robots give too much importance on finding the optimal solution seems to jeopardize the performance of the team since some collisions or communication ruptures may delay or even interfere with the collective intelligence. This phenomenon may be explained due to the prioritization of the robots' objectives. Although robotic teams are designed for specific applications (*e.g.*, finding a gas leak), the requirements to fulfil such applications (*e.g.*, ensuring *MANET* connectivity) need to be ensured for collective cooperation to emerge. Nevertheless, it is impossible to withdraw more specific conclusions about the influence of using a different parameterization or population on the algorithm performance by only looking at the median of the best solution over time. Therefore, a more exhaustive statistical analysis needs to be carried out so as to assist the design of robotics network dynamic partitioning algorithms for similar scenarios.

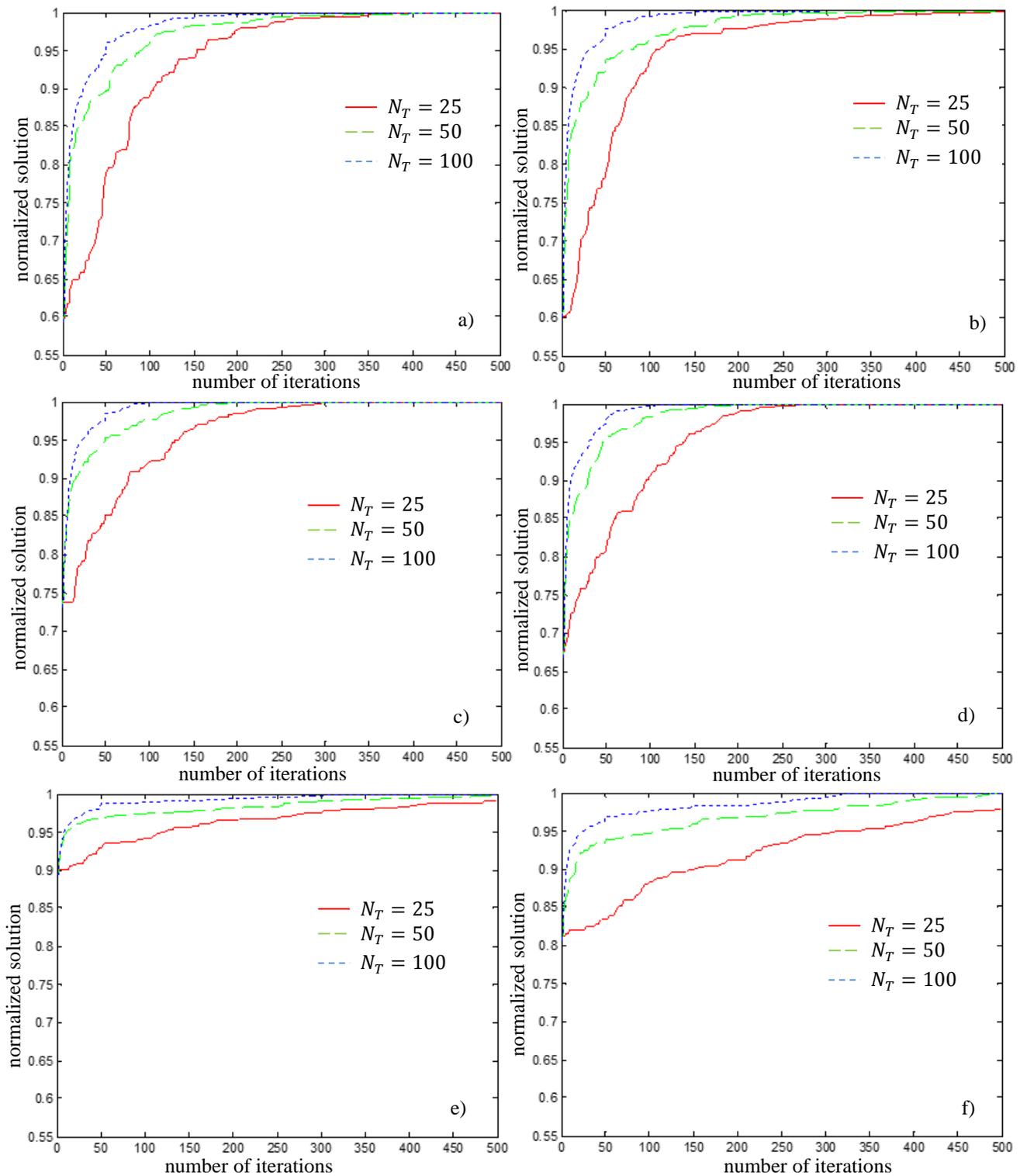


Figure 7.16. *RDPSO* evaluation changing the number of robots N_T for each objective function and set of parameters: a) $f_1(x, y)$ and S_1 ; b) $f_1(x, y)$ and S_2 ; c) $f_2(x, y)$ and S_1 ; d) $f_2(x, y)$ and S_2 ; e) $f_3(x, y)$ and S_1 ; and f) $f_3(x, y)$ and S_2 .

Section 4.7.1 presented a statistical analysis of the *RDPSO* using the *MANOVA* in order to evaluate the relationship between the population of robots and their maximum communication distance

(Maroco, 2010; Pallant, 2011). Similarly, we herein present a *MANOVA* analysis to further understand the impact of using a conservative and greedy behaviour (*i.e.*, set of parameters) while increasing the number of robots within the population. Therefore, the significance of the set of parameters and the number of robots (*i.e.*, independent variables) to the final solution and the runtime (*i.e.*, dependent variables) was analysed using a two-way *MANOVA* for each target distribution (*i.e.*, objective function). As in section 4.7.1, the assumption of multivariate normality was validated (Maroco, 2010; Pallant, 2011). The assumption about the homogeneity of variance/covariance matrix in each group was examined with the *Box's M Test* in the three objective functions. When the *MANOVA* detected significant statistical differences, we proceeded to the *ANOVA* for each dependent variable followed by the *Tukey's HSD Post Hoc*. This analysis was performed using *IBM SPSS Statistics* for a significance level of 5%.

As Table 7.3 depicts, the *MANOVA* revealed that the number of robots had a medium effect and significant on the multivariate composite independently on the objective function (with *p-value* = 0.001 and *Power* = 1.000). This indicates that the population of robots, as expected, has a crucial influence in the *RDPSO* performance.

Table 7.3. Multivariate test for the number of robots.

	$f_1(x, y)$	$f_2(x, y)$	$f_3(x, y)$
<i>Pillai's Trace</i>	0.139	0.201	0.203
<i>Partial Eta Squared</i> η_p^2	0.069	0.101	0.117

Table 7.4 shows that the set of parameters had a small effect, yet significant, on the multivariate composite except for the *Gaussian* function $f_1(x, y)$. In this last one, the different set of parameters does not present any statistically significant differences. Hence, the proposed set of parameters seems to have a minor influence over the *RDPSO* performance. However, at this point, it is still not clear if it has a positive or negative influence. Finally, the interaction between the two independent variables only had a small effect, yet significant, on the multivariate composite in *Rosenbrock* function ($f_3(x, y)$) (*Pillai's Trace* = 0.032; η_p^2 = 0.016; *p-value* = 0.001; *Power* = 0.957).

Table 7.4. Multivariate test for the set of parameters.

	$f_1(x, y)$	$f_2(x, y)$	$f_3(x, y)$
<i>Pillai's Trace</i>	0.006	0.015	0.045
<i>Partial Eta Squared</i> η_p^2	0.006	0.015	0.045
<i>p-value</i>	0.192	0.011	0.001
<i>Power</i>	0.350	0.800	0.998

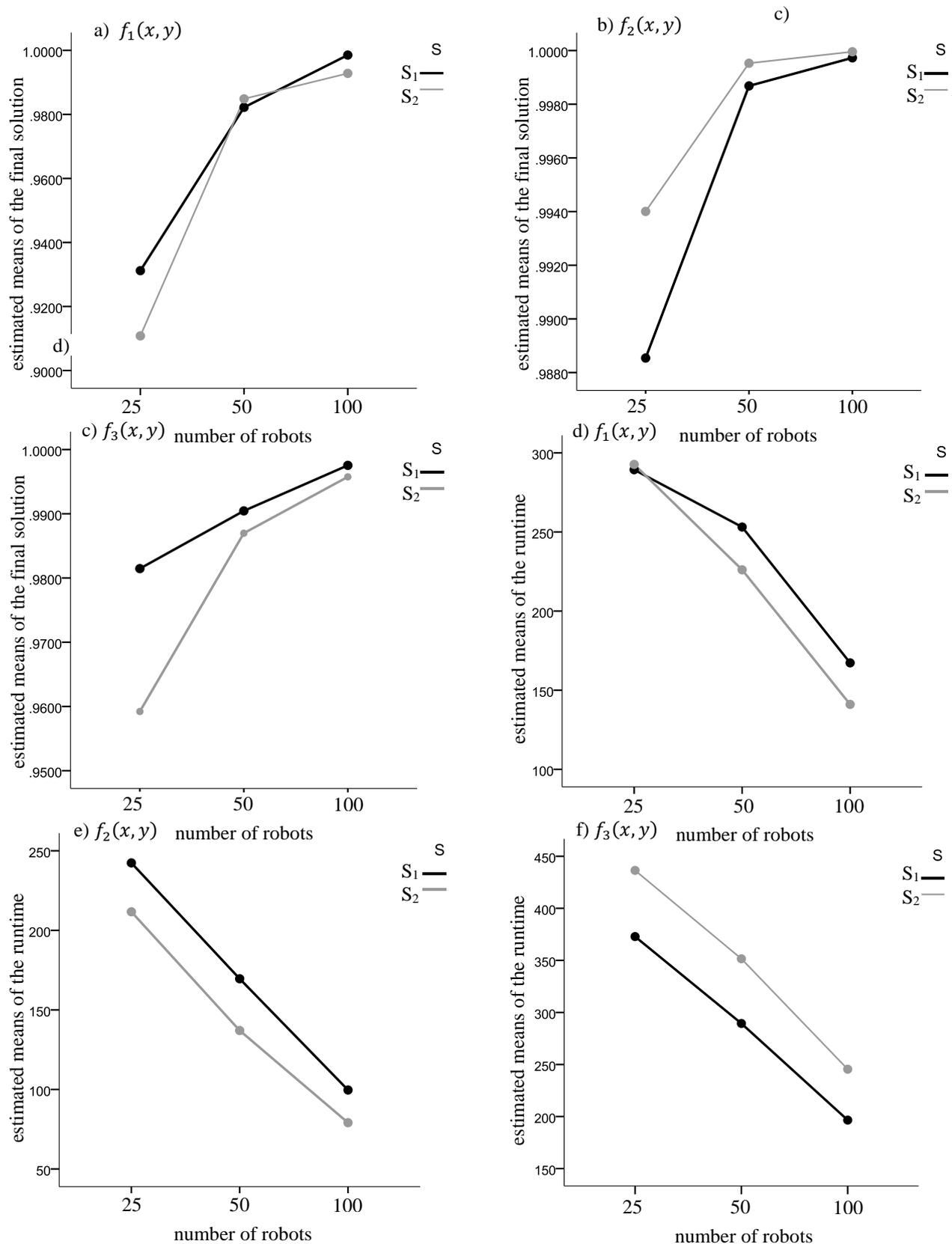


Figure 7.17. Estimated marginal means of the RDPSO performance for the: a) final solution using $f_1(x, y)$; b) final solution using $f_2(x, y)$; c) final solution using $f_3(x, y)$; d) runtime using $f_1(x, y)$; e) runtime using $f_2(x, y)$; f) runtime using $f_3(x, y)$.

After observing the multivariate significance in the number of robots and set of parameters, a univariate *ANOVA* for each dependent variable followed by the *Tukey's HSD Test* was carried out (Figure 7.17). It can be concluded that, in general, increasing the number of robots from 50 to 100 does not significantly improve the final solution of the *RDPSO* for such applications and target distribution. In other words, the algorithm is able to find the optimal solution with 50 robots in most situations. However, the runtime improves significantly as the number of robots increases (Figure 7.17d-f). There is an almost inverse linear relationship between the runtime and the number of robots. As for the set of parameters, the performance of the *RDPSO*, for both final solution and runtime, does not follow any tendency. For instance, a greedy behaviour (S_2) decreases the runtime for functions $f_1(x, y)$ and $f_2(x, y)$ and increases it for function $f_3(x, y)$ (Figure 7.17d-f). Despite using similar parameter values inside the previously defined particular attraction domain \mathcal{A}_p (Figure 7.1), it can be observed that small differences between both sets may result in considerable differences, mainly, in the algorithm convergence rate (*i.e.*, runtime). On the other hand, in most situations, one can slightly overcome the negative effect of a poor choice of parameters (within \mathcal{A}_p), by increasing the population of robots. However, for the second set of parameters (S_2), robots happen to collide and sometimes they cannot maintain the maximum communication distance between them. Even within the second set, this could be avoided if parameters were not constant values throughout the search. In other words, there are some situations in which robots should adapt their behaviour (Liu & Winfield, 2010). For instance, if a robot is near collision, the obstacle susceptibility weight ρ_3 should instantaneously increase, hence ignoring the mission and communication constraints.

Having studied the parameters' influence within the *RDPSO* algorithm, let us now present some experiments to evaluate the adaptive strategy herein proposed.

7.4.2 Dynamic Environments in Simulation

The use of simulated robots allows to evaluate the adaptive *RDPSO* within large populations of robots and larger dynamic scenarios. All of the experiments were carried out in a simulated scenario of 600×600 meters with an initial *Gaussian* target distribution, herein denoted as $F(x, y)$, with obstacles randomly deployed at each trial, *i.e.*, the same distribution presented in Figure 7.15a with twice the scenario dimensions. Test groups of 100 trials and 500 iterations each were considered for $N_T = \{25, 50, 100\}$ robots. Also, a minimum, initial and maximum number of 2, 5 and 8 subgroups were used. The maximum travelled distance between iterations was set as 0.750 meters, *i.e.*, $\Delta x_{max} = 0.75$, while the maximum communication distance between robots was set to $d_{max} = 15$ meters.

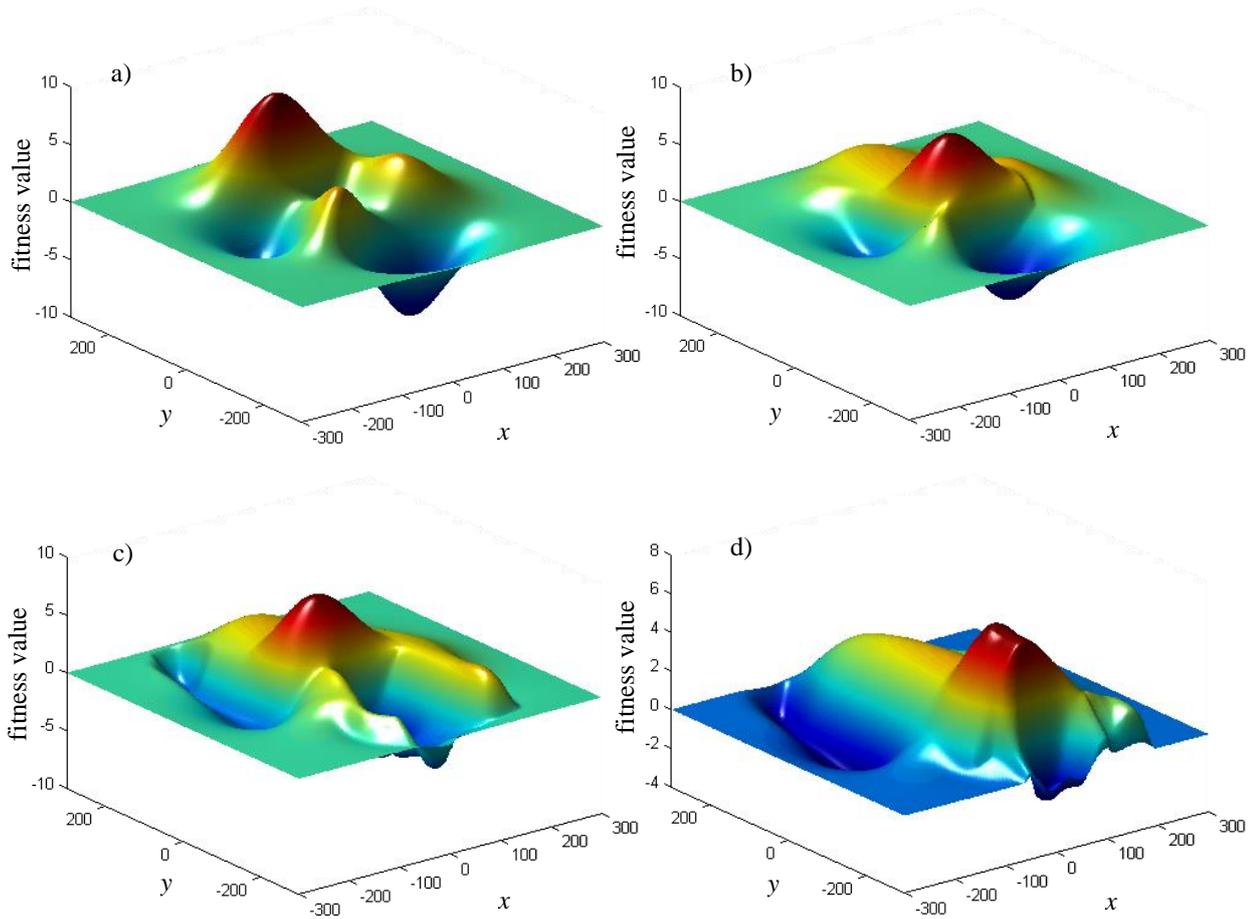


Figure 7.18. Planar motion of $F_1(x, y)$ peaks based on Forced Duffing Oscillator. a) $t = 0$; b) $t = 150$; c) $t = 300$; d) $t = 450$ iterations.

From the previous experiments, one can observe that both non-adaptive and adaptive algorithms present a high efficiency since the intrinsic features of the *RDPSO* (*i.e.*, social exclusion and inclusion presented on section 4.5) allows avoiding sub-optimal solutions in most situations. Therefore, to further compare both approaches, a dynamically changing environment is considered. Due to the continual changes of such environments, the optimal solution in the environment will also change over time. This demands that the *RDPSO* needs to be able not only find the solution in a short time, but also track the trajectory of the optimal solution in the dynamic environment. Non-adaptive algorithms, such as the regular *RDPSO*, usually present several drawbacks in dynamic problems since they lack the ability to track the non-stationary optimal solution in the dynamically changing environment (Carlisle & Dozier, 2000) (Cui & Potok, 2007).

Chaotic functions are the most common and well-studied way to generate non-stationary functions such as logistic functions (Morrison & De Jong, 1999). In this work, a general way to dynamically change the peaks location based on *Forced Duffing Oscillator (FDO)* is used (Tan & Kang,

2001). Hence, the function $F(x, y)$ is defined as a dynamic Gaussian function (*cf.*, Figure 7.15a) that changes over time based on Algorithm 7.1. A sequence of the $F_1(x, y)$ peaks' motion is represented in Algorithm 7.1. Dynamic *FDO* function generator.

. The motion of each peak can be configured through the tuple $\{\gamma_{FDO}, \omega_{FDO}, \varepsilon_{FDO}, \Gamma_{FDO}, \Omega_{FDO}\}$, wherein γ_{FDO} controls the size of the damping, ω_{FDO} controls the size of the restoring force, ε_{FDO} controls the amount of non-linearity in the restoring force, Γ_{FDO} controls the amplitude of the periodic driving force, and Ω_{FDO} controls the frequency of the periodic driving force. Although the tuple $\{\gamma_{FDO}, \omega_{FDO}, \varepsilon_{FDO}, \Gamma_{FDO}, \Omega_{FDO}\}$ may be randomly defined for a more unexpected and chaotic behaviour, to better understand the experimental results, it was defined with the constants $\{0.1, 1, 0.25, 1, 0.5\}$. To soften the surface, a circular averaging filter was also applied.

```

[X, Y, Z] = generate_surface() // the initial surface depends on user's choice; for instance, the MatLab peaks() function
returns a Gaussian distribution with 5 peaks (3 with positive amplitude and 2 with negative amplitude)
[zi, zj] = find(|Z| > zT) // detect the peaks; the threshold depends on the peaks' maximum amplitude – 20% of the abso-
lute maximum amplitude may be an option
Zb = zeros(size(Z)) // initialize the binary surface as a matrix of zeros
Zb(zi, zj) = 1 // the surface peaks are identified by logical ones
[B, L, N] = boundaries(Zb) // detect the region boundaries in the binary surface; for instance, the MatLab bwboundaries()
function returns the label matrix L as the second output argument and the number of peaks N as the third argument.
For i = 1:N
    [xP{i}, yP{i}] = find(L == i) // detect the peaks; the threshold depends on the peaks' maximum amplitude
– 20% of the absolute maximum amplitude may be an option
     $\dot{x}\{i\}(t) = -\gamma_{FDO}x(t) + \omega_{FDO}^2x(t) - \varepsilon_{FDO}x(t)^3 + \Gamma_{FDO} \cos(\Omega_{FDO}t)$  // chaotic motion based on Forced
Duffing Oscillator (FDO); one can use the MatLab ode45() to solve this differential equation problem
     $\dot{y}\{i\}(t) = x(t)$  // the  $\{\gamma_{FDO}, \omega_{FDO}, \varepsilon_{FDO}, \Gamma_{FDO}, \Omega_{FDO}\}$  tuple may be randomly defined for each peak
ZP = Z // initialize the modified surface as the original one
For t = 1:T // the total time (i.e., number of iterations) of the motion is defined by T
    For i = 1:N
        Zaux = ZP(xP{i}, yP{i}) // save the current peak in a temporary variable
        xP{i} = xP{i} + round(x{xP{i}}(t)) // the elements of the matrix needs to be integer values
        yP{i} = yP{i} + round(y{yP{i}}(t))
        ZP(xP{i}, yP{i}) = Zaux // new position of the peak
    ZP = filter(ZP) // a filter may be used for a more soften surface; the MatLab filter2() and fspecial() func-
tions may be used for that purpose
    visualize(X, Y, ZP) // optionally, one can visualize the new surface; the MatLab surf() function may be
used

```

Algorithm 7.1. Dynamic *FDO* function generator.

Similarly as before, Figure 7.19 depicts the performance of the non-adaptive and adaptive *RDPSO* under a dynamic environment. Analysing Figure 7.19, it is clear that the proposed mission can be accomplished by any number of robots greater or equal than 25. However, one can observe that the adaptive strategy improves the convergence of the *RDPSO*. That difference is more visible than in the previous static example, both for the median value and the variability of the solution. In the adaptive *RDPSO*, this last one is lower than the non-adaptive *RDPSO* and the difference increases as the number of robots increases. It is also clear that the non-adaptive *RDPSO* seems to be unable to

always successfully track the optimal solution, thus increasing the inconsistency of the final result obtained (larger interquartile range). Moreover, it is interesting to observe that, in the adaptive *RDPSO*, the line representing the third quartile (top solid blue line) gets closer to the one representing the median value (darker solid blue line). In other words, the data distribution turns out to be negatively skewed (*i.e.*, the mean is smaller than the median). This means that, in this case, as the goal is to maximize the normalized objective function, approximately 50% of the trials are around the desired objective value for the adaptive *RDPSO* under a dynamic environment.

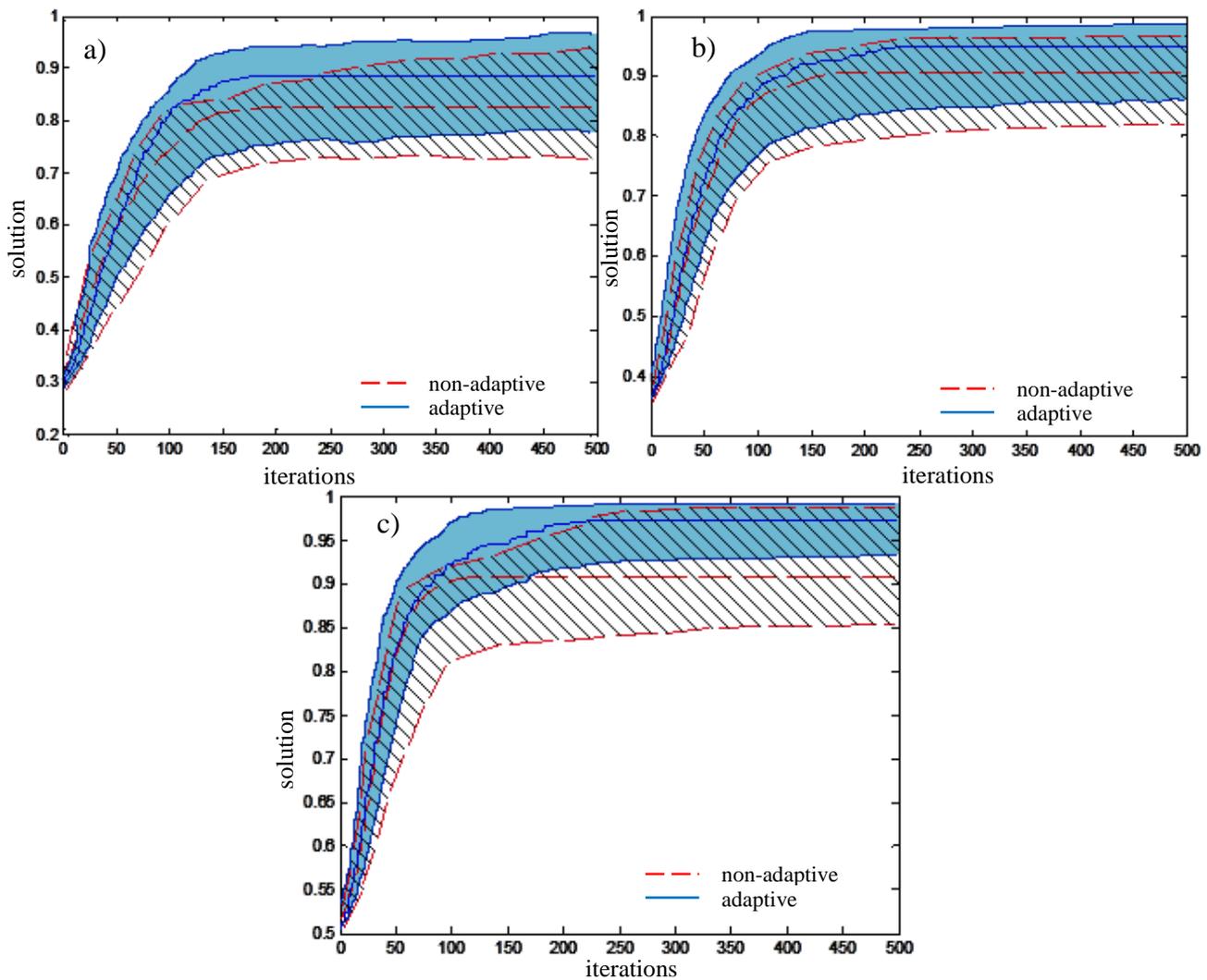


Figure 7.19. Performance of the non-adaptive and adaptive *RDPSO* under a dynamic environment for: a) $N_T = 25$ robots; b) $N_T = 50$ robots; c) $N_T = 100$ robots.

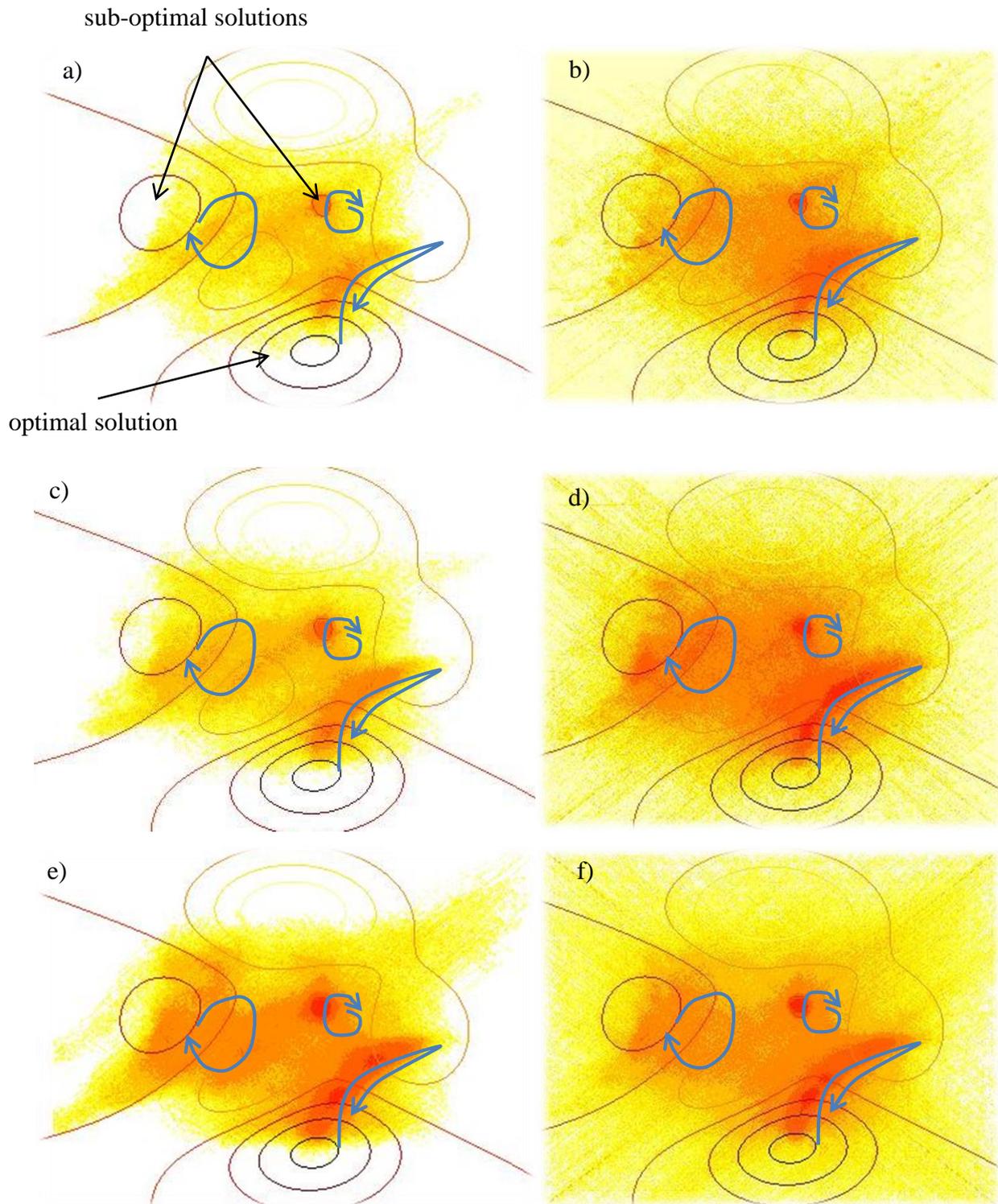


Figure 7.20. Heat maps representation of robot's trajectory in a dynamic environment. The blue arrows indicate the trajectory of the sub-optimal and optimal solutions. a) Non-adaptive $N_T = 25$; b) Adaptive $N_T = 25$; c) Non-adaptive $N_T = 50$; d) Adaptive $N_T = 50$; e) Non-adaptive $N_T = 100$; f) Adaptive $N_T = 100$.

To further improve the comparison between non-adaptive and adaptive strategies, heat maps were used (Figure 7.20). Heat maps can be designed to indicate how robots tend to be grouped together as well as reflecting the overall quality of the teams. The blue arrows represent the trajectory carried out by the sub-optimal and optimal solutions during the 500 iterations. Figure 7.20 presents

the heat map of evolutionary trajectories over the 100 trials of 500 iterations each for both non-adaptive and adaptive *RDPSO*, under a dynamic objective function, for each population of $N_T = \{25, 50, 100\}$ robots. Although the most visited regions correspond to the vicinities of the solutions, the algorithm is unable to effectively track the exact trajectory in some specific situations (*cf.*, Figure 7.20a). Once again, one may observe that the adaptive *RDPSO* presents a higher exploration behaviour keeping a high level of exploitation as the hot (*i.e.*, darkest) colours are more concentrated around the solutions' trajectories.

7.5 Discussion

At the beginning of this Thesis, we stated that *coopetition* was a fundamental phenomenon towards the survival of a given specie (chapter 1). Despite being fundamental at the macroscopic level, at the microscopic level (*i.e.*, the individual agent) one needs to consider another vital mechanism from the evolutionary cycle, namely, the mutation. Such mutation may occur at the genetic level or not but, regardless of how it occurs, it seeks to promote adaptive changes. Going back to the dinosaurs' case from chapter 1, some works represent dinosaurs as evolutionary failures, stating how unfitted they were for our world and how they seized to evolve and take advantage of their changing environment during their last 50 million years on Earth. Using *supertrees* computational methods to chart the evolution of species, researchers discovered that, while plants, birds and other species evolved swiftly, the dinosaurs plodded behind (Pisani, Yates, Langer, & Benton, 2001). As a consequence, a short time later they were extinct. As a counterexample, most of other creatures on earth have been evolving over the years and adapting to the multiple contextual changes around them, thus inspiring the creation of biological computational algorithms. For that reason, those bio-inspired algorithms have been employed in situations wherein conventional optimization techniques cannot find a satisfactory solution, for instance, when the optimization function is discontinuous, non-differentiable, or presents too many nonlinearly related parameters (Bonabeau, Dorigo, & Theraulaz, 1999). Nevertheless, the use of bio-inspired algorithms, including the particular case of swarm intelligence domain, comes with an additional cost: the superior parameterized complexity. This has been a known problem from computational complexity theory and, although the literature presents several strategies to solve it for fixed (constant) parameters, the same cannot be said for non-fixed (adaptive) parameters (Downey & Fellows, 1999).

This chapter intended to promote the tuning of the *RDPSO* collective behaviour by presenting a rationale behind its parameterization. To that end, the first section of this chapter focused on studying the stability of the *RDPSO* algorithm, so as to define a set of conditions where the convergence of

robots toward the solution is guaranteed (section 7.1). By doing so, it was possible to obtain an attraction domain that, for all intents and purposes, simply confines the relationship between the *RDPSO* parameters to a small region (Figure 7.1). This is highly important since it significantly reduces the complexity on settling the *RDPSO* algorithm, without resorting to arbitrary parameters that would not ensure its convergence and adequate performance.

The next scientific contribution consisted on outlining a set of fuzzy rules in which the outputs corresponding to the *RDPSO* parameters would systematically adapt based on the inputs defined by several context-based evaluation metrics. To that end, those context-based evaluation metrics were defined step-by-step and experimentally corroborated by a phase space analysis to the trajectories carried out by a pair of robots (section 7.2). Afterwards, the insights from section 3.1.3 were placed into practice toward the design of a control architecture to adapt robots' behaviour on-the-fly (section 7.3). As an example, when a given robot does not sense any obstacle within its sensing radius R_w , then the fuzzy system will reduce the obstacle susceptibility weight ρ_3 to zero, thus increasing the range of other parameters.

To evaluate the methodologies presented in this chapter, experimental results were conducted. Virtual agents (*i.e.*, without considering robot's dynamics and radio frequency propagation) were simulated to evaluate parameters' influence on large swarm populations, large and diverse scenarios and a large number of experiments for a statistically significant analysis (section 7.4.1). Those preliminary results obtained in section 7.4.1 fostered the proposal of some guidelines in the process of designing robotics network dynamic partitioning algorithms for similar scenarios. For instance, a more conservative behaviour (S_1 from Table 7.1) with special attention to obstacle avoidance and communication constraints may lead to better results in both terms of performance and runtime as the collective performance highly depends on the information shared between robots. Regarding the number of robots, it was expected that this would be a crucial variable in designing swarm algorithms. A better performance was achieved in a short amount of time as the number of robots increases. Moreover, a larger population of robots does not significantly disturb the communication network as the *RDPSO* is endowed with dynamic partitioning properties. However, if the main objective resides in fulfilling the mission regardless on the time needed, then a rationale on the size of the population of robots needs to be carried out. For instance, for the three simulated scenarios of 300×300 meters, a number of 50 robots proves to be enough regardless of the target distribution and obstacles' location.

Considering these results, it was possible to go a step further by comparing a non-adaptive *RDPSO* (*i.e.*, with constant parameters but yet within the attraction domain defined in Figure 7.1) with an adaptive *RDPSO* (*i.e.*, following the methodology from this chapter). Experiments with virtual agents were carried out for such comparison and it was possible to observe the superiority of the

adaptive version. Moreover, the *fuzzified* systematic parameters adjustment represented in the architecture from Figure 7.10 was able to improve the *RDPSO* responsiveness to time-varying source tracking. The heat maps from Figure 7.20 clearly show that robots under the adaptive *RDPSO* influence present a higher exploration level, thus increasing their capacity to track the exact trajectory of the sources.

7.6 Summary

The *RDPSO* was presented in chapter 4 as a sociobiologically inspired parameterized swarm algorithm that takes into account real-world *MRS* characteristics. The particularity of depending upon several parameters motivated the work presented on this chapter, by first proposing a novel methodology to study the convergence of swarm robotic algorithms with concepts from stability theory. A subgroup of two physical platforms was used to evaluate constraints such as robot dynamics, obstacles and communication, thus allowing defining metrics at both microscopic and macroscopic levels. Afterwards, those context-based metrics were used as inputs of a fuzzy system to systematically adapt the *RDPSO* algorithms.

Experimental results showed that the adaptive version of the algorithm presents an improved convergence when compared to the traditional one. Also, the distribution of target locations, *i.e.*, main objective function, does not greatly affect the adaptive algorithm performance. Even within a dynamic distribution, the adaptive *RDPSO* is able to track the optimal solution easier than in the non-adaptive case.

The success of such endeavour to improve the *RDPSO* algorithm with adaptive capabilities, despite remarkable, it is not the necessary unique feature of swarm strategies designed for real-world applications. In fact, several other methods have been proposed in the literature and identified as promising approaches for realistic swarm robotic applications. Therefore, and due to the flexibility of the herein proposed final solution of this ever-improving *RDPSO*, next chapter further evaluates and compares it with other state-of-the-art swarm algorithms under realistic exploration applications (*e.g.*, multi-robot mapping).

Benchmark

IN nature, some complex group behaviours arise in biological systems composed of swarms that are observed in a variety of simple social organisms (*e.g.*, ants, bees) (Bonabeau, Dorigo, & Theraulaz, 1999). One of the most relevant topics in *MRS* is the modelling and control of the population. Hence, the design of such bio-inspired *MRS* requires the analysis of the social characteristics and behaviours of insects and animals.

This chapter puts the *RDPSO* side-by-side with other state-of-the-art alternatives, thus classifying and discussing the theoretical advantages and disadvantages of the existing studies. Given the foundations of this work introduced in section 1.1, as well as the characteristics of the *RDPSO* previously presented, the benchmark will only consider algorithms that fall within the following summarized description:

- Belong to the domain of swarm robotics;
- Benefit from explicit communication;
- Are fully distributed.

That being said, let us then enumerate the contributions of this chapter (Couceiro, Vargas, Rocha, & Ferreira, 2014):

- i)* A selection of four state-of-the-art swarm algorithms for cooperative exploration tasks is described and theoretically compared with the herein proposed *RDPSO* (section 8.1);
- ii)* Such comparison is supported by a set of simulation experiments to evaluate the five algorithms under different configurations (*i.e.*, number of robots and maximum communication range), in which the three best performing algorithms are afterwards further compared in a source localization problem (section 8.2);

Sections 8.3 and 8.4 outline the discussion and main conclusions of this chapter.

8.1 Theoretical Comparison

Due to the successive improvements of the *RDPSO* and its positive outcome on several search tasks as presented in the previous sections, now comes the time to benchmark it with state-of-the-art alternatives. Over the past few years, some algorithms initially designed to solve tasks such as optimization problems have been adapted to embrace the principles associated to real robots. Within that list,

and including the aforementioned *RDPSO*, the following ones were found as the most promising for realistic search task applications:

- i) *Robotic Darwinian Particle Swarm Optimization (RDPSO)*;
- ii) *Extended Particle Swarm Optimization (EPSO)* (Pugh & Martinoli, 2006; Pugh & Martinoli, 2007);
- iii) *Physically-embedded Particle Swarm Optimization (PPSO)* (Hereford & Siebold, 2008; Hereford & Siebold, 2010);
- iv) *Glowworm Swarm Optimization (GSO)* (Krishnanand & Ghose, 2009a; Krishnanand & Ghose, 2009b);
- v) *Aggregations of Foraging Swarm (AFS)* (Gazi & Passino, 2003; Gazi & Passino, 2004).

Next sections systematically compares and discusses the *RDPSO* algorithm over the alternative swarm robotic algorithms.

8.1.1 Extended Particle Swarm Optimization (*EPSO*)

One of the first adapted versions of the *PSO* to handle real world constraints, such as obstacles, was presented by Pugh and Martinoli (Pugh & Martinoli, 2006; Pugh & Martinoli, 2007) (Algorithm 8.1). The main difference between the algorithm presented by those authors, denoted hereafter as *Extended PSO (EPSO)*, and the classical *PSO* is that each robot (or particle) only takes into consideration the information of the robots within a fixed radius R_r (omnidirectional communication). Hence, contrarily to the *RDPSO*, the *EPSO* algorithm does not use multi-hop connectivity and does not constrain robots' motion so as to ensure some degree of communication network connectedness.

Moreover, and also contrarily to the *RDPSO* algorithm in which obstacle avoidance behaviour is integrated in the main equations of robots' motion (as equation (4.5) depicts), the authors used the *Braitenberg* obstacle avoidance algorithm (Braitenberg, 1984). Hence, if a robot is executing a step of the algorithm and avoids an obstacle, it will continue moving in its new direction but will not modify its internal velocity representation. Although such methodology makes it possible to decouple the high level behaviour of robots from collision avoidance routines, such strategy may be unfeasible if one needs to study the stability of the algorithm considering obstacles influence over robots, or even define adaptive methodologies to systematically adjust all the algorithm parameters based on contextual information (chapter 7).

```

Initialize pose  $\langle x_n[0], \varphi_n[0] \rangle$  randomly defined
Loop:
  Evaluate the robot individual solution  $h_n[t]$ 
  If  $h_n[t] > h_{best}$  // robot has improved
     $h_{best} = h_n[t]$ 
     $\chi_1[t] = x_n[t]$ 
  Exchange information with the  $N_s$  neighbours about the individual solution  $h_n[t]$  and current position  $x_n[t]$ 
  Build a vector  $H[t]$  containing the individual solution of all  $N_s$  robots within a fixed radius  $R_r$ 
  If  $\max H[t] > H_{best}$  // swarm has improved
     $[H_{best}, j] = \max H[t]$  //  $j$  will return the best neighbor
     $\chi_2[t] = x_j[t]$ 
   $v_n[t + 1] = wv_n[t] + \sum_{i=1}^2 \rho_i r_i (\chi_i[t] - x_n[t])$ 
   $x_n[t + 1] = x_n[t] + v_n[t + 1]$ 
   $x_n[t + 1] = \text{Braitenberg\_obst\_avoid}(x_n[t + 1])$ 
until stopping criteria (convergence/time)

```

Algorithm 8.1. *EPSO* algorithm for robot n .

Pugh and Martinoli evaluated the performance of their learning technique for a simple task in robot groups of various sizes (Pugh & Martinoli, 2006; Pugh & Martinoli, 2007). The authors analysed how the performance of the standard *PSO* neighbourhood structure was affected by adapting it to a more realistic model which considered limited communication abilities. Experimental results obtained using the *Webots* simulator (Michel, 2004) showed that the adapted version of the *PSO* maintained good performance for groups of robots of various sizes when compared to other bio-inspired methods such as Genetic Algorithms. However, contrarily to the presented *RDPSO* algorithm, all bio-inspired methods used in this work, including the adapted *PSO*, tend to get trapped in sub-optimal solutions, *i.e.*, the authors did not present any strategy to avoid sub-optimal solutions.

8.1.2 Physically-embedded Particle Swarm Optimization (*PPSO*)

Similarly, Hereford and Siebold presented a *Physically-embedded PSO (PPSO)* in swarm platforms (Hereford & Siebold, 2008; Hereford & Siebold, 2010) (Algorithm 8.2). As in *RDPSO*, there is no central agent to coordinate the robots movements or actions. The authors constrained the movement of particles within a limited cone to avoid the omnidirectionality inherent to the common *PSO*. Although this strategy seems practical, this could be achieved by considering the dynamical characteristics of robots. For instance, the *RDPSO* benefits from fractional calculus of order α to avoid drastic changes in a robot's direction (see section 4.2).

The algorithm presented by Hereford and Siebold also assumed the synchronization of robots, such that robots would only compute a novel position after all other robots exchange the necessary information (*e.g.*, individual solutions) (Hereford & Siebold, 2008; Hereford & Siebold, 2010). Also, robots would only share their position if their own solution is the best solution in the whole swarm. This makes it possible to reduce the amount of communication traffic, however it also requires that robots stop after each iteration in order to handle all relevant information. This is an interesting strategy

when using broadcasting mechanisms since robots can share information among themselves without requiring a large amount of communication traffic. Nevertheless, such strategy will not significantly improve the algorithm performance if the team benefits from ad hoc communication with multi-hop properties.

```

Initialize pose  $\langle x_n[0], \varphi_n[0] \rangle$  randomly defined
Loop:
  Evaluate the robot individual solution  $h_n[t]$ 
  If  $h_n[t] > h_{best}$  // robot has improved
     $h_{best} = h_n[t]$ 
     $\chi_1[t] = x_n[t]$ 
  Exchange information with the  $N_s$  neighbours about the individual solution  $h_n[t]$ 
  Build a vector  $H[t]$  containing the individual solution of all  $N_s$  robots within a fixed radius  $R_r$ 
  If  $\max H[t] == h_n[t]$  // it is the best robot
    Exchange information of the current position  $x_n[t]$  if it is the best performing robot within a fixed radius  $r_c$ 
  If  $\max H[t] > H_{best}$  // swarm has improved
     $[H_{best}, j] = \max H[t]$  //  $j$  will return the best neighbor
     $\chi_2[t] = x_j[t]$ 
   $v_n[t + 1] = wv_n[t] + \sum_{i=1}^2 \rho_i r_i (\chi_i[t] - x_n[t])$ 
   $x_n[t + 1] = x_n[t] + v_n[t + 1]$ 
   $x_n[t + 1] = \text{constrain\_movement}(x_n[t + 1])$ 
  while  $\text{collision}() == 1$ 
     $x_n[t + 1] = \text{go\_back\_turn\_right}(x_n[t + 1])$ 
until stopping criteria (convergence/time)

```

Algorithm 8.2. *PPSO* algorithm for robot n .

Despite the potentialities of the physically-embedded *PSO* presented by Hereford and Siebold (Hereford & Siebold, 2008; Hereford & Siebold, 2010), experimental results were carried out using a population of only three robots, performing a distributed search in a scenario without sub-optimal solutions. Also, although authors present experimental results with one and two obstacles, the collision avoidance behaviour was not considered within the algorithm's equation. Instead, once a robot got stuck or collided, it was programmed to go back and turn right.

8.1.3 Glowworm Swarm Optimization (*GSO*)

A distributed biologically algorithm inspired on glowworm behaviour was presented and applied in *MRS* by Krishnanand and Ghose (Krishnanand & Ghose, 2009a; Krishnanand & Ghose, 2009b)(Algorithm 8.3). Similarly to the *RDPSO*, the *Glowworm Swarm Optimization (GSO)* algorithm features an adaptive decision domain which enables the formation of subgroups in the population where the goal is to partition the population of robots to track multiple sources concurrently. Nevertheless, and contrarily to the *RDPSO* that uses a set of fuzzy rules and performance evaluation of both robots and swarms of robots (*cf.*, chapter 7), the *GSO* acts more like a *PSO* with best neighbourhood solution information. In fact, to begin a search, a robot chooses a neighbour to be its leader

and moves toward it. The most probable choice for the leader is the one with the highest *luciferin*³¹ value (*i.e.*, individual solution), thus corresponding to the most probable direction of the source. As a result of this leader selection, subgroups form within the population and begin searching for nearby solutions. In other words, as no evolutionary techniques are used, it is shown that all members of a single cluster will converge to the leader at some finite time, and members of overlapping clusters will converge to one of the leaders asymptotically.

```

Initialize pose  $\langle x_n[0], \varphi_n[0] \rangle$  and luciferin level  $l_n[0]$  randomly defined
Loop:
   $l_n[t] = (1 - \rho)l_n[t - 1] + \gamma h_n[t]$  // update the luciferin level
   $N_n = \{j: d_{nj}[t] < r_d^n[t]; l_n[t] < l_j[t]\}$  //determine neighbours of glowworm  $n$  in the local-decision range and with higher luciferin levels
  Exchange information with the  $N_n$  selected neighbors about the individual solution  $l_n[t]$  and current position  $x_n[t]$ 
   $L_n[t] = \frac{l_j[t] - l_n[t]}{\sum_{k \in N_n} l_k[t] - l_n[t]}$  // calculate probability of selecting neighbour  $j$  from the  $N_s$  neighbors
   $[L_{best}, j] = \max L[t]$  //  $j$  will return the best neighbor glowworm
   $x_n[t + 1] = x_n[t] + s \left( \frac{x_j[t] - x_n[t]}{\|x_j[t] - x_n[t]\|} \right)$  // move toward neighbour  $j$ 
   $r_d^n[t + 1] = \min\{r_c, \max\{0, r_d^n[t] + \beta(\eta_t - |N_n|)\}\}$  // update local-decision range based on specified number of neighbours
until stopping criteria (convergence/time)

```

Algorithm 8.3. GSO algorithm for robot n .

Similarly to Pugh and Martinoli (Pugh & Martinoli, 2006; Pugh & Martinoli, 2007) and Hereford and Siebold (Hereford & Siebold, 2008; Hereford & Siebold, 2010), the authors also incorporated a low-level obstacle avoidance model, thus allowing robots to turn away from detected obstacles to prevent collisions.

Unfortunately, and despite the algorithm potentialities, the experiments were carried out using only four wheeled physical robots and a target location using a single sound source.

8.1.4 Aggregations of Foraging Swarm (AFS)

Another interesting approach was presented by Gazi and Passino in which the swarm is modelled based on attractant/repellent profiles as aggregations of foraging swarm (AFS) (Gazi & Passino, 2003; Gazi & Passino, 2004) (Algorithm 8.4). This kind of attractant/repellent profiles are consistent with biological observations where the inter-individual attraction/repulsion is based on an interplay between attractive and repulsive forces, with the attractive dominating on large distances and the repulsive dominating on short distances (Warburton & Lazarus, 1991). As in the *RDPSO*, the authors

³¹ According to *Merriam-Webster* dictionary, *luciferin* may be defined as “any of various organic substances in luminescent organisms (as fireflies) that upon oxidation produce a virtually heatless light”.

presented a stability and convergence analysis of their algorithm (section 7.1). To that end, the authors carried out a behavioural analysis followed by several simulation experiments so as to define the most adequate values of the system parameters. This is worth mentioning since most of the works define the parameters using a trial-and-error mechanism. Hence, some sort of mathematical formalism, such as stability analysis, is required to enable obtaining such comparable performance. Despite this, the authors did not present any mechanism for sub-optimal solutions avoidance. Therefore, the convergence of the swarm cannot be proved in the general case, thus demonstrating the difficulty of obtaining general guarantees for progress properties.

In this approach, the authors consider obstacles as a part of the objective function of the swarm. In other words, if robots need to maximize a given measure (*e.g.*, finding the larger density of victims in a catastrophic incident), obstacles are considered as global minima of their objective function. This is not too different from the *RDPSO* case that benefits from another component to define obstacles, *i.e.*, a monotonic and positive *sensing function* that depends on the sensing information (section 4.3). Nevertheless, the approach presented by Gazi and Passino does not make it possible to adjust robots' behaviour depending on the presence or absence of obstacles. Put differently, the swarm behaviour is limited to convergence in the vicinity of a solution or divergence from the neighbourhood of a sensed obstacle, being unable to adapt to the adequate contextual information.

Initialize pose $\langle x_n[0], \varphi_n[0] \rangle$ randomly defined

Loop:

Evaluate the robot individual solution $h_n[t]$ and distance to obstacles $g_n[t]$
 Exchange information with the N_T robots from the population about the individual solution $h_n[t]$ and current position $x_n[t]$
 $\sigma_n[t] = \gamma g_n[t] - \rho h_n[t]$ // build the attractant/repellent “ σ -profile” of attractant substances (main mission objective) and repellent substances (obstacles)

$$J(x_n[t]) = \sum_{k \in N_T} -(x_n[t] - x_k[t]) \left[a - b e^{-\frac{\|x_n[t] - x_k[t]\|^2}{c}} \right]$$

$v_n[t + 1] = -\nabla \sigma_n[t] + J(x_n[t])$ // compute the velocity of robot n based on the information of all N_T robots from the population and its own “ σ -profile”

$$x_n[t + 1] = x_n[t] + v_n[t + 1]$$

until stopping criteria (convergence/time)

Algorithm 8.4. *AFS* algorithm for robot n .

Although the work of Gazi and Passino does not assume any specificities about communication constraints, their model controls agents individually but each agent needs to know the positions of all other agents in the swarm (Gazi & Passino, 2003; Gazi & Passino, 2004). Therefore, we will consider that this approach requires multi-hop communication and the same principles assumed for the *RDPSO* will be considered.

8.1.5 Theoretical comparison

For theoretical comparison purposes, a summary of the previously presented algorithms is presented in Table 8.1, thus highlighting the most pertinent features for *MRS* applications. An empty cell in the table indicates that the algorithm does not benefit from that feature or there is no pertinent information in the literature to support it.

Table 8.1. Summary of swarm foraging algorithms used in the benchmarking study.

	<i>RDPSO</i>	<i>EPSO</i>	<i>PPSO</i>	<i>GSO</i>	<i>AFS</i>
robot dynamics	fractional calculus		constrained movements		
obstacle avoidance	artificial repulsion	low-level control	low-level control	low-level control	artificial repulsion
initial deployment	<i>EST</i> approach	random	random	random	random
communication	ad hoc multi-hop	broadcast	broadcast	broadcast	ad hoc multi-hop
fault-tolerance	multi-connectivity				
parameterization	stability analysis				stability analysis
avoid sub-optima	punish-reward mechanism based on natural selection				
multiple and dynamic sources	dynamic partitioning & fuzzy adaptive behaviour			partitioning	
computational complexity	$\mathcal{O}(2N_S)$	$\mathcal{O}(N_S)$	$\mathcal{O}(N_S)$	$\mathcal{O}(N_S)$	$\mathcal{O}(N_T)$
memory complexity	$\mathcal{O}(r_\alpha)$	$\mathcal{O}(1)$	$\mathcal{O}(1)$	$\mathcal{O}(1)$	$\mathcal{O}(1)$
communication complexity	$\mathcal{O}(N_S)$	$\mathcal{O}(N_S)$	$\leq \mathcal{O}(N_S)$	$\mathcal{O}(N_S)$	$\mathcal{O}(N_T)$

Robot dynamics consists of constraining agents' dynamics to fulfil the requirements inherent to the limited mobility of robots. From the previously presented works, only two considers this feature. The *PPSO* presents a simple rule to constrain robots' movements within a limited cone, while the *RDPSO* uses fractional calculus to include memory properties within the kinematical equation (section 4.2).

All the presented works handle *obstacles avoidance* with basically two strategies: *i*) low-level control (*EPSO*, *PPSO* and *GSO*); and *ii*) artificial repulsion mechanisms (*RDPSO* and *AFS*). Despite using different algorithms within such strategies, the main idea remains the same. Low-level control strategies trigger routines whenever robots sense obstacles, thus allowing decoupling the high-level

behaviour of robots from collision avoidance routines. Nevertheless, contrarily to the artificial repulsion mechanisms, low-level control routines do not support the integration of collision avoidance susceptibility within the algorithm behaviour (*cf.*, section 4.3).

One of the common approaches in the *initial deployment* of mobile robots is using a random distribution along the scenario (*EPSO*, *PPSO*, *GSO* and *AFS*). This methodology is the simplest way of deploying robots since that, in most situations, the distribution of the points of interest is usually random. However, in real situations, it is necessary to ensure several constraints of the system (*e.g.*, *MANET* connectivity), hence increasing the complexity of the random distribution. In addition, random deployment may cause unbalanced deployment and, therefore, increase the mission cost. Alternatively, section 5.1 presented the *EST* initial deployment strategy applied to the *RDPSO* algorithm that also shares some random properties. However, this methodology secures that the robots from the same subgroup (*i.e.*, cluster of the swarm population) are initially and autonomously deployed in an unknown environment, while avoiding areas of no interest (*i.e.*, obstacles) and maintaining *MANET* multiple connectivity.

Most of the works consider broadcast *communication* with purely local interactions over some specified range, in which robots only cooperate with their neighbours (*EPSO*, *PPSO* and *GSO*). Although this is the classical approach, recently many works suggested some kind of global communication without any pre-existent infrastructure, denoted as multi-hop ad hoc communication (*RDPSO* and *AFS*). This makes it possible for robots to communicate with other robots outside their direct (*i.e.*, one-hop) range and in the absence of a communication infrastructure. It is noteworthy that such strategy increases the communication overhead of the system. Nevertheless, if combined with partitioning strategies (*cf.*, section 4.5) and adequate communication rationale (*cf.*, chapter 6), it becomes possible to reduce the number of robots within each team, and the advantages inherent to it are countless when compared to broadcast communication.

As one can easily imagine, ensuring *MANET*'s connectivity and robustness is much more demanding than in infrastructured networks. As a result, to prolong the *MANET* lifetime and prevent loss of connectivity, *fault-tolerance* strategies are needed. A simple but efficient strategy is the one presented in section 5.2, wherein robots' movements within the *RDPSO* are controlled to allow significant node redundancy guaranteeing a multi-connectivity strategy. This means that, in the worst case, a multi-connected *MANET* requires the failure of multiple robots to become disconnected. All remaining algorithms do not present any fault-tolerance strategy.

Algorithms' *parameterization* enables the computation of values, or range of values, that would result in an improved performance. Most of the works in optimization or swarm applications present

trial-and-error methodologies, not benefiting from a formal mathematical analysis. Among the previously presented algorithms, only the *RDPSO* (see section 7.1) and the *AFS* from Gazi and Passino (Gazi & Passino, 2003; Gazi & Passino, 2004) presented a formal analysis of their algorithms, thus restricting the parameters' definition to a range of values.

Bio-inspired algorithms usually benefit from evolutionary techniques to *avoid sup-optimal solutions*. The *RDPSO* handles such problem perfectly by mimicking natural selection through the principles of social exclusion and inclusion, *i.e.*, adding and removing robots to subgroups as explained in section 4.5. In brief, as a recap, socially active robots from the same subgroup cooperate in the search task toward maximizing a given objective function (*e.g.*, gas leak, fire outbreak, number of victims, among others). Socially excluded robots randomly wander in the scenario instead of searching for the objective function's optimal solution like the other robots in the active subgroups. However, they are always aware of their individual solution and the global solution of the socially excluded group. This approach improves the algorithm, making it less susceptible to becoming trapped in sub-optimum solutions. The other algorithms do not consider any specific technique to avoid sub-optimal solutions.

Similarly, only the *RDPSO* and Krishnanand and Ghose's *GSO* are fitted to handle *multiple and dynamic sources* (Krishnanand & Ghose, 2009a; Krishnanand & Ghose, 2009b). They both use partitioning techniques for that end. Moreover, the *RDPSO* also uses an adaptive control system to systematically adjust its parameters based on contextual information (see section 7.3). This kind of adaptive mechanism is used, for instance, to control the swarm activity, balancing the exploitation and exploration levels of the group or each individual agent (Yasuda, Iwasaki, Ueno, & Aiyoshi, 2008; Wakasa, Tanaka, & Nishimura, 2010). These phenomena were explained in section 7.2.

The *computational complexity* refers to the system requirements for algorithm computation. The total number of robots, *i.e.*, population, is represented by N_T . If an algorithm benefits from partitioning features or local interactions, then the number of robots within a subgroup or the broadcast signal is represented by N_S , wherein $N_S \leq N_T$. All the previously mentioned algorithms, except the *AFS*, are endowed with partitioning techniques. Nevertheless, the *RDPSO* presents twice the computational complexity of the other algorithms that are endowed with partitioning techniques. This is due to the fault-tolerance characteristics (*cf.*, section 5.2) that require the computation of a sorting algorithm (Bhalchandra, Deshmukh, Lokhande, & Phulari, 2009).

The *memory complexity* refers to the system requirements in terms of data storage. Contrarily to the other algorithms that only require information about the previous iteration, *i.e.*, $\mathcal{O}(1)$, the *RDPSO* exhibits a memory complexity that depends on the truncation of the fractional order series r (section 4.2.1). Nevertheless, this is a difference that may be neglected because r is usually small and depends

on the requirements of the application and the features of the robots. For instance, as proved in section 4.2.1, a $r = 4$ for the *eSwarBot*, leads to results of the same type as for a $r > 4$, hence leading to a memory complexity of the *RDPSO* algorithm of $\mathcal{O}(4)$.

The *communication complexity* refers to the local and/or global communication overhead. The algorithms that benefit from partitioning or communication broadcast strategies present a communication complexity smaller than the ones that are not endowed with such features. From the previously presented algorithms, only the *AFS* is not endowed with any of those strategies, thus resulting in a higher communication complexity, *i.e.*, all robots within the population need to communicate with each other. Note that in this chapter we do not consider the size of the message itself, since all algorithms roughly share the same type of information. However, note that only the *RDPSO*, as proposed in section 6.1, and the *PPSO* (section 8.1.2) consider optimizing the messages based on some heuristic rules. Moreover, the *RDPSO* goes even further by optimizing the *AODV* routing protocol based on *RDPSO* dynamics (section 6.2).

The following section presents experiments with simulated and real robotic platforms so as to experimentally assess and compare the performance of these five algorithm in exploration tasks.

8.2 Experimental Results

Let us now support the theoretical comparison with both simulation and real-world experiments.

8.2.1 Simulation Experiments

The *Multi-Robot Simulator (MRSim)* was used to evaluate and compare the five previously presented swarm techniques. All algorithms were evaluated while changing the number of robots within the population $N_T = \{10, 20, 30\}$ and the maximum communication range $d_{max} = \{30, 100\}$ meters. The communication range was based on common values presented in the literature for both *ZigBee* and *WiFi* communication (Couceiro, Rocha, & Ferreira, 2011b). To significantly test and compare the different algorithms, 30 trials of 500 iterations for each (N_T, d_{max}) combination were conducted. Also, to perform a straightforward comparison between the algorithms, robots were randomly deployed in the scenario presented in Figure 3.9 from section 3.2.4 with an area of $A = 2975 \text{ m}^2$.

Exploring and building a map of the scenario was used as the mission objective to evaluate the five algorithms. Hence, the objective function of the team of robots was defined as a cost function in which robots need to minimize the map's entropy, *i.e.*, the uncertainty about the map. Please refer to Rocha *et al.* for a more detailed description (Rocha, Ferreira, & Dias, 2008).

Each robot n computes its best frontier cell as:

$$m_i^s = \operatorname{argmax}_{m_i \in \mathcal{N}(x_n[t], R_w)} [\psi(x_n[t], m_i) \|\vec{\nabla} H(m_i)\|], \quad (8.1)$$

wherein $\mathcal{N}(x_n[t], R_w)$ represents the set of frontier cells located in the neighbourhood of robot n with sensing radius R_w . The coefficient $\psi(x_n[t], m_i) \in [0, 1]$ measures if the cell m_i is in line-of-sight from a position $x_n[t]$, which also implies that cell m_i is likely to be empty. Moreover, the entropy of the cell m_i is represented by $H(m_i)$ and may be calculated as:

$$H(m_i) = -p(m_i) \log[p(m_i)] - (1 - p(m_i)) \log_2[1 - p(m_i)], \quad (8.2)$$

being $p(m_i)$ the probability that a grid cell is occupied. The performance metric used is the exploration ratio of the scenario over time (number of iterations). The exploration ratio can be obtained by normalizing the mapped scenario as it follows:

$$\eta_{exp}[t] = \frac{A_e[t]}{A_a}, \quad (8.3)$$

wherein A_a is the useful area of the scenario only considering free cells, while $A_e[t]$ is the scenario explored up to time, or iteration, t .

As Figure 8.1 depicts, the median of the best solution over the 500 trials was taken as the final output for each (N_T, d_{max}) combination. As it is possible to observe, the *RDPSO* outperforms the other methods for all (N_T, d_{max}) configurations tested. Nevertheless, such difference when compared to the *AFS* and the *GSO* decreases as the population of robots increases. For instance, for the configuration of $(N_T, d_{max}) = (30, 30)$, *i.e.*, Figure 8.1e, the *GSO* presents a better performance than the *RDPSO* during the first iterations while the *AFS* closely follows the same performance as the *RDPSO*.

To facilitate a straightforward comparison and since some of the algorithms present a similar performance, the area under the curve (*AUC*) can be used (Couceiro, et al., 2013). This is a common measure used to analyse the accuracy of receiver operating characteristic (*ROC*) curves that represent the performance of classifiers.

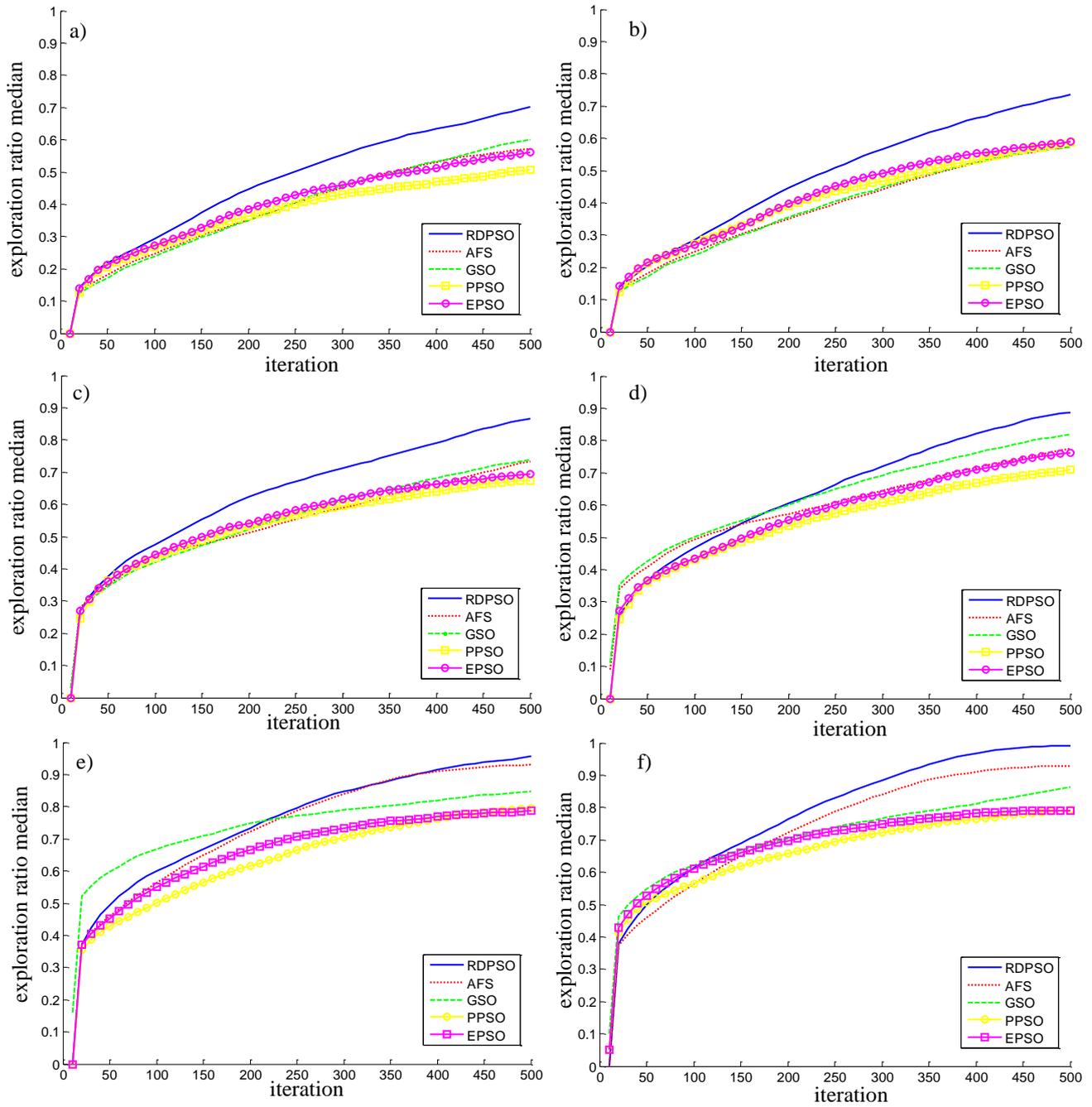


Figure 8.1. Median of the exploration ratio $\eta_{exp}[t]$ over the 500 iteration for each method. a) $(N_T, d_{max}) = (10, 30)$; b) $(N_T, d_{max}) = (10, 100)$; c) $(N_T, d_{max}) = (20, 30)$; d) $(N_T, d_{max}) = (20, 100)$; e) $(N_T, d_{max}) = (30, 30)$; f) $(N_T, d_{max}) = (30, 100)$.

As the exploration ratio $\eta_{exp}[t]$ is a discrete function with $t \in \mathbb{N}_0$, the *AUC* may be computed by the sum of each value over the 500 iterations. Moreover, one can normalize the *AUC* by dividing it by 500, thus resulting in a representation of the probability that a team of robots under a given algorithm has to explore the whole scenario. Hence, the normalized *AUC* may be calculated as:

$$AUC = \frac{1}{500} \sum_{k=0}^{500} \eta_{exp}[k], \quad (8.4)$$

The *AUC* of each set of trials is represented using boxplot charts. As one may observe in Figure 8.2, the influence of the population is more significant than the communication range. This should be expected as swarm intelligent algorithms perform well for larger population of robots, *i.e.*, it is possible to observe a higher degree of collective emergent behaviours as the population grows (Beni, 2004). Nevertheless, it is still possible to observe that, in most methods, an increase in the maximum communication range results in a minor improvement in the exploration ratio accuracy and a significant one in its precision, *i.e.*, smaller interquartile range. In other words, the outcome becomes more predictable and regular as the maximum communication range increases. Regarding the comparison between algorithms, it is possible to observe that both *PPSO* and *EPSO* present a similar performance with a probability of successfully exploring the whole scenario of almost 70% for a population of 30 robots.

The same may be observed for both *AFS* and *GSO* algorithms, in which a superior performance of almost 75% may be observed for such population. Finally, the *RDPSO* outperforms the other methods depicting a probability of successfully exploring the whole scenario of approximately 80% for the maximum population. This 5% difference may be generalized for all other (N_T, d_{max}) configurations tested. Nevertheless, such a difference is not linear and although the *GSO* presents a slightly better performance than the *AFS* for smaller populations, it seems that the *AFS* is able to overcome the *GSO* as the number of robots increases. Also, and Figure 8.1 depicts, the *AFS* presents a similar performance to the *RDPSO* for larger populations of robots.

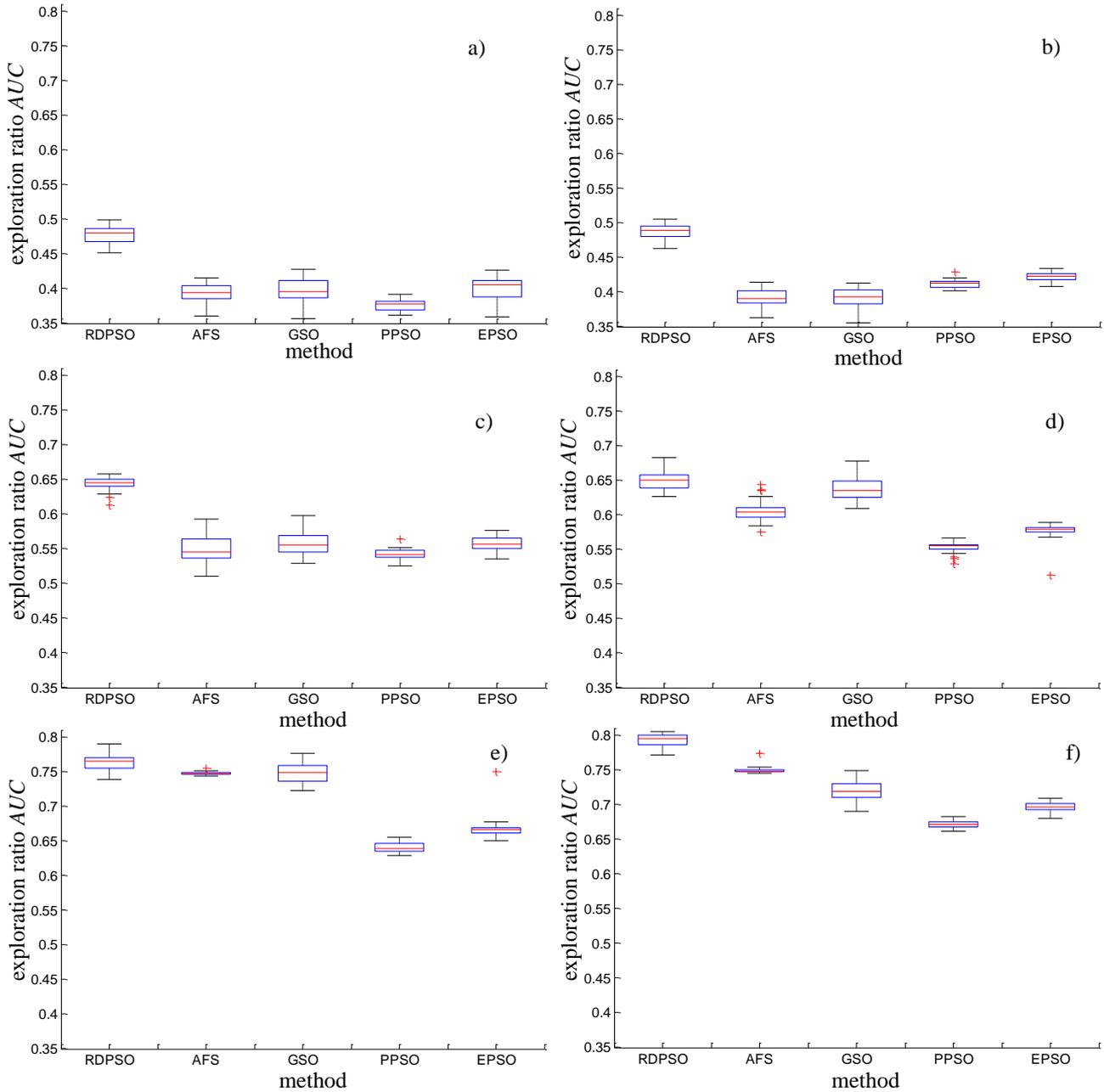


Figure 8.2. AUC of the exploration ratio $\eta_{exp}[t]$ over the 500 iteration for each method. a) $(N_T, d_{max}) = (10, 30)$; b) $(N_T, d_{max}) = (10, 100)$; c) $(N_T, d_{max}) = (20, 30)$; d) $(N_T, d_{max}) = (20, 100)$; e) $(N_T, d_{max}) = (30, 30)$; f) $(N_T, d_{max}) = (30, 100)$.

Hence, for further evaluation, next section compares the three best performing algorithms, namely *RDPSO* herein proposed, *AFS* (Gazi & Passino, 2003; Gazi & Passino, 2004) and *GSO* (Krishnanand & Ghose, 2009a; Krishnanand & Ghose, 2009b), using 14 mobile physical robots.

8.2.2 Real-World Experiments

In this section, the effectiveness of using the three best performing algorithms from the previous simulation experiments on swarms of *e-pucks* (Mondada, et al., 2009) equipped with *Gumstix Overo COM* turret to benefit from inter-robot *WiFi* communication is further explored (*cf.*, section 3.2.3). Due to the limitations of those turrets, all communication was centralized into a single server by means of *TCP/IP* sockets. To that end, an e-puck network manager was created on the server side to forward the data between the e-pucks and to store the necessary information to evaluate the *RDPSO*, *AFS* and *GSO* algorithms (see section 3.2.3). Although this does not enable the comparison of the algorithms under different communication ranges and even paradigms (*e.g.*, single-hop *vs* multi-hop communication), the previous experiments already considered this variable. Moreover, by not considering the *MANET* constraints, here we will only focus on evaluating the behavioural aspect of the algorithms. With the purpose on maintaining the scope around *SaR* applications, these experiments consisted of collectively finding 2 “victims” emulated by *e-pucks* on a 2.0×1.8 meters ($A = 3.6 \text{ m}^2$) scenario (Figure 8.3a).

The *e-pucks* are equipped with three omnidirectional microphones that acquired data at a maximum acquisition speed of 33 *kHz* (A/D frequency of 100 *kHz* divided by three) (Mondada, et al., 2009). They are also equipped with a speaker on top of them connected to an audio codec. Combined with the microphones, the speaker can create a communication network for peers’ location. Unfortunately, the lack of sensitivity regarding *e-pucks*’ microphones makes it hard to use them for sound source localization purposes. For instance, Figure 8.3b depicts the intensity values $F(x, y)$ with a maximum amplitude of one byte, obtained sweeping the whole scenario with a single *e-puck*. As one may see, the e-pucks are only able to distinguish sound from noise at a distance to the sound source (*i.e.*, victims *e-pucks*) of approximately 30 centimetres. However, such limitation favours the realistic applicability of the herein evaluated algorithms to *SaR* applications. For instance, to the similarity as the scenario used to evaluate the algorithms on simulations (*cf.*, Figure 3.9), if one would consider a large basement garage (*e.g.*, parking of a shopping mall), the laboratorial scenario from Figure 8.3 could easily be on a scale of 1:100. As a consequence, robot rescuers would be able to “hear” victims (receiver sensitivity) at a distance of 30 meters from them. Several sources would confirm that a human call for help may achieve a level between 72 and 78 *dB* at approximately 1 *m* away from the source, *i.e.*, from very loud voice to shouting voice (Truax, 1999). Moreover, as a rule of thumb, for every doubling of the distance from the source, the sound pressure level is reduced by 6 *dB*. According to Truax (Truax, 1999), one may expect average ambient sound levels between 40 and 55 *dB* in underground structures and medium density urban environment. This significantly reduces the ability to identify a call for help to distances between approximately 7 and 58 meters, depending on the

source and the ambient noise level, thus making the 30 meters sensitivity of robot rescuers a realistic constraint.

Two *e-pucks* were programmed to emulate victims, by playing the same sound. The “rescuers” *e-pucks* were programmed with the *RDPSO*, *AFS* and *GSO* algorithms with the main objective of collectively maximizing the input retrieved by the microphones. Contrarily to the previous experiments in which sub-optimality should be avoided to navigate towards the direction of maximum entropy at each iteration, the objective here is to find both victims. Hence, as both *RDPSO* and *GSO* have the particularity of avoiding sub-optimality, this feature was ignored by using a simple heuristic rule to stop when retrieving a sound amplitude of 100, *i.e.*, in the vicinities of the victims (hot colours from Figure 8.3b). This also intends to emulate the rescuing phase in which robots that found a victim should now either monitor or save it, thus being unavailable to search for other victims.

Due the complex nature of the problem considered in this section, the *Webots* simulator was used to first calibrate the real experiments. As such, the *swis2d* plugin found in *Webots* and developed by the *Swarm-Intelligent Systems Group* at the *École Polytechnique Fédérale de Lausanne*, was used to simulate the sound propagation. However, due to the limitations in *Webots* and *e-puck*'s cross-compatibility *APIs*, and as previously stated on section 3.2, several other features were developed so as to first validate the proposed approach under simulation experiments. Afterwards, all *e-pucks* were directly programmed using *Webots* cross-compilation tool without any sort of adjustment over the simulation experiments.

Since the 3 algorithms are stochastic, they may lead to a different trajectory convergence whenever they are executed. Therefore, test groups of 10 trials of 300 seconds each were considered for 14 *e-pucks*, *i.e.*, $N_T = 14$, placed in an initial configuration as presented in Figure 8.3a.

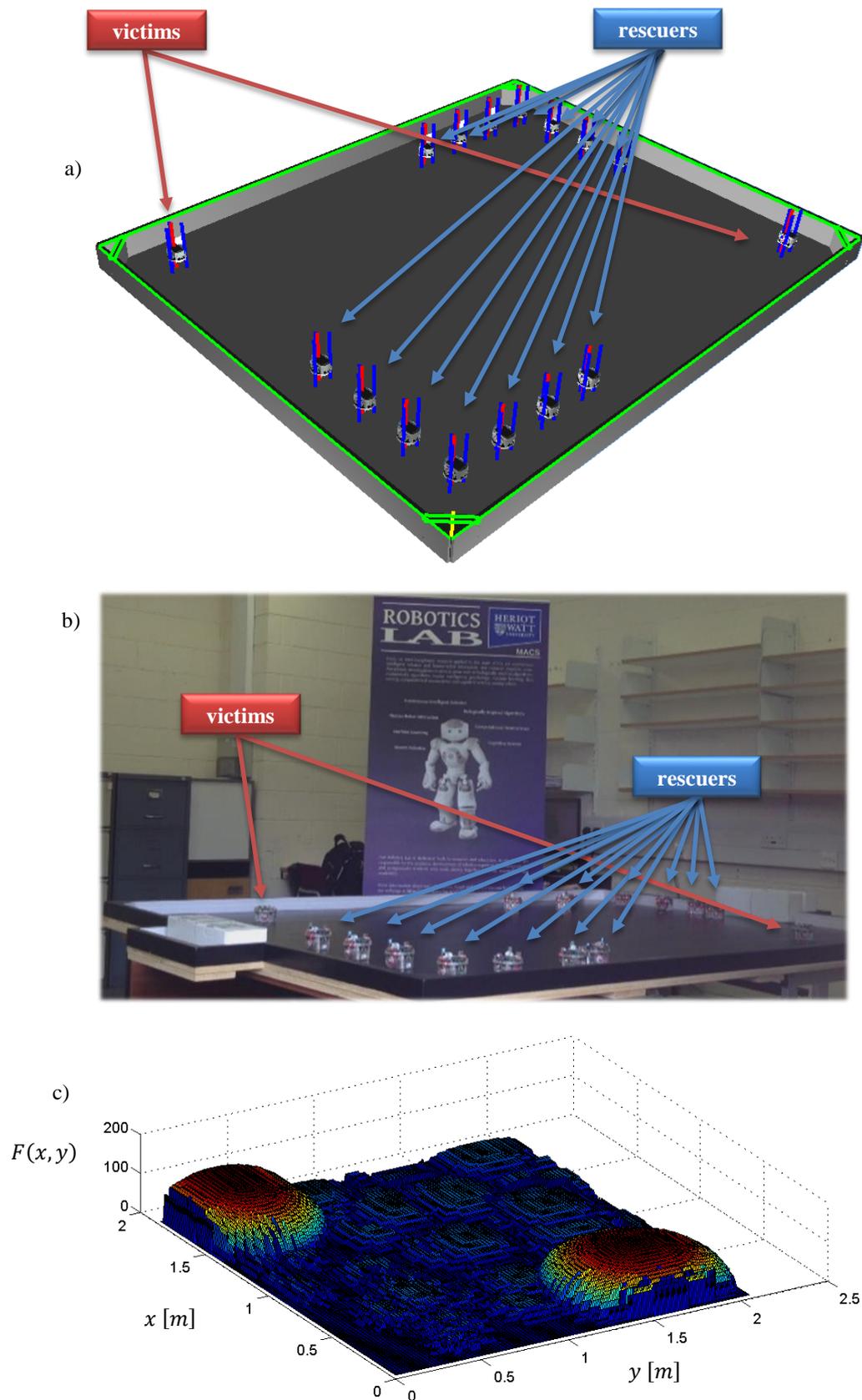


Figure 8.3. Experimental setup in an arena of 2.0×1.8 meters. a) Virtual representation in *Webots*; b) Real representation; b) Representation of the sound distribution.

In the case of the *RDPSO*, two subgroups were initially defined dividing the whole population into two equal parts of 7 *e-pucks* each. Note that due to *RDPSO* properties, both the number of subgroups and *e-puck* within each subgroup varied during the mission based on their individual and collective performance (see section 4.5). In the case of both *AFS* and *GSO*, all robots belonged to the same swarm. However, in the *GSO* the local-decision range varied according to the *luciferin* level, thus mimicking the same dynamical sub-division effect as the *RDPSO*.

All results from the 10 trials of each algorithm are summarized in Figure 8.4. The outcome from each algorithm is represented by a different colour and marker explained on the figure's caption. The axis correspond to the required time to save each victim. Markers located at the borders corresponding to the 300 seconds depict the unsaved victims. For instance, in any of the trials the rescuers failed at finding at least one victim, as there was no marker on position (300, 300) seconds. In other words, the performance of the algorithm increased as closer to the origin (0, 0) the markers were.

As one may observe, the 3 algorithms fail at finding the 2 victims at some point over the 10 trials of 300 seconds each in which they were evaluated in. The *RDPSO* was able to find only one victim in 2 trials, followed by the *GSO* in 4 trials and, lastly, the *AFS* in 7 trials. The outperformance of both *RDPSO* and *GSO* over the *AFS* regarding the partition of the population to multiple optimal solutions was expected due to their dynamic principles (*cf.*, Table 8.1). Despite not being able to always find the 2 victims, the *AFS* presents a faster convergence as rescuers were able to find the victim(s) in the first half of the mission time (≈ 150 seconds). Nevertheless, this early convergence may also be the reason why rescuers could not find the second victim since the *AFS* does not provide any partitioning or adaptive mechanism to balance the already existing exploitation level of agents with higher exploration capabilities.

The performance of the *RDPSO* is closely followed by the *GSO*. The average and standard deviation times necessary to find both victims for the *RDPSO*, *AFS* and *GSO* were 205 ± 64 , 259 ± 63 and 225 ± 70 seconds, respectively. Although both *RDPSO* and *GSO* can find all victims within a finite time due to their evolutionary mechanism to avoid stagnation, the *GSO* fails more often. As previously explained in section 8.1.3, the *GSO* benefits from a *luciferin* mechanism that, contrarily to all other algorithms, does not only depend on the sensed solution (*e.g.*, amplitude of the emitted sound by the victim). In fact, the *luciferin* value of a given robot decreases over time, thus avoiding its stagnation within a given region. We could make the analogy with nature by defining a limited quantity of oxygen in each discrete position the glowworm is in. In other words, to produce light, the glowworm requires oxygen (or water) for the *enzymatic oxidation* of the *luciferin* to occur. If the glowworm has a limited amount of oxygen in a certain position (represented by the sound amplitude in these experiments), then it needs to move to another position to maintain, or even increase, its

emitted light. This is a particularly interesting mechanism applied on swarm intelligence that ensures the convergence of robots to multiple solutions in an enclosed environment within a limited amount of time. However, this also plays the role of a “*double-edged sword*”. If the robot is unable to converge fast enough within the vicinities of a solution to maintain or increase its current *luciferin* level, then it may decide upon the wrong direction. This is likely to happen under noisy and nonlinear measures such as sound propagation with an increased complexity added by the lack of sensitivity of e-pucks’ microphones. This phenomenon was observed in some occasions during the experiments in which clusters of robots within the *GSO* got close enough to listen to the victim but still depicted a poor convergence, when converging at all. It is noteworthy that this could possibly be overcome by tuning parameters ρ and γ from Algorithm 4 though little insights are introduced in (Krishnanand & Ghose, 2009a; Krishnanand & Ghose, 2009b) regarding those.

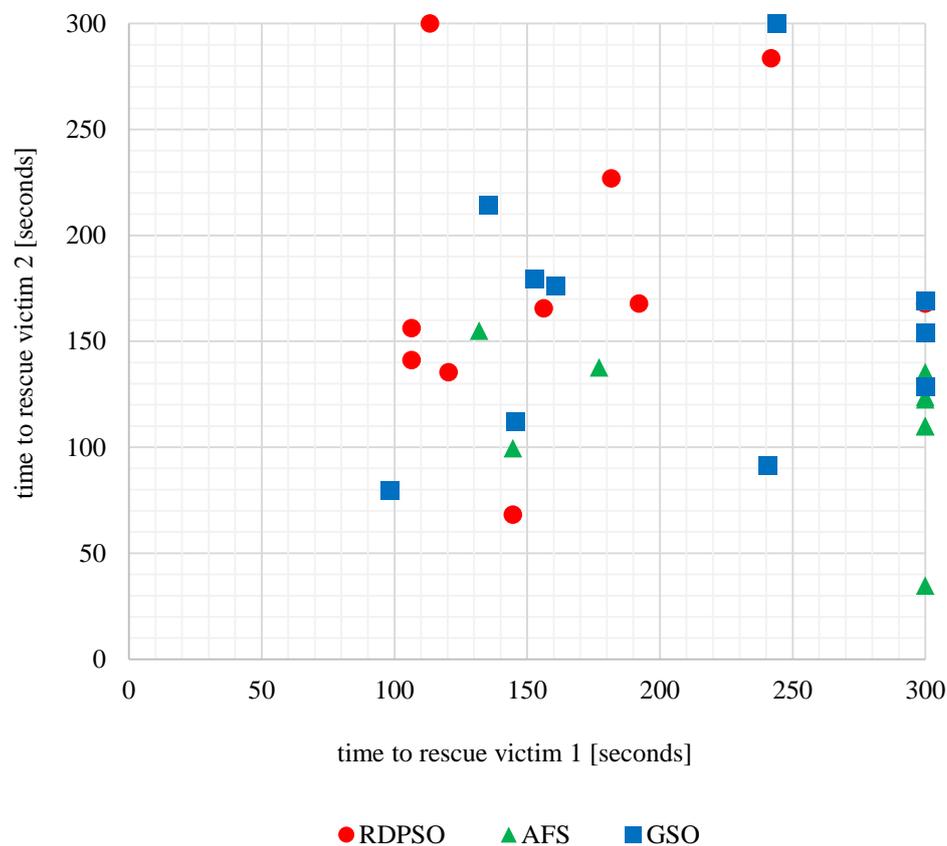


Figure 8.4. Representation of the rescue success of the *RDPSO*, *AFS* and *GSO* algorithms. Each marker corresponds to a different trial under a different algorithm. As closer the markers are from the origin (0, 0), the fastest robots were able to find the victims. Markers located on the border lines of the 300 seconds means that only 1 victim was found during that trial.

A video of the experiments is provided to better understand the typical behaviour of the 3 algorithms under these experiments³².

8.3 Discussion

The authors would like to discuss the take-home message this chapter brings forth. First, the primary motivation for this work was to find a group of swarm robotic algorithms with the potential of fulfilling realistic search tasks such as *SaR* operations. From that initial theoretical survey, five algorithms were chosen, namely: the *RDPSO* (proposed in this Thesis), the *EPSO* (Pugh & Martinoli, 2006; Pugh & Martinoli, 2007), the *PPSO* (Hereford & Siebold, 2008; Hereford & Siebold, 2010), the *GSO* (Krishnanand & Ghose, 2009a; Krishnanand & Ghose, 2009b) and the *AFS* (Gazi & Passino, 2003; Gazi & Passino, 2004). Table 8.1 was the leading step towards a detailed comparison of the five algorithms, thus describing the most relevant features one should expect under such tasks. From that table, it was possible to conclude that the *RDPSO* touch upon all desired features for a higher computational and memory cost. The main features outstanding the *RDPSO* from the alternatives are its ability to avoid sub-optimality by benefitting from a “punish-reward” mechanism based on natural selection (section 4.5) and the fault-tolerance assessment using a multi-connectivity strategy (section 5.2). Such outcome promotes the use of the *RDPSO* algorithm in applications affected by multiple sub-optimal and dynamic solutions in which the communication may be susceptible to failures. However, both computational power and the memory of the robotic platforms need to be well-weighted due to the requirements of the *RDPSO*.

Going deeper into the “*rabbit hole*”, a large number of simulation experiments was conducted to study the effect of the number of robots and the communication constraints of the five algorithms. The mission consisted of exploring and mapping a 2000 m^2 scenario in which robots needed to minimize the map’s entropy (Rocha, Ferreira, & Dias, 2008). More than to just state the obvious phenomenon that a larger population of robots improves the overall performance, those experiments were useful to understand the influence of a more constrained communication network on the five swarm algorithms. Through Figure 8.2 it was possible to observe a lower variability of the exploration ratio for a larger maximum communication distance feasible between robots, *i.e.*, the outcome became more consistent for a less constrained communication network. Such phenomenon was more percep-

³² http://www2.isr.uc.pt/~michaelcouceiro/media/RDPSO_AFS_GSO.mp4

tible using the *EPSO* and *PPSO* algorithms, thus suggesting their higher susceptibility over the communication constraints. Associating this aspect to the fact that both algorithms work on a broadcast communication basis (Table 8.1), the authors dissuade the use of those algorithms on applications that may require a larger number of robots (above 20 in the experiments in section 8.2.1) or too limited communication constraints (below an inter-robot distance of 100 meters in the experiments in section 8.2.1).

Those results paved the way to an insightful evaluation of the three best performing algorithms, namely, the *RDPSO*, the *GSO* and the *AFS*. This new evaluation was carried out using real platforms: the well-known *e-puck* robots equipped with *WiFi* technology for inter-robot communication (section 3.2.3). Instead of a mapping mission that would be typical of a *reconnaissance* phase, those experiments were consistent with the next phase of the firefighting operation: the *rescuing* (*cf.*, section 2.2.2). In brief, these experiments consisted of collectively finding 2 “victims” by benefiting from *e-pucks*’ speakers and microphones. To complement the previous experiments in which the size of the population and the communication constraints were studied, these experiments were conducted to evaluate the behavioural aspect, and even evolutionary features, of the algorithms. The results fostered even more the use of the *RDPSO* for such tasks with 80% success of finding both victims over the 300 seconds. Nevertheless, the *GSO* was able to closely follow the *RDPSO* due to its evolutionary *luciferin* mechanism for stagnation avoidance. Such a result proves to be crucial since the *GSO* presents itself as a “*low cost*” alternative to the *RDPSO* in terms of computational and memory requirements. Although, in general, the *RDPSO* presented better results than the *GSO*, it is noteworthy that the *GSO* would achieve a similar final outcome if one could benefit from a larger mission time.

All that being said, one may state that it is still difficult at this point to find a simple answer to the question:

“Which is the best swarm robotic algorithm for my application?”

However, we argue that this chapter provides a preliminary rationale on the most fitted swarm robotic algorithm for search applications. Such choice should consider some predefined assumptions, such as the number of available robots, the existing wireless communication and other mission-related features, *e.g.*, existence of dynamic sources, number of sub-optimal solutions to find, among others.

8.4 Summary

One of the main questions regarding swarm robotic algorithms is whether the full-scale deployment of these systems in real-world application environments would fit the necessary mission requirements. Despite the outstanding accomplishment of such algorithms in optimization or any other task unconstrained by real world features, such as robot dynamics, obstacles interference or communication failures, the reality gap still needs to be crossed for most of them. To address this issue, this chapter outlined an initial benchmark regarding the outcome from five swarm robotic algorithms under different configurations (*e.g.*, number of robots) and search tasks. Such results can be used to apply swarm robotic concepts to real world applications, such as *SaR*.

The list of swarm robotic algorithms herein compared is by no means exhaustive and a deeper research should be conducted based on the insights provided in this chapter. It is, however, possible to make a proper selection of the most desired algorithm based on the requirements of the application and hardware limitations (*e.g.*, wireless technology).

The experimental results essentially show the advantages of using evolutionary algorithms over non-evolutionary ones, starting with simulation experiments in which robots needed to cooperatively map an unknown environment, and all the way to real experiments in which a group of *e-pucks* needed to find the location of victims through sound. With a small increase of the computational complexity, the *RDPSO* algorithm depicted an improved convergence which was also better fitted to handling multiple and dynamic sources.

Given the advantages of the *RDPSO* algorithm, a deeper analysis should be conducted under hundreds or even thousands of robotic agents. Nevertheless, due to the computational complexity of such experiments in nowadays robotic simulators and available hardware, an alternative strategy is to analytically estimate the *RDPSO* outcome can be deliberated. Thereunto, a macroscopic model of the *RDPSO* is proposed in the next chapter in order to predict teams' performance for a given task. By doing this, one may be able to choose the most correct configuration (*e.g.*, number of robots within each team) without resorting to exhaustive and unsuitable experimentation.

Macroscopically Modelling

OVERCOMING the scalability problem within real-world swarm robotics presents a superior challenge than most other *MRS* domains. It is somehow obvious that, ideally, the collective performance of swarms increases with the number of agents in the society. In fact, studies that include only a small number of robots (inferior to 10) do not aim for scalability, thus falling outside swarm robotics context (Sahin E. , 2005). Nevertheless, in real situations, this assumption does not generally hold. As the number of robots within the same workspace increases, the interference between them also increases (Agassounon, Martinoli, & Easton, 2004) (Lerman & Galstyan, 2002) (Bjerknes & Winfield, 2010). Moreover, in swarm models requiring robots to explicitly share information with their teammates, the communication constraints (*e.g.*, signal quality) bring the interference concept to a whole new level of complexity (Couceiro, Rocha, & Ferreira, 2013b). Therefore, it is important answering the following crucial question that many works are still trying to study within *MRS* context:

“What is the ideal number of robots for a given task?”

Or else, more unambiguously:

“What is the output one should expect for a given number of robots under a given task?”

Many works on swarm robotics have been trying to answer such questions. However, most of them tried it empirically by means of experimentation. Despite being a straightforward strategy, this requires extensive experimental validation due to algorithms stochasticity. Thus, several alternative works came closer to propose analytical models that could explain, at some extent, the stochasticity inherent to swarm intelligence without resorting to trial-and-error strategies.

The recent work of Chen and Chen (Chen & Chen, 2011), despite not being in the context of swarm robotics, presented a statistical model to obtain theoretical results on the convergence of the well-known *PSO* (Kennedy & Eberhart, 1995). Yet, the authors considered only the facet of particles' interactions (*i.e.*, the social component of the swarm). Even though it is a good approach to predict the collective phenomena of swarms, adapting this methodology to swarm robotics would require a

completely new formulation so as to consider the many real world aspects (*e.g.*, obstacles, communication).

Alternatively, Lerman and Galstyan presented and analysed a mathematical model of foraging in a group of robots (Lerman & Galstyan, 2002). The class of models introduced by the authors was able to describe swarm intelligence in which each agent's future state only depends on its current state and the time the agent has spent in it. Such models are denoted as *Markov chains* or *semi-Markov chains* (see section 3.1.4).

Those probabilistic representations, as the most common probabilistic finite state machines (*PFSM*), have been showing their potential toward representing the macroscopic model of swarm robots. For instance, one of the most self-consistent works in trying to estimate swarm's outcome using Markov chains was presented by (Agassounon, Martinoli, & Easton, 2004). Despite the discrepancies in some predictions, their model was quite accurate and able to closely estimate the foraging task outcome. Nevertheless, some unrealistic assumptions were still considered by the authors. For instance, the authors considered obstacle-free squared scenarios that utterly refute the estimation process in real applications. Moreover, the macroscopic model was only evaluated for simple foraging tasks to collect and gather objects in a single cluster, without any evolutionary properties that might drastically change robots' behaviour during the course of the mission.

Considering these limitations in the state-of-the-art, this chapter key contributions are divided as it follows (Couceiro M. S., Martins, Rocha, & Ferreira, 2013b (Under Review)):

- i*) Formal statement of the problem that will be studied in this chapter (section 9.1);
- ii*) The necessary features and assumptions are introduced so as to establish some rules and input requirements that one should consider before estimating the most adequate robotics' team configurations based on the proposed model (section 9.2);
- iii*) Subsequently, the evolutionary rates defining the transition probabilities between states of the semi-Markov chain are described step-by-step (section 9.3);
- iv*) To start with the macroscopic modelling, two simplified semi-Markov chains are designed to embody the behaviour of robots within socially active subgroups and socially excluded subgroups, respectively (section 9.4);
- v*) Finally, an evolutionary *RDPSO* macroscopic semi-Markov model is proposed as the analytical methodology to estimate the performance of a family of evolutionary robotic swarms performing exploration tasks, dealing with physical constraints posed by real scenarios, including robots' dynamics, obstacles and communication constraints (section 9.5);

- vi) Besides the simulation experiments used to justify each choice made throughout this chapter, the macroscopic *RDPSO* model is closely compared to its microscopic counterpart proposed in this Thesis on a source localization problem (section 9.6).

Sections 9.7 and 9.8 outline the discussion and main conclusions of this chapter.

9.1 Problem Formulation

Consider a swarm of N_T robots that has just been deployed with the purpose of exploring an unknown scenario. It is assumed that this team has no central controller, being based on emergent cooperative behaviors arising from simple local interactions between individual robots. The problem addressed in this chapter is to accurately estimate the steady-state regime of the robot swarm through a macroscopic analytical model, *i.e.*, to determine the average number of robots within each possible state (*e.g.*, number of robots avoiding obstacles) and the collective outcome after a certain number of iterations t . The macroscopic model should consider the features of both the scenario (*e.g.*, dimension, density of obstacles) and robots (*e.g.*, obstacle sensing range, maximum communication range). Moreover, the key evolutionary features of the *RDPSO* algorithm presented in section 4.5 also need to be considered.

9.2 Features and Assumptions

This section fully explores the necessary requirements for the adequate implementation of the semi-Markov model with a step-by-step analysis of each variable. All features and assumptions, as well as the macroscopic model itself, will be primarily evaluated by means of a set of simulations.

9.2.1 Simulation setup

The *Multi-Robot Simulator (MRSim)* was used to evaluate the macroscopic model, thus continuously comparing it to the microscopic *RPDSO* (section 3.2.4). The same scenario presented on Figure 3.9 from section 3.2.4, the *ISR-UC* garage, was considered as an example throughout this chapter. This was the test case scenario used to exploit the model properties as it is a large area of $A = 2975 \text{ m}^2$ with a considerable density of obstacles (*e.g.*, pillars).

Next section introduces some important parameters regarding the world and the robotic agents.

9.2.2 Robot's features

Although all computational and experimental results were carried out using differential wheeled robots so far, *e.g.*, *eSwarBot* platforms presented in section 3.2.1, the same modelling could be applied to any other driving mechanism. It is however worth mentioning that the *RDPSO* was formalized based on planar motion, *i.e.*, dimensionality \mathbb{R}^{ϖ} with $\varpi = 2$. This is a first assumption that will be considered but that still does not limit the macroscopic model over the microscopic one previously proposed on chapter 4.

Robots are equipped with a distance sensor having a maximum range $R_w > 0$ [m] and an angular field-of-view $\varphi_w \in [0, 2\pi[$. The sensor cannot penetrate obstacles and, therefore, the presence of surrounding obstacles decreases robots' search capabilities. The 2π upper limit was assumed to allow for an easier computation of the average open space among the obstacle distribution.

Robots are also endowed with wireless communication capabilities that enable multi-hop messaging within a maximum one-hop range of d_{max} from which the communication interference range $R_r > 0$ [m] is related to. As the *RDPSO* algorithm ensures the *MANET* connectivity, this adds up an interference effect between robots from the same subgroup. As stated in section 2.1, a higher number of robots is likely to cause a higher interference. For the sake of simplicity, one can define the communication interference range as the desired communication distance $R_r = d_{max}$.

As a probabilistic macroscopic model, we assume a spatial uniformity. That is to say, after some time, robots uniformly cover the scenario in such a way that a specific robot trajectory is not considered. In a steady-state regime, this returns a similar average distance covered by each robot. Such assumption holds as long as robots maintain a similar average velocity $\bar{v}_n > 0$ [$m \cdot s^{-1}$]. Taking into account the *RDPSO* properties and the homogeneity between agents, this can barely be considered a hard assumption. Considering the *DE* system formed by equations (4.5), (4.6) and (4.8), the velocity of robot n will vary depending on several stochastic factors weighted with coefficients ρ_i and r_i , wherein r_i is uniformly distributed between 0 and 1, in such a way that:

$$E(r_i) = \frac{1}{2}, i = 1, \dots, 4, \quad (9.1)$$

represents the expected value of r_i . Replacing r_i by its expected value would yield a deterministic version of the *RDPSO*, rewriting the equation of the velocity in (4.5) as:

$$\bar{v}_n[t + 1] = \bar{w}_n[t] + \frac{1}{2} \sum_{i=1}^4 \rho_i (\chi_i[t] - x_n[t]), \quad (9.2)$$

wherein $\bar{w}_n[t]$ is a function of the previous velocities, *i.e.*, $\bar{v}_n[t], \dots, \bar{v}_n[t + 1 - r]$ (see the *DE* in (4.8)). In other words, in a steady-state regime wherein $\chi_i[t] = x_n[t]$, $i = 1, 2, 3, 4$, robots will depict an average velocity of $|\bar{v}_n[t + 1]| = \frac{1}{2}v_{max}$.

Table 9.1 summarizes the physical parameters of robots that will be considered throughout this section. Those will merely be used as an example to allow for a straightforward analysis of the proposed model.

Table 9.1. Inputs of the robot model.

$\bar{v} [m \cdot s^{-1}]$	$R_w [m]$	$R_r [m]$
1	3	15

Similarly as presented in this section, the following section discusses the physical features of the world.

9.2.3 World features

As this work aims at realistic swarm robotic applications such as *SaR*, the knowledge regarding a scenario cannot be assured. For instance, let us suppose a fire outbreak within a large basement garage. Such urban fire requires a prompt response because of life hazard in highly populated zone and the high risk of fire propagation to buildings and parked cars in the vicinity. Under such situation, firefighters should coordinate properly in order to manage and respond quickly to mitigate the disaster. In fact, the probability of successfully rescuing victims greatly decreases over a short period of time in the first minutes of the operation. Therefore, firefighters' action plan cannot be prearranged and grounded upon the building blueprints, especially because most of the time such blueprints are unavailable or require an exhaustive bureaucratic procedure. The same observation could be conducted for almost any other real situation under hostile environments, *SaR*, or disaster recovery.

Nevertheless, there are, at least, two crucial variables that one should know to efficiently predict robotic teams' performance under such scenarios: *i)* the area of the scenario; and *ii)* the density of obstacles. As both variables may need to be estimated (*e.g.*, by human observations), one needs to ensure that the macroscopic model is not highly susceptible to small discrepancies in these inputs.

Therefore, our scenario will be characterized in Table 9.2. Once again, those values are only used as an example.

Table 9.2. Inputs of the world model.

$A [m^2]$	ρ_w
2975	0.0493

being ρ_w the estimated density of obstacles in the scenario, this yields a useful area in the scenario of $A_a = A(1 - \rho_w) \approx 2828 m^2$ to be explored. Besides the robot and world models, only a small consideration presented in the next section needs to be respected regarding the *RDPSO* algorithm.

9.2.4 Algorithm's features

The performance of the *RDPSO* algorithm, as any other parameterized stochastic algorithm, greatly depends on the choice of its parameters, namely, coefficients ρ_i , $i = 1,2,3,4$, and the inertial fractional coefficient α . In chapter 7, the effect of those parameters were studied by carrying out a convergence analysis, thus retrieving an attraction domain in which parameters may be defined in such a way that robots can find the optimal solution while avoiding obstacles and ensuring *MANET* connectivity (see section 7.1). Subsequently, sections 7.2 and 7.3 presented a fuzzy method to systematically adapt those parameters to the contextual information.

Hence, all the results presented in this chapter should follow the insights previously presented. In other words, if the *RDPSO* is parameterized in an arbitrary way or without considering the aforementioned adaptive mechanism (*cf.*, chapter 7), the macroscopic model herein presented will fail to estimate the adequate outcome. Therefore, the *RDPSO* parameters should vary according to the fuzzy adaptive scheme from chapter 5 and the attraction domain depicted in Figure 7.1. The relationship between N_{min} and N_{max} with $N_s[0]$ can be approximated based on the previous *RDPSO* parameters tables (*e.g.*, Table 7.2).

Besides the *RDPSO* parameters from the *DE* (4.5), the following parameters were also considered based on the results retrieved from previous works:

Table 9.3. Inputs of the *RDPSO* algorithm.

$SC_{max} [s]$	N_{min}	N_{max}
30	$\left\lfloor \frac{1}{2} N_s[0] \right\rfloor$	$2N_s[0]$

wherein $N_s[0]$ is the initial number of robots in subgroup s .

Given the description of the most relevant inputs required by our macroscopic model, the following section describes the evolutionary rates based on such inputs.

9.3 Evolutionary Rates

It is expected that the transition probabilities between states will depend on the previously presented parameters. In other words, a robot's current state will depend on several aspects, such as its sensing and interaction capabilities or the scenario's dimensions (Ijspeert, Martinoli, Billard, & Gambardella, 2001; Agassounon, Martinoli, & Easton, 2004). Considering the computation of transition probabilities based on encountering rates, as described by equation (3.6), let us now define the following evolutionary rates:

- Robots' communication interference rate γ_r ;
- Obstacles' encountering rate γ_w ;
- Mission-related exploration rate γ_e ;
- Social exclusion rate γ_{exc} ;
- Social inclusion rate γ_{inc} .

This section will thoroughly describe each of those evolutionary rates.

Although Table 9.1 to Table 9.3 described the necessary inputs one would need to implement in the system, these inputs inevitably influence the necessary time a robot needs to spend on each different state, *i.e.*, the necessary time to circumvent an obstacle, to maintain network connectivity with its teammates, to explore the scenario within its sensing range, and to be socially excluded or included from the active groups. This usually induces a delay state (*i.e.*, dwell time) that simply represent a particular behaviour that the robot performs for a certain duration. Table 9.4 summarizes the values of delays used in the proposed macroscopic model, namely, the delay spent: on the *Search (Wandering)* state T_e , on the *Obstacle Avoidance* state T_w , on the *Communication Interference* state T_r , as socially excluded T_{exc} and as socially active T_{inc} . As previously stated in section 3.1.4, a discretization interval of $\Delta t = 1$ second was chosen, thus resulting in integer delays. The most common values (*i.e.*, mode values) were obtained running 90 simulation experiments equally distributed between 5, 10 and 15 socially active robots and 90 simulation experiments equally distributed between 5, 10 and 15 socially excluded robots. For these experiments, the evolutionary properties inherent to the *RDPSO* algorithm were removed so as to independently analyse each different model separately.

Table 9.4. Average delay of robots within each state.

<i>Status</i>	T_r [s]	T_w [s]	T_e [s]	T_{exc} [s]	T_{inc} [s]
<i>Active</i>	1	2	5	8	-
<i>Excluded</i>	2	4	15	-	19

It is important to note that those values did not suffer changes for significant variations of $\pm 10\%$ for each input presented in Table 9.1 and Table 9.2. This justifies the applicability of the delays presented in Table 9.4 for similar configurations.

It is also important to note that those delays change as the mission advances, in particular the exploration T_e , exclusion T_{exc} and inclusion rates T_{inc} . This is only natural since the more robots advance in the exploration of the scenario the more difficult it is for them to find unexplored regions, thus increasing the exploration time T_e . As a consequence, this also results in a significant increase in the social exclusion time T_{exc} and decrease in the social inclusion time T_{inc} . However, in this work, the delay within each state was considered to be constant. This is a valid assumption as most evolutionary rates are time-variant, thus resulting in time-variant transition probabilities and, as a result, in a nonlinear time-delayed DE system for the $RDPSO$ full aggregation.

9.3.1 Robots' communication interference rate

Robots' communication interference with respect to communication constraints is perhaps the most straightforward one to be defined. It is easy to come to an agreement that an area of the scenario is relevant since it directly influences robotic teams' coverage magnitude *i.e.*, the time needed to cover a scenario usually grows with its size. Nevertheless, this is far from being a linear relationship. As argued by Agassounon *et al.* (Agassounon, Martinoli, & Easton, 2004), a larger scenario may also decrease the interference between robots since they can move more freely.

Bearing this idea in mind, and generalizing the definition presented by Agassounon *et al.* (Agassounon, Martinoli, & Easton, 2004), one may outline the rate a robot may interfere with its teammates in a subgroup of N_s robots. Since the number of robots to form a subgroup within the evolutionary properties of the $RDPSO$ algorithm might vary over time (*cf.*, section 4.5), let us generalize the rate a robot may interfere with its teammates as:

$$\gamma_r[t] = 2\bar{v}(N_s[t] - 1) \frac{R_r}{A_a}, \quad (9.3)$$

wherein R_r is the communication range and $A_a \leq A$ represents the useful area of the scenario, *i.e.*, the total area of the scenario subtracted by the area occupied by obstacles (*cf.*, section 9.3.2 later). In other words, a robot will be constrained by any of its $N_s[t] - 1$ teammates at time t at a rate of $\gamma_r[t]$. As one may observe from equation (9.3), the interference rate γ_r increases with the number of robots within the same subgroup N_s , the average velocity of robots \bar{v} and the desired communication range R_r , while it decreases with the useful area of the scenario A_a .

Note that this rate introduces a nonlinear factor to the system as it varies over time with the number of robots in the same subgroup. However, this only takes effect for the *RDPSO* full aggregation due to its social exclusion and inclusion properties.

9.3.2 Obstacles' encountering rate

Despite the outstanding accomplishment, the authors in Agassounon *et al.* (Agassounon, Martinoli, & Easton, 2004) completely ignored the effect of the obstacles density since they only considered obstacle-free scenarios. As this assumption is not passive to be taken under realistic scenarios, the density of occupied space needs to be estimated. Hence, and as one may not assume to have a deeper knowledge regarding the scenario characteristics, let us present a general modelling based on the work of Rañó and Minguez (Rañó & Minguez, 2006). In other words, one needs to engender a generic representation based on a simple characterization of the real scenario. This representation allows retrieving some necessary information. The main idea consists of describing the scenario as a unit circle wherein obstacles are smaller circles within the unit circle. This results in an area of the normalized scenario given by $\hat{A} = \pi$.

At this point, two assumptions need to be considered:

- i) Obstacles will have the same normalized area \hat{A}_w , being one thousand times smaller than the area of the scenario, *i.e.*, $\hat{A}_w = \frac{\pi}{1000}$;
- ii) Obstacles will be uniformly distributed throughout the scenario.

Assumption (i) was considered based on an average value retrieved from the set of empirical results from Rañó and Minguez (Rañó & Minguez, 2006). Such statement means that an existence of 1000 obstacles of $\hat{A}_w = \frac{\pi}{1000}$ area would fill the whole scenario, thus disabling robots' navigation. Assumption (ii), on the other hand, was considered based on the principle that the distribution of obstacles is usually random. As this assumption may not always hold, this scenario generalization will foment more toward scenarios that fall within such description. As this work focuses on indoor scenarios representing large basement garages (*e.g.*, Figure 3.9), this assumption does not fall apart in reality as obstacles (*e.g.*, pillars) are usually evenly distributed.

Under such assumptions, one may define the density of obstacles as:

$$\rho_w = n_w \frac{\hat{A}_w}{\hat{A}}, \rho_w \in [0, 1], \quad (9.4)$$

wherein n_w represents the estimated number of obstacles, $n_w \in [0,1000]$. Note that a $\rho_w = 0$ represents a free-obstacle scenario, while a $\rho_w = 1$ represents a scenario completely occupied without any free space for the robot motion.

The density ρ_w measures the amount of space occupied by obstacles. This is an intrinsic and global property of the environment that one should try to obtain (*e.g.*, by human observations). By obtaining a good estimation on the density of obstacles ρ_w , and considering assumption (i), one may rewrite equation (9.4) as:

$$n_w = \lfloor 1000\rho_w \rfloor. \quad (9.5)$$

For instance, to our specific situation, the scenario on Figure 3.9 corresponds to a density of obstacles of approximately $\rho_w = 0.0493$, *i.e.*, 4.93% of the scenario is occupied by obstacles (Table 9.2). This leads to $n_w = 49$ obstacles. One may observe on Figure 3.9 that the real scenario presents a number of 56 obstacles (57 considering the north wall separating the garage from a small store room). Nevertheless, this is still a good approximation as one may also observe that some obstacles are not evenly deployed.

Afterwards, the number of obstacles n_w may make it possible to retrieve the “clearness” of the scenario. The clearness is related to the open space among the obstacle distribution. Note that this descriptor depends on the obstacle sensing radius R_w as robots with a larger obstacle detection range will be more susceptible to obstacles. One may resort to dispersion metrics to measure the open space within the scenario (Niederreiter, 1992). Hence, the following four steps methodology (see example on Figure 9.1) will be followed:

1. Create a grid of points representing the location of the n_w obstacles uniformly distributed within a unit square;
2. Map the obstacles location to a unit circle;
3. Dilate the obstacles considering robot’s normalized obstacle detection range \hat{R}_w ;
4. Calculate the average open space within the normalized circle \hat{S}_w and convert it to real coordinates, S_w .

The first step may be easily accomplished by creating a cartesian grid dividing the unit space between -1 and 1 into 1000 cells. This value was chosen based on the assumption regarding the size of each obstacle. Afterwards, a resample of such division is carried out based on the round of the square root of n_w , *i.e.*, $\lfloor \sqrt{n_w} \rfloor$, so as to choose the cells that will be occupied. Let us identify each of those occupied cells as C_i , $i = 1, \dots, n_w$.

The second step may be easily fulfilled by following Proposition 9.1.

Proposition 9.1. *Let x_{1i} and x_{2i} be the cartesian location of obstacle i within the unit square retrieved from step 1, i.e., the position of each C_i :*

$$\begin{aligned} -1 &\leq x_{1i} \leq 1, \\ -1 &\leq x_{2i} \leq 1. \end{aligned} \tag{9.6}$$

Then obstacles location may be mapped to a unit circle following the coordinate transformation:

$$\begin{cases} \widehat{x}_{1i} = x_{1i} \sqrt{1 - \frac{x_{2i}^2}{2}} \\ \widehat{x}_{2i} = x_{2i} \sqrt{1 - \frac{x_{1i}^2}{2}} \end{cases}. \tag{9.7}$$

The following proof is built upon (Nowell, 2005).

Proof: Let

$$\frac{\widehat{x}_{1i}^2}{a^2} + \frac{\widehat{x}_{2i}^2}{b^2} = 1, \tag{9.8}$$

be the general equation of an ellipse in which a, b are the radii on the \widehat{x}_{1i} and \widehat{x}_{2i} axes, respectively.

For a constant x_{1i} , $-1 \leq x_{1i} \leq 1$, one can rewrite equation (9.8) as:

$$\frac{\widehat{x}_{1i}^2}{x_{1i}^2} + \frac{\widehat{x}_{2i}^2}{b^2} = 1, \tag{9.9}$$

To map the unit square to unit circle (i.e., with radius of 1) one can resort to the concepts of polar coordinates in which the following relation can be obtained:

$$\begin{cases} \widehat{x}_{1i} = \cos \varphi \\ \widehat{x}_{2i} = \sin \varphi \end{cases} \Leftrightarrow \begin{cases} \widehat{x}_{1i} = \cos \varphi \\ \widehat{x}_{2i} = \sqrt{1 - \cos^2 \varphi} \end{cases}, \tag{9.10}$$

wherein $\cos \varphi$ can be given following the basic trigonometric principles as the ratio of the length of the adjacent side (*i.e.*, constant x_{1i}) to the length of the hypotenuse (*i.e.*, half the unit square's diagonal) as:

$$\cos \varphi = \frac{x_{1i}}{\sqrt{2}}, \quad (9.11)$$

thus resulting in:

$$\begin{cases} \widehat{x}_{1i} = \frac{x_{1i}}{\sqrt{2}} \\ \widehat{x}_{2i} = \sqrt{1 - \frac{x_{1i}^2}{2}} \end{cases}. \quad (9.12)$$

By replacing equation (9.12) in (9.9), one can obtain b as:

$$b = \sqrt{2 - x_{1i}^2}. \quad (9.13)$$

Carrying out a similar procedure to a constant x_{2i} , $-1 \leq x_{2i} \leq 1$, one can write the following system of equations:

$$\begin{cases} \frac{\widehat{x}_{1i}^2}{x_{1i}^2} + \frac{\widehat{x}_{2i}^2}{2-x_{1i}^2} = 1 \\ \frac{\widehat{x}_{1i}^2}{2-x_{2i}^2} + \frac{\widehat{x}_{2i}^2}{x_{2i}^2} = 1 \end{cases}. \quad (9.14)$$

Solving the system of equations (9.14) in order to the new obstacles coordinates \widehat{x}_{1i} and \widehat{x}_{2i} , we obtain the system of equations in (9.7). ■

The third step is crucial as it considers the robot's obstacle detection range. The radius of each obstacle will then be enlarged by the radius of the robot's sensing capabilities R_w . As we are working on the normalized unit space, one needs to first normalize R_w . This can be calculated considering one of the real dimensions known – the area of the scenario. As we are approximating the scenario to a square in the first place, one may calculate its side length as $L = \sqrt{A}$. Considering the area of $A = 2975 \text{ m}^2$, this would yield a side of $L = 54.54 \text{ m}$. As one needs to map the square into a circle, the

side length corresponds to the circle diameter, in which one directly obtain its radius $R = 27.27 \text{ m}$. As the normalized radius corresponds to $\hat{R} = 1$, one can directly calculate the robot's normalized obstacle detection range as:

$$\hat{R}_w = \frac{R_w}{R}, \quad (9.15)$$

which results in $\hat{R}_w = 0.11$ for our specific situation.

The third step may now be concluded by using Minkowski addition (also known as dilation) on each obstacle as $\tilde{C}_i = C_i \oplus C_{\hat{R}_w/2}$, where \oplus is the Minkowski sum and $C_{\hat{R}_w/2}$ is the sphere with radius $\frac{\hat{R}_w}{2}$ (Meyer & Minkowski, 1969). The average clearness, *i.e.*, the mean distance travelled before encountering an obstacle, may now be calculated as:

$$\hat{S}_w = \frac{1}{n_p} \sum_{j \in n_p} \min_{i \in n_w} \|P_j - \tilde{C}_i\|, \quad \hat{S}_w \in [0, 1], \quad (9.16)$$

where $\|\cdot\|$ is the Euclidean distance from cell P_j to the sphere \tilde{C}_i and n_p is the number of free cells. When there are no obstacles the value of the dispersion \hat{S}_w is one. However, as the number of obstacles increases the dispersion \hat{S}_w drops to zero. This characteristic captures the notion of clearness (open space) since it represents the average allowable distance for the robot to manoeuvre.

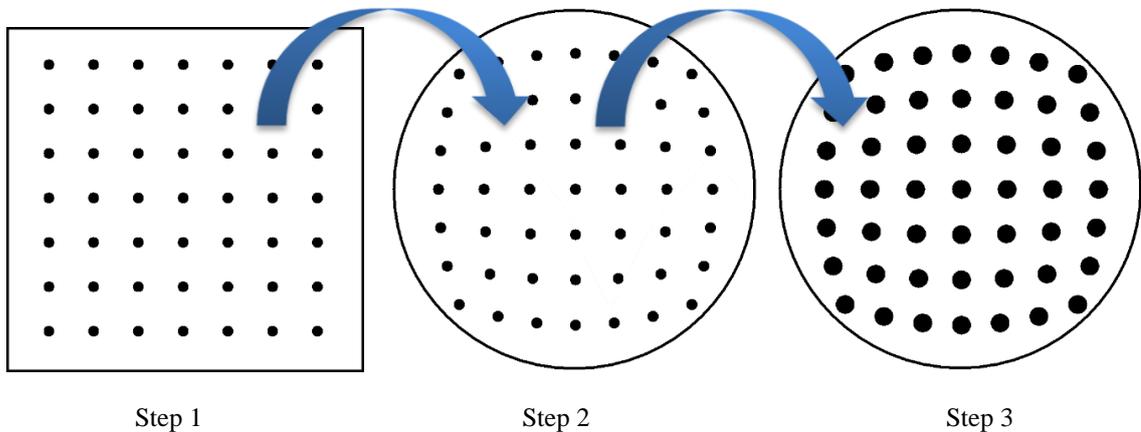


Figure 9.1. Illustration of the normalization process for the scenario depicted in Figure 3.9. Step 1) Create a grid of points identifying obstacles' location uniformly distributed within the unit square based on the density of obstacles; Step 2) Map the obstacles location to a unit circle; and Step 3) Dilate the obstacles considering robot's obstacle detection range \hat{R}_w .

Finally, the forth step consists of a simple conversion following the same reasoning as in equation (9.15), *i.e.*, $S_w = \hat{S}_w R$, thus resulting in an average distance travelled before encountering an obstacle of $S_w = 22.59 \text{ m}$ to our case study.

This whole process is depicted in Figure 9.1. As a consequence of the above presented method to obtain the average distance travelled before encountering an obstacle, one may mathematically define the encountering rate with obstacles as:

$$\gamma_w = \bar{v}_n \frac{1}{S_w}. \quad (9.17)$$

Note that, contrarily to the interference rate γ_r , the obstacle detection rate γ_w does not depend on the number of robots in the subgroup and, as a result, is constant. Although this may seem like a hard assumption, one should consider that robots within the same subgroup under the *RDPSO* algorithm never interfere as obstacles with each other. This is achieved by ensuring a certain “ideal” distance d_{max} or signal quality q_{min} between robots so as to maintain *MANET* connectivity (*cf.*, section 4.4 and 5.2). Those “ideal” measures give rise to an interference radius $R_r > 0$ in which $R_r > R_w$. Moreover, although robots from different subgroups of the swarm could in fact interfere with each other, one may neglect such effect for large scenarios which is in accordance to most real situations (*e.g.*, catastrophic incidents in large areas).

9.3.3 Mission-related detection rate

As the previously defined rates, the mission-related one will depend on the inputs that define the mission. In other words, a different mission objective may greatly affect the whole definition of the mission-related detection rate. To avoid a too generalized solution that would fit every possible mission with some hard constraints, this chapter, as the Thesis itself, mainly focuses on the exploration of an unknown scenario (*e.g.*, mapping, source localization, among others). Note that this is a realistic application in which robots can be of an invaluable help if they are equipped with sensors that enable their navigation where humans are highly hampered (*e.g.*, lack of visibility in indoor fire outbreaks). In this context, robots may provide a systematically updated map of the environment by means of multi-robot simultaneous localization and mapping (*SLAM*) strategies (Leung, Barfoot, & Liu, 2012). Those strategies are outside the scope of this Thesis and, therefore, it is assumed that robots can perform *SLAM* under such scenarios.

It is noteworthy that the average velocity, under such application, will directly influence the area a robot can sweep over time (Lerman & Galstyan, 2002). If a robot travels through the area at an

average speed \bar{v}_n , it sweeps out a detection region that depends on the sensing radius R_w during the time interval Δt , with $\Delta t = 1$. Hence, the typical exploration rate (*e.g.*, seed encounter) presented in works such as Lerman and Galstyan (Lerman & Galstyan, 2002) or Agassounon *et al.* (Agassounon, Martinoli, & Easton, 2004) has been described as:

$$\gamma_e = 2\bar{v}_n \frac{R_w}{A_a}. \quad (9.18)$$

However, the exploration rate cannot be considered as constant. As a rule of thumb, the more robots explore a given scenario, the more difficult it is to further explore it. This is only natural as the rate between the explored area and the total area increases over time, thus making it hard, in a non-linear fashion, for robots to find unexplored regions. Moreover, both socially excluded and active robots in the *Wandering [Exploration]* $N_{exc}[t]$ and *Search [Exploration]* $N_{inc}[t]$ states, respectively, contribute to the explored area. For more details about those states please refer to section 9.5.

By considering these properties, one can generalize equation (9.18) to a more realistic model through the following proposition.

Proposition 9.2. *Let the exploration rate at time t of all $N_{inc}[t] + N_{exc}[t]$ exploring robots be defined as:*

$$\gamma_e[t + 1] = (1 - A_e[t])(N_{inc}[t + 1] + N_{exc}[t + 1]) \times 2\bar{v}_n \frac{R_w}{A_a}, \quad (9.19)$$

such that

$$A_e[t] = \sum_{k=0}^t \gamma_e[k], \quad (9.20)$$

wherein $A_e[t]$ is the proportion of the explored area at time t which corresponds to the cumulative sum of the exploration rate up to time t .

Proof: In accordance to Lerman and Galstyan (Lerman & Galstyan, 2002) and Agassounon *et al.* (Agassounon, Martinoli, & Easton, 2004), one robot is able to sense a proportion of $2\bar{v}_n \frac{R_w}{A_a}$ from the scenario, wherein A_a is the area to be explored following equation (9.18). Also, immediately after the

mission begins ($t = 1$), the probability that robots have to find unexplored regions is 1. Hence, considering $N_{inc}[t] + N_{exc}[t]$ exploring robots, the exploring rate, *i.e.*, proportion of the scenario explored by the robots, immediately after the mission begins, is:

$$\gamma_e[1] = (N_{inc}[1] + N_{exc}[1])2\bar{v}_n \frac{R_w}{A_a}, \quad (9.21)$$

thus resulting in an unexplored proportion of the scenario, *i.e.*, probability to find an unexplored region, for $t = 2$ given by $1 - \gamma_e[1]$. By considering this, one can calculate the exploring rate at time $t = 2$ as:

$$\gamma_e[2] = (1 - \gamma_e[1])(N_{inc}[2] + N_{exc}[2])2\bar{v}_n \frac{R_w}{A_a}. \quad (9.22)$$

Similarly as before, this results in an unexplored proportion of the scenario, *i.e.*, probability to find an unexplored region, for $t = 3$ given by $1 - (\gamma_e[1] + \gamma_e[2])$. Simplifying by means of equation (9.20), the unexplored proportion of the scenario for time $t = 3$ can be given by $1 - \sum_{k=0}^2 \gamma_e[k] = 1 - A_e[2]$. Generalizing, the probability to find an unexplored region at time $t + 1$ is described as $1 - A_e[t]$, with $A_e[0] = 0$ and $\lim_{t \rightarrow \infty} A_e[t] = 1$. Following the same reasoning for $t = 1$ (9.21) and $t = 2$ (9.22), the generalization in equation (9.19) is achieved. ■

Note that the collective exploration rate immediately after the mission begins follows equation (9.18) for all exploring agents, as equation (9.21), and tends to zero as they explore the scenario as $\lim_{t \rightarrow \infty} \gamma_e[t] = 0$. As one may observe from equation (9.19), the more the scenario is explored, the more difficult it is to further explore it due to the $(1 - A_e[t])$ component as $\lim_{t \rightarrow \infty} (1 - A_e[t]) = 0$. Moreover, the contribution of all robots that are not trying to avoid obstacles or maintain the *MANET* connectivity is explained by the $N_{inc}[t] + N_{exc}[t]$ component.

It is noteworthy that although the same detection rate is defined for both socially active and excluded robots, one should consider that the relation between the transition probabilities greatly differ as the average delay within each state is different (*cf.*, Table 9.4).

9.3.4 Social exclusion rate

As previously stated, the *RDPSO* algorithm mimics the natural selection in Darwin's theory by benefiting from a "punish"- "reward" mechanism (see section 4.5). In short, misbehaving robots are "punished" by being socially excluded while ever-improving robots are "rewarded" by either increasing their team size or by creating new groups from the socially excluded robots. However, this social reevaluation of a robot depends on the following variables:

- Number of times the robot was unable to improve (*i.e.*, stagnancy counter $SC_s[t]$ and threshold SC_{max});
- Minimum N_{min} , maximum N_{max} and current $N_s[t]$ number of robots within its socially active subgroup;
- Number of currently available socially excluded robots $N_s^\times[t]$;
- Number of times its socially active subgroup was punished (*i.e.*, punishing counter $N_s^{kill}[t]$).

It is noteworthy that, in practice, the social exclusion and inclusion rates depend on several other aspects, such as the size of the scenario, the robots' features and, especially, the mission objective. However, those rates do not need to include all those variables due to the properties of the semi-Markov chains. In fact, the dependency between states is settled upon the several evolutionary rates that already depend on all those features which, as a result, indirectly influence the social exclusion and inclusion rates. Hence, considering the dependencies described above, the probability that robots may be socially excluded or included will change over time. As a consequence, the transitions between social statuses, $p_{exc}[t]$ and $p_{inc}[t]$, are time-variant.

As described in section 4.5, the number of times a subgroup s evolves without finding an improved objective is tracked with a stagnancy counter SC_s . If the subgroup improves, then the stagnancy threshold is set to zero. If the subgroup's stagnancy counter exceeds a maximum critical threshold SC_{max} , the subgroup is punished by excluding the worst performing robot, which is added to the socially excluded group. In this situation, the subgroup's stagnancy counter is then reset to a value near SC_{max} that is calculated by means of equation (4.3), wherein $N_s^{kill}[t]$ is a punishing counter that, at the microscopic level, is represented by the difference between the number of robots excluded from subgroup s and included in subgroup s . Observing equation (4.3), one can conclude that the more robots are socially excluded, the more socially active subgroups are susceptible to losing their robots. Hence, at a macroscopic level, *i.e.*, without considering any specific socially active subgroup, one may define an exponentially increasing normalizing punishing counter N_s^{kill} with the number of socially excluded robots as:

$$N_s^{kill}[t] = \frac{N_s^\times[t]}{N_s[t]^2}, \quad (9.23)$$

wherein $N_s^\times[t]$ and $N_s[t]$ is the total number of socially excluded and active robots at time t , respectively. Note that, in the beginning of the mission, all robots are socially active, *i.e.*, $N_s[0] = N_T$ and $N_s^\times[0] = 0 \therefore N_s^{kill}[0] = 0$. However, as previously stated, the more robots advance in the mission, the more difficult it is for them to improve. This will theoretically yield to the exclusion of all robots at some point, *i.e.*, $\lim_{t \rightarrow \infty} N_s[t] = 0$ and $\lim_{t \rightarrow \infty} N_s^\times[t] = N_T \therefore \lim_{t \rightarrow \infty} N_s^{kill}[t] = \infty$.

Therefore, following the reasoning from section 4.5 and considering the inputs from Table 9.3, as well as equations (4.3) and (9.23), one can describe the social exclusion rate as:

$$\gamma_{exc}[t] = \frac{SC_s[t]+1}{SC_{max}^2+1} = \frac{1}{SC_{max}^2+1} - \frac{SC_{max}(N_s[t]-N_T)}{(SC_{max}^2+1)(N_s[t]^2-N_s[t]+N_T)}. \quad (9.24)$$

The addition of the number 1 in the numerator and denominator act as a seed to initialize the social exclusion rate. Otherwise, the transition probability between social statuses would always be zero and robots would never get socially excluded. The choice of raising the stagnancy threshold to the power of 2 in the denominator is not only to cancel its effect on the microscopic definition of the stagnancy counter $SC_s[t]$, previously defined in equation (4.3), but to also give the desired susceptibility to it on the social exclusion rate. By doing so, it is possible to conclude that a larger stagnancy threshold SC_{max} and the population of robots N_T decrease the social exclusion rate. In opposition, the number of socially excluded robots increases the social exclusion rate. Please see Figure 9.2 from Example 9.1 for a better understanding of $\gamma_{exc}[t]$ as a function of $N_s[t]$. This is in accordance to the concepts inherent to the *RDPSO* algorithm (chapter 4). Let us now present a similar rationale for the social inclusion rate.

9.3.5 Social inclusion rate

In the beginning of the mission, socially including robots in active subgroups should be generally easier than excluding them. This rate is related with the likelihood of improving the current solution. If the mission objective comprises an exploration task (*e.g.*, mapping), robots are able to easily find unexplored regions in the beginning. However, and as already stated in this chapter, the exploration rate decreases over time (section 9.3.3) and, as a consequence, the social inclusion rate should also decrease.

As described in section 4.5, socially active subgroups are able to immediately call new members as long as their stagnancy counter is zero, *i.e.*, $SC_s[t] = 0$. Moreover, under those conditions they are even able to spawn new socially active subgroups formed by socially excluded robots with a probability proportional to $\frac{N_s[t]}{N_{max}}$ as defined in equation (4.17). However, those are only valid if there are available socially excluded robots (Table 4.1).

As previously stated, we aim at describing the macroscopic behaviour of the *RDPSO* and, therefore, the difference between socially active subgroups should be neglected. In other words, all robots are considered socially active in the beginning, *i.e.*, $N_s[0] = N_T$. In the first stage, and considering the relation between N_{max} and $N_s[0]$ depicted on Table 9.3, one can define the probability of social active subgroup creation as:

$$p_{spawn}[t] = \frac{N_s[t]}{2N_T}, \quad (9.25)$$

thus translating this ability as proportional to the number of socially active robots.

As a second step, one only needs to remember that the “reward” mechanism is only valid for a punishing counter of zero, *i.e.*, $N_s^{kill}[t] = 0$, and if there are socially excluded robots, *i.e.*, $N_s^\times[t] > 0$. Hence, considering equation (4.3) and (9.25) one can describe the social inclusion rate as:

$$\gamma_{inc}[t] = \left[\frac{p_{spawn}[t] N_T}{SC_s[t]+1} \frac{N_T}{2} + \frac{p_{spawn}[t]}{SC_s[t]+1} \frac{N_T - N_s[t]}{N_T^2} \right] = \frac{N_s[t](N_s[t] - N_T)(N_T + 2)(N_s[t]^2 - N_s[t] + N_T)}{4N_T^3(N_s[t] - N_T + SC_{max}(N_s[t] - N_T) - N_s[t]^2)} \quad (9.26)$$

The first fraction of equation (9.26) within square brackets corresponds to the capability of spawning a new socially active subgroup of $N_{min} = \frac{N_T}{2}$ robots (Table 9.3) while the second fraction within square brackets corresponds to the inclusion of one socially excluded robot within the active subgroup. As one may observe, the probability of inclusion changes nonlinearly with the number of socially active and excluded robots at time t mainly through parameter N_s^{kill} , accordingly to chapter 4.

To clarify the previous definitions of both evolutionary rates $\gamma_{exc}[t]$ and $\gamma_{inc}[t]$, let us present the following example.

Example 9.1. Considering a population of 15 robots ($N_T = 15$) bounded by the RDPSO “punish-reward” rules described in section 4.5, the social exclusion $\gamma_{exc}[t]$ and inclusion $\gamma_{inc}[t]$ rates vary according to the number of socially active robots $N_s[t]$ as:

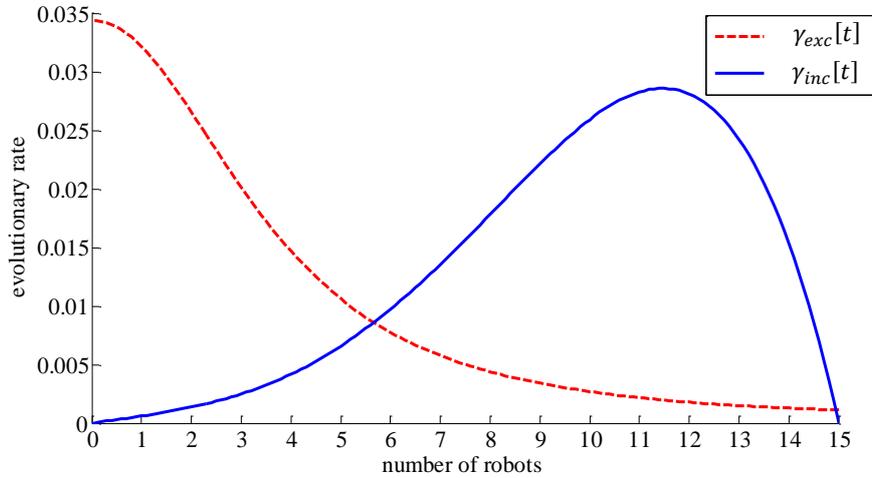


Figure 9.2. Social exclusion $\gamma_{exc}[t]$ and inclusion $\gamma_{inc}[t]$ rates for a population of $N_T = 15$ robots.

As one may observe, when all robots are socially active $N_s[t] = 15$, which usually occurs at the beginning of the mission, robots have a small probability of being excluded proportional to $\frac{1}{SC_{max}^2 + 1}$ as suggested by equation (9.24). As robots are excluded, the probability of exclusion increases, thus increasing the number of socially excluded robots. As a consequence, as excluded robots become available, the probability of inclusion also grows. However, this inclusion rate only grows until the number of socially active robots is large enough to improve over the number of socially excluded robots (for $N_s[t] \gtrsim 11$). Afterwards, the inclusion rate will drop and the exclusion rate will increase. Note that for $N_s[t] \lesssim 6$ the social exclusion rate even outgrows the social inclusion rate, $\gamma_{exc}[t] > \gamma_{inc}[t]$, thus resulting in more and more excluded robots over time as a consequence of the lack of improvement of socially active robots.

The necessary inputs and encountering rates being described, now comes the time to assess them by presenting the preliminary semi-Markov models that will constitute the whole RDPSO macroscopic system.

9.4 Simplified Macroscopic Models

As previously described in chapter 4, the *RDPSO* algorithm comprises two different types of swarms: socially active subgroups and the socially excluded subgroup (section 4.5). Active subgroups represent a key feature from the collective model that allows robots to explore the scenario. On the other hand, robots within the excluded subgroup are able to avoid stagnation which may be represented by a deadlock. Before fully describing the whole *RDPSO* system, one should first study those two different behaviours separately.

The two simplified macroscopic models are evaluated with sets of 30 trials for each different configuration of number of robots $N_T = \{5,10,15\}$. At this point, the mission objective and, as a consequence, the mission-related detection rate γ_e , social exclusion rate γ_{exc} and social inclusion rate γ_{inc} , will be neglected as the *RDPSO* requires merging both social statuses. Moreover, without the *RDPSO* evolutionary properties, the robots' communication interference rate will be constant, *i.e.*, $\gamma_r[t] = \gamma_r \forall t \in \mathbb{N}_0$, since the number of initially deployed robots N_T corresponds to the number of robots from the unique subgroup (or swarm) N_s . Therefore, only the number of robots N_j in each different state j , $j = \{e, w, r\}$, will be analyzed at this point.

To do so, a steady-state analysis should be conducted. The steady-state values under simulation were obtained by averaging each variable over a 1000 steps time window. Since the time step was defined as $\Delta t = 1$ s (please refer to section 3.1.4), this corresponds to a mission time of 16.6 minutes. This is a realistic and considerably large mission time when considering rescue operations in, for instance, indoor urban fires. Due to the rapid evolution of an urban fire, the rescue operation does not take long. After a short period of time, the firefighters need to regroup to establish the means of action and proceed to the firefighting phase. For that reason, the exploration and rescue phase typically lasts between 10 to 15 minutes³³. Yet, to justify the conservative choice regarding a time window of 1000 steps per simulation, Figure 9.3 depicts the average number of robots within each different state over time using a socially active subgroup of $N_s = 15$ robots.

As one may observe, due to the linear dependency between the number of robots as

$$N_e[t] = N_s - (N_w[t] + N_r[t]), \quad (9.27)$$

³³ Information collected during several interviews conducted in the *Coimbra Fire Department* (Portugal) for the *CHOPIN* R&D project (<http://chopin.isr.uc.pt/>).

a settling time of approximately 360 iterations (36 seconds) is enough to reach a steady-state regime in each state. Next section closely compares these results with the predictions retrieved from the macroscopic model.

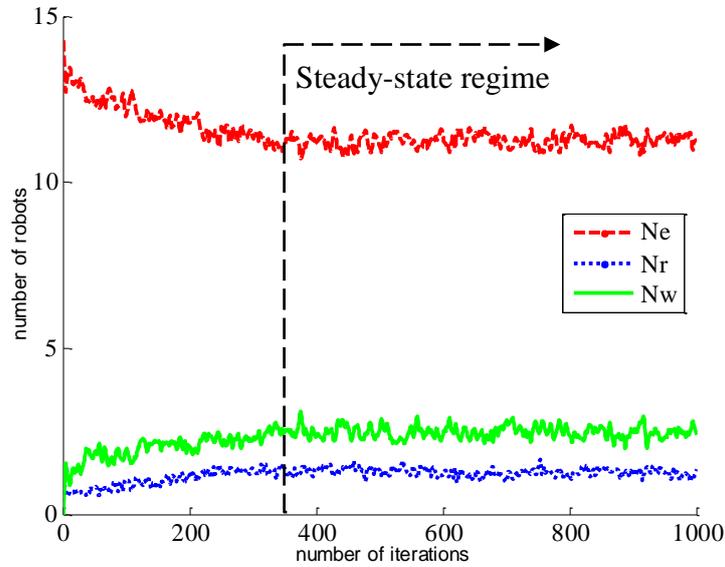


Figure 9.3. Evolution of the average number of robots in each different state over time using a socially active subgroup of $N_s = 15$ robots.

9.4.1 Socially Active Subgroups

As previously described, in socially active subgroups robots' default state is the *Search* state. While in this state, the robot may either encounter an obstacle or suffer from the interference with other robots with probability ρ_w and ρ_r , respectively. Upon such situations, the robot enters either the *Obstacle Avoidance* state or the *Communication Interference* state, remaining on it for a duration of T_w or T_r , respectively. By considering Definition 3.4 and Definition 3.5 from section 3.1.4, this yields to the following finite state semi-Markov macroscopic model of socially active subgroups (Figure 9.4).

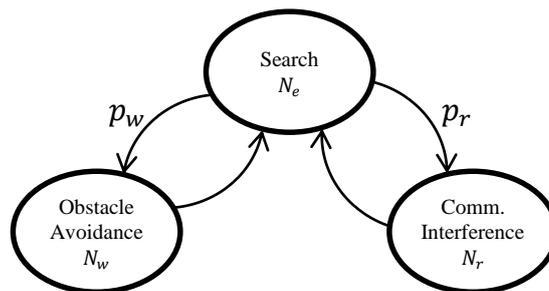


Figure 9.4. Finite state semi-Markov model of socially active subgroups.

From the *PFSM* presented in Figure 9.4 one can extract a *DE* system, one for each state, namely *Search* (9.28), *Obstacle Avoidance* (9.29) and *Communication Interference* (9.30) (Martinoli & Easton, 2003). The *DE* system in (9.28)-(9.30) represents the collective dynamics of socially active subgroups at the macroscopic level:

$$N_e[t + 1] = N_e[t] - (p_w + p_r)N_e[t] + p_w N_e[t - T_w] + p_r N_e[t - T_r], \quad (9.28)$$

$$N_w[t + 1] = N_w[t] + p_w N_e[t] - p_w N_e[t - T_w], \quad (9.29)$$

$$N_r[t + 1] = N_s - N_e[t] - N_w[t]. \quad (9.30)$$

Since the discretization interval is $\Delta t = 1$ s, $N_j[t]$ represents the number of robots in each different state j , $j = \{e, w, r\}$ at time t seconds, $t \in \mathbb{N}_0$. For instance, the *DE* (9.28) models how the number of robots in the *Search* state $N_e[t]$ at iteration t seconds decreases as the number of robots in the *Obstacle Avoidance* $N_w[t]$ and *Communication Interference* states $N_r[t]$ at iteration t seconds increases. It is assumed that in the beginning all robots start in the *Search* state, *i.e.*, $N_e[0] = N_s$ and $N_w[0] = N_r[0] = 0$, and the robots are not in any of the available states before the mission starts, *i.e.*, $N_e[t] = N_w[t] = N_r[t] = 0$, $\forall t < 0$. As one can observe in Figure 9.4, this is an acceptable convention.

To analyse the steady-state regime of the linear *DE* system in (9.28)-(9.30), one can resort to Theorem 3.2 and Theorem 3.3 of the *Z*-Transform (section 3.1.5), thus obtaining the following system:

$$zN_e(z) - N_e(0) = N_e(z) - (p_w + p_r)N_e(z) + p_w z^{-T_w} N_e(z) + p_r z^{-T_r} N_e(z), \quad (9.31)$$

$$zN_w(z) - N_w(0) = N_w(z) + p_w N_e(z) - p_w z^{-T_w} N_e(z), \quad (9.32)$$

$$zN_r(z) - N_r(0) = N_s - N_e(z) - N_w(z). \quad (9.33)$$

One can now find the steady-state of each state $N_j^* = \lim_{t \rightarrow \infty} N_j[t]$, $j = \{e, w, r\}$, *i.e.*, equilibrium, solving the *Z*-Transformed *DE* system (9.31)-(9.33) by employing the final value theorem introduced on Theorem 3.4. Note that solving the limit inherent to the final value theorem on (9.31)-(9.33) results in the indeterminate form of $\frac{0}{0}$. To manipulate the expression so that the limit can be evaluated, one can simply apply the *L'Hôpital's rule* (*aka, Bernoulli's rule*) (Stewart, 2007). This yields the following *DE* system in the steady-state regime:

$$\begin{cases} N_e^* = \lim_{z \rightarrow 1} (z-1)N_e(z) = \frac{N_s}{1+p_w T_w + p_r T_r} \\ N_w^* = \lim_{z \rightarrow 1} (z-1)N_w(z) = \frac{p_w T_w N_s}{1+p_w T_w + p_r T_r} \\ N_r^* = \lim_{z \rightarrow 1} (z-1)N_r(z) = \frac{p_r T_r N_s}{1+p_w T_w + p_r T_r} \end{cases} \quad (9.34)$$

In brief, manipulating the inputs from the robot model (Table 9.1) and the world model (Table 9.2) results in a different steady-state distribution of robots N_j^* within each state $j = \{e, w, r\}$.

Let us now place side-by-side the results obtained from simulation experiments on *MRSim* with the analytical results retrieved from the steady-state analysis of socially active subgroups. Boxplot charts were used to present the results of each set of 30 trials for each configuration graphically (Figure 9.5).

The width ends of the blue boxes and the horizontal red line in between correspond to the first and third quartiles and the median values of the simulation results, respectively. The dark dots represent the steady-state result retrieved from the macroscopic model for each configuration using the values from Table 9.2. The dark linking line is only used to illustrate how the distribution varies with the number of robots. The grey regions around the dark dots are of strategic importance since they represent the range of results obtained using the macroscopic model while varying both area and density of obstacles to $\pm 25\%$ of the real values depicted on Table 9.2, *i.e.*, $A \in [2231, 3719]$ and $\rho_w \in [0.0370, 0.0616]$. This makes it possible to understand the sensibility of the proposed model to variations of those parameters since they may be susceptible to flaws inherent to human agents' perception.

The results presented on Figure 9.5 point toward a good match between the macroscopic model predictions and the simulation results for all steady-state variables regardless of the variations in the scenario characteristics. It is however noteworthy that the macroscopic model still achieves a superior prediction when using the exact values (dark dots). This also indicates that the mapping between the microscopic *RDPSO* behaviour and its macroscopic counterpart without evolutionary properties is straightforward and the individual behaviour of each robot may be neglected in order to obtain an average number of robots in a given state. From the predictions achieved by the macroscopic model, the number of robots avoiding inter-robot *Communication Interference* N_r seems to be the one showing some minor discrepancies. In fact, one may observe that when the population goes from 10 to 15 robots, the quasi-linear relation of the distribution of robots is clearly affected, *i.e.*, while the distribution of robots in the *Search* and *Obstacle Avoidance* states seem to almost grow linearly with the number of robots, the distribution of robots in the robots' *Communication Interference* state does not

follow that trend. Despite still not falling utterly apart from the inter-quartile range, it still presents some incorrect predictions for significant variations of the scenario's features.

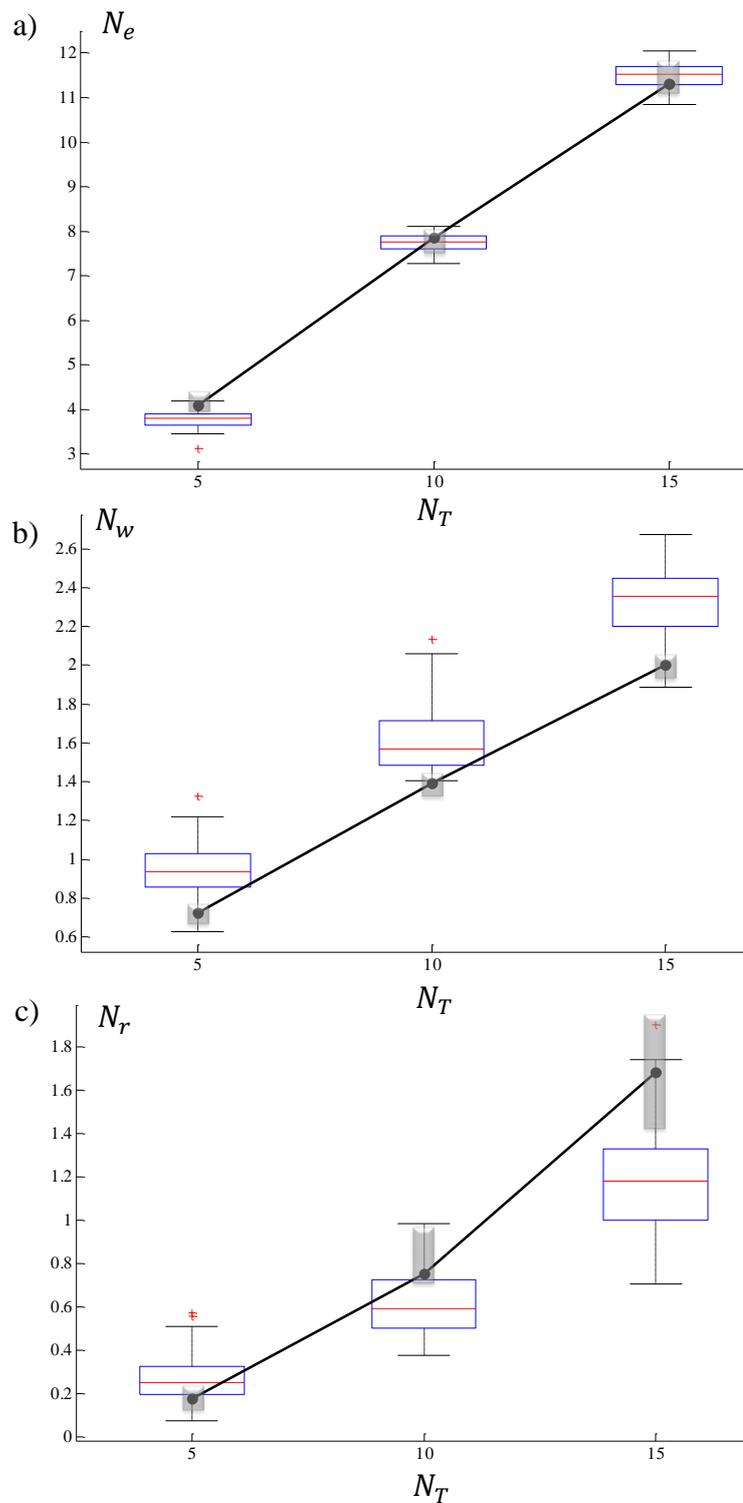


Figure 9.5. Steady-state analysis of socially active subgroups on both simulated and analytical results using the proposed macroscopic model. a) number of robots exploring the scenario; b) number of robots avoiding obstacles; and c) number of robots constrained by communication interference.

9.4.2 Socially Excluded Subgroup

The major difference between socially active subgroups and the socially excluded subgroup resides in the fact that robots within the socially excluded subgroup do not focus on the main mission objective. Instead, they randomly wander in the scenario, thus avoiding stagnation from sub-optimal solutions. For the sake of simplicity and without lack of generality, the same subscript e will be used to identify the *Wandering* state of excluded robots (Figure 9.6). It is also noteworthy that, although robots do not intentionally explore the scenario, they still have a detection rate γ_e .

As one may observe, the finite state semi-Markov macroscopic model of the socially excluded subgroup depicted in Figure 9.6 is essentially the same as the one from active subgroups (Figure 9.5). Therefore, from the *PFSM* on Figure 9.6 one can extract the same *DE* system in (9.28)-(9.30).

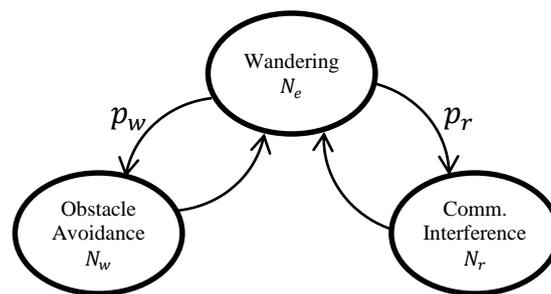


Figure 9.6. Finite state semi-Markov model of socially excluded subgroups.

Once again, following the same reasoning as in section 9.4.1, the *DE* system in the steady-state regime will correspond to the one presented in (9.34). Note, however, that although the inputs from the robot model (Table 9.1) and the world model (Table 9.2) are the same as before, the delay of robots within the *Obstacle Avoidance* T_w and *Communication Interference* T_r states is different (Table 9.4). As a consequence, this will result in a different steady-state distribution of robots N_j^* within each state $j = \{e, w, r\}$.

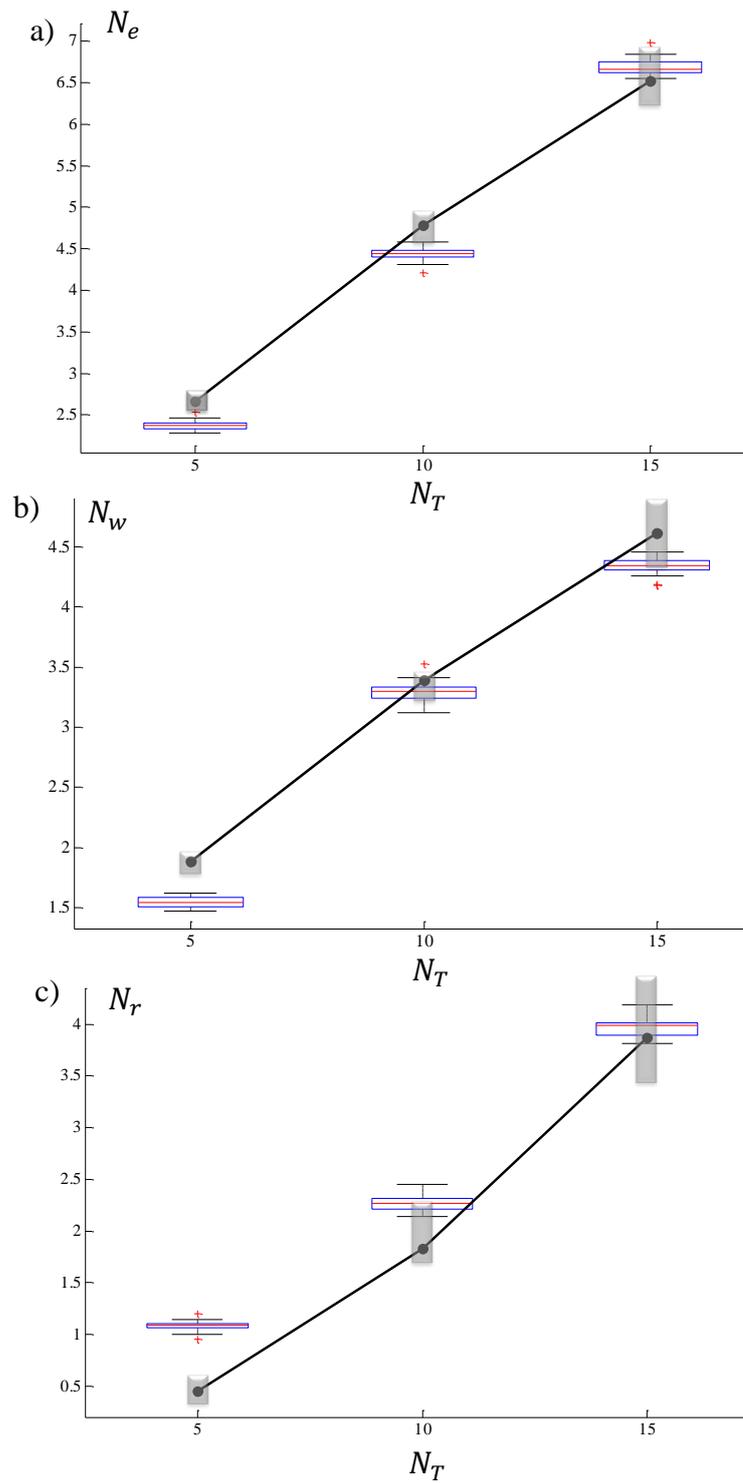


Figure 9.7. Steady-state analysis of the socially excluded subgroup on both simulated and analytical results using the proposed macroscopic model. a) number of robots exploring the scenario; b) number of robots avoiding obstacles; and c) number of robots constrained by communication interference.

As before, let us now compare the results obtained from simulation experiments with the analytical results retrieved from the steady-state analysis of the socially excluded subgroup. Figure 9.7 depicts the boxplot charts of the results of each set of 30 trials for each configuration and the analytical results retrieved from the previous section. By observing the results from Figure 9.7 one may immediately conclude that the behaviour of socially excluded robots is harder to predict than the socially active ones. Although the differences are still minor, in some occasions the prediction does not fall within the inter-quartile range. For instance, the worst case occurs for a small population of 5 robots in which the macroscopic model is unable to accurately predict the number of robots within the robots' *Communication Interference* state. Although this difference is mitigated as the number of robots grows, this is still worth discussing. Moreover, the number of robots within the robots' *Communication Interference* state is still more susceptible to variations in the scenario features.

These discrepancies on steady-state regime will be further explored after presenting the evolutionary *RDPSO* macroscopic model that merges both of the previously presented models.

9.5 Evolutionary Macroscopic Model

This section intends to fully address the macroscopic modelling of the *RDPSO* algorithm by merging the two finite state semi-Markov models presented on Figure 9.4 and Figure 9.6. As previously stated, the sets of *DEs* from those two simplified systems at the macroscopic level are linear. However, the macroscopic model that comprises the full collective and evolutionary behaviour presented in this section will correspond to a set of nonlinear time-delayed *DEs* due to the dynamic clustering properties of the *RDPSO*.

To keep up the analogy with the previously presented semi-Markov chains from Figure 9.4 and Figure 9.6, let us define the same states for each social status, *i.e.*, *Search (Wandering)*, *Obstacle Avoidance* and *Communication Interference*. For the sake of simplicity, the \times symbol will be used as superscript to identify the common variables (*e.g.*, states, transition probabilities and others) associated to the socially excluded subgroups.

Contrarily to the previous models in which the mission objective (*i.e.*, exploration ratio) was overlooked due to the simplified macroscopic models, this section intends to consider such variable. Therefore, the *Search (Wandering)* states were divided into two sub-states, namely, *Exploring* and *Found*, in such a way that the number of robots in the *Search* state is given by $N_{inc} + N_e$ and in the *Wandering* state by $N_{exc} + N_e^\times$. The N_{inc} (N_{exc}) robots within the *Search (Wandering)* [*Exploring*] state are either socially active or excluded, respectively, while traveling to an unexplored regions.

Complementarily, N_e and N_e^\times robots within the *Search* (*Wandering*) [*Found*] states are either socially active or excluded, respectively, while scanning an unexplored region.

The transition between both sub-states is described by the mission-related transition probability $p_e[t]$ and $p_e^\times[t]$ as a function of the exploration rate defined in equation (9.19). Within the *RDPSO* full aggregation, most of the transition probabilities, such as the transitions between social statuses, $p_{exc}[t]$ and $p_{inc}[t]$, the mission-related transition $p_e[t]$ and $p_e^\times[t]$ and the communication interference transition $p_r[t]$ and $p_r^\times[t]$, are time-varying. This introduces nonlinear coupling factors among the equations, thus resulting in a nonlinear time-delayed *DE* system (Figure 9.8).

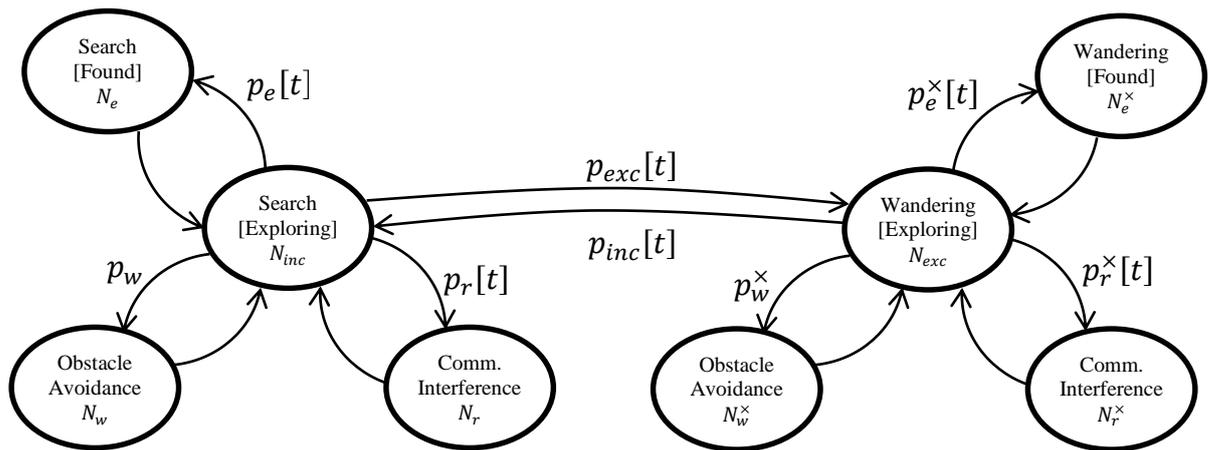


Figure 9.8. Finite state semi-Markov model of the complete evolutionary *RDPSO* algorithm. The \times superscript identifies the states and transition probabilities associated to the socially excluded subgroups.

From the *PFSM* presented in Figure 9.8 one can write the following *DE* system representative of the evolutionary *RDPSO* collective dynamics at the macroscopic level:

$$\begin{aligned} N_{inc}[t+1] = & N_{inc}[t] - (p_w + p_r[t] + p_e[t] + p_{exc}[t])N_{inc}[t] + \\ & + p_w N_{inc}[t - T_w] + p_r[t - T_r]N_{inc}[t - T_r] + p_e[t - T_e]N_{inc}[t - T_e] + \\ & p_{inc}[t - T_{exc}^\times]N_{exc}[t - T_{exc}], \end{aligned} \quad (9.35)$$

$$N_e[t+1] = N_e[t] + p_e[t]N_{inc}[t] - p_e[t - T_e]N_{inc}[t - T_e], \quad (9.36)$$

$$N_w[t+1] = N_w[t] + p_w N_{inc}[t] - p_w N_{inc}[t - T_w], \quad (9.37)$$

$$N_r[t+1] = N_s[t] - N_{inc}[t] - N_w[t] - N_e[t], \quad (9.38)$$

$$\begin{aligned} N_{exc}[t+1] = & N_{exc}[t] - (p_w^\times + p_r^\times[t] + p_e^\times[t] + p_{inc}[t])N_{exc}[t] + \\ & + p_w^\times N_{exc}[t - T_w^\times] + p_r^\times[t - T_r^\times]N_{exc}[t - T_r^\times] + p_e^\times[t - T_e^\times]N_{exc}[t - \\ & T_e^\times] + p_{exc}[t - T_{inc}]N_{inc}[t - T_{inc}], \end{aligned} \quad (9.39)$$

$$N_e^\times[t+1] = N_e^\times[t] + p_e^\times[t]N_{exc}[t] - p_e^\times[t - T_e^\times]N_{exc}[t - T_{exc}], \quad (9.40)$$

$$N_w^\times[t+1] = N_w^\times[t] + p_w^\times N_{exc}[t] - p_w^\times N_{exc}[t - T_w^\times], \quad (9.41)$$

$$N_r^\times[t+1] = N_s^\times[t] - N_{exc}[t] - N_w^\times[t] - N_e^\times[t]. \quad (9.42)$$

As previously stated, all robots start in the *Search [Exploring]* state, *i.e.*, $N_{inc}[0] = N_s$ and $N_j[0] = 0, \forall j \neq inc$, and the robots are not in any of the available states before the mission starts, *i.e.*, $N_j[t] = 0$, for $\forall t < 0$ and with $j = \{inc, e, w, r, exc, e^\times, w^\times, r^\times\}$.

The *DE* system represented in (9.35)-(9.42) can be interpreted as the *DE* system from (9.28)-(9.30). For instance, adding equations (9.35) and (9.36) tells that the number of robots exploring the scenario decreases when robots interfere with each other, when they encounter obstacles and when they are socially excluded. Furthermore, adding equations (9.35) to (9.38) returns the number of socially active robots $N_s[t+1]$ while, as opposed, adding equations (9.39) to (9.42) returns the number of socially excluded robots $N_s^\times[t+1]$, such that one can write the following relation:

$$N_s[t+1] = N_{inc}[t+1] + N_e[t+1] + N_w[t+1] + N_r[t+1], \quad (9.43)$$

$$N_s^\times[t+1] = N_{exc}[t+1] + N_e^\times[t+1] + N_w^\times[t+1] + N_r^\times[t+1], \quad (9.44)$$

$$N_T = N_s[t+1] + N_s^\times[t+1]. \quad (9.45)$$

Note that the population of robots N_T is time invariant. Theoretically, following the principles of the *RDPSO* “punish-reward” mechanisms (section 4.5), it is expected that the number of socially active robots decreases over time, thus consequently increasing the number of socially excluded robots as equation (9.45) suggests, in such a way that $\lim_{t \rightarrow \infty} N_s[t] = 0 \therefore \lim_{t \rightarrow \infty} N_s^\times[t] = N_T$. How exactly the number of robots within each state varies can once again be assessed through the steady-state regime.

9.5.1 Steady-state analysis

To analyse the steady-state regime of the nonlinear time-delayed *DE* system in (9.35)-(9.42), one can once again resort to the *Z*-Transform (section 3.1.5). However, contrarily to the previous *DE* macroscopic systems, the collective system of the full aggregation described by (9.35)-(9.42) adds a new level of complexity due to the products between time-variant functions (*e.g.*, $p_r[t - T_r]N_{inc}[t - T_r]$). Note that in the frequency *Z*-domain, the product of two functions is translated into a curvilinear integral of the product of both *Z*-Transformed functions within their radius of convergence. The *Z*-

transform of the product of two discrete time domain functions is beyond the scope of this Thesis, but for further information please refer to (Kliger & Lipinski, 1964).

This additional aspect considerably raises the level of complexity of the Z -Transform and its analysis since the transition probability functions depend on several parameters of which, in turn, some of them are also time-varying functions (*e.g.*, $N_s[t]$). Therefore, to bypass the mathematical complexity inherent to the Z -Transform of a product of two functions comprising of a large set of inputs such as time t , let us focus on the main purpose of such transformation in the context of this work: the steady-state analysis. By limiting the DE system analysis from (9.35)-(9.42) to its steady-state regime, one can define the time-varying transition probabilities with their “final value”, *i.e.*, for $t \rightarrow \infty$ (Theorem 3.4).

First, let us recall a couple of already mentioned final values inherent to the $RDPSO$ algorithm:

- All robots within the population tend to be excluded as time goes by in such a way that $\lim_{t \rightarrow \infty} N_s^\times[t] = N_T$ and $\lim_{t \rightarrow \infty} N_s[t] = 0 \therefore \lim_{t \rightarrow \infty} N_s^{kill}[t] = \infty$;
- The more the scenario is explored, the more difficult it is to further explore it in such a way that $\lim_{t \rightarrow \infty} \gamma_e[t] = 0$.

At last, considering the above properties and equations (9.3), (9.19), (9.24) and (9.26), yields to the following transition probabilities “final values”:

$$\lim_{t \rightarrow \infty} p_r[t] = \lim_{t \rightarrow \infty} p_e[t] = \lim_{t \rightarrow \infty} p_e^\times[t] = \lim_{t \rightarrow \infty} p_{inc}[t] = 0, \quad (9.46)$$

$$\lim_{t \rightarrow \infty} p_{exc}[t] = \frac{SC_{max}+1}{SC_{max}^2+1} T_{exc} \approx \frac{1}{SC_{max}} T_{exc}, \quad (9.47)$$

$$\lim_{t \rightarrow \infty} p_r^\times[t] = 2\bar{v}(N_T - 1) \frac{R_r}{A_a} T_r^\times. \quad (9.48)$$

Note that the approximation from equation (9.47) is reasonable as the stagnancy threshold is considerably superior to 1, *i.e.*, $SC_{max} \gg 1$. Replacing the corresponding transition probabilities “final values” in the DE system (9.35)-(9.42), one can easily apply the Z -Transform (section 3.1.5). Afterwards, by employing the final value theorem (Theorem 3.4) it becomes possible to find the steady-state of each state $N_j^* = \lim_{t \rightarrow \infty} N_j[t]$, $j = \{inc, e, w, r, exc, e^\times, w^\times, r^\times\}$ as:

$$\left\{ \begin{array}{l} N_{inc}^* = \lim_{z \rightarrow 1} (z-1)N_{inc}(z) = 0 \\ N_e^* = \lim_{z \rightarrow 1} (z-1)N_e(z) = 0 \\ N_w^* = \lim_{z \rightarrow 1} (z-1)N_w(z) = 0 \\ N_r^* = \lim_{z \rightarrow 1} (z-1)N_r(z) = 0 \\ N_{exc}^* = \lim_{z \rightarrow 1} (z-1)N_{exc}(z) = \frac{N_T}{1+p_w^\times T_w^\times + p_r^\times T_r^\times}, \\ N_e^{\times*} = \lim_{z \rightarrow 1} (z-1)N_e^\times(z) = 0 \\ N_w^{\times*} = \lim_{z \rightarrow 1} (z-1)N_w^\times(z) = \frac{p_w^\times T_w^\times N_T}{1+p_w^\times T_w^\times + p_r^\times T_r^\times} \\ N_r^{\times*} = \lim_{z \rightarrow 1} (z-1)N_r^\times(z) = \frac{p_r^\times T_r^\times N_T}{1+p_w^\times T_w^\times + p_r^\times T_r^\times} \end{array} \right. \quad (9.49)$$

with $p_r^\times = 2\bar{v}(N_T - 1)\frac{R_r}{A_a}T_r^\times$. Considering the *DE* systems in (9.43)-(9.45) and (9.49), the final number of robots socially active and excluded, herein denoted as N_s^* and $N_s^{\times*}$, respectively, can be calculated as:

$$\left\{ \begin{array}{l} N_s^* = N_{inc}^* + N_e^* + N_w^* + N_r^* = 0 \\ N_s^{\times*} = N_{exc}^* + N_e^{\times*} + N_w^{\times*} + N_r^{\times*} = N_T \end{array} \right. \quad (9.50)$$

As one may observe, in accordance to the *RDPSO* “punish-reward” mechanism, as time tends to infinity all robots turn out to be socially excluded, *i.e.*, $N_s^* = 0$ and $N_s^{\times*} = N_T$. In addition, and regardless of the social status, robots are also unable to find unexplored regions as time tends to infinity, *i.e.*, $N_e^* = N_e^{\times*} = 0$. Finally, the relationship between the *Wandering [Exploring]*, the *Obstacle Avoidance* and *Communication Interference* states follows the same principles as in the previously presented macroscopic models from section 9.4.

9.5.2 Preliminary Results

Let us now evaluate the *RDPSO* full macroscopic model herein proposed with *MRSim* simulation experiments under different configurations. As before, 30 trials for each configuration of population of robots $N_T = \{5,10,15\}$ were carried out. As before, let us first analyze the number of robots within states. However, and considering the *DE* system in (9.49), a different visualization method is required. Note that following the same boxplots from Figure 9.5 and Figure 9.7 would yield a large variability (inter-quartile range) as, contrarily to before, the number of robots within each social status varies over time, thus changing the whole proportion between states of each social status.

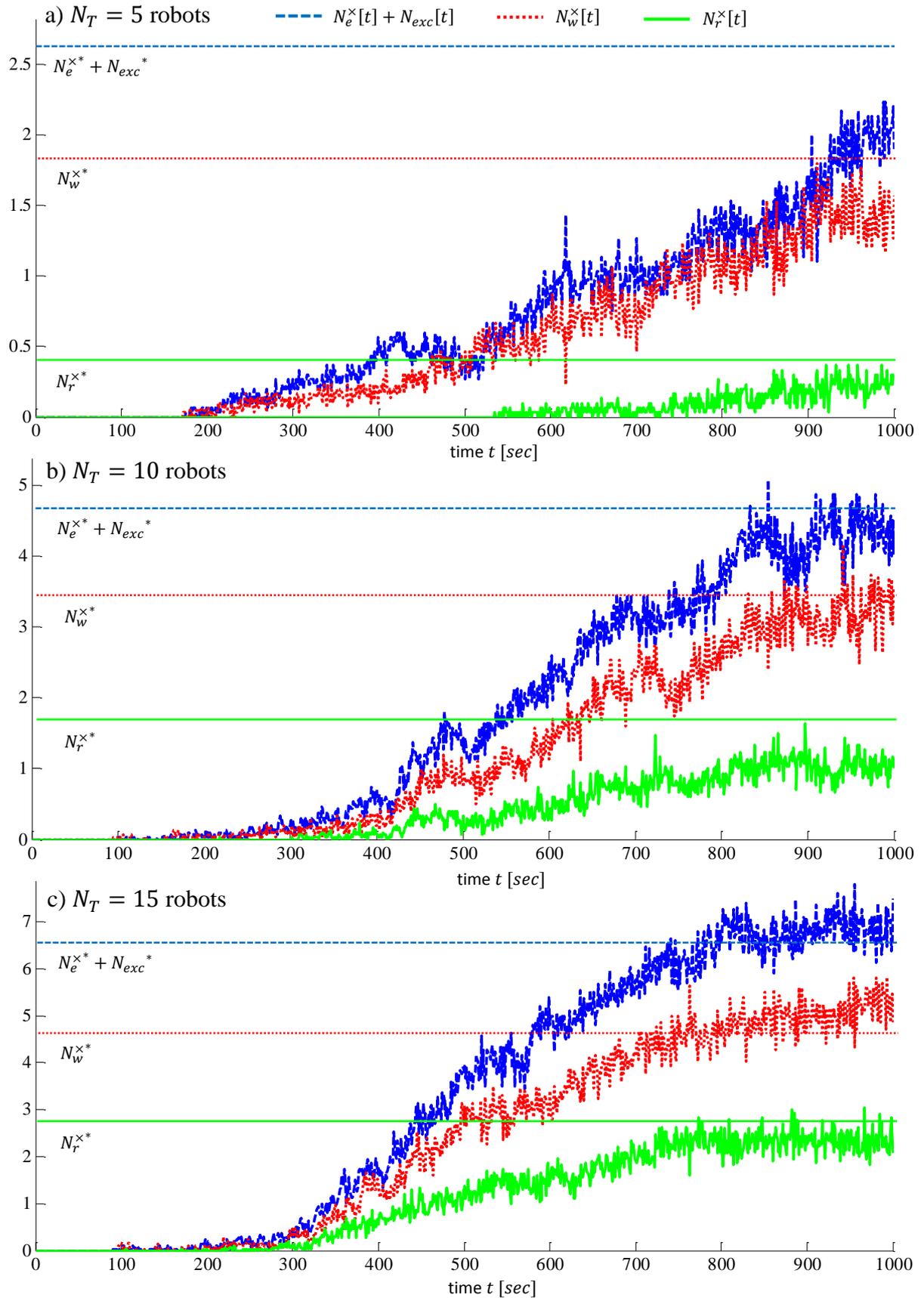


Figure 9.9. Evolution of the average number of socially excluded robots in each different state over time. The final values (horizontal lines) retrieved from equation (9.49) are represented by the subscript *.

Therefore, instead of showing the distribution of robots in the steady-state regime, Figure 9.9 depicts the average distribution of robots within the main states of the social exclusion status, *i.e.*, *Search* ($N_e^\times[t] + N_{exc}[t]$), *Obstacle Avoidance* $N_w^\times[t]$ and *Communication Interference* $N_r^\times[t]$. The steady-state number of robot within each state calculated by means of the *DE* system in (9.49), namely N_{exc}^* , $N_w^{\times*}$ and $N_r^{\times*}$, were represented in the same chart as horizontal lines.

As previously stated, all robots start as socially active. As they advance in the mission and it gets harder to improve, they get excluded by following the principles inherent to the *RDPSO* rules (section 4.5). Regardless of the population of robots one can easily observe that the distribution tends to the final values retrieved from the full *RDPSO* macroscopic model. However, such similarity increases with the number of robots within the population. For instance, although the number of robots from each state converges to the steady-state value in the case of a population of $N_T = 5$ robots, such convergence is slower than with a larger population. Regardless of that aspect, the estimation regarding the distribution of robots performing a given task can be forecasted by the macroscopic model from Figure 9.8.

As previously mentioned, the main difference between the previous two macroscopic models from section 9.4 and the full macroscopic model of the *RDPSO* algorithm herein presented is that the probabilities with which robots switch states are functions of time rather than being time-invariant. More precisely, the aggregation process modifies robots' communication interference probability, while including the social exclusion and inclusion rates and, most importantly, the mission-related exploration rate that varies over time. This last variable can be considered to be the *RDPSO* full aggregation system most important outcome as it may decide upon the choice of the desired teams' configuration (*e.g.*, number of robots).

For that reason, Figure 9.10 compares the exploration ratio, or proportion of the explored area, $A_e[t]$, retrieved from the simulation experiments with the one estimated by the herein proposed macroscopic model. The coloured zones between the solid lines represent the interquartile range of the best solution in the 30 trials that was taken as the final output for each different configuration. The dashed line correspond to the estimated proportion of the explored area $A_e[t]$ using equation (9.20) and the macroscopic model from Figure 9.8.

As one may observe, the estimation of the explored area is within the inter-quartile range for all the evaluated configurations. Although the macroscopic estimation gets near the first quartile in the situation of $N_T = 5$ robots, it seems that its accuracy grows with the population. For instance, for $N_T = 15$ robots, the estimated outcome is almost the same as the median of the proportion of the

explored area retrieved over the 30 trials. This is of major importance as one can predict the performance of the robotic teams under the *RDPSO* algorithm without resorting to simulations or any other kind of experimental evaluation.

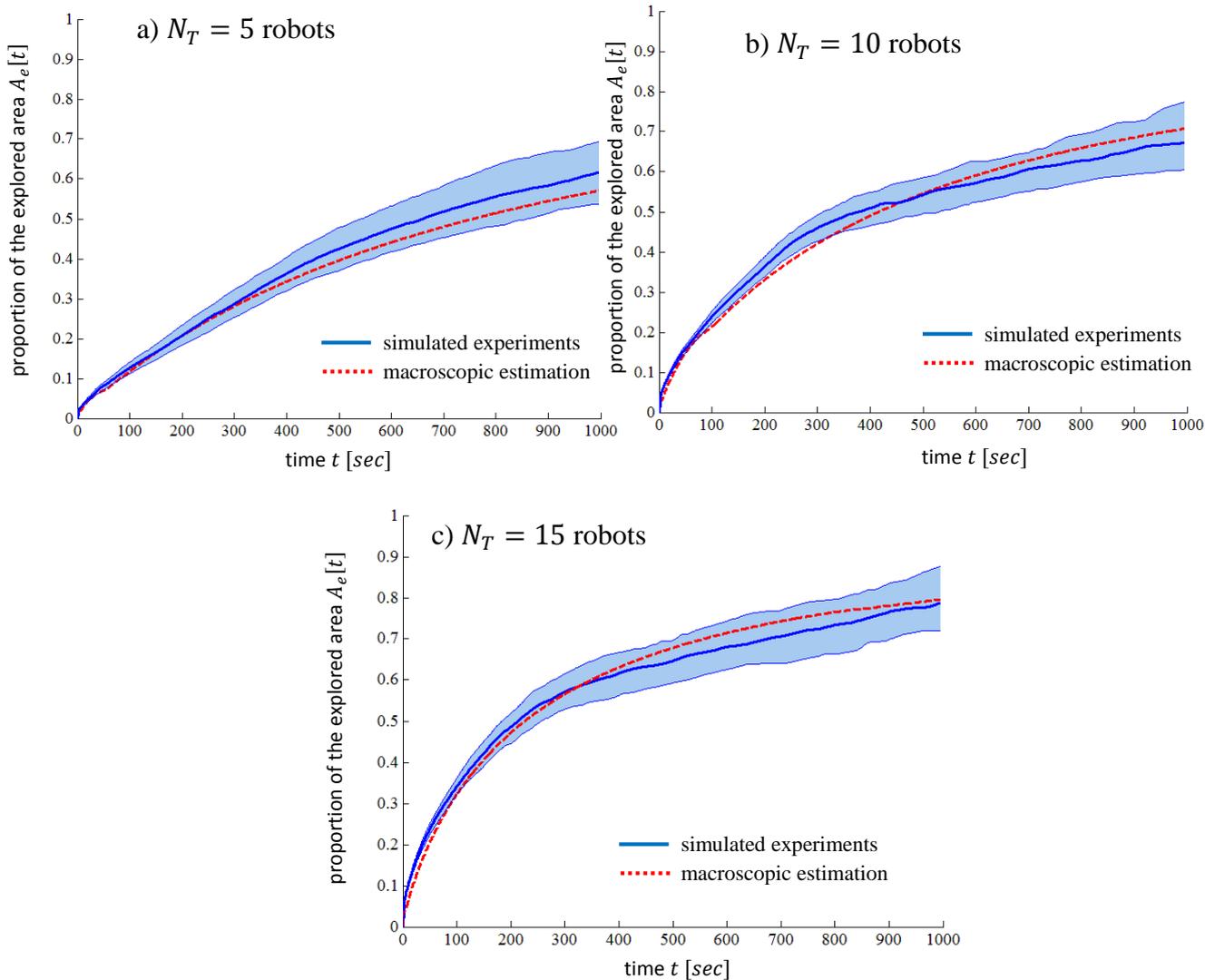


Figure 9.10. Proportion of the explored area over time, $A_e[t]$, for both simulated experiments and macroscopic estimation.

To further evaluate the herein proposed macroscopic model, next section explores its prediction accuracy for a totally different scenario and application with real robotic platforms.

9.6 Experimental Results

This section explores the prediction accuracy of the macroscopic *RDPSO* using swarms of the *Arduino*-based *eSwarBots* (see section 3.2.1), while performing a collective exploration task in a realistic scenario populated with obstacles and under communication constraints.

As previously described in section 3.2.1, all the sensing information of the robot comes from a single sonar range finder connected to *Arduino* analog input. Although the sonar is accurate and provides readings up to approximately 6 meters, there are some errors (*i.e.*, outliers) that occur when the robot moves. Regardless of those errors, robots with readings inferior to 6 meters would be within the obstacle avoidance state, *i.e.*, $R_w = 6\text{ m}$. Also, inter-robot communication within the same subgroups was carried out by benefiting from the *XBee* modules described in section 3.2.1. The ideal communication signal between *eSwarBots* from the same subgroup was settled with a *RSSI* of -75 dBm , thus corresponding to inter-robot distances between 2 and 4 meters. In other words, robots within that range would be in robots' communication interference state. For the sake of simplicity, let us define the interference radius as the average value of such range, *i.e.*, $R_r = 3\text{ m}$.

eSwarBot relevant features to the macroscopic model are summarized in Table 9.5. Note that the values in Table 9.5 greatly differ from the ones in Table 9.1. In practice, this could yield to a different average delay of robots within each state (Table 9.4). However, the same values from Table 9.4 will be used so as to evaluate the generalization of the macroscopic model to any application of the *RDPSO* algorithm.

Table 9.5. Inputs of the *eSwarBot* robot model.

\bar{v} [$m \cdot s^{-1}$]	R_w [m]	R_r [m]
0.2	6	3

The real experiments used for comparison purposes are the same carried out in section 5.3.1, in which the *EST* strategy was adopted for a swarm of 15 *eSwarBots* deployed in a 10×20 meters sports pavilion ($A = 200\text{ m}^2$) in which obstacles were randomly deployed (see Figure 5.7). As previously, the experimental arena contained two sites represented by an illuminated spot uniquely identifiable by controlling the brightness of the light. Therefore, contrarily to the mapping objective previously used as case study throughout this work, the robots' main objective was to find the brighter site (optimal solution). Table 9.6 summarizes the relevant features of the experimental arena.

Table 9.6. Inputs of the real world model.

A [m^2]	ρ_w
200	0.0175

As the mission objective drastically changes from mapping to light source localization, the definition of the exploration rate γ_e needs to be slightly adjusted. While previously the exploration rate

was proportional to the obstacle detection sensing range, now the mission-related sensing capabilities only rely on the light sensor LDR on top of the robots. As the $eSwarBot$ has a radius of $R_{robot} = 63 \text{ mm}$ (section 3.2.1), let us redefine the detection radius R_w , only from equation (9.19), with that value.

Due to the stochasticity inherent to the $RDPSO$ algorithm, 20 trials of 360 seconds each were considered for 15 $eSwarBots$, *i.e.*, $N_T = 15$. A minimum, initial and maximum number of 0 (all robots socially excluded), 3 and 6 subgroups were used (in accordance with the conditions from Table 9.3), thus resulting in an initial subgroup size of $N_S[0] = 5 \text{ eSwarBots}$.

Figure 9.11 depicts the inter-quartile range of the normalized solution over time. The dashed line corresponds to the estimated proportion of the explored area $A_e[t]$ using equation (9.20) and the macroscopic model from Figure 9.8. Note that the macroscopic model was previously created, and closely compared to simulation experiments, within an exploration task that consisted of mapping an unknown scenario. In this section, the $RDPSO$ was evaluated under a source localization task and compared to the same macroscopic model from Figure 9.8. Despite being centred on a totally different task, the macroscopic model of the $RDPSO$ still presents a considerably acceptable estimation of the outcome. Although one may observe a lack of accuracy in the first two minutes of the mission, the macroscopic estimation $A_e[t]$ converges to the median value retrieved from the experiments.

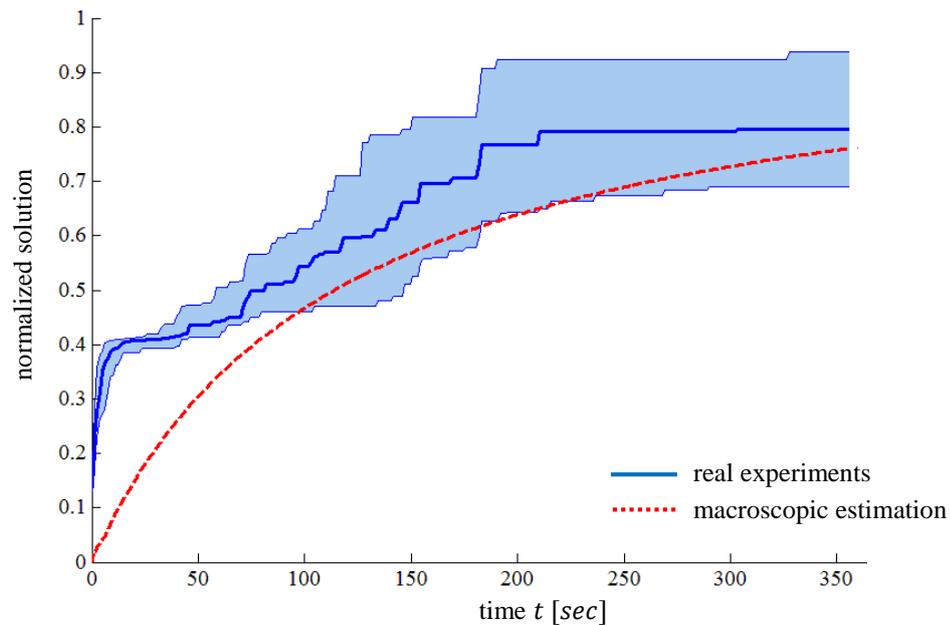


Figure 9.11. Comparison between the normalized best intensity of light sensed by the swarm of robots over time and the macroscopic estimation $A_e[t]$.

9.7 Discussion

For all intents and purposes, predicting how biological swarms, either formed by bees, locusts, or any other creature, behave is impossible. In fact, the mathematical model of the collective behaviour, *aka* macroscopic model, gets more accurate as the data about a certain swarm grows. For instance, the authors from Buhl *et al.* (Buhl, et al., 2006) studied the reasons that could lead to the formation of locust bands predicting the transitions from disordered “*solitarious*” movements to ordered “*gregarious*” movements. From that study, it was possible to conclude that a small number of 8 locusts was enough to form a group with a directed collective motion. In the same way that this and other similar works aimed at understanding such behaviours to, for instance, control mobile swarming insect pests such as the desert locust, this work ambitions lays in a similar principle: identifying the necessary number of robots to form a swarm with “gregarious” actions to fulfil a given task.

Following the footsteps of the successive improvements and benchmark around the *RDPSO* algorithm (previous chapters), the contribution of this chapter was in proposing a macroscopic model that could represent the collective behaviour of swarms. Such macroscopic model can be used to predict the teams’ performance under specific situations and, henceforth, find the most rightful configuration for a given application (*e.g.*, number of robots within each team), thus providing an analytical method to synthesize robot swarms using the *RDPSO* exploration algorithm.

The macroscopic model inspired by semi-Markov concepts was developed throughout the chapter and compared side-by-side with experiments. The dependencies (inputs) of the macroscopic model comprises on the robots’ features (*i.e.*, sensing range R_w , communication range R_r and average velocity \bar{v}) and the world’s features (*i.e.*, area of the scenario A and density of obstacles ρ_w), *cf.*, Table 9.1 and Table 9.2.

At first, the analysis was carried out separately for each social status, *i.e.*, for robots within socially active subgroups (section 9.4.1) and for robots within the socially excluded subgroup (section 9.4.2). With those preliminary results it was possible to verify the capability of the simplified macroscopic models to predict the average number of robots under one of the following tasks: *Search*, *Obstacle Avoidance* and *Communication Interference*. Despite some minor discrepancies observed on the average number of robots located in the *Communication Interference* state, the simplified macroscopic models were successful into answering the question:

“How many (R_w, R_r, \bar{v}) -robots from a swarm will be performing a given task in a (A, ρ_w) -scenario?”

To further exploit the relevance of predicting the *RDPSO* performance, the social statuses were integrated within the same macroscopic model (section 9.5). This global macroscopic model (Figure 9.8) encompassed the full collective and evolutionary behaviour inherent to the dynamic clustering properties *RDPSO* algorithm. As a consequence, and contrarily to the previous simplified models from section 9.4, this macroscopic model was defined by a set of nonlinear time-delayed *DEs* due to the time-varying transition probabilities between states of the semi-Markov chain. Despite the complexity of the semi-Markov chain, more than just predicting the steady-state regime associated to the number of robots performing a given task, the macroscopic model was able to return a more accurate estimation of the scenario explored by swarms over time $A_e[t]$. The conclusion withdrawn from section 9.5.2 was that such outcome estimation improves with the number of robots in the swarm. For instance, by observing Figure 9.10c, it can be concluded that at least 15 robots are required to map 75% of the scenario depicted in

Figure 3.9 within the mandatory time limit of 1000 seconds. In sum, the macroscopic model of the full *RDPSO* aggregation was successful into answering the question:

“How many (R_w, R_r, \vec{v}) -robots from a swarm are necessary to map a (A, ρ_w) -scenario?”

With the purpose of further validating the proposed macroscopic model, experiments with 15 real robots were carried out on a scenario with $A = 200 \text{ m}^2$ (section 9.6). The task went from mapping an unknown scenario to a source localization problem. Nevertheless, despite this key difference, the macroscopic model was still able to successfully predict the outcome of the robotic teams with some discrepancies in the first two minutes of exploration.

9.8 Summary

In the previous chapters, the *RDPSO* was proposed as a complete solution to accomplishing exploration tasks. The key feature of the *RDPSO* algorithm is its dynamical partitioning mechanism of the whole population of robots by means of an evolutionary social exclusion based on Darwin’s survival-of-the-fittest. This evolutionary property raises the level of complexity associated with the stochasticity already inherent to swarm robotic algorithms, thus making it hard to predict the *RDPSO* performance under specific situations and, henceforth, to synthesize a robot swarm that fulfils the application requirements, without resorting to trial-and-error upon numerical simulations carried out before the swam deployment in the real scenario.

This chapter proposed a macroscopic model of the *RDPSO* algorithm based on semi-Markov chains with the purpose of estimating the distribution of robots within certain tasks and their performance without resorting to experimentation. To that end, the *RDPSO* algorithm was thoroughly studied and a set of evolutionary rates were defined. These evolutionary rates, most of which were time-varying, gave rise to the transition probabilities on which the semi-Markov model depends to switch between states. The semi-Markov macroscopic model was successively evaluated side-by-side with simulation experiments around a mapping task and, later, with real experiments under a source localization task. Despite some discrepancies, the experimental results clearly prove the success of the macroscopic model in estimating the performance of the *RDPSO* algorithm under different tasks, number of robots and scenario characteristics.

All that being said, one may state that it is still difficult at this point to find a simple answer to the underlying research question. However, we argue that this chapter provides an accurate and precise macroscopic model of the evolutionary *RDPSO* without unconceivable nor unrealistic assumptions. The considered assumptions revolve essentially around human reasoning to obtain an estimation of the area of the scenario and the density of obstacles where the mission occurs. However, even despite the inconsistencies that may arise therefrom, the macroscopic model seems to generally achieve a good prediction of the final result, both regarding the number of robots within each state and, more importantly, the mission outcome (*e.g.*, explored area).

Next chapter brings this Ph.D. Thesis to an ending by summarizing its contributions and proposing future directions to solve its limitations.

Conclusion

NARROWING down all that was previously presented to a sentence, the focus of this Ph.D. Thesis was the bottom-up applicability of swarm robotic strategies into real-world operations, such as search and rescue (*SaR*). This final chapter summarizes the research covered in the previous chapters, regarding the research questions presented in chapter 1. After discussing the presented contributions, and considering their advantages and limitations, it points out perspectives on future research.

10.1 Fulfilment of the Objectives

In chapter 1, a set of objectives was listed considering the research questions that this work tries to answer through its several chapters (*cf.*, section 1.3). All those objectives were accomplished on top of the core objective of this Thesis that was described in chapter 4 – the *Robotic Darwinian Particle Swarm Optimization (RDPSO)*. This Multi-Robot System (*MRS*) approach adopts the concepts of the well-known Particle Swarm Optimization (*PSO*) (Kennedy & Eberhart, 1995) to real-world robotic applications. This was done by considering robots' dynamics using fractional calculus tools and communication in addition to obstacles constraints. Moreover, simple attraction and repulsion mechanisms, as well as social exclusion and inclusion concepts, are used to avoid stagnation and sub-optimality.

Although each chapter has a summary section that thoroughly discusses its accomplishments, let us summarize the level of achievement of each one of these objectives. Note that the key research question presented in section 1.2 is answered by means of several subsidiary questions enumerated in the same section.

10.1.1 Subsidiary Objective 1: Autonomous Deployment of Robots

The autonomous deployment of robots was addressed in section 5.1 and a novel approach was introduced and denoted as *Extended Spiral of Theodorus (EST)*. One of the main concerns in this approach was the efficiency in deploying exploring robots, therein denoted as scouts, using auxiliary platforms, denoted as rangers. This efficiency should not only ensure that scouts are scattered throughout the scenario as much as possible for a larger diversity of initial solutions, but also ensure that they can communicate between themselves so as to share such solutions. Taking a step further, section 5.1.3

generalized the approach to consider that each swarm corresponds to a fault-tolerant Mobile Ad hoc Network (*MANET*) with k -connectivity properties, in such a way that each scout is deployed while maintaining the connectivity with, at least, k other neighbours.

The approach was evaluated in section 5.3.1 using 3 rangers (*TraxBots* from section 3.2.2) and 15 scouts (*eSwarBots* from section 3.2.1). The *TraxBots* were adapted to transport a subgroup of 5 *eSwarBots* each and autonomously decide where to deploy each one of them in a large indoor scenario – a sports pavilion of 200 m^2 . The *EST* deployment strategy was compared with the typical state-of-the-art random initial deployment. Due to the small number of robots, a scalability evaluation was carried out in section 5.3.2 by benefiting from *MRSim* (section 3.2.4) under the same (virtualized) scenario. In both real and simulated experiments, the performance of the *EST* in terms of both distribution of robots and on maintaining the communication constraints was undoubtedly superior.

10.1.2 Subsidiary Objective 2: Communication in Faulty Environments

In line with the fault-tolerant deployment strategy, the fault-tolerance assessment of the *RDPSO* algorithm was addressed in section 5.2. Since creating a fault-tolerant *MANET* is only the beginning, an approach based on attractive and repulsive forces was proposed so as to ensure that the *MANET* remains k -connected throughout the mission, *i.e.*, that each scout is capable of directly communicating with, at least, k other neighbours at any one time. It is proved that each scout is able to achieve this level of decision-making without increasing the memory requirements (see section 5.2.3).

The standard *RDPSO* simply based on the mechanisms presented in chapter 4 is compared to this k -fault-tolerant *RDPSO* using a swarm of 15 scouts under the same real indoor scenario in which the deployment strategy was evaluated. It was possible to observe from the experimental results that a larger level of connectivity k jeopardizes the performance of the swarm. However, such handicap can be tackled by benefiting from a strategic initial deployment of scouts. For instance, one can also conclude from the experimental results that a biconnected *MANET* ($k = 2$) under the *EST* approach is able to achieve the same level of performance of a non-fault-tolerant *MANET* ($k = 1$) under a random initial distribution.

10.1.3 Subsidiary Objective 3: Efficient Sharing of Information

The *RDPSO* communication mechanisms are deeply studied in section 6.1 in which a set of heuristic rules is proposed so as to address the efficient inter-robot communication. The approach consisted of understanding the “*why*” behind a robot’s necessity to share information with its teammates and rendering the *RDPSO* as scalable as possible in terms of communication overhead. This was further

accomplished in section 6.2 by adapting the *Ad hoc On Demand (AODV)* reactive routing protocol considering the *RDPSO* dynamics, thus establishing more stable routes between robots and, as a consequence, reducing the overhead aggravated by *route discovery* messages. It is noteworthy that this last contribution increases the memory requirements of the exploring agents. This is a necessary limitation as each robot needs to remember neighbours' previous position to some extent so as to estimate their current position.

Benefiting from the same conditions as before (*i.e.*, same hardware, scenario and simulation experiments), this “optimized” version of the *RDPSO* with efficient communication management was compared with the common communication architecture inherent to its mechanisms presented in chapter 4. Experimental results evidenced that the advantages of a careful communication management manifest themselves more and more as the swarm size grows and robots advance in the mission. For instance, while an efficient communication results in the decrease of approximately 50% of the number of packets exchanged for a small swarm of 15 *eSwarBots* (section 6.3.1), it was able to decrease to approximately 80% for a swarm of 60 agents (section 6.3.2).

10.1.4 Subsidiary Objective 4: Adaptability to Dynamic Environments

A methodology to endow the swarm robots with adaptive mechanisms based on contextual information is addressed in chapter 7. Given the complexity of incorporating an adaptive mechanism in a stochastic parameterized algorithm such as the *RDPSO*, this was divided into two main components. The first part simply studies the convergence of the *RDPSO* by considering the system of difference equations (*DEs*) that mathematically model each robot's motion (*cf.*, section 7.1). By adopting concepts from stability theory, such as Jury-Marden's Theorem (Barnett, 1983), an attraction domain that illustrates the relationship between *RDPSO* parameters is defined. In the second part of the chapter (see sections 7.2 and 7.3), several context-based metrics are proposed and a rationale is given on how each one of them should influence robot's decision-making. To that end, and to support the theoretical concepts, the motions of two *eSwarBots* under different situations and with different sets of parameters are studied with a phase space analysis. This makes it possible to tune the *RDPSO* to each different context and foster the design of a fuzzy logic architecture (Zadeh, 1965) to systematically adjust robot's behaviour.

Given the complexity of this approach and to evaluate it under harsh conditions (large number of sub-optimal solutions and dynamic environments) and large scenarios ($A = 90000 \text{ m}^2$ and $A = 360000 \text{ m}^2$), this was achieved by means of numerical experiments. Experimental results clearly showed the advantages of an adaptive version of the *RDPSO* over a non-adaptive one. Not only it ensured a faster convergence of robots towards the optimal solution, but also presented high dynamic

source tracking capabilities. The heat maps from section 7.4.2 illustrate the capability of robots to track down the trajectory of optimal solutions. However, it also shows that this is achieved by increasing the energy consumption of robots, as their level of exploration is higher and, as a consequence, they spend more energy travelling.

10.1.5 Subsidiary Objective 5: Estimation of a Provable Convergence

The last objective of this Thesis in which the *RDPSO* is analytically studied at a macroscopic level is addressed in chapter 9. To that end, mathematical tools such as Semi-Markov chains and the *Z-Transform* were adopted to provably study the convergence of the *RDPSO* under given situations. Each social status was first evaluated separately to predict the number of robots within each different state under a certain task (*e.g.*, avoiding obstacles) on the steady-state regime. Afterwards, this mathematical methodology, reinforced side-by-side with exhaustive computational experiments, shaped an evolutionary non-linear *RDPSO* macroscopic semi-Markov model. This allows for an estimation of the performance of robotic swarms to be made during the course of a given mission including dealing with physical constraints posed by real world scenarios, regarding robots' dynamics, obstacles and communication constraints.

Besides being successfully evaluated in a collective mapping task under simulation, the outcome of the macroscopic model was compared to a source localization task using 15 *eSwarBots* in a 200 m^2 scenario. Although the model was not successful on synthesizing robots' outcome in the first two minutes of the mission, the differences between estimation and reality were mitigated as robots advanced into the mission. As a result, one can predict the *RDPSO* outcome under certain situations with minimal initial requirements, without ever running exhaustive simulations or real tests that are, in most situations, infeasible to consider.

10.2 Main Contributions and Achievements

The main scientific contributions of the thesis are described in the following sections.

10.2.1 New Platforms and Tools

Starting by the lowest level, this Thesis contributes to the development of a new robotic platform ideal for swarm applications due to its miniaturization and low-cost, denoted as Educational Swarm Robot (*eSwarBot*). Although the platform was not fully described in section 3.2.1, the paper of Couceiro *et al.* (Couceiro M. S., Figueiredo, Luz, Ferreira, & Rocha, 2011) presents the development

and evaluation of the *eSwarBot* considering odometry limitation, sensing capabilities, battery autonomy and explicit wireless communication mechanism.

Still around robotic platforms, we proposed the design of a conveyor kit for the mobile robot *TraxBot* (section 3.2.2). Once again, the development of the platform is better explained in Couceiro *et al.* (Couceiro M. S., Figueiredo, Portugal, Rocha, & Ferreira, 2012). The conveyor kit is easily adapted to the *TraxBot* platform, or any other platform with similar dimensions, and is capable of carrying 5 *eSwarBots*, or any other small-sized robots as long as the maximum overall weight is inferior to 4.5 *Kg*. The high accuracy and precision, as well as the small incremental energy consumptions, of the conveyor kit are also worth mentioning as a unique step motor controls the driving pulley with a step resolution inferior to 1 *mm*.

In terms of simulation tools, two major activities were carried out throughout this work. The first one was the design of a multi-robot simulator for *MatLab*, denoted as *MRSim* (see section 3.2.4). This is, so far, the only multi-robot simulator available for *MatLab* with several real-world phenomena, such as radio frequency propagation. Although it does not possess portability capabilities such as *ROS* or *Webots*, it is inspired by agent-based programming wherein each agent (robot) has its own algorithm, in a distributed manner, to the similarity of physical platforms.

The second major contribution regarding multi-robot simulation was the extension of *Webots* simulator and *e-pucks*' robotic platforms cross-compatibility (section 3.2.5). By benefitting from the developed *APIs*, *Webots* users can access all previously inaccessible features from the *e-puck*, such as the microphone, the speakers, the inter-robot *Bluetooth* communication and, most importantly, the *WiFi* communication from *Linux Gumstix Overo turret*. Besides that, several new tools that were already included in *Webots* simulator but ignored during cross-compilation, were integrated, such as the *printf* function for debugging and the access to a unique robot identifier depending on its hardware.

For a better understanding around the development and features of those new robotic platforms and simulation tools, please refer to the Technical Reports section at the end of this document.

10.2.2 Applicability of Mathematical Concepts

This Thesis, and all its several phases, make use of mathematical formulations and tools, not only for evaluation purposes, but for the development of new concepts and methodologies. Many of the algorithms and methodologies herein presented, on the other hand, share the same mathematical foundation around discrete-time analysis. Furthermore, there is a serious amount of theoretical rationale provided to build the bridge between mathematical theory and its applicability to real-world robotic strategies.

Due to such complexity, most of the mathematical formalism was centralized in section 3.1 from chapter 3. From *fractional calculus* to *fuzzy logic*, and all the way to *Semi-Markov chains*, this work provides rules and strategies on how to apply these for the design of real-world swarm robotic systems or any other *MRS* domain for that matter.

Fractional calculus was introduced, for the first time, as a way to consider robot's motion in which the intrinsic memory properties from a given trajectory are considered. Note that although this was applied here to explain the trajectory of a robot as an adjusting inertial factor, the same concepts were recently adopted in the context of the football game to estimate the position of a players based on his trajectory (Couceiro, Clemente, & Martins, 2013).

Jury-Marden's Theorem in stability theory (Barnett, 1983) was used, for the first time, to study the convergence of a swarm robotic algorithm. Although it was applied to the *RDPSO* case study, its applicability is not restricted to it since all parameterized method that can be described by *DE* may benefit from a similar analysis. This provides a significantly important step to limiting the range wherein parameters may be defined to an *attraction domain* without resorting arbitrary values.

Fuzzy logic was used as a decision-making mechanism to systematically adapt the *RDPSO* parameters within the mathematically retrieved *attraction domain*. Using fuzzy logic to extend a given algorithm or method is not new. From classical control theory to improve the efficiency of proportional-integral-derivative (*PID*) controllers (Couceiro M. S., Figueiredo, Ferreira, & Machado, 2010) to decision-making applied as a warning system against zombies (Couceiro M. S., Figueiredo, Luz, & Delorme, 2014 (In Press)), fuzzy logic applicability as an auxiliary tool that has been improving over the years. However, to the authors' knowledge, this is the first time it was used to adapt a swarm robotic algorithm to contextual information. Due to the low computational complexity of fuzzy logic strategies and the positive results retrieved so far, the same architecture may be reproduced to fit any other method, in robotics or not, that requires a systematic adaptive mechanism.

Semi-Markov chains, together with *Z-Transform* concepts, were considered to analytically describe the *RDPSO*. A similar, yet simplistic, methodology was considered by Agassounon *et al.* (Agassounon, Martinoli, & Easton, 2004) without considering real-world constraints (*e.g.*, obstacles) and evolutionary properties. Given the satisfactory results from this Thesis, and considering that the *RDPSO* is stochastic, evolutionary and hardly predictable, the methodology herein proposed can be adopted to almost any system, discrete or not, that may be represented by a probabilistic finite state machine (*PFSM*).

10.2.3 Benchmark on Swarm Robotics

Although benchmarking was not an explicit objective of this Thesis, its implementation was inevitable given the novelty of the work. Chapter 8 compared the herein proposed *RDPSO* with four other state-of-the-art swarm robotic methods under the same features and assumptions (section 1.1). Even though the list of algorithms was far from being exhaustive, it was the first step of proposing a comparison strategy for swarm algorithms applied to real-world situations. As a result, we focused on features such as communication, computational and memory complexity, as well as fault-tolerant strategies and the capability to track multiple and dynamic sources.

As an additional benefit, the *RDPSO* was proclaimed the best performing algorithm under the real and simulation experiments conducted. However, this does not go off without a hitch. From the five algorithms, the *RDPSO* was also the one presenting a larger computational and memory complexity. In spite of this, a rationale was carefully presented and alternatives, such as the *Glowworm Swarm Optimization (GSO)* from (Krishnanand & Ghose, 2009a), were proposed depending on the limitations of the design architecture.

10.3 Prospects

In this Thesis, we have contributed to the state-of-the-art on swarm robotics for real applications by developing novel algorithms and approaches via numerical and analytical mathematical solutions. However, despite the main contributions, this Thesis also has some limitations. In the following sections, and in spite of these limitations, we briefly discuss possible extensions of the presented work and future lines of research.

10.3.1 Future Work

The most obvious limitation is concerned with the particularization of all strategies to planar problems, *i.e.*, the explored physical space is bidimensional. Although this is a common particularization in most works presented in the literature (Agassounon, Martinoli, & Easton, 2004; Hereford & Siebold, 2008; Marques, Nunes, & Almeida, 2006), some more recent studies have been exploring the capabilities of swarm robotics in three-dimensional spaces by using swarms of flying or underwater robots (Lee, Nishimura, Tatara, & Chong, 2010; Stirling, Roberts, Zufferey, & Floreano, 2012).

In most situations, generalizing the problem to a three-dimensional space does not represent a complex challenge. However, considering the work presented in this Thesis, this could yield to the

complete remodelling of certain strategies, the most evident being the initial deployment of exploring robots.

As described in chapter 5, the initial deployment proposed is a geometric method inspired by the *Spiral of Theodorus* – a typical two-dimensional spiral. In geometry, when moving from two-dimensional space to three-dimensional, a spiral is often represented as *conical spiral* or *conical helix*. One of the most typical conical spirals is based on a generalization of the *Archimedean Spiral* (Chen, Yang, Guo, & Xu, 2008). On the other hand, and since the *Spiral of Theodorus* may be seen as a discretization of *Archimedean Spiral* composed of contiguous right triangles, the same discretization could be accomplished in the three-dimensional space, thus considering communication constraints (Figure 10.1a). Alternatively to a conic spiral, one could benefit from the geometry of spherical spirals described by a *rhumb line*. This is a common concept used for navigation purposes in which a *rhumb line* (aka, *loxodrome*) crosses all meridians of longitude at the same angle (Figure 10.1b). The effect of a *rhumb line* was studied for the first time by the Portuguese mathematician from the *University of Coimbra* Pedro Nunes³⁴ in 1537.

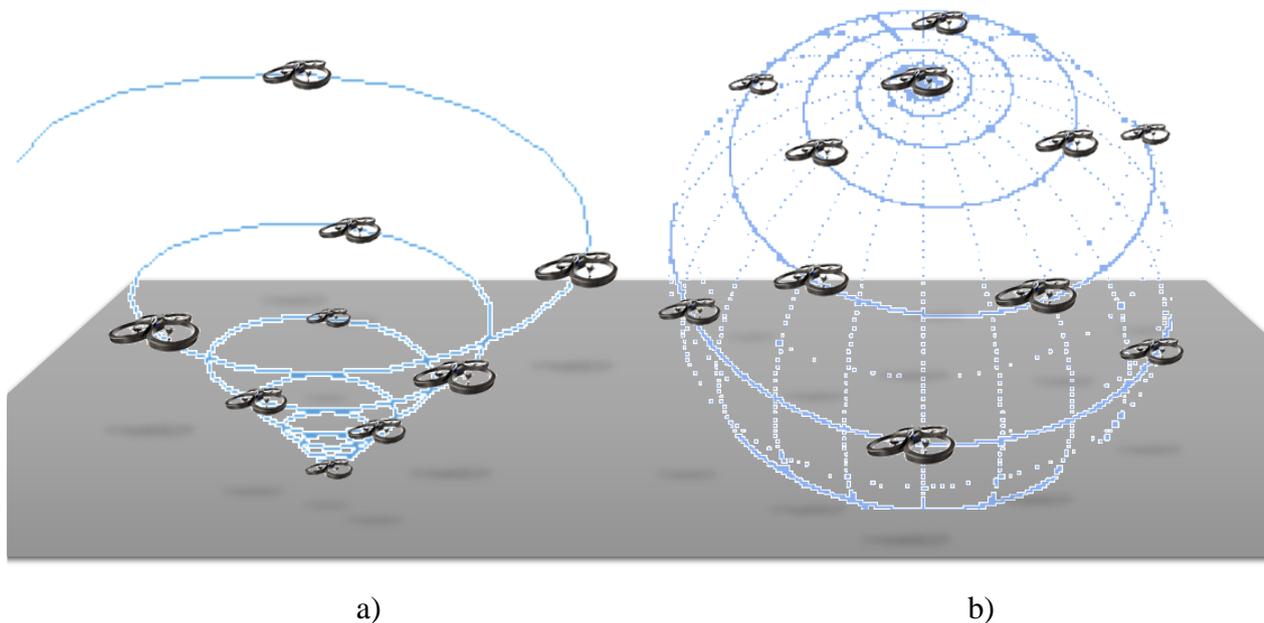


Figure 10.1. Examples of spiral-based three-dimensional deployment of UAVs. a) *conical spiral*; b) *rhumb line*.

Apart from the initial deployment, the three-dimensional generalization would also need to be considered throughout the remaining strategies that depend on bidimensional measures, such as the

³⁴ <http://www-history.mcs.st-and.ac.uk/Biographies/Nunes.html>

area of the scenario or the sensing circle around robots. Despite the fact that those are minor adjustments that may be easily accomplished, addressing more carefully the dimensionality problem would be an important extension of the contributions proposed herein.

A first step in that direction has been made by two BSc *Biomedical Engineering* students from *ISEC-IPC* during their honours project under our supervision. The undergraduate students implemented a simplistic version of the *RDPSO* algorithm in snake-like robots to navigate in an aquatic three-dimensional environment in *V-REP* (Figure 10.2). The main idea consisted on designing a benchmark environment representative of the inside of the human body to evaluate swarm nanorobotic strategies for cancer-fighting or other oncological diseases (Al-Hudhud, 2012). A video of those preliminary experiments is provided to present a general overview of a simplified version of the *RDPSO* performance in a three-dimensional environment³⁵.

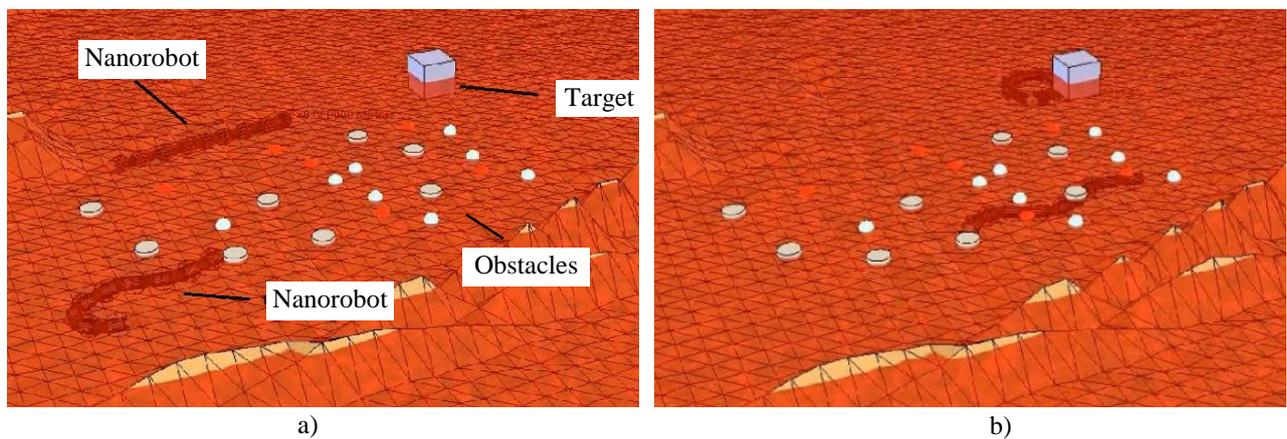


Figure 10.2. Illustrative sequence of a swarm of two snake-like nanorobots finding a target (*e.g.*, cancer cell) using a simplified version of the *RDPSO* algorithm. a) Mission start; b) One of the nanorobots is able to find its target while the other is still exploring.

The second major limitation of this work is that all of the presented strategies consider swarms of homogeneous agents. Robotics has been traditionally used to function either as singular agents for a specific task or within homogeneous teams in which agents share the same hardware and, as a consequence, the same communication protocols, range of abilities and programming environment, among others (Parker L. E., 1998). However, over the past few years, the research on *MRS* has been conducted towards the common goal of enabling different robotic platforms to cooperatively execute tasks. From swarm robotics (Dorigo, et al., 2011), passing through *marsupial* systems (Couceiro M.

³⁵ <http://www2.isr.uc.pt/~michaelcouceiro/media/NanoRobots.wmv>

S., Figueiredo, Portugal, Rocha, & Ferreira, 2012), and all the way to social robotics (Hristoskova, Agüero, Veloso, & Turck, 2013), heterogeneous teams have been significantly increasing the scope of applicability of robotics in real-world applications.

Note that, although the initial deployment strategy herein presented is an exception to this limitation, it only uses heterogeneous teams during the deployment phase. However, the same support platforms (denoted as rangers) that transport and deploy the swarm robots (previously identified as scouts) could be used during the mission to, for instance, access remote locations (Dellaert, et al., 2002) or provide power to the exploring robots (Murphy, Ausmus, Bugajska, & Ellis, 1999; Murphy, 2000). In spite of this, different platforms could be used for different purposes. Nevertheless, heterogeneous robot teams require a higher level of coordination since not all tasks can be performed by all team members. Therefore, such heterogeneous architectures require a proper mapping of each robot subtasks based on its own capabilities and performance. Such consideration gives rise to many complications that must be addressed by means of explicit communication so as to achieve the desirable level of cooperation within the *MRS* (Parker L. E., 1998). On the other hand, this would significantly affect the whole *RDPSO* design.

One way to generalize the *RDPSO* with heterogeneous capabilities, at a first stage, would be to divide the main task into several subtasks – one for each different subgroup of homogeneous robots. This is not a new strategy in *MRS* even though it has not been fully explored on swarm robotics. One of the few works on task partitioning in swarm robotics using explicit communication was recently presented in a technical report by Pini *et al.* (Pini, Brutschy, Scheidler, Dorigo, & Birattari, 2012). Similarly to this work, the future line of work would need to reside in the decomposition of a main task into several subtasks that can be tackled by different subgroups of robots. This decomposition not only facilitates the exploitation of heterogeneous teams but also results in an increased overall efficiency. For instance, in nature, large workers of the leaf-cutting ant *Atta laevigata* climb trees and cut leaves at the petioles that are then dropped to the ground, where smaller individuals cut them in pieces and transport them to the nest. As the petioles are harder to cut than the leaves, leaves harvesting is usually performed by larger individuals (Vasconcelos & Cherrett, 1996). Therefore, as many other insects in nature, partitioning the main task (*e.g.*, collection of leaves) into smaller subtasks (*e.g.*, gathering and clustering of leaves) removes the need to repeatedly climb the tree, resulting in an increased energy efficiency (Ratnieks & Anderson, 1999). Note that although this kind of gathering and clustering is a typical and well-explored collective aggregation in swarm robotics (Agassounon, Martinoli, & Easton, 2004), it has been exclusively focused on homogeneous teams of robots.

The previous example suggests that the division of the *RDPSO* swarm, besides considering the performance of individuals and their subgroup, should also consider their type. As an example, in a *SaR* mission, the main task, as the acronym suggests, is to search and rescue victims. The swarm, comprising of small *UAVs* and robust *UGVs*, could be initially divided into multiple dynamic subgroups of exploring agents, regardless of their type, by relying upon the “punish-reward” rules of social exclusion and inclusion (section 4.5). As the swarm finds victims, subgroups of adequately fitted agents would be created uniquely based on their type. In other words, while some subgroups would still comprise both types of robots to search for victims, other subgroups would be exclusively formed by *UGVs* to transport them.

As one may agree, by setting up a team of cooperating heterogeneous robots, it would be possible to significantly increase the range of applications and degrees of freedom. However, the difficulties in creating these teams arise from the fact that most developed robots are not designed to be cross-platform compatible. Previous projects, such as *Swarmanoid* (Dorigo, et al., 2011), involved producing a variety of robots that were designed and built to work together, thus inevitably leading to a high production cost. Along the several benefits of adapting robots that were not designed or built for cooperation purposes is that it eliminates the need for producing new units. Also, the implications of making completely independent robots to communicate with each other are much wider spreading. The ability to expand the robot team with off-the-shelf robots and, thereby, expand the team’s skill base would be extremely convenient.

To address this challenge, a preliminary work was carried out in the context of this Thesis at the *Robotics Lab (MACS-HWU)*. The work consisted of the development of a heterogeneous robot team comprising of *e-pucks* (small *UGVs*), *Aldebaran NAOs* (humanoid robots) and *Parrot AR. Drones 2.0 (UAVs)*. Despite the lack of documentation and solutions available by the community, it was possible to create a *MANET* in which all robots (nodes of the *MANET*) were able to explicitly communicate between themselves, passing data and instructions via *WiFi*, so as to coordinate their actions and cooperatively fulfil the necessary tasks. This pioneering work, despite still in undergoing development, had an admirable recognition from *MACS-HWU* community and other associated laboratories. A video of those preliminary experiments is provided in which the *NAO* robot controls an *AR. Drone* through ad hoc communication³⁶.

In brief, the generalization of the methodologies presented in this Thesis to support three-dimensional applications as well as heterogeneous swarms is the main short-term future line of research.

³⁶ <http://www2.isr.uc.pt/~michaelcouceiro/media/heterogeneousRobotTeams.mp4>

Next section goes a step further into proposing a long-term prospect, yet already starting to be discussed in the literature.

10.3.2 Further Expectations

This section presents a further insight to the future of this project and the applicability of swarm robotics to real-world situations in general. It is the author's perspective that was put together based on the experience acquired while working with other researchers from different domains throughout his Ph.D. studies and must be addressed as a futuristic (or not so much) vision at this point.

Going deeper into the heterogeneity of teams, researchers at the *MRL* from *ISR-UC* and at the *Robotics Lab (MACS-HWU)* have been focusing their efforts around *Human-Machine Interface (HMI)* (Enz, Diruf, Spielhagen, Zoll, & Vargas, 2011; Prado, Simplício, Lori, & Dias, 2012). *HMI* is a driving research field for the successful application of a variety of robotic technologies to further understand the capabilities of robots and how they execute their tasks (Rasmussen, 1986). Moreover, it fosters the cooperation between humans and robots to achieve ever improving solutions. Although merging the concepts from *HMI* with *MRS*, more specifically, swarm robotics, has not yet received the proper attention, some studies are starting to identify such union as an important research topic for producing responsive and robust systems for complex real-world problems (Steinberg, 2011).

As previously stated in this Thesis, at the highest level we could consider cooperative robots as a non-biological society (section 2.1.2). If so, this innovative vision would represent the cooperation between both robotic society and human society towards the same goal – this concept of inter-society cooperation is, once again as many other concepts, not new in nature, known as *protocooperation*. According to *Merriam-Webster* dictionary, *protocooperation* is the “*automatic or involuntary interaction by different kinds of organisms through which they mutually benefit*”. This form of mutualism, although not utterly necessary for the survival of a given society, has been observed within several societies in nature. For instance, hermit crabs often pick up a sea anemone to attach to their shell forming a typical prey and predator relationship (Figure 10.3a). While the sea anemone spreads out long stinging threads over the hermit crab to protect it from large predators, the hermit crab does the work of capturing food and leaving leftovers for the sea anemone. Another example of *protocooperation* can be seen between ants and aphids. Ants usually stimulate aphids to secrete honeydew straight into their mouths for the exchange of protection from natural predators (Figure 10.3b). Several mammals such as buffalos, antelopes, giraffes, and rhinos also benefit from *protocooperating* behaviours observed with Egyptian plover birds (Figure 10.3c). At the same time those birds remove insect pests from the backs of the mammals, consequently increasing their longevity, they benefit from an unlimited source of food. Another interesting example of *protocooperation* is observed between dolphins

and our own society (Figure 10.3d). Without any training, certain bottlenose dolphins in Brazil round up the fish and alert fishermen with signals as to when and where nets should be thrown.

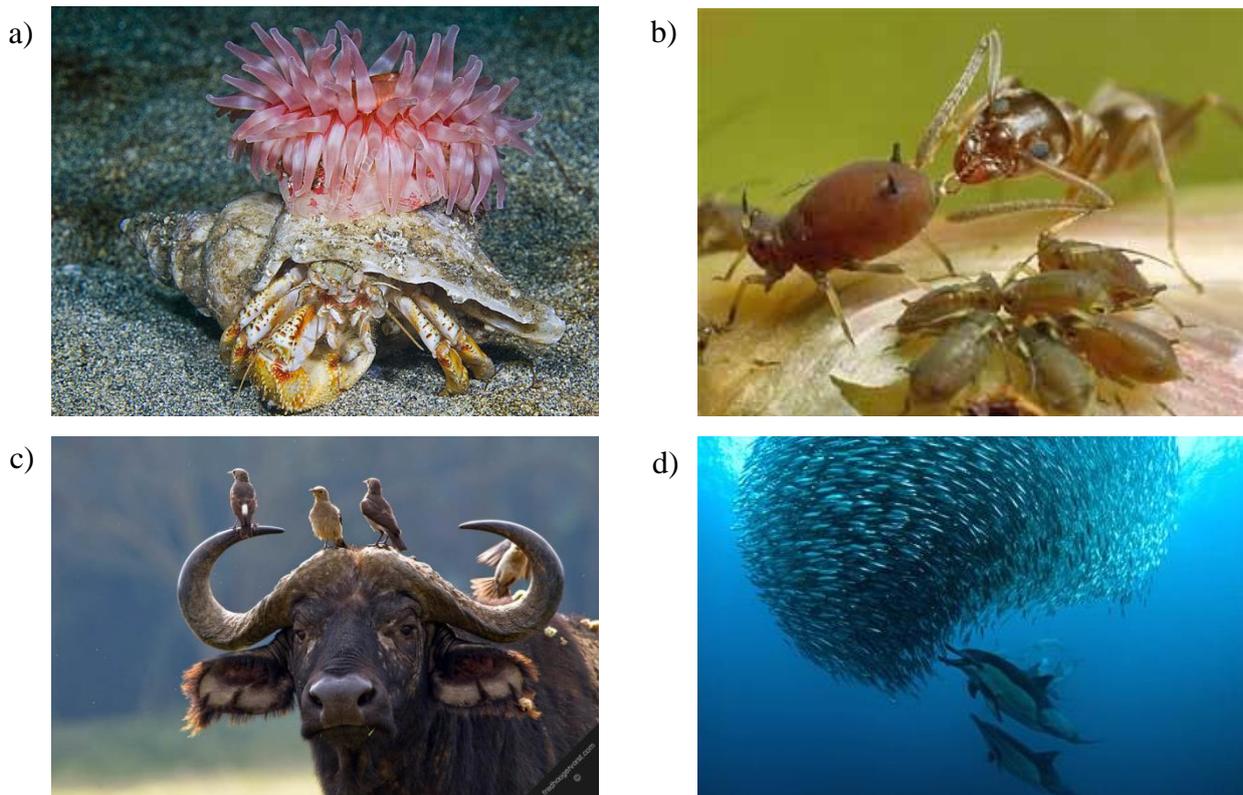


Figure 10.3. *Protocooperation* in nature: a) Hermit crab and sea anemone; b) Ants and aphids; c) Buffalo and Egyptian plover birds; d) Humans and dolphins.

Despite the advantages of this *protocooperation* mechanism, its applicability to robotics is far from being straightforward. In fact, most of the inter-society cooperation previously described is still not fully reported in the literature. For instance, scientists are not certain if some inter-society communications are mechanical or chemical³⁷. As a result, the best way to promote the communication between humans and robots is still far from being completely solved (Klingspor, Demiris, & Kaiser, 1997). Although some recent works already proposed preliminary strategies for multi-robot multi-human communication, there is still the need for more experimentation to explore the real-world unpredictability and complexity (Briggs & Scheutz, 2012). The few works focused around this topic evaluated their approaches in simulation or laboratorial scenarios without any constraints (*e.g.*, noise)

³⁷ <http://animals.pawnation.com/relationship-between-hermit-crabs-sea-anemones-1857.html>

and with only two or three agents (*e.g.*, two robots and one human). Furthermore, most existing approaches merging *HMI* with *MRS* use centralized modes of influence such as broadcasting parameter changes to all agents or implementing design-time consensus algorithms to support very specific collective behaviour (McLurkin, et al., 2006). However, when adding human influence in teams of swarm robots, it is imperative that both decentralized implementation and distributed communication described at the beginning of this Thesis (section 1.1) are preserved (Goodrich, Pendleton, Kerman, & Sujit, 2012).

Within the *CHOPIN R&D* project, of which the author is a research team member, one of the main goals consists of proposing architectures and models for cooperation in teams of humans and in teams of mobile robots (*cf.*, section 2.2.2). The motivation relies on the fact that teams of autonomous and cooperative mobile robots can provide human teams with an extension of sensing, inference and actuation capabilities in hazardous areas where human activity should be avoided (*e.g.*, incident response zones, contaminated areas, nuclear facilities decommissioning, among others). Although the work does not solely focus on *HMI*, such *protooperative* heterogeneous architectures would require, at the very end, a multi-robot multi-human interface to ensure cooperation and coordination among teams. This interface would provide humans with an improved situation awareness by benefiting from context (Baldauf, Dustdar, & Rosenberg, 2007), context awareness (Gellersen, Schmidt, & Beigl, 2002) and collaborative context awareness (Salkham, Cunningham, Senart, & Cahill, 2006).

For instance, let us suppose the deployment of a *protooperative* heterogeneous human-robot team in a *SaR* mission comprising of humanoid (legged) robots, *UGVs* and *UAVs* as illustrated in Figure 10.4. The different levels of mobility of the several platforms allow us to achieve a higher degree of freedom and, consequently, to significantly increase the success of the mission: while the humanoid robots can walk over the debris identified by the large number of low-cost *UGVs* during their swarm exploration, or even directly interact with first responders through *HMI* (right part of the image), the *UAVs* can significantly increase the coverage area of the rescuing operation and reach high places such as buildings (left part of the image). Nonetheless, for cooperation to emerge, a common wireless communication medium (*e.g.*, *WiFi* technology) between all agents is required. Moreover, such medium cannot rely on a pre-existent communication infrastructure since it may be inexistent under such scenarios. In other words, all agents, both robots and humans, need to act simultaneously as end nodes and routers of a *MANETs*.

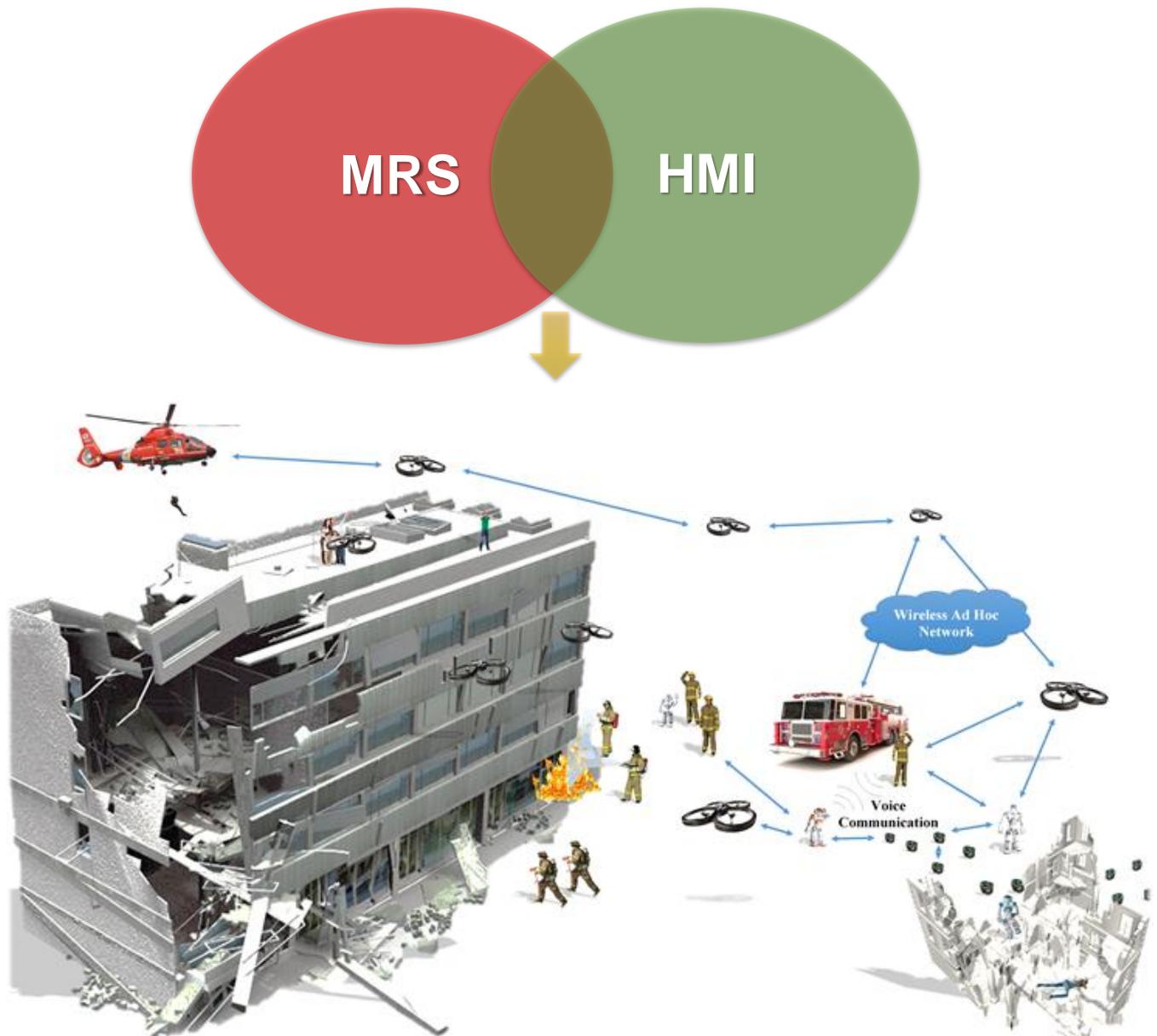


Figure 10.4. An example application of using a heterogeneous robot team to cooperatively fulfil a *SaR* mission with first responders. Small low-cost robots represented by the *e-pucks* could be used as scouts to explore the environment by benefiting from swarm robotic algorithms such as the *RDPSO* herein proposed. Humanoid robots represented by the *NAO* platforms could be used to walk over debris due to their higher mobility over the wheeled platforms and to build a bridge between human first responders and the robotic agents by benefiting from *HMI* algorithms (Casper & Murphy, 2003). The *UAVs* represented by the *AR.Drone* quadcopters could be used to significantly increase the coverage area of the rescuing operation and reach places that *UGVs* cannot (*e.g.*, buildings), thus increasing the mission success (Julian, Angermann, Schwager, & Rus, 2012). Both *UAVs* and humanoid robots could deploy the smaller scouts following the same principles previously addressed in this Thesis.

References and Bibliography

- Abbasi, A., Hossain, L., Hamra, J., & Owen, C. (2010). Social Networks Perspective of Firefighters' Adaptive Behaviour and Coordination among them. *2010 IEEE/ACM International Conference on Green Computing and Communications & 2010 IEEE/ACM International Conference on Cyber, Physical and Social Computing*, (pp. 819-824). Hangzhou, China.
- Abd-El-Wahed, W. F., Mousa, A. A., & El-Shorbagy, M. A. (2011). Integrating particle swarm optimization with genetic algorithms for solving nonlinear optimization problems. *Journal of Computational and Applied Mathematics*, 235(5), 1446-1453.
- Abedi, O., Fathy, M., & Taghiloo, J. (2008). Enhancing AODV routing protocol using mobility parameters in VANET. *IEEE/ACS International Conference on Computer Systems and Applications, AICCSA2008*, (pp. 229-235). Doha, Qatar.
- Abichandani, P., Benson, H. Y., & Kam, M. (2011). Constraints, Decentralized Multi-Vehicle Path Coordination under Communication. *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS2011*, (pp. 2306-2313). San Francisco, CA, USA.
- Agassounon, W., Martinoli, A., & Easton, K. (2004). Macroscopic Modeling of Aggregation Experiments using Embodied Agents in Teams of Constant and Time-Varying Sizes. *Autonomous Robots*, 17(2-3), 163-191.
- Al-Hudhud, G. (2012). On Swarming Medical Nanorobots. *International Journal of Bio-Science and Bio-Technology*, 4(1), 75-90.
- Alrashidi, M. R., & El-Hawary, M. E. (2006). A Survey of Particle Swarm Optimization Applications in Power System Operations. *Electric Power Components and Systems*, 34(12), 1349-1357.
- Anderson, S. O., Simmons, R., & Goldberg, D. (2003). Maintaining line of sight communications networks between planetary rovers. *Proceedings of the 2003 IEEE/RSJ Intl. Conference on Intelligent Robots and Systems*, (pp. 2266-2272). Las Vegas, NV.
- Antonelli, G., Arrichiello, F., & Chiaverini, S. (2010). The NSB control: a behavior-based approach for multi-robot systems. *PALADYN Journal of Behavioral Robotics*, 1(1), 48-56.
- Aras, R., Dutech, A., & Charpillet, F. (2004). Stigmergy in Multi Agent Reinforcement Learning. *Loria, Inst. Nat. de Recherche en Inf. et Autom.*, (pp. 468-469). Nancy, France.
- Araújo, A. (2012). *ROSint - Integration of a mobile robot in ROS architecture*. M.Sc. Thesis, Institute of Systems and Robotics, Department of Electrical and Computer Engineering. Coimbra: University of Coimbra.
- Araújo, A., Portugal, D., Couceiro, M. S., Figueiredo, C., & Rocha, R. P. (2012). TRAXBOT: Assembling and Programming of a Mobile Robotic Platform. *Proceedings of the 4th International Conference on Agents and Artificial Intelligence (ICAART'2012)*, (pp. 301-304). Vilamoura, Algarve, Portugal.

- Arkin, R., & Diaz, J. (2002). Line-of-sight constrained exploration for reactive multiagent robotic teams. *AMC 7th International Workshop on Advanced Motion Control*, (pp. 455-461). Maribor, Slovenia.
- Arrichiello, F., Chiaverini, S., & Fossen, T. I. (2006). Formation control of marine surface vessels using the Null-Space-based Behavioral Control. In K. Y. Pettersen, T. Gravdahl, & H. Nijmeijer (Eds.), *Group Coordination and Cooperative Control* (pp. 1-19). Springer-Verlag's Lecture Notes in Control and Information Systems series.
- Arrichiello, F., Das, J., Heidarsson, H., Pereira, A., Chiaverini, S., & Sukhatme, G. S. (2009). Multi-robot collaboration with range-limited communication: Experiments with two underactuated ASVs. *Proceedings 2009 International Conference on Field and Service Robots*, (pp. 443-453). Cambridge, MA, USA.
- Arrichiello, F., Heidarsson, H., Chiaverini, S., & Sukhatme, G. S. (2010). Cooperative caging using autonomous aquatic surface vehicles. *IEEE International Conference on Robotics and Automation, ICRA2010*, (pp. 4763-4769). Anchorage, AK.
- Arvin, F., Samsudin, K., & Ramli, A. R. (2009). Development of a Miniature Robot For Swarm Robotic Application. *International Journal of Computer and Electrical Engineering*, 1(4), 436-442.
- Asama, H., Ozaki, K., Matsumoto, A., Ishida, Y., & Endo, I. (1992). Development of Task Assignment System Using Communication for Multiple Autonomous Robots. *Journal of Robotics and Mechatronics*, 2(4), 122-127.
- Asenov, H., & Hnatyshin, V. (2009). GPS-Enhanced AODV routing. In *Proceedings of the International Conference on Wireless Networks (ICWN'09)*. Las Vegas, NV, USA.
- Asimov, I. (1982). *The Complete Robots: I, Robot. Voyager*.
- Ayash, M., Mikki, M., & Kangbin, Y. (2012). Improved AODV Routing Protocol to Cope with High Overhead in High Mobility MANETs. *6th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS)*, (pp. 244-251). Palermo, Italy.
- Bagnall, B. (2007). *Maximum LEGO NXT: Building Robots with Java Brains*. Variant Press.
- Bahramgiri, M., Hajiaghayi, M., & Mirrokni, V. (2006). Fault-tolerant and 3-Dimensional Distributed Topology Control Algorithms in Wireless Multi-hop Networks. *Wireless Networks*, 12(2), 179-188.
- Balch, T., & Hybinette, M. (2000). Social Potentials for Scalable Multi-Robot Formations. *Proceedings of the 2000 IEEE International Conference on Robotics, ICRA*. San Francisco, CA, USA.
- Baldauf, M., Dustdar, S., & Rosenberg, F. (2007). A survey on context-aware systems. *International Journal of Ad Hoc and Ubiquitous Computation*, 2(4), 263-277.
- Barnett, S. (1983). *Polynomials and Linear Control Systems*. New York, USA: Marcel Dekker, Inc.
- Bartolini, N., Calamoneri, T., Fusco, E. G., Massini, A., & Silvestri, S. (2008). Snap and Spread: A Self-deployment Algorithm for Mobile Sensor Networks. *Proceedings of the 4th IEEE*

- international conference on Distributed Computing in Sensor Systems (DCOSS08)*, (pp. 451-456). Santorini Island, Greece.
- Beckers, R., Holland, O., & Deneubourg, J. (1994). From Local Actions to Global Tasks: Stigmergy and Collective Robotics. In R. Brooks (Ed.), *Proceedings of the 14th International Workshop on Synthesis and Simulation of Living Systems* (pp. 181-189). Cambridge, MA, USA: P. Maes (MIT Press).
- Beni, G. (2004). From swarm intelligence to swarm robotics. In E. Sahin, & W. M. Spears (Ed.), *Proceedings of the Swarm Robotics Workshop* (pp. 1-9). Heidelberg, Germany: Springer-Verlag.
- Benkoski, S. J., Monticino, M. G., & Weisinger, J. R. (1991). A Survey of the Search Theory Literature. *Naval Research Logistics*, 31(4), 469-494.
- Bertocchi, F., Bergamo, P., Mazzini, G., & Zorzi, M. (2003). Performance comparison of routing protocols for ad hoc networks. *IEEE Global Communications Conference (GLOBECOM)*, 2, pp. 1033-1037. San Francisco, CA, USA.
- Bhalchandra, P., Deshmukh, N., Lokhande, S., & Phulari, S. (2009). A Comprehensive Note on Complexity Issues in Sorting Algorithms. *Advances in Computational Research*, 1(2), 1-9.
- Bjerknes, J. D., & Winfield, A. (2010). On Fault Tolerance and Scalability of Swarm Robotic Systems. In *Proceedings of 10th International Symposium on Distributed Autonomous Robotic Systems (DARS2010)*, (pp. 431-444). Lausanne, Switzerland.
- Blackwell, T., & Bentley, P. (2002). Don't push me! Collision-avoiding swarms. *Proceedings of the IEEE Congress on Evolutionary Computation*, 2, pp. 1691-1696. Honolulu, HI, USA.
- Bonabeau, E., Dorigo, M., & Theraulaz, G. (1999). *Swarm Intelligence: From Natural to Artificial Systems*. New York: Oxford University Press.
- Bonani, M., Longchamp, V., Magnenat, S., Rétornaz, P., Burnier, D., Roulet, G., . . . Mondada, F. (2010). The MarXbot, a Miniature Mobile Robot Opening new Perspectives for the Collective-robotic Research. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2010)*, (pp. 4187-4193). Taipei, Taiwan.
- Borghoff, U. M., & Schlichter, J. H. (2000). *Computer-Supported Cooperative Work: Introduction to Distributed Applications*. USA: Springer.
- Braitenberg, V. (1984). *Vehicles. Experiments in synthetic psychology*, The MIT Press.
- Bräunl, T. (2008). *Embedded Robotics* (3rd ed.). Germany: Springer-Verlag.
- Briggs, G., & Scheutz, M. (2012). Multi-modal belief updates in multi-robot human-robot dialogue interaction. *Proceedings of the AISB/IACAP Symposium on Linguistic and Cognitive Approaches to Dialogue Agents (LaCATODA)*, (pp. 67-72). Birmingham, UK.
- Broch, J., Maltz, D. A., Johnson, D. B., Hu, Y.-C., & Jetcheva, J. (1998). A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols. *Proceedings of the 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom '98)*, (pp. 85-97). Dallas, TX, USA.

-
- Brüggemann, B., & Schulz, D. (2010). Coordinated Navigation of Multi-Robot Systems with Binary Constraints. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2010)*, (pp. 3854-3859). Taipei, Taiwan, China.
- Buhl, J., Sumpter, D. J., Couzin, I. D., Hale, J. J., Despland, E., Miller, E. R., & Simpson, S. J. (2006). From Disorder to Order in Marching Locusts. *Science*, *312*(5778), 1402-1406.
- Burchardt, T. (2000). Social exclusion: concepts and evidence. In D. G. P. Townsend (Ed.), *Breadline Europe: The measurement of poverty*. Bristol: The Policy.
- Calisi, D., Iocchi, L., Nardi, D., Randelli, G., & Zuparo, V. A. (2009). Improving Search and Rescue Using Contextual Information. *Journal on Advanced Robotics*, *23*(9), 1199-1216.
- Cao, Y., Fukunaga, A., & Kahng, A. (1997). Cooperative Mobile Robotics: Antecedents and Directions. *Autonomous Robots*, *4*(1), 1-23.
- Carlisle, A., & Dozier, G. (2000). Adapting particle swarm optimization to dynamic environments. *Proceedings of the International Conference on Artificial Intelligence*, (pp. 429-433). Las Vegas, USA.
- Carpin, S., Lewis, M., Wang, J., Balakirsky, S., & Scrapper, C. (2007). USARSim: a robot simulator for research and education. *IEEE International Conference on Robotics and Automation (ICRA2007)*, (pp. 1400-1405). Roma, Italy.
- Casper, J., & Murphy, R. (2003). Human-robot interaction during the robot-assisted urban search and rescue response at the World Trade Center. *IEEE Transaction on Systems, Man, and Cybernetics*, *33*(3), 367-385.
- Casteigts, A., Albert, J., Chaumette, S., Nayak, A., & Stojmenovic, I. (2010). Biconnecting a Network of Mobile Robots using Virtual Angular Forces. *IEEE 72nd Vehicular Technology Conference Fall (VTC2010)*, (pp. 1-5). Ottawa, ON, USA.
- Chen, C., Yang, F., Guo, C., & Xu, J. (2008). Simulation of Monifilar Archimedean conical spiral antennas. *International Conference on Microwave and Millimeter Wave Technology (ICMMT08)*, (pp. 1038-1040). Nanjing, China.
- Chen, C.-H., & Chen, Y.-P. (2011). Convergence Time Analysis of Particle Swarm Optimization Based on Particle Interaction. *Advances in Artificial Intelligence*, *2011*(204750), 1-7.
- Cheriyian, J., Vempala, S., & Vetta, A. (2003). An approximation algorithm for the minimum-cost k-vertex connected subgraph. *SIAM Journal on Computing*, *32*(4), 1050-1055.
- Chia-Feng, J. (2004). A hybrid of genetic algorithm and particle swarm optimization for recurrent network design. *IEEE Trans. Syst., Man, Cybern., Part B: Cybern.*, *34*(2), 997-1006.
- Cianci, C. M., Raemy, X., Pugh, J., & Martinoli, A. (2006). Communication in a Swarm of Miniature Robots: The e-Puck as an Educational Tool for Swarm Robotics. In E. Sahin, W. M. Spears, & A. F. Winfield, *Proceedings of Simulation of Adaptive Behavior (SAB-2006)*, *Swarm Robotics Workshop* (pp. 103-115). Rome, Italy: Lecture Notes in Computer Science (LNCS) 4433.

- Clerc, M., & Kennedy, J. (2002). The particle swarm - explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation*, 6(1), 58-73.
- Colman, A. M. (1995). *Game Theory and its Applications in the Social and Biological Sciences* (Vol. 2nd Edition). Oxford: Butterworth-Heinemann.
- Cornejo, A., & Lynch, N. (2010). Fault-Tolerance Through k-Connectivity. *IEEE International Conference on Robotics and Automation (ICRA2010), Workshop on Network Science and System Issues in Multi-Robot Autonomy*. Anchorage, Alaska.
- Correll, N., Bachrach, J., Vickery, D., & Rus, D. (2009). Ad-hoc Wireless Network Coverage with Networked Robots that Cannot Localize. *IEEE International Conference on Robotics and Automation Kobe International Conference Center*, (pp. 3878-3885). Kobe, Japan.
- Couceiro, M. S., Clemente, F. M., & Martins, F. M. (2013). Analysis of football player's motion in view of fractional calculus. *Central European Journal of Physics*, 11(3), 1-15.
- Couceiro, M. S., Fernandes, A., Rocha, R. P., & Ferreira, N. M. (2013 (In Press)). Understanding the Communication Complexity of the Robotic Darwinian PSO. *Robotica Journal of Cambridge*.
- Couceiro, M. S., Ferreira, N. M., & Machado, J. T. (2011). Fractional Order Darwinian Particle Swarm Optimization. *3th Symposium on Fractional Signals and Systems, FSS'2011*, (pp. 127-136). Coimbra, Portugal.
- Couceiro, M. S., Ferreira, N. M., & Machado, J. T. (2012). Hybrid adaptive control of a dragonfly model. *Communications in Nonlinear Science and Numerical Simulation*, 17(2), 893-903.
- Couceiro, M. S., Figueiredo, C. M., Ferreira, N. M., & Machado, J. T. (2010). Electric Vehicle Drive System with Adaptive PID Control. *29th IASTED International Conference on Modelling, Identification and Control (MIC2010)*. Innsbruck, Austria.
- Couceiro, M. S., Figueiredo, C. M., Fonseca Ferreira, N. M., & Tenreiro Machado, J. A. (2009). Biological Inspired Flying Robot. *Proceedings of IDETC/CIE ASME International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*. San Diego, USA.
- Couceiro, M. S., Figueiredo, C. M., Luz, J. A., Ferreira, N. M., & Rocha, R. P. (2011). A Low-Cost Educational Platform for Swarm Robotics. *International Journal of Robots, Education and Art*, 2(1), 1-15.
- Couceiro, M. S., Figueiredo, C. M., Luz, M. A., & Delorme, M. J. (2014 (In Press)). Zombie Infection Warning System Based on Fuzzy Decision-Making. In R. Smith?, *Mathematical Modelling of Zombies*. Ottawa, Canada: University of Ottawa Press.
- Couceiro, M. S., Figueiredo, C. M., Portugal, D., Rocha, R. P., & Ferreira, N. M. (2012). Initial Deployment of a Robotic Team - A Hierarchical Approach Under Communication Constraints Verified on Low-Cost Platforms. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2012)*, (pp. 4614-4619). Vilamoura, Algarve, Portugal.

-
- Couceiro, M. S., Figueiredo, C. M., Rocha, R. P., & Ferreira, N. M. (2014 (In Press)). Darwinian Swarm Exploration under Communication Constraints: Initial Deployment and Fault-Tolerance Assessment. *Robotics and Autonomous Systems*.
- Couceiro, M. S., Luz, J. M., Figueiredo, C. M., & Ferreira, N. M. (2012). Modeling and Control of Biologically Inspired Flying Robots. *Robotica Journal of Cambridge*, 30(1), 107-121.
- Couceiro, M. S., Luz, J. M., Figueiredo, C. M., Ferreira, N. M., & Dias, G. (2010). Parameter Estimation for a Mathematical Model of the Golf Putting. In V. M. Marques, C. S. Pereira, & A. Madureira (Ed.), *Proceedings of WACI-Workshop Applications of Computational Intelligence* (pp. 1-8). Coimbra, Portugal: ISEC - IPC.
- Couceiro, M. S., Machado, J. T., Rocha, R. P., & Ferreira, N. M. (2012). A Fuzzified Systematic Adjustment of the Robotic Darwinian PSO. *Robotics and Autonomous Systems*, 60(12), 1625-1639.
- Couceiro, M. S., Martins, F. M., Rocha, R. P., & Ferreira, N. M. (2012). Introducing the Fractional Order Robotic Darwinian PSO. *Proceedings of the 9th International Conference on Mathematical Problems in Engineering, Aerospace and Sciences (ICNPAA2012)*, (pp. 242-251). Vienna, Austria.
- Couceiro, M. S., Martins, F. M., Rocha, R. P., & Ferreira, N. M. (2013a (Under Review)). Mechanism and Convergence Analysis of a Multi-Robot Swarm. *Journal of Intelligent & Robotic Systems*.
- Couceiro, M. S., Martins, F. M., Rocha, R. P., & Ferreira, N. M. (2013b (Under Review)). On a Macroscopic Model of an Evolutionary Swarm Robotics Exploration Approach. *IEEE Transactions on Evolutionary Computation*.
- Couceiro, M. S., Mendes, R., Fonseca Ferreira, N. M., & Tenreiro Machado, J. A. (2009). Control Optimization of a Robotic Bird. *European Workshop on Movement Science (EWOMS'09)*. Lisbon, Portugal.
- Couceiro, M. S., Portugal, D., & Rocha, R. P. (2013). A Collective Robotic Architecture in Search and Rescue Scenarios. *Proceedings of the 28th Symposium On Applied Computing (SAC2013)*, (pp. 64-69). Coimbra, Portugal.
- Couceiro, M. S., Portugal, D., Gonçalves, N., Rocha, R., Luz, J. M., Figueiredo, C. M., & Dias, G. (2013). A Methodology for Detection and Estimation in the Analysis of the Golf Putting. *Pattern Analysis and Applications*, 16(3), 459-474.
- Couceiro, M. S., Rocha, R. P., & Ferreira, N. M. (2011a). A Novel Multi-Robot Exploration Approach based on Particle Swarm Optimization Algorithms. *Proceedings of the IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR2011)*, (pp. 327-332). Kyoto, Japan.
- Couceiro, M. S., Rocha, R. P., & Ferreira, N. M. (2011b). Ensuring Ad Hoc Connectivity in Distributed Search with Robotic Darwinian Swarms. *Proceedings of the IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR2011)*, (pp. 284-289). Kyoto, Japan.
- Couceiro, M. S., Rocha, R. P., & Ferreira, N. M. (2013a). A PSO Multi-Robot Exploration Approach Over Unreliable MANETs. *Advanced Robotics*, 27(16), 1-14.

- Couceiro, M. S., Rocha, R. P., & Ferreira, N. M. (2013b). Fault-Tolerance Assessment of a Darwinian Swarm Exploration Algorithm under Communication Constraints. *Proceedings of IEEE International Conference on Robotics and Automation (ICRA2013)*, (pp. 2000-2005). Karlsruhe, Germany.
- Couceiro, M. S., Rocha, R. P., Ferreira, N. M., & Vargas, P. A. (2013). Darwinian Robotic Swarms for Exploration with Minimal Communication. *IEEE Congress on Evolutionary Computation (CEC'2013), Special session on Evolutionary Robotics, 2013*. Cancún, México, USA.
- Couceiro, M. S., Rocha, R. P., Figueiredo, C. M., Luz, J. A., & Ferreira, N. M. (2012). Multi-Robot Foraging based on Darwin's Survival of the Fittest. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2012)*, (pp. 801-806). Vilamoura, Algarve, Portugal.
- Couceiro, M. S., Vargas, P. A., Rocha, R. P., & Ferreira, N. M. (2014). Benchmark of Swarm Robotic Distributed Techniques in a Search Task. *Robotics and Autonomous Systems*, 62(2), 200-213.
- Craighead, J. J., Sumner, J. S., & Mitchell, J. A. (1995). *The grizzly bears of Yellowstone: Their ecology in the Yellowstone ecosystem*. Washington, D.C., USA: Island Press.
- Crispin, Y. J. (2009). Cooperative Control of Multiple Swarms of Mobile Robots with Communication Constraints. In LNCIS, & M. Hirsch (Ed.), *Optimization and Cooperative Control* (pp. 207-220). Berlin, Heidelberg: Springer-Verlag.
- Cui, X., & Potok, T. E. (2007). Distributed Adaptive Particle Swarm Optimizer in Dynamic Environment. *IEEE International Parallel and Distributed Processing Symposium (IPDPS'07)*, (pp. 1-7). Long Beach, CA, USA.
- Cummins, J., Azhar, M. Q., & Sklar, E. (2008). Using Surveyor SRV-1 Robots to Motivate CS1 Student. *4th Artificial Intelligence for Interactive Digital Entertainment Conference, AIIDE'08*, (pp. 23-27). Stanford, USA.
- Darwin, C. (1872). *On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life*. London: Public Domain Books.
- Dean, M. (1913). *Book of Proverbs*. Catholic Encyclopedia. Adapted from Holman Bible Handbook on Proverbs.
- Dellaert, F., Balch, T., Kaess, M., Ravichandran, R., Alegre, F., Berhault, M., . . . Walker, D. (2002). The Georgia Tech yellow jackets: A marsupial team for urban search and rescue. *AAAI Mobile Ro-bot Competition Workshop*. Edmonton, Alberta, Canada.
- Demirbas, M. (2004). *Scalable Design of Fault-Tolerance for Wireless Sensor Networks*. Ohio State University. Columbus.
- Deneubourg, J., Goss, S., Sandini, G., Ferrari, F., & Dario, P. (1990). Self-organizing collection and transport of objects in unpredictable environments. *Japan-U.S.A. Symposium on Flexible Automation*, (pp. 1093-1098). Kyoto, Japan.
- Derbakova, A., Correll, N., & Rus, D. (2011). Decentralized Self-Repair to Maintain Connectivity and Coverage in Networked Multi-Robot Systems. *IEEE International Conference on Robotics and Automation (ICRA2011)*, (pp. 3863-3868). Shanghai, China.

- Doniec, A., Bouraqadi, N., Defoort, M., Le, V. T., & Stinckwich, S. (2009). Distributed constraint reasoning applied to multi-robot exploration. *IEEE International Conference on Tools with Artificial Intelligence (ICTAI2009)*, (pp. 159-166). Newark, NJ, USA.
- Dorigo, M., & Sahin, E. (2004). Swarm Robotics - Special Issue. *Autonomous Robots*, 17(2-3), 111-113.
- Dorigo, M., & Stützle, T. (2004). Ant Colony Optimization. *MIT Press, Cambridge*, 23(6), 815-815.
- Dorigo, M., Floreano, D., Gambardella, L. M., Mondada, F., Nolfi, S., Baaboura, T., . . . Foster, A. (2011). *Swarmanoid: a novel concept for the study of heterogenous robotic swarms*. Bruxelles, Belgium: IRIDIA.
- Dorigo, M., Trianni, V., Sahin, E., Gross, R., Labella, T. H., Baldassarre, G., . . . Gambardella, L. M. (2004). Evolving Self-Organizing Behaviors for a Swarm-bot. *Autonomous Robots, special Issue on Swarm Robotics*, 17(2-3), 223-245.
- Doroodgar, B., Ficocelli, M., Mobedi, B., & Nejat, G. (2010). The Search for Survivors: Cooperative Human-Robot Interaction in Search and Rescue Environments using Semi-Autonomous Robots. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA2010)*, (pp. 2858-2863). Anchorage, AK, USA.
- Downey, R. G., & Fellows, M. R. (1999). *Parameterized Complexity*. Springer.
- Dudek, G., Jenkin, M., Milios, E., & Wilkes, D. (1995). Experiments in sensing and communication for robot convoy navigation. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2003)*, (pp. 268-273). Las Vegas, Nevada, USA.
- Elaydi, S. (2005). *An Introduction to Difference Equations* (3rd ed.). Hong Kong, China: Springer, Science & Business Media, Inc.
- Ellis, C. A., Gibbs, S. J., & Rein, G. L. (1991). Groupware - Some Issues and Experiences. *Communications of the ACM*, 34(1), 38-58.
- Enz, S., Diruf, M., Spielhagen, C., Zoll, C., & Vargas, P. (2011). The Social Role of Robots in the Future - Explorative Measurement of Hopes & Fears. *International Journal of Social Robotics*, 3(3), 263-271.
- Ferreira, N. M. (2006). *Sistemas Dinâmicos e Controlo de Robôs Cooperantes*. PhD Thesis, University of Trás-os-Montes e Alto Douro.
- Ferworn, A., Hough, G., Manca, R., Antonishek, B., Scholtz, J., & Jacoff, A. (2006). Expedients for Marsupial Operations of USAR Robots. *Proceedings of the IEEE International Workshop on Safety, Security and Rescue Robotics (SSRR)*. Gaithersburg, MD, USA.
- Floreano, D., & Mattiussi, C. (2008). *Bio-Inspired Artificial Intelligence: Theories, Methods, and Technologies*. Cambridge, MA: MIT Press.
- Foster, K. R., & Xavier, J. B. (2007). Cooperation: Bridging Ecology and Sociobiology. *Current Biology*, 17, pp. 319-321.
- Franchi, A., Freda, L., Oriolo, G., & Vendittelli, M. (2009). The sensor-based random graph method for cooperative robot exploration. *IEEE/ASME Transactions on Mechatronics*, 14(2), 163-175 .

- Freese, M., Singh, S., Ozaki, F., & Matsuhira, N. (2010). Virtual Robot Experimentation Platform V-REP: A Versatile 3D Robot Simulator. In N. Ando, S. Balakirsky, T. Hemker, M. Reggiani, & O. v. Stryk, *2nd International Conference on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR'2010)* (pp. 51-62). Darmstadt, Germany: Lecture Notes in Computer Science, Springer.
- Friis, H. (1946). A note on a simple transmission formula. *Proceedings of the IRE*, 34(5), 254-256.
- Fuks, H., Raposo, A. B., Gerosa, M. A., & Pereira de Lucena, C. J. (2003). O Modelo de Colaboração 3C e a Engenharia de Groupware. *Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio)*.
- Gancet, J., Motard, E., Naghsh, A., Roast, C., Arancon, M. M., & Marques, L. (2010). User Interfaces for Human Robot Interactions with a Swarm of Robots in Support to Firefighters. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA2010)*, (pp. 2846-2851). Anchorage, AK, USA.
- Garibaldi, J. M., & Ifeachor, E. C. (1999). Application of simulated Annealing Fuzzy Model Tuning to Umbilical Cord Acid-base Interpretation. *IEEE Transactions on Fuzzy Systems*, 1(7), 72-84.
- Gazi, V., & Passino, K. M. (2003). Stability Analysis of Swarms. *IEEE Transactions on Automatic Control*, 48(4), 692-697.
- Gazi, V., & Passino, K. M. (2004). Stability Analysis of Social Foraging Swarms. *IEEE Transactions on Systems, Man and Cybernetics*, 34(1), 539-557.
- Gebali, F. (2008). *Analysis of Computer and Communication Networks*. Canada: Springer.
- Gellersen, H., Schmidt, A., & Beigl, M. (2002). Multi-sensor context awareness in mobile devices and smart artifacts. *Mobile Networks and Applications*, 7(5), 341-351.
- Gerkey, B. P., Vaughan, R. T., & Howard, A. (2003). The Player/Stage Project: Tools for Multi-Robot and Distributed Sensor Systems. *Proceedings of the International Conference on Advanced Robotics (ICAR2003)*, (pp. 317-323). Coimbra, Portugal.
- Ghamisi, P., Couceiro, M. S., Benediktsson, J. A., & Ferreira, N. M. (2012). An Efficient Method for Segmentation of Images Based on Fractional Calculus and Natural Selection. *Expert Systems with Applications, Elsevier*, 39(16), 12407-12417.
- Gleick, J. (1987). *Chaos: Making a New Science*. Viking Penguin.
- Goodrich, M. A., Pendleton, B., Kerman, S., & Sujit, P. B. (2012). What Types of Interactions do Bio-Inspired Robot Swarms and Flocks Afford a Human? *Proceedings of Robotics, Science and Systems (RSS)*, (pp. 105-112). Sydney, Australia.
- Groß, R., O'Grady, R., Christensen, A. L., & Dorigo, M. (2011). The Swarm-bot experience: Strength and mobility through physical cooperation. In S. Kernbach, & S. Kernbach (Ed.), *Handbook of Collective Robotics* (pp. 49-76). Pan Stanford Publishing.
- Guyot, L., Heiniger, N., Michel, O., & Rohrer, F. (2011). Teaching robotics with an open curriculum based on the e-puck robot, simulations and competitions. *Proceedings of 2nd International*

-
- Conference on Robotics in Education (RiE'2011)* (pp. 53-58). Vienna, Austria: INNOC - Austrian Society for Innovative Computer Sciences.
- Hahn, H. K., & Schoenberger, K. (2007). The Ordered Distribution of Natural Numbers on the Square Root Spiral. *The Journal of Business*, 1-35.
- Hattenberger, G., Lacroix, S., & Alami, R. (2010). Formation Flight: Evaluation of autonomous configuration control algorithms. *Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, (pp. 2628-2633). San Diego, CA, USA.
- Hauert, S., Winkler, L., Zufferey, J. C., & Floreano, D. (2008). Ant-based Swarming with Positionless Micro Air Vehicles for Communication Relay. *Swarm Intelligence*, 2(2-4), 167-188.
- Hauert, S., Zufferey, J. C., & Floreano, D. (2009). Evolved swarming without positioning information: an application in aerial communication relay. *Autonomous Robots*, 26(1), 21-32.
- Heliövaara, S. (2007). *Computational Models for Human Behavior in Fire Evacuations*. MSc Thesis, Helsinki University of Technology, Department of Engineering Physics and Mathematics, Helsinki, Finland.
- Hereford, J. M., & Siebold, M. A. (2010). Bio-Inspired Search Strategies for Robot Swarms. In E. M. Martin, *Swarm Robotics, From Biology to Robotics* (pp. 1-27).
- Hereford, J., & Siebold, M. (2008). Multi-robot search using a physically-embedded Particle Swarm Optimization. *International Journal of Computational Intelligence Research*, 4(2), 197-209.
- Herman, I. (2007). Sound, hearing and the human voice. In I. Herman, *Physics of the Human Body* (pp. 87-98). Springer.
- Hollinger, G., & Singh, S. (2010). Multi-Robot Coordination with Periodic Connectivity. *IEEE International Conference on Robotics and Automation, ICRA2010*, (pp. 4457-4462). Anchorage, Alaska, USA.
- Howard, A., Mataric, M. J., & Sukhatme, G. S. (2002b). Mobile sensor network deployment using potential fields: A distributed scalable solution to the area coverage problem. *Proceedings of the 6th International Symposium on Distributed Autonomous Robotic Systems, DARS02*, (pp. 299-308). Fukuoka, Japan.
- Howard, A., Mataric, M., & Sukhatme, G. (2002a). An Incremental Deployment Algorithm for Mobile Robot Teams. *Proceedings of the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems*, (pp. 2849-2854). Lausanne, Switzerland.
- Hristoskova, A., Agüero, C. E., Veloso, M., & Turck, F. D. (2013). Heterogeneous Context-Aware Robots Providing a Personalized Building Tour. *International Journal of Advanced Robotic Systems*, 10(14), 1-13.
- Hsieh, M. A., Kumar, R. V., Cowley, A., & Taylor, C. J. (2008). Maintaining network connectivity and performance in robot teams. *Special Issue on Search and Rescue Robots in the Journal of Field Robotics*, 25(1-2), 111-131.
- Huber, M. J., & Durfee, E. (1995). Deciding when to commit to action during observation-based coordination. *Proceedings of the First International Conference on Multi-Agent Systems*, (pp. 163-170).

- Ijspeert, A. J., Martinoli, A., Billard, A., & Gambardella, L. M. (2001). Collaboration through the Exploitation of Local Interactions in Autonomous Collective Robotics: The Stick Pulling Experiment. *Autonomous Robots*, 11(2), 149-171.
- Janssen, M., & Papanikolopoulos, N. (2007). Enabling Complex Behavior by Simulating Marsupial Actions. *Proceedings of the 15th Mediterranean Conference on Control & Automation*, (pp. 1-6). Athens, Greece.
- Jatmiko, W., Sekiyama, K., & Fukuda, T. (2006). Modified Particle Swarm Robotic Odor Source Localization in Dynamic Environments. *International Journal of Intelligent Control and Systems*, 11(2), 176-184.
- Jennings, N. (1995). Controlling Cooperative Problem Solving in Industrial Multi-agent Systems using Joint Intentions. *Artificial Intelligence*, 75(2), 195-240.
- Julian, B. J., Angermann, M., Schwager, M., & Rus, D. (2012). Distributed robotic sensor networks: An information-theoretic approach. *International Journal of Robotics Research (IJRR)*, 31(10), 1134-1154.
- Jung, D. (1998). *An Architecture for Cooperation among Autonomous Agents*. PhD Thesis, University of Wollongong, Dep. of Computer Science, Wollongong, Australia.
- Kadirkamanathan, V., Selvarajah, H., & Fleming, P. (2006). Stability analysis of the particle dynamics in particle swarm optimizer. *IEEE Transactions on Evolutionary Computation*, 10(3), 245-255.
- Kannan, S., Slochanal, S., & Padhy, N. (2004). Application of particle swarm optimization technique and its variants to generation expansion problem. *ELSERVIER Electric Power Syst. Res.*, 70(3), 203-210.
- Kazadi, S. (2003). Extension of Plume Tracking Behavior to Robot Swarms. *Proceedings of the 7th World Multi Conference on Systematics, Cybernetics and Informatics, SCI2003, Special Session on Swarm Engineering*. Orlando, Florida, USA.
- Kennedy, J., & Eberhart, R. (1995). A new optimizer using particle swarm theory. *Proceedings of the IEEE 6th International Symposium on Micro Machine and Human Science*, (pp. 39-43). Nagoya, Japan.
- Kernbach, S., Häbe, D., Kernbach, O., Thenius, R., Radspieler, G., Kimura, T., & Schmickl, T. (2013). Adaptive collective decision-making in limited robot swarms without communication. *The International Journal of Robotics Research*, 32(1), 35-55.
- Keshavdas, S., & Kruijff, G. J. (2012). Functional Mapping for Human-Robot Collaborative Exploration. *Proc. of the 2012 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR 2012)*, (pp. 1-6). College Station, TX, USA.
- Kim, Y. D., Kang, J. H., Sun, D. H., Moon, J. I., Ryuh, Y. S., & An, J. (2010). Design and Implementation of User-Friendly Remote Controllers for Rescue Robots Used at Fire Sites. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2010)*, (pp. 377-382). Taipei, Taiwan, China.

-
- Klavins, E. (2002). Communication Complexity of Multi-Robot Systems. *Algorithmic Foundations of Robotics*, 5(7), 1-17.
- Kliger, I., & Lipinski, W. (1964). The Z transform of a product of two functions. *IEEE Transactions on Automatic Control*, 9(4), 582-583.
- Klingspor, V., Demiris, J., & Kaiser, M. (1997). Human-Robot Communication and Machine Learning. *Applied Artificial Intelligence Journal*, 11, 719-74.
- Kloetzer, M., & Belta, C. (2006). Hierarchical abstractions for robotic swarms. *Proceedings in IEEE International Conference on Robotics and Automation (ICRA2006)*, (pp. 952-957). Orlando, FL, USA.
- Kotz, D., Newport, C., & Elliott, C. (2003). *The mistaken axioms of wireless network research*. Technical Report, Dartmouth College, Hanover, NH, USA.
- Krink, T., Vesterstrom, J., & Riget, J. (2002). Particle swarm optimization with spatial particle extension. *Proceedings of the IEEE Congress on Evolutionary Computation*, 2, 1474-1479.
- Krishnanand, K. N., & Ghose, D. (2009a). A Glowworm Swarm Optimization Based Multi-robot System for Signal Source Localization. In *Design and Control of Intelligent Robotic Systems - Studies in Computational Intelligence* (pp. 49-68).
- Krishnanand, K. N., & Ghose, D. (2009b). Glowworm swarm optimization for simultaneous capture of multiple local optima of multimodal functions. *Swarm Intelligence*, 3(2), 87-124.
- Kruijff, G. J., Janicek, M., Keshavdas, S., Larochele, B., Zender, H., Smets, N., . . . Gianni, M. (2012). Experience in System Design for Human-Robot Teaming in Urban Search & Rescue. *Proceedings of the 8th International Conference on Field and Service Robotics (FSR2012)*, (pp. 1-14). Matsushima, Miyagi, Japan.
- Kube, C. R., & Zhang, H. (1993). Collective Robotics: From Social Insects to Robots. *Adaptive Behavior*, 2, 189-219.
- Kulkarni, R. V., & Venayagamoorthy, G. K. (2010). Bio-inspired Algorithms for Autonomous Deployment and Localization of Sensor Nodes. *IEEE Transactions on Systems, Man and Cybernetics*, 40(6), 663-675 .
- Lee, G., Nishimura, Y., Tatara, K., & Chong, N. Y. (2010). Three Dimensional Deployment of Robot Swarms. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2010)*, (pp. 5073-5078). Taipei, Taiwan, China.
- Lee, J. S., Su, Y. W., & Shen, C. C. (2007). A comparative study of wireless protocols: bluetooth, UWB, ZigBee, and Wi-Fi. *Proceedings of the 33rd Annual Conference of the IEEE Industrial Electronics Society (IECON '07)*, 46-51.
- Lee, S. J., Gerla, M., & Toh, C. K. (1999). A simulation study of table-driven and on-demand routing protocols for mobile ad hoc networks. *Network, IEEE*, 13(4), 48-54.
- Lerman, K., & Galstyan, A. (2002). Mathematical Model of Foraging in a Group of Robots: Effect of Interference. *Autonomous Robots*, 13(2), 127-141.

- Leung, K. Y., Barfoot, T. D., & Liu, H. H. (2012). Decentralized Cooperative SLAM for Sparsely-Communicating Robot Networks: A Centralized-Equivalent Approach. *Journal of Intelligent and Robotic Systems*, 66(3), 321-342.
- Levy, P. (1954). Systems Semi-Markoviens Ayant au plus une Infinite Denombrable d'Etats Possibles. *Proceedings of the International Congress on Mathematics*, (pp. 416-426). Amsterdam, Netherlands.
- Lewis, P. O. (2001). A likelihood approach to estimating phylogeny from discrete morphological character data. *Systematic Biology*, 50(6), 913-925.
- Li, M., Alvarez, A., De Pellegrini, F., Prabhakaran, B., & Chlamtac, I. (2007). ROBOTRAK: a centralized real-time monitoring, control, and coordination system for robot swarms. *Proceedings of the 1st international conference on Robot communication and coordination (RoboComm '07)*, IEEE Press, (pp. 1-37). Athens, Greece.
- Liao, E., Hollinger, G., Djughash, J., & Singh, S. (2006). Preliminary Results in Tracking Mobile Targets Using Range Sensors from Multiple Robots. *Proceedings of the International Symposium on Distributed Autonomous Robotic Systems (DARS2006)*, (pp. 125-134). Minneapolis, MN, USA.
- Liu, H., & Abraham, A. (2009). Chaos and Swarm Intelligence. In L. Kocarev, Z. Galias, & S. Lian, *Intelligent Computing Based on Chaos, Studies in Computational Intelligence* (pp. 197-212). Springer Berlin Heidelberg.
- Liu, H., Abraham, A., & Zhang, W. (2007). A fuzzy adaptive turbulent particle swarm optimisation. *International Journal of Innovative Computing and Applications*, 1(1), 39-47.
- Liu, W., & Winfield, A. F. (2010). Modeling and Optimization of Adaptive Foraging in Swarm Robotic Systems. *The International Journal of Robotics Research, IJRR*, 29(14), 1743-1760.
- Lovbjerg, M., & Krink, T. (2002). Extending particle swarms with self-organized criticality. *Proceedings of the IEEE Congress on Evolutionary Computation*, (pp. 1588-1593). Honolulu, HI, USA.
- Luca, D. D., Mazzenga, F., Monti, C., & Vari, M. (2006). Performance Evaluation of Indoor Localization Techniques Based on RF Power Measurements from Active or Passive Devices. *EURASIP Journal on Applied Signal Processing*, 2006(74796), 1-11.
- Machado, J. T., Silva, M. F., Barbosa, R. S., Jesus, I. S., Reis, C. M., Marcos, M. G., & Galhano, A. F. (2010). Some Applications of Fractional Calculus in Engineering. *Hindawi Publishing Corporation Mathematical Problems in Engineering*, 2010(1), 1-34.
- Macwan, A., Nejat, G., & Benhabib, B. (2011). Optimal Deployment of Robotic Teams for Autonomous Wilderness Search and Rescue. *Proceedings in IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS11*, (pp. 4544-4549). San Francisco, CA, USA.
- Marjovi, A., Marques, L., & Penders, J. (2009). Guardians Robot Swarm Exploration and Firefighter Assistance. *Workshop on NRS in IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS2009)*. St Louis, MO, USA.

- Maroco, J. (2010). *Análise Estatística com utilização do SPSS*. Lisboa, Portugal: Edições Silabo.
- Marques, L., Nunes, U., & Almeida, A. (2006). Particle swarm-based olfactory guided search. *Autonomous Robots*, 20(3), 277-287.
- Martínez-de Dios, J. R., Merino, L., Caballero, F., Ollero, A., & Viegas, D. X. (2006). Experimental Results of Automatic Fire Detection and Monitoring with UAVs. *Forest Ecology and Management*, 234(1), 1-10.
- Martinoli, A., & Easton, K. (2003). Modeling Swarm Robotic Systems. In *Experimental Robotics VIII* (pp. 297-306). Springer-Verlag.
- Maslow, A. H. (1943). A Theory of Human Motivation. *Psychological Review*, 50(4), 370-396.
- Mataric, M. J. (1995). Issues and Approaches in the Design of Collective Autonomous Agents. *Robotics and Autonomous Systems*, 16(1), 321-331.
- McLurkin, J. D. (2004). *Stupid Robot Tricks: A Behavior-Based Distributed Algorithm Library for Programming Swarms of Robots*. MSc Thesis in Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, Cambridge, MA, USA.
- McLurkin, J., Smith, J., Frankel, J., Sotkowitz, D., Blau, D., & Schmidt, B. (2006). Speaking swarmish: Human-robot interface design for large swarms of autonomous mobile robots. *Association for the Advancement of Artificial Intelligence Spring Symposium (AAAI)*, (pp. 72-75). Stanford, CA, USA.
- Mei, Y., Lu, Y.-H., Hu, Y. C., & Lee, C. G. (2005). Deployment Strategy for Mobile Robots with Energy and Timing Constraints. *Proceedings of the IEEE International Conference on Robotics and Automation, ICRA05*, (pp. 2827-2832). Barcelona, Spain.
- Mei, Y., Lu, Y.-H., Hu, Y. C., & Lee, C. G. (2006). Deployment of Mobile Robots with Energy and Timing Constraints. *IEEE Transaction on Robotics*, 22(3), 507-521.
- Menezes, P. (1999). *Navegação de Robôs Móveis*. MSc Thesis, University of Coimbra.
- Meyer, W. J., & Minkowski, H. (1969). *Minkowski addition of convex sets*. Madison, USA: University of Wisconsin-Madison.
- Mi, Z., Yang, Y., & Liu, G. (2011a). Coverage Enhancement of Mobile Multi-agent Networks while Preserving Global Connectivity. *IEEE International Conference on Robotics and Automation, ICRA2011*, (pp. 5381-5386). Shanghai, China.
- Mi, Z., Yang, Y., & Liu, G. (2011b). HERO: A Hybrid Connectivity Restoration Framework for Mobile Multi-Agent Networks. *IEEE International Conference on Robotics and Automation, ICRA2011*, (pp. 1702-1707). Shanghai, China.
- Michael, N., Zavlanos, M. M., Kumar, V., & Pappas, G. J. (2009). Maintaining Connectivity in Mobile Robot Networks. *The 11th International Symposium Experimental Robotics, STAR 54*, (pp. 117-126). Athens, Greece.
- Michel, O. (2004). Webots: Professional Mobile Robot Simulation. *Journal of Advanced Robotic Systems*, 1(1), 39-42.

- Miller, L. E. (2001). Multihop Connectivity of Arbitrary Networks. In *Wireless Communication Technologies Group, NIST*.
- Miranda, V., & Fonseca, N. (2002). New evolutionary particle swarm algorithm (EPSO) applied to voltage/VAR control. *Proceedings of the 14th Power Systems Computation Conference*, (pp. 1-6). Seville, Spain.
- Mohan, Y., & Ponnambalam, S. G. (2009). An Extensive Review of Research in Swarm Robotics. *World Congress on Nature & Biologically Inspired Computing (NaBIC'09)*, (pp. 140-145). Coimbatore, India.
- Molga, M., & Smutnicki, C. (2005). Test functions for optimization needs. *Engineering Optimization: An Introduction with Metaheuristic Applications*, 1-43.
- Mondada, F., Bonani, M., Raemy, X., Pugh, J., Cianci, C., Klaptoz, A., . . . Martinoli, A. (2009). The e-puck - a Robot Designed for Education in Engineering. *Proceedings of the 9th Conference on Autonomous Robot Systems and Competitions*, (pp. 59-65). Castelo Branco, Portugal.
- Morrison, R. W., & De Jong, K. A. (1999). A Test Problem Generator for Non-Stationary Environments. *Proceedings of the Congress on Evolutionary Computation (CEC'99)*, (pp. 2047-2053). Washington DC, USA.
- Mosteo, A. R., Montano, L., & Lagoudakis, M. G. (2008). Multi-Robot Routing under Limited Communication Range. *Proceedings of the IEEE/RSJ international conference on Intelligent robots and systems (ICRA2008)*, (pp. 1531-1536). Pasadena, CA, USA.
- Murphy, R. R. (2000). Marsupial and shape-shifting robots for urban search and rescue. *IEEE Intelligent Systems and Their Applications*, 15(2), 14-19.
- Murphy, R. R. (2004). Human-robot interaction in rescue robotics. *IEEE Transactions on Systems, Man, and Cybernetics*, 34(2), 138-153.
- Murphy, R. R., Ausmus, M., Bugajska, M., & Ellis, T. (1999). Marsupial-like Mobile Robot Societies. *ACM Autonomous Agents*.
- Murphy, R. R., Tadokoro, S., Nardi, D., Jacoff, A., Fiorini, P., Choset, H., & Erkmen, A. M. (2008). Search and Rescue Robotics. In B. Siciliano, & O. Khatib (Eds.), *Springer Handbook of Robotics* (pp. 1151-1173). Springer Verlag.
- Natesapillai, K., Palanisamy, V., & Duraiswamy, K. (2009). A Performance Evaluation of Proactive and Reactive Protocols Using NS2 Simulation. *International Journal of Engineering Research and Industrial Applications*, 2(11), 309-326.
- Niccolini, M., Innocenti, M., & Pollini, L. (2010). Near Optimal Swarm Deployment using Descriptor Functions. *Proceedings of the IEEE International Conference on Robotics and Automation, ICRA10*, (pp. 4952-4957). Anchorage, Alaska, USA.
- Nicholson, T. (2008). *Symposium on Social Inclusion Down Under* (pp. 1-5). Melbourne, Australia: Brotherhood of St Laurence.
- Niederreiter, H. (1992). *Random Number Generation and Quasi-Monte Carlo Methods*. Philadelphia: Society for Industrial and Applied Mathematics.

- Nowell, P. (2005). *Mapping a Square to a Circle*. Retrieved from Math Proofs - Interesting mathematical results and elegant solutions to various problems: <http://mathproofs.blogspot.pt/2005/07/mapping-square-to-circle.html>
- Ohgai, A., Gohnai, Y., Ikaruga, S., Murakami, M., & Watanabe, K. (2005). Cellular Automata Modeling for Fire Spreading as a Tool to Aid Community-Based Planning for Disaster Mitigation. In J. P. Van Leeuwen, & H. J. Timmermans (Eds.), *Recent Advances in Design & Decision Support Systems in Architecture and Urban Planning* (pp. 193-209). Dordrecht, Netherlands: Kluwer Academic Publishers.
- Onn, S., & Tennenholtz, M. (1997). Determination of Social Laws for Multi-agent Mobilization. *Artificial Intelligence*, 95, 155-167.
- Ortigueira, M. D., & Machado, J. A. (2003). Special Issue on Fractional Signal Processing. *Signal Process and Applications*, 83(11), 2285-2286.
- Ostalczyk, P. W. (2009). A note on the Grünwald–Letnikov fractional-order backward-difference. *Physica Scripta*, 136, 1-5.
- Pallant, J. (2011). *SPSS Survival Manual* (4th ed.). Open University Press.
- Park, I. W., & Kim, J. O. (2011). Philosophy and Strategy of Minimalism-based User Created Robots (UCRs) for Educational Robotics - Education, Technology and Business Viewpoint. *International Journal of Robots, Education and Art*, 1(1), 26-38.
- Parker, L. E. (1994). *Heterogeneous Multi-Robot Cooperation*. PhD Thesis, MIT, EECS Dept.
- Parker, L. E. (1998). ALLIANCE: An Architecture for Fault-Tolerant Multi-Robot Cooperation. *IEEE Transactions on Robotics and Automation*, 14(2), 220-240.
- Parker, L. E. (2008a). Distributed Intelligence: Overview of the Field and its Application in Multi-Robot Systems. *Journal of Physical Agents*, 2(1), 5-14.
- Parker, L. E. (2008b). Multiple Mobile Robot Systems. In *Springer Handbook of Robotics* (pp. 921-941).
- Parker, L. E., Schneider, F., & Schultz, A. (2005). Multi-Robot Systems: From Swarms to Intelligent Automata. *Proceedings from the International Workshop on Multi-Robot Systems*. Dordrecht, Netherlands: Springer.
- Pasqualetti, F., Durham, J., & Bullo, F. (2012). Cooperative Patrolling via Weighted Tours: Performance Analysis and Distributed Algorithms. *IEEE Transactions on Robotics*, 9, 1-8.
- Pedrosa, A. C., & Gama, S. M. (2004). *Introdução Computacional à Probabilidade e Estatística*. Portugal: Porto Editora.
- Pei, Y., Mutka, M. W., & Xi, N. (2010). Coordinated Multi-Robot Real-Time Exploration With Connectivity and Bandwidth Awareness. *IEEE International Conference on Robotics and Automation (ICRA2010)*, (pp. 5460-5465). Anchorage, Alaska, USA.
- Pereira, G. A., Das, A. K., Kumar, V., & Campos, M. F. (2003). Decentralized motion planning for multiple robots subject to sensing and communication constraints. *Proceedings of the Second Multi-Robot Systems Workshop*, (pp. 267-278). Washington, DC, USA.

- Perkins, C. E., Royer, E. M., & Das, S. R. (1999). *Ad hoc on demand distance vector (AODV) routing*. Internet Draft, Mobile Ad Hoc Networking Working Group of the Internet Engineering Task Force (IETF), Nokia Research Center. Retrieved from <http://www.cs.ucsb.edu/~ebelding/txt/aodv/aodvidv4.txt>
- Pestana, M. H., & Gageiro, J. N. (2008). *Análise de dados para Ciências Sociais - A complementaridade do SPSS* (5th ed.). Lisboa, Portugal: Edições Sílabo.
- Pini, G., Brutschy, A., Scheidler, A., Dorigo, M., & Birattari, M. (2012). *Task partitioning in a robot swarm: retrieving objects by transferring them directly between sequential sub-tasks*. Université Libre de Bruxelles. Bruxelles, Belgium: Institut de Recherches Interdisciplinaires et de Développements en Intelligence Artificielle (IRIDIA).
- Pires, E. J., Machado, J. A., Cunha, P. B., & Mendes, L. (2010). Particle swarm optimization with fractional-order velocity. *Journal on Nonlinear Dynamics*, 61(1-2), 295-301.
- Pires, E. J., Oliveira, P. B., Machado, J. A., & Cunha, J. B. (2006). Particle Swarm Optimization versus Genetic Algorithm in Manipulator Trajectory Planning. *7th Portuguese Conference on Automatic Control*. Lisbon, Portugal.
- Pisani, D., Yates, A. M., Langer, M. C., & Benton, M. J. (2001). A genus-level supertree of the Dinosauria. *Proceedings of the Royal Society*, 269(1494), 915-921.
- Podlubny, I. (1999). *Fractional Differential Equations* (198 ed., Vol. 198). San Diego, California: Academic Press.
- Portugal, D., & Rocha, R. P. (2012). Decision Methods for Distributed Multi-Robot Patrol. *Proceedings of the IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR2012)*, (pp. 1-6). Texas, USA.
- Prado, J., Simplicio, C., Lori, N. F., & Dias, J. (2012). Visuo-Auditory Multimodal Emotional Structure to Improve Human-Robot-Interaction. *International Journal of Social Robotics*, 4(1), 29-51.
- Pugh, J., & Martinoli, A. (2006). Multi-Robot Learning with Particle Swarm Optimization. *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems*, (pp. 441-448). New York, USA.
- Pugh, J., & Martinoli, A. (2007). Inspiring and Modeling Multi-Robot Search with Particle Swarm Optimization. *Proceedings of the IEEE Swarm Intelligence Symposium*, (pp. 332-339). Honolulu, HI, USA.
- Quigley, M., Gerkey, B., Conley, K., Faust, J., Foote, T., Leibs, J., . . . Ng, A. (2009). ROS: an open-source Robot Operating System. *International Conference on Robotics and Automation (ICRA'09) Workshop on Open Source Software*. Kobe, Japan.
- Rañó, I., & Minguez, J. (2006). Steps toward the automatic evaluation of robot obstacle avoidance algorithms. *Workshop of Benchmarking in Robotics in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2006)*. Beijing, China.

- Rapaic, M. R., Kanovic, Ž., & Jelicic, Z. D. (2009). A Theoretical and Empirical Analysis of Convergence Related Particle Swarm Optimization. *WSEAS Transactions on Systems and Control*, 4(11), 541-550.
- Rasmussen, J. (1986). *Information Processing and Human-Machine Interaction. An Approach to Cognitive Engineering*. North-Holland: University of Michigan.
- Ratnieks, F. L., & Anderson, C. (1999). Task partitioning in insect societies. *Insectes Sociaux*, 46(2), 95-108.
- Reddy, P. P., & Veloso, M. M. (2011). RSSI-based Physical Layout Classification and Target Tethering in Mobile Ad-hoc Networks. *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS2011*, (pp. 2327-2332). San Francisco, CA, USA.
- Reynolds, C. W. (1987). Flocks, herds, and schools: A distributed behavioral model. *14th Annual Conference on Computer Graphics and Interactive Techniques*, 21, pp. 25-34. New York, USA.
- Rocha, R. P. (2006a). *Building Volumetric Maps with Cooperative Mobile Robots and Useful Information Sharing: a Distributed Control Approach based on Entropy*. PhD Thesis, Faculty of Engineering of University of Porto, Porto, Portugal.
- Rocha, R. P. (2006b). Efficient Information Sharing and Coordination in Cooperative Multi-Robot Systems. In *Proceedings of II European-Latin-American Workshop on Engineering Systems (SELASI'2006)*. Porto.
- Rocha, R. P., Ferreira, F., & Dias, J. (2008). Multi-Robot Complete Exploration using Hill Climbing and Topological Recovery. *Proc. of 2008 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'2008)*, (pp. 1884-1888). Nice, France.
- Rocha, R., Dias, J., & Carvalho, A. (2005). Cooperative Multi-Robot Systems: a study of Visionbased 3-D Mapping using Information Theory. *Robotics and Autonomous Systems*, 53(3-4), 282-311.
- Rohrer, F. (2008). *Curriculum of exercises for e-puck and Webots*. Swiss: EPFL.
- Rooker, M., & Birk, A. (2006). Communicative Exploration with Robot Packs. In *Lecture Notes in Artificial Intelligence* (Vol. 4020, pp. 267-278). Berlin: Springer.
- Rooker, M., & Birk, A. (2007). Multi-robot exploration under the constraints of wireless networking. *Control Engineering Practice*.
- Ruan, D., & Kerre, E. E. (2002). On if-then-else inference rules. *IEEE International Conference on Systems, Man, and Cybernetics*, (pp. 1420-1425). Beijing, China.
- Rybski, P. E., Papanikolopoulos, N. P., Stoeter, S. A., Krantz, D. G., Yesin, K. B., Gini, M., . . . Erickson, M. D. (2000). Enlisting Rangers and Scouts for Reconnaissance and Surveillance. *IEEE Robotics & Automation Magazine*, 7(4), 14-24.
- Sabatier, J., Agrawal, O. P., & Machado, J. A. (2007). *Advances in Fractional Calculus - Theoretical Developments and Applications in Physics and Engineering*. Berlin, Germany: Springer.

- Sabattini, L., Chopra, N., & Secchi, C. (2011a). On decentralized connectivity maintenance for mobile robotic systems. *50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, (pp. 988-993). Orlando, Florida, USA.
- Sabattini, L., Chopra, N., & Secchi, C. (2011b). Distributed Control of Multi-Robot Systems with Global Connectivity Maintenance. *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS2011*, (pp. 2321-2326). San Francisco, CA, USA.
- Sahin, C. S., Urrea, E., Uyar, M. U., Conner, M., Hokelek, I., Bertoli, G., & Pizzo, C. (2008). Self-deployment of Mobile Agents in MANETs for Military Applications. *Army Science Conference*, (pp. 1-8). Orlando, Florida, USA.
- Sahin, E. (2005). Swarm robotics: From sources of inspiration to domains of application. *Lecture Notes in Computer Science, 3342*(Swarm Robotics), 10-20.
- Saikishan, D., & Prasanna, K. (2010). *Multiple robot co-ordination using particle swarm optimisation and bacteria foraging algorithm*. BTech thesis, Department of Mechanical Engineering, National Institute of Technology, Rourkela.
- Salkham, A., Cunningham, R., Senart, A., & Cahill, V. (2006). A taxonomy of collaborative context-aware systems. *Proceedings of the International Workshop on Ubiquitous Mobile Information and Collaboration Systems (UMICS)*. Luxemburg.
- Schwager, M., Rus, D., & Slotine, J.-J. (2010). Unifying Geometric, Probabilistic, and Potential Field Approaches to Multi-robot Deployment. *The International Journal of Robotics Research, 30*, 371-383.
- Scutella, R., Wilkins, R., & Horn, M. (2009). *Measuring Poverty and Social Exclusion in Australia: A Proposed Multidimensional Framework for Identifying Socio-Economic Disadvantage* (Vol. 4). Victoria, Australia: Melbourne Institute Working Paper Series.
- Secrest, B., & Lamont, G. (2003). Visualizing particle swarm optimization - Gaussian particle swarm optimization. *Proceedings of the IEEE Swarm Intelligence Symposium*, (pp. 198-204). Indianapolis, Indiana, USA.
- Shah, K., & Meng, Y. (2007). Communication-Efficient Dynamic Task Scheduling for Heterogeneous Multi-Robot Systems. *Proceedings of the 2007 IEEE International Symposium on Computational Intelligence in Robotics and Automation*, (pp. 230-235). Jacksonville, FL, USA.
- Shannon, C. E. (1949). *The Mathematical Theory of Communication*. University of Illinois Press.
- Shaw, I. S. (1998). *Fuzzy Control of Industrial Systems: Theory and Applications*. Kluwer Academic Publishers.
- Sheng, W., Yang, Q., Tan, J., & Xi, N. (2006). Distributed multi-robot coordination in area exploration. *Robotics and Autonomous Systems, 54*(2006), 945-955.
- Shi, Y., & Eberhart, R. (2001). Fuzzy adaptive particle swarm optimization. *Proceedings of the IEEE Congress on Evolutionary Computation, 1*, 101-106.
- Sklar, B. (1997). Rayleigh fading channels in mobile digital communication systems .I. Characterization. *IEEE Communications Magazine, 35*(7), 90-100.

- Staranowicz, A., & Mariottini, G. L. (2011). A Survey and Comparison of Commercial and Open-Source Robotic Simulator Software. *Proceedings of the 4th International Conference on Pervasive Technologies Related to Assistive Environments (PETRA '11)*, (pp. 1-8). New York, USA.
- Steinberg, M. L. (2011). Biologically-inspired approaches for self-organization, adaptation, and collaboration of heterogeneous autonomous systems. *Proceedings of SPIE Volume on Defense Transformation and Net-Centric Systems, 8062*, pp. 1-13. Orlando, Florida, USA.
- Stewart, J. (2007). *Calculus* (5th ed.). Cengage Learning.
- Stirling, T., Roberts, J., Zufferey, J.-C., & Floreano, D. (2012). Indoor Navigation with a Swarm of Flying Robots. *IEEE International Conference on Robotics and Automation (ICRA2012)*, (pp. 4641-4647). Saint Paul, MN, USA.
- Suarez, J., & Murphy, R. (2011). A Survey of Animal Foraging for Directed, Persistent Search by Rescue Robotics. *Proceedings of the 2011 IEEE International Symposium on Safety, Security and Rescue Robotics*, (pp. 314-320). Kyoto, Japan.
- Suresh, K., Ghosh, S., Kundu, D., Sen, A., Das, S., & Abraham, A. (2008). Inertia-Adaptive Particle Swarm Optimizer for Improved Global Search. *Eighth International Conference on Intelligent Systems Design and Applications, ISDA '08*, (pp. 253 - 258). Kaohsiung.
- Sweeney, J. D., Grupen, R. A., & Shenoy, P. (2004). Active QoS flow maintenance in controlled mobile networks. *Proceedings of the Fourth International Symposium on Robotics and Automation (ISRA2004)*, (pp. 1-9). Queretaro, Mexico, USA.
- Takahashi, R., Ise, H., Konno, A., Uchiyama, M., & Sato, D. (2008). Hybrid simulation of a dual-arm space robot colliding with a floating object. *IEEE International Conference on Robotics and Automation (ICRA2008)*, (pp. 1201-1206). Pasadena, CA, USA.
- Tambe, M. (1997). Towards flexible teamwork. *Journal of Artificial Intelligence Research*, 7, 83-124.
- Tan, C. A., & Kang, B. S. (2001). Chaotic motions of a Duffing oscillator subjected to combined parametric and quasiperiodic excitation. *International Journal of Nonlinear Sciences and Numerical Simulation*, 2(4), 353-364.
- Tang, J., Zhu, J., & Sun, Z. (2005). A novel path panning approach based on appart and particle swarm optimization. *Proceedings of the 2nd International Symposium on Neural Networks, LNCS 3498*, (pp. 253-258).
- Tardioli, D., & Villarroel, J. L. (2007). Real Time Communications over 802.11: RT-WMP. *IEEE Internatonal Conference on Mobile Adhoc and Sensor Systems*, (pp. 1-11). Pisa, Italy.
- Tardioli, D., Mosteo, A. R., Riazuelo, L., Villarroel, J. L., & Montano, L. (2010). Enforcing Network Connectivity in Robot Team Missions. *The International Journal of Robotics Research*, 29(4), 460-480.
- Thibodeau, B. J., Fagg, A. H., & Levine, B. N. (2004). *Signal strength coordination for cooperative mapping*. Technical Report, University of Massachusetts, Department of Computer Science, Amherst, MA, USA.

- Tijms, H. C. (2003). *A First Course in Stochastic Models*. Chichester, England: John Wiley & Sons Ltd. The Atrium.
- Tillett, J., Rao, T. M., Sahin, F., Rao, R., & Brockport, S. (2005). Darwinian Particle Swarm Optimization. In B. Prasad (Ed.), *Proceedings of the 2nd Indian International Conference on Artificial Intelligence*, (pp. 1474-1487). Pune, India.
- Truax, B. (1999). *Handbook for Acoustic Ecology* (2nd ed.). World Soundscape Project, Simon Fraser University and ARC Publications. Retrieved from <http://www.sfu.ca/sonic-studio/handbook/>
- Tsai, W. (2002). Social Structure of “Coopetition” Within a Multiunit Organization: Coordination, Competition, and Intraorganizational Knowledge Sharing. *Organization Science*, 13(2), 179-190.
- Tuan, L. V., Bouraqadi, N., Moraru, V., Stinckwich, S., & Doniec, A. (2009). Making Networked Robot Connectivity-Aware. *Proceedings of the IEEE International Conference on Robotics and Automation, ICRA2009*, (pp. 3502-3507). Kobe, Japan.
- Turner, R. M. (1998). Context-mediated behavior for intelligent agents. *International Journal of Human-Computer Studies*, 48(3), 307-330.
- Valle, Y. d., Venayagamoorthy, G. K., Mohagheghi, S., Hernandez, J. C., & Harley, R. (2008). Particle swarm optimization: Basic concepts, variants and applications in power systems. *IEEE Transactions on Evolutionary Computation*, 2(2), 171-195.
- Vasconcelos, H. L., & Cherrett, J. M. (1996). The effect of wilting on the selection of leaves by the leaf-cutting ant *Atta laevigata*. *Entomologia Experimentalis et Applicata*, 78(2), 215-220.
- Veloso, M., Uther, W., Fujita, M., Asada, M., & Kitano, H. (1998). Playing soccer with legged robots. *Proceedings of IEEE/RSJ International Workshop on Intelligent Robots and Systems (IROS1998)*, (pp. 437-442). Victoria, Canada.
- Venter, G. (2002). Particle swarm optimization. *Proceedings of 43rd AIAA/ASME/ASCE/AHS/ASC Structure, Structures Dynamics and Materials Conference*, (pp. 22-25). Denver, Colorado, USA.
- Vidal, R., Shakernia, O., Kim, H., Shim, D., & Sastry, S. (2002). Probabilistic pursuit-evasion games: theory, implementation, and experimental evaluation. *IEEE Transactions on Robotics and Automation*, 18(5), 662-669.
- Wakasa, Y., Tanaka, K., & Nishimura, Y. (2010). Control-Theoretic Analysis of Exploitation and Exploration of the PSO Algorithm. *IEEE International Symposium on Computer-Aided Control System Design, IEEE Multi-Conference on Systems and Control*, (pp. 1807-1812). Yokohama, Japan.
- Warburton, K., & Lazarus, J. (1991). Tendency-distance models of social cohesion in animal groups. *Journal of Theoretical Biology*, 150(4), 473-488.
- Werger, B. B. (1999). Cooperation without deliberation: A minimal behavior-based approach to multi-robot teams. *Artificial Intelligence*, 110(2), 293-320.
- Williams, R. L., & Wu, J. (2010). Dynamic Obstacle Avoidance for an Omnidirectional Mobile Robot. (S. Arimoto, Ed.) *Journal of Robotics*, 2010(901365), 1-15.

-
- Wilson, M., Melhuish, C., Sendova-Franks, A., & Scholes, S. (2004). Algorithms for building annular structures with minimalist robots inspired by brood sorting in ant colonies. *Autonomous Robots*, 17(2-3), 115-136.
- Winfield, A. F. (2000). Distributed Sensing and Data Collection via Broken Ad Hoc Wireless Connected Networks of Mobile Robots. (L. E. Parker, G. Bekey, & J. Barhen, Eds.) *Distributed Autonomous Robotic Systems*, 4, 273-282.
- Wu, X., Xu, H., Sadjadpour, H. R., & Garcia-Luna-Aceves, J. J. (2008). Proactive or Reactive Routing: A Unified Analytical Framework in MANETs. *Proceedings of 17th International Conference on Computer Communications and Networks, ICCCN '08*, (pp. 1-7). St. Thomas, US Virgin Islands.
- Xue, S. D., & Zeng, J. C. (2008). Control over swarm robots search with swarm intelligence principles. *Journal of System Simulation*, 20(13), 3449-3454.
- Yasuda, K., Iwasaki, N., Ueno, G., & Aiyoshi, E. (2008). Particle Swarm Optimization: A Numerical Stability Analysis and Parameter Adjustment Based on Swarm Activity. *IEEE Transactions on Electrical and Electronic Engineering, Wiley InterScience*, 3, 642-659.
- Zadeh, L. A. (1965). Fuzzy Sets. *Information and Control*, 8(3), 338-353.
- Zhang, W., & Xie, X. (2003). DEPSO: Hybrid particle swarm with differential evolution operator. *IEEE International Conference on Systems, Man and Cybernetics*, 4, pp. 3816-3821. Washington, DC, USA.
- Zhong, X., & Zhou, Y. (2012). Maintaining wireless communication coverage among multiple mobile robots using fuzzy neural network. *IEEE/ASME International Conference on Mechatronics and Embedded Systems and Applications (MESA)*, (pp. 35-41). Suzhou, China.

Publications Relevant to the Thesis

What follows is a list of publications directly related to the research contribution of this PhD thesis. Note that some of them are still under review and the number between brackets represent the review cycle they were at the time this Thesis was written, *e.g.*, a paper identified as “*Under Review (2)*” means that a first iteration from the reviewers was already considered by the authors and the paper was resubmitted for the second review cycle.

Refereed Scientific Journals

1. Micael S. Couceiro, Fernando M. L. Martins, Rui P. Rocha & Nuno M. F. Ferreira. “*On a Macroscopic Model of an Evolutionary Swarm Robotics Exploration Approach*”, IEEE Transactions on Evolutionary Computation, *Under Review (1)*.
2. Micael S. Couceiro, Fernando M. L. Martins, Rui P. Rocha & Nuno M. F. Ferreira. “*Mechanism and Convergence Analysis of a Multi-Robot Swarm Approach Based on Natural Selection*”, Journal of Intelligent & Robotic Systems, *Under Review (3)*.
3. Micael S. Couceiro, Carlos M. Figueiredo, Rui P. Rocha & Nuno M. F. Ferreira. “*Darwinian Swarm Exploration under Communication Constraints: Initial Deployment and Fault-Tolerance Assessment*”, Robotics and Autonomous Systems, 2014 (*In Press*).
4. Micael S. Couceiro, Patricia A. Vargas, Rui P. Rocha & Nuno M. F. Ferreira. “*Benchmark of Swarm Robotic Distributed Techniques in a Search Task*”, Robotics and Autonomous Systems, 2014 (*In Press*).
5. Micael S. Couceiro, David Portugal, Rui P. Rocha & Nuno M. F. Ferreira. “*Marsupial Teams of Robots: Deployment of Miniature Robots for Swarm Exploration under Communication Constraints*”, Robotica Cambridge Journal, 2014 (*In Press*).
6. Micael S. Couceiro, Amadeu Fernandes, Rui P. Rocha & Nuno M. F. Ferreira. “*Understanding the Communication Complexity of the Robotic Darwinian PSO*”, Robotica Cambridge Journal, 2014 (*In Press*).
7. Micael S. Couceiro, Rui P. Rocha & Nuno M. F. Ferreira. “*A PSO Multi-Robot Exploration Approach Over Unreliable MANETs*”, Advanced Robotics, Vol. 27, Issue 16, 2013.
8. Micael S. Couceiro, J. A. Tenreiro Machado, Rui P. Rocha & Nuno M. F. Ferreira. “*A fuzzified systematic adjustment of the robotic Darwinian PSO*”, Robotics and Autonomous Systems, Vol. 60, Issue 12, pp. 1625-1639, 2012.
9. Micael S. Couceiro, Fernando M. L. Martins, Rui P. Rocha & N. M. Fonseca Ferreira. “*Analysis and Parameter Adjustment of the RDPSO - Towards an Understanding of Robotic*

Network Dynamic Partitioning based on Darwin's Theory", International Mathematical Forum, Hikari, Ltd., Vol. 7, Issue 32, pp. 1587-1601, 2012.

10. Micael S. Couceiro, Rui P. Rocha, N. M. Fonseca Ferreira & J. A. Tenreiro Machado. "Introducing the Fractional Order Darwinian PSO", Signal, Image and Video Processing, Fractional Signals and Systems Special Issue, Springer, Vol. 6, Issue 3, pp. 343-350, 2012.
11. Micael S. Couceiro, Carlos M. Figueiredo, J. Miguel A. Luz, Nuno M. F. Ferreira & Rui P. Rocha. "A Low-Cost Educational Platform for Swarm Robotics", International Journal of Robots, Education and Art, IJREA, Volume 2, Issue 1, February, pp. 1-15, 2012.

Book Chapters

1. Micael S. Couceiro, Andria R. Lopes, N. M. Fonseca Ferreira, Anabela G. Ferreira & Rui Rocha. "Toward the Concept of Robot Society: A Multi-Robot SLAM Case Study", Mathematical Methods in Engineering Book, Springer, 2013.
2. Micael S. Couceiro, Fernando M. L. Martins, Filipe Clemente, Rui P. Rocha & Nuno M. F. Ferreira. "Towards a Further Understanding of the Robotic Darwinian PSO", Computational Intelligence and Decision Making - Trends and Applications, From Intelligent Systems, Control and Automation: Science and Engineering Bookseries, Springer Verlag, pp. 17-26, 2012.

Refereed Conferences

1. Micael S. Couceiro, Fernando M. L. Martins, Rui P. Rocha & Nuno M. F. Ferreira. "A Predictive Model for an Evolutionary Swarm Robotic Algorithm" In Proc. of 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2014), Chicago, USA, September 14-18, 2014, *Under Review (1)*.
2. Micael S. Couceiro, Rui P. Rocha, Nuno M. F. Ferreira & Patricia A. Vargas. "Darwinian Robotic Swarms for Exploration with Minimal Communication", IEEE Congress on Evolutionary Computation (CEC 2013), Special Session on Evolutionary Robotics, June 20-23, Cancún, Mexico, 2013.
3. Micael S. Couceiro, Rui P. Rocha & Nuno M. F. Ferreira. "Fault-Tolerance Assessment of a Darwinian Swarm Exploration Algorithm under Communication Constraints", In Proc. of 2013 IEEE International Conference on Robotics and Automation (ICRA 2013), Karlsruhe, Germany, May 6-10, 2013.
4. Micael S. Couceiro, Carlos M. Figueiredo, David Portugal, Rui P. Rocha & Nuno M. F. Ferreira. "Initial Deployment of a Robotic Team - A Hierarchical Approach Under Communication Constraints Verified on Low-Cost Platforms", IROS'12 - IEEE/RSJ International Conference on Intelligent Robots and Systems, October 7-12, Vilamoura, Portugal, 2012.

5. Micael S. Couceiro, Rui P. Rocha, Carlos M. Figueiredo, J. Miguel A. Luz & N. M. Fonseca Ferreira. “*Multi-Robot Foraging based on Darwin’s Survival of the Fittest*”, IROS’12 - IEEE/RSJ International Conference on Intelligent Robots and Systems, October 7-12, Vilamoura, Portugal, 2012.
6. Micael S. Couceiro, Fernando M. L. Martins, Rui P. Rocha & Nuno M. F. Ferreira. “*Introducing the Fractional Order Robotic Darwinian PSO*”, Proceedings of the 9th International Conference on Mathematical Problems in Engineering, Aerospace and Sciences - ICNPAA’2012, Vienna University of Technology, Austria, July 10-14, 2012.
7. André Araújo, David Portugal, Micael S. Couceiro, Carlos M. Figueiredo & Rui P. Rocha. “*Small and Compact Mobile Robots - Surveying and Comparing Platforms*”, International Conference Automatics’2012, FA, Anniversary conference “50 Years Education in Automatics, Sozopol, Bulgaria, June 01–04, 2012.
8. André Araújo, David Portugal, Micael S. Couceiro, Carlos M. Figueiredo & Rui P. Rocha. “*TraxBot: Assembling and Programming of a Mobile Robotic Platform*”, Proceedings of the 4th International Conference on Agents and Artificial Intelligence - ICAART’2012, pp. 301-304, Vilamoura, Portugal, February 6-8, 2012.
9. Micael S. Couceiro, N. M. Fonseca Ferreira, Rui P. Rocha, Fernando M. L. Martins & Filipe Clemente. “*Statistical Significance Analysis of the R-DPSO - Towards an Understanding of the Relationship between the Population of Robots and the Maximum Communication Distance*”, ISCIES 2011 - International Symposium on Computational Intelligence for Engineering Systems, November 16-18, Coimbra, Portugal, 2011.
10. Micael S. Couceiro, Rui P. Rocha & Nuno M. F. Ferreira. “*Ensuring Ad Hoc Connectivity in Distributed Search with Robotic Darwinian Particle Swarms*”, IEEE International Symposium on Safety, Security and Rescue Robotics, November 1-5, Kyoto, Japan, 2011.
11. Micael S. Couceiro, Rui P. Rocha & Nuno M. F. Ferreira. “*A Novel Multi-Robot Exploration Approach based on Particle Swarm Optimization Algorithms*”, IEEE International Symposium on Safety, Security and Rescue Robotics, November 1-5, Kyoto, Japan, 2011.
12. Micael S. Couceiro, Nuno M. F. Ferreira & Rui Rocha. “*O Conceito da Sociedade Robótica – Evidências Extraídas a Partir de GridSLAM Cooperativo em Player/Stage*”, 6º Congresso Luso-Moçambicano de Engenharia, August 29 – September 02, Maputo, Moçambique, 2011.
13. Micael S. Couceiro, Nuno M. F. Ferreira & Rui Rocha. “*Multi-Robot Exploration based on Swarm Optimization Algorithms*”, ENOC’2011 - 7th European Nonlinear Dynamics Conference, July 24-29, Rome, Italy, 2011.
14. Micael S. Couceiro, Andria R. Lopes, N. M. Fonseca Ferreira, Anabela G. Ferreira & Rui Rocha. “*The Concept of Robot Society: Evidence Taken From Multi-Robot SLAM*”, MME’10 - Mathematical Methods in Engineering International Symposium, ISEC-IPC, October, Coimbra, Portugal, 2010.

Technical Reports

1. Micael S. Couceiro. “*Bridging the Gap between Webots and e-puck Robots - Towards an Improved Compatibility*”. Technical Report developed under the supervision of Prof. Patricia Vargas and Prof. Rui Rocha at the Robotics Lab, School of Mathematical and Computer Sciences from Heriot-Watt University, United Kingdom, 2013. Available at http://www2.isr.uc.pt/~michaelcouceiro/media/ReportFinal_1.pdf.
2. Micael S. Couceiro. “*Endowing e-pucks with WiFi Communication - Fostering Multi-Robot Cooperative Experiments*”. Technical Report developed under the supervision of Prof. Patricia Vargas and Prof. Rui Rocha at the Robotics Lab, School of Mathematical and Computer Sciences from Heriot-Watt University, United Kingdom, 2013. Available at http://www2.isr.uc.pt/~michaelcouceiro/media/ReportFinal_2.pdf.
3. Micael S. Couceiro. “*MRSim - Multi-Robot Simulator v1.0*”. Tutorial documentation of the developed MatLab multi-robot simulator MRSim, Institute of Systems and Robotics, University of Coimbra, Portugal, 2012. Available at <http://www2.isr.uc.pt/~michaelcouceiro/media/help/helpMRSim.htm>.