UNIVERSITÀ DELLA CALABRIA

**UNIVERSITA' DELLA CALABRIA**

Dipartimento di Ingegneria Informatica, Modellistica, Elettronica e Sistemistica

**Dottorato di Ricerca in**

Information and Communication Engineering for Pervasive Intelligent Environments

**CICLO**

XXIX

# BIO-INSPIRED TECHNIQUES APPLIED TO THE COORDINATION OF A SWARM OF ROBOTS INVOLVED IN MULTIPLE TASKS

**Settore Scientifico Disciplinare ING-INF/03**

**Coordinatore:** Prof. Felice Crupi

Firma

**Tutor:** Prof. Salvatore Marano

Firma

**Co -Tutor:** Prof. Xin-She Yang

Firma

**Dottorando:** Nunzia Palmieri

Firma

I would like to dedicate this thesis to my family

# Acknowledgements

Firstly, I would like to express my gratitude to my advisor, Prof. Salvatore Marano for his guidance and his support.

Special thanks to Prof. Xin-She Yang, he was my guidance at Middlesex University, London. He directed my research with many suggestions, advices and constant help.

I also thank Dr. Wei Li for interesting chats at University during my visit in London as a visiting research student.

Thanks to the colleagues in the Department, for sharing joys and sorrows and always ready to help me.

Last but not least, I would like to thank my family, my friends and my husband. They all encouraged me and accompanied me during these years.

# Abstract

La tematica di ricerca trattata in questa tesi riguarda il problema di coordinamento di robot attraverso l'utilizzo di algoritmi decentralizzati che usano meccanismi basati sulla Swarm Intelligence. Tali tecniche hanno lo scopo di migliorare le capacità di ogni robot, ciascuno dei quali ha risorse limitate, nel prendere decisioni su dove muoversi o su cosa fare basandosi su semplici regole ed interazioni locali. Negli ultimi anni, infatti, c'è un crescente interesse a risolvere alcuni problemi nell'ambito della robotica attraverso algoritmi che traggono ispirazione da fenomeni naturali e da alcuni animali in natura che esibiscono comportamenti sociali sviluppati e con una notevole capacità di adattamento ambientale. Nel campo della robotica, un aspetto cruciale è la coordinazione dei robot affinché possano compiere dei task in maniera cooperativa. La coordinazione deve essere tale da permettere agli agenti di adattarsi alle condizioni dinamiche dell'ambiente circostante conferendo al sistema caratteristiche di robustezza, flessibilità e affidabilità. Più dettagliatamente, lo scenario di riferimento è un'area nella quale sono disseminati degli oggetti, e dove operano un certo numero di robot che hanno come scopo quello di rilevare gli oggetti stessi e manipolarli. Ciascun robot non conosce la posizione di tali oggetti e non ha conoscenza né dell'ambiente che lo circonda, né della posizione degli altri robot.

Il problema è diviso in due sotto-problemi. Un primo problema riguarda l'esplorazione dell'area e l'altro la manipolazione degli oggetti. Essenzialmente, ogni robot esplora in maniera indipendente l'ambiente basandosi sulla propria posizione attuale e sulla posizione degli altri mediante un meccanismo di comunicazione indiretta (stigmergia). Nella fase di manipolazione degli oggetti, invece, è utilizzato un meccanismo di comunicazione diretta attraverso l'uso di una comunicazione wireless.

L'algoritmo di esplorazione dell'area trae ispirazione dal comportamento di alcuni tipi di insetti in natura, come le formiche,che utilizzano l'ambiente nel quale vivono come mezzo di comunicazione (stigmergia).Successivamente, quando un robot rileva la presenza di un oggetto, sono stati proposti due approcci. Nel primo caso le informazioni sono diffuse tra i robot secondo un meccanismo di comunicazione"one hop"ed alcune meta-euristiche di derivazione naturale sono state utilizzate come meccanismo decisionale e di coordinamento.

Il secondo approccio fa riferimento ad una comunicazione “multi-hop” ed è stato proposto un protocollo di coordinamento, anche esso di derivazione biologica.

Entrambi gli approcci si basano su meccanismi decentralizzati dove non esiste nessun leader che dia direttive gerarchiche e ciascun robot prende le sue decisioni in maniera autonoma sulla base degli eventi che accadono nell’ambiente. Globalmente si ha un sistema auto organizzato, flessibile ed altamente adattabile. Per testare gli approcci è stato costruito un simulatore sul quale sono stati sviluppati numerosi studi allo scopo di valutare gli algoritmi proposti, la loro efficienza nonché stimare come le principali variabili ed i parametri del modello possono influenzarela soluzione finale.

The research described in this thesis focuses on the problem of multiple robots coordination in search and rescue mission. In particular, decentralized swarm algorithms, that use mechanisms based on Swarm Intelligence, are presented. Such techniques aim to improve the capabilities of each mobile robot, that is considered as an individual decision maker, in defining motion directives, making decisions about next actions, self-adaptability, using only simple rules and local interactions.

In recent years, indeed, there is an increasing interest in taking inspiration from natural phenomena for solving computational problems in robotics. Recent works show a potential in designing algorithms and appropriate models for robotic systems that, mimicking insect behaviour in nature, can solve complex tasks. Animals in nature, are able to adapt to dynamic changes that can occur in the environment, and through simple local interactions they can solve complex problems that are crucial for their survival.

In more detail, the work considers a team of simple mobile robots that have to explore an 2D unknown area in order to rescue and handle cooperatively some distributed hazardous targets. The desired behavior of the robotic system entails in multiple requirements, which may also be conflicting thus a good trade off must be found. The problem is formulated by defining an optimization model and then considering two-sub problems.
Firstly, the environment is incrementally explored by the robots using a modified version of Ant Colony Algorithm, that uses only indirect communication among the team. Then, when a robot detects a target, a recruiting mechanism is carried out in order to recruit a certain number of robots to deal with the found target together. For this latter purpose, the dissertation proposes, essentially, two approaches. The first uses an one-hop communication mechanism to spread locally the information among the robots and different bio-inspired meta-heuristics are proposed and compared. The second approach, is based on a multi-hop communication mechanism allowing, potentially, global information among the robots and trying to minimize the interactions among them and the wastage of resources. At this purpose, a decentralized bio-inspired protocol is developed and tested.

Both approaches have fully distributed intelligence without any leaders or central information distributor. In other words, they turn simplistic robots with limited capabilities into an intelligent, self-organized system with distributed intelligence for effective and efficient solutions to achieve collaboration and coordination among the members.

A computational study and extensive simulations have been carried out to assess the behaviour of the proposed approaches and to analyze their flexibility and effectiveness, studying also how the main variables of the problem can influence the solutions in various scenarios.

The proposal presented in this dissertation has paved the way for exploring new bio-inspired techniques for the coordination of a swarm of robots. It can be expected that it will inspire more active research in this exciting area with potentially more realistic real-world applications.

# Table of contents

**List of figures** xv

**List of tables** xix

**1 Introduction** 1

1.1 Multi-robot systems . . . 3
1.2 Swarm Intelligence . . . 4
1.3 Swarm Robotics . . . 6
1.3.1 Nature Inspired Robotics . . . 7
1.3.2 Advantages to Classical Approaches . . . 10
1.3.3 Application of swarm robotics . . . 10
1.3.4 Aspect of communication . . . 12
1.3.5 Features of Swarm Robotic Algorithms . . . 13
1.4 Main Contributions . . . 14
1.5 Structure of the thesis . . . 17

**2 Literature Review** 19

2.1 Multi-robot Exploration . . . 19
2.2 Multi-robot Allocation and Coordination . . . 22

**3 Preliminaries** 27

3.1 Assumptions of the Model . . . 28
3.1.1 Environment modeling . . . 28
3.1.2 Robots Properties . . . 30
3.1.3 Robot's actions . . . 32
3.2 Decision Making as Optimization in multi-robot teams . . . 34
3.2.1 Mathematical Model . . . 35
3.2.2 Energy Model . . . 37

**4 Searching Task** **39**
4.1 Ant-inspired techniques . . . . . . . . . . . . . . . . . . . . . . . . . . 40
4.2 Ant-Based Team Strategy for Robot Exploration (ATS-RE) . . . . . . . . . 41
4.2.1 Probabilistic Decision Making . . . . . . . . . . . . . . . . . . . . 43
4.2.2 ATS-RE Algorithm . . . . . . . . . . . . . . . . . . . . . . . . . . 45
4.3 Computational Experiments . . . . . . . . . . . . . . . . . . . . . . . . 45
4.3.1 Metrics . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 46
4.3.2 Case study 1: Influence of the parameters on the performance . . . 46
4.3.3 Case study 2: Influence of the number of robots on the performance 50
4.3.4 Case study 3: Influence of the dimension of the area on the performance 54
4.3.5 Comparing ATS-RE to other algorithms . . . . . . . . . . . . . . . 54
4.4 Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 56

**5 Recruitment Task** **59**
5.1 One-hop Communication . . . . . . . . . . . . . . . . . . . . . . . . . . 60
5.1.1 Firefly Algorithm . . . . . . . . . . . . . . . . . . . . . . . . . . . 60
5.1.2 Firefly based Team Strategy for Robots Recruitment (FTS-RR) . . . 62
5.1.3 Implementation of Robot Decision Mechanism . . . . . . . . . . . 62
5.1.4 FTS-RR Algorithm . . . . . . . . . . . . . . . . . . . . . . . . . . 64
5.1.5 Particle Swarm Optimization . . . . . . . . . . . . . . . . . . . . . 66
5.1.6 Particle Swarm Optimization for Robot Recruitment (PSO-RR) . . 66
5.1.7 PSO-RR Algorithm . . . . . . . . . . . . . . . . . . . . . . . . . . 67
5.1.8 Artificial Bee Colony Algorithm . . . . . . . . . . . . . . . . . . . 69
5.1.9 Artificial Bee Colony Algorithm for Robot Recruitment (ABC-RR) 70
5.1.10 ABS-RR Algorithm . . . . . . . . . . . . . . . . . . . . . . . . . . 72
5.2 Simulation Experiments . . . . . . . . . . . . . . . . . . . . . . . . . . . 72
5.3 Evaluation of the Performance in Static Conditions . . . . . . . . . . . . . 73
5.3.1 Test Parameters and Metrics . . . . . . . . . . . . . . . . . . . . . 73
5.3.2 Simulation Experiments I: Evaluation of the Time to complete the mission . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 73
5.3.3 Simulation Experiments II: Evaluation of the total energy consumed by the system . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 75
5.3.4 Case study 1: Influence of the size of the swarm and the dimension of the area on the energy consumption . . . . . . . . . . . . . . . . 78
5.3.5 Case study 2: Influence of the number of targets on the energy consumption . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 79

5.3.6 Case Study 3: Influence of the wireless range on the energy consumption . . . . . 81
5.3.7 Statistical tests . . . . . 83
5.3.8 Robot in dynamic scenario . . . . . 87
5.3.9 Influence of the coordination strategies introducing energy constraints 88
5.4 Balance the two tasks in Pareto Sense . . . . . 92
5.5 Multi-Objective Optimization Formulation . . . . . 92
5.6 Computational Experiments . . . . . 94
5.6.1 Evaluation of the Weights under Static Conditions . . . . . 94
5.6.2 Case study 1 . . . . . 94
5.6.3 Case study 2 . . . . . 95
5.6.4 Evaluation of weights under dynamic conditions . . . . . 99
5.7 Summaries . . . . . 107
5.8 Multi hop Communication . . . . . 108
5.9 ACO routing algorithms . . . . . 109
5.9.1 Data structures of ants . . . . . 110
5.10 Ant-Based Task Robot Coordination Protocol (ATRC) . . . . . 112
5.10.1 ATRC Communication Structure . . . . . 112
5.10.2 Forwarding mechanism of FANT and BANT . . . . . 115
5.10.3 Task Requesting BANT and FANT Management . . . . . 117
5.10.4 Recruitment FANT and BANT Management . . . . . 118
5.11 Simulations . . . . . 119
5.12 One Hop Communication VS Multi Hop Communication . . . . . 127
5.13 Summary . . . . . 129

**6 Conclusion and future work** **131**

# List of figures


3.1 A representation of the considered environment. . . . . . . . . . . . . . . . . 29
3.2 Possible movements for a robot . . . . . . . . . . . . . . . . . . . . . . . . 29
3.3 Recruitment's Process . . . . . . . . . . . . . . . . . . . . . . . . . . . 30
3.4 Local coalitions of robots formed through the recruitment's processes . . . 31
3.5 Robot's State change . . . . . . . . . . . . . . . . . . . . . . . . . . . 33
3.6 State transition for a robot . . . . . . . . . . . . . . . . . . . . . . . . 34

4.1 Example of pheromone diffusion. . . . . . . . . . . . . . . . . . . . . . . 42
4.2 The flow chart of Exploration task for a robot . . . . . . . . . . . . . . . . 44
4.3 Simulator Front-end . . . . . . . . . . . . . . . . . . . . . . . . . . . . 47
4.4 Impact of $\rho$ on the total time steps . . . . . . . . . . . . . . . . . . . . . 48
4.5 Impact of the dispersion coefficient on the total time steps . . . . . . . . . . 49
4.6 Total time steps need to explore the area (a) 30x30 grid cells (b) 50x50 grid cells (c) 100x100 grid cells . . . . . . . . . . . . . . . . . . . . . . . . 51
4.7 Total energy consumed by the system to explore the area (a) 30x30 grid cells (b) 50x50 grid cells (c) 100x100 grid cells . . . . . . . . . . . . . . . . . . 52
4.8 Performance of the ATS-RR considering a grid area 30x30 with and without obstacles (a) Total Time Steps (b) TESC (c) Average number of accesses for a cell . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 53
4.9 Influence of the dimension of the area on total time steps . . . . . . . . . . 54
4.10 ATR-RE vs IAS-SS (a) Free Environment (b) Obstacle Environment . . . . 55
4.11 Comparison among exploration algorithms in terms of total time steps . . . 56

5.1 Example of an overlap region . . . . . . . . . . . . . . . . . . . . . . . . 63
5.2 A possible selected cell. . . . . . . . . . . . . . . . . . . . . . . . . . . 64
5.3 Evaluation of Total Time Steps for performing 3 targets varying the number of robots needed to handle a target in an area 20x20 cells (a) 2 robots (b) 3 robots (c) 4 robots. . . . . . . . . . . . . . . . . . . . . . . . . . . . 76

5.4 Evaluation of the number of accesses for a cell for performing 3 targets in a grid area 20x20 varying the number of robots involved in the mission and varying the number of robots needed to handle a target in a area 20x20 cells (a) 2 robots (b) 3 robots (c) 4 robots. . . . . . . . . . . . . . . . . . . . . . 77
5.5 Evaluation of the Total Time Steps considering 30x30 grid area, 5 targets and 4 robots needed to handle a target. . . . . . . . . . . . . . . . . . . . . . 78
5.6 Evaluation of the Total Time Steps considering 40x40 grid area, 10 targets and 4 robots needed to handle a target. . . . . . . . . . . . . . . . . . . . . . 79
5.7 Evaluation of Total-Energy-System-Consumed (TESC) for performing 3 targets varying the number of robots involved in the mission considering 3 robots needed to deal with a single target (a) 40x40 grid (b) 50x50 grid (c) 60x60 grid. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 80
5.8 Evaluation of Total-Energy-System- Consumed (TESC) for performing 3,5,7,10 targets and 3 robots needed to perform a target (a) 40x40 grid (b) 50x50 grid (c) 60x60 grid. . . . . . . . . . . . . . . . . . . . . . . . 82
5.9 Evaluation of Total-Energy-System-Consumed (TESC) for performing 7,10,15 targets and 3 robots needed to perform a target in 50x50 grid (a) $R_t = 6$ (b) $R_t = 8$ (c) $R_t = 10$. . . . . . . . . . . . . . . . . . . . . . . . . . . . . 84
5.10 Percentage of Explored Cells evaluation for resolving 7 targets in a grid 50x 50 (a) 2 robots needed to disarm a target (b) 3 robots needed to resolve a target (c) 4 robots needed to resolve a target. . . . . . . . . . . . . . . . . . 90
5.11 Number of handled targets in a grid 50x 50 and 3 robots needed to resolve a target (a) 7 targets (b) 10 targets (c) 15 targets. . . . . . . . . . . . . . . . 91
5.12 Evaluation of the Total Time Steps to complete the mission and 3 robots needed to handle a target. (a) 50x50 grid area, 10 targets to disarm (b) 100x100 grid area, 20 targets to be disarmed. . . . . . . . . . . . . . . . . 96
5.13 Evaluation of the Total-Energy-System- Consumed (TESC) (a) 50x50 grid area, 10 mines to disarm and 3 robots needed to handle a target. (b) 100x100 grid area, 20 targets to disarm and 3 robots needed to handle a target. . . . 97
5.14 Evaluation of the total time steps to execute the mission in 50x50 squares and 40 robots and 3 robots to handle a target. (a) 15 targets to be handled. (b) 20 targets to be handled. (c) 35 targets to be handled. . . . . . . . . . . 98
5.15 Evaluation of the Total energy consumed by the system in 50x50 squares and 40 robots and 3 robots to treat a target. (a) 15 targets (b) 20 targets (c) 35 targets . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 100

5.16 Evaluation of the total time steps in a grid area 50x50. (a) 15 targets. (b) 20 targets. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 101
5.17 Evaluation of the total time steps in a 100x100 grid area, varying the dimension of the swarm with (a) 20 targets and (b) 30 targets. . . . . . . . . . . . 102
5.18 Evaluation of the total time steps varying $R_{min}$ in an area with 50x50 squares, 40 robots and 20 targets. . . . . . . . . . . . . . . . . . . . . . . . . . . 102
5.19 Percentage of unexplored cells in a 50x50 grid area, varying the dimension of the swarm of robots and 3 robots needed to handle a target. (a) 10 targets (b) 15 targets (c) 20 targets . . . . . . . . . . . . . . . . . . . . . . . . . 104
5.20 Percentage of disarmed targets in a 50x50 grid area, varying the dimension of the swarm of robots and 3 robots needed to handle a target. (a) 10 targets (b) 15 targets (c) 20 targets . . . . . . . . . . . . . . . . . . . . . . . . . 105
5.21 Percentage of alive robots in a 50x50 grid area, varying the dimension of the swarm of robots and 3 robots needed to handle a target. (a) 10 targets (b) 15 targets (c) 20 targets . . . . . . . . . . . . . . . . . . . . . . . . . . . 106
5.22 The Flow Chart of a Forager Robot . . . . . . . . . . . . . . . . . . . . 114
5.23 The flow Chart of a Coordinator Robot . . . . . . . . . . . . . . . . . . 115
5.24 The Flow Chart of a Recruited Robot . . . . . . . . . . . . . . . . . . . 115
5.25 The Flow Chart of Request FANT and BANT . . . . . . . . . . . . . . . 117
5.26 The Flow Chart of a Recruitment FANT and BANT . . . . . . . . . . . . 118
5.27 Influence of the $R_t$ on the total time steps in an area without obstacles (a) 30x30 grid area and 3 targets (b) 50x50 grid area and 5 targets. . . . . . . 120
5.28 Influence of the $R_t$ on the total time steps in an area with obstacles (a) 30x30 grid area with 3 targets (b) 50x50 grid area with 5 targets. . . . . . . . . . 121
5.29 Influence of the $R_t$ on the sent packets in an area without obstacles (a) 30x30 grid area and 3 targets (b) 50x50 grid area and 5 targets. . . . . . . . . . 122
5.30 Influence of the $R_t$ on the sent packets in an area with obstacles (a) 30x30 grid area and 3 targets (b) 50x50 grid area and 5 targets. . . . . . . . . . 123
5.31 Evaluation of the performance of ATRC in 30x30 grid area (a) Total Time Steps (b) Number of sent packets . . . . . . . . . . . . . . . . . . . . . . 124
5.32 Evaluation of the influence of the dimension of the area (a) Total Time Steps (b) Packets (c) Bit . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 125
5.33 Evaluation of average energy consumed by a robots varying the number of targets (a) 40x40 grid cell (b) 50x50 grid cell (c) 60x60 grid cell . . . . . . 126

# List of tables


5.1 Parameters used in the exploration algorithm. . . . . . . . . . . . . . . . . . 74
5.2 Cost related to the wireless communication. . . . . . . . . . . . . . . . . . . 74
5.3 Parameters used in the coordination algorithms. . . . . . . . . . . . . . . . . 74
5.4 Results of p value in t Test for FTS-RR and PSO-RR. . . . . . . . . . . . . 83
5.5 Results of p value in t Test for FTS-RR and ABC-RR. . . . . . . . . . . . . 83
5.6 Results of p value in t Test for PSO-RR and ABC-RR. . . . . . . . . . . . 85
5.7 Results of p value Test for FTS-RR, PSO-RR . . . . . . . . . . . . . . . . . 85
5.8 Results of p value Test for FTS-RR and ABC-RR. . . . . . . . . . . . . . . 85
5.9 Results of p value Test for PSO-RR and ABC-RR. . . . . . . . . . . . . . . 85
5.10 Results of p value in t Test for FTS-RR and PSO-RR. . . . . . . . . . . . . 85
5.11 Results of p value in t Test for FTS-RR and ABC-RR. . . . . . . . . . . . . 85
5.12 Results of p value in t Test for PSO-RR and ABC-RR. . . . . . . . . . . . 86
5.13 Results of p value in t Test for FTS-RR and PSO-RR related to the wireless range . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 86
5.14 Results of p value in t Test for FTS-RR and ABC-RR related to the wireless range . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 86
5.15 Results of p value in t Test for PSO-RR and ABC-RR related to the wireless range . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 86
5.16 Comparison of FTS-RR and ATRC: Total Time Steps . . . . . . . . . . . . 127
5.17 Comparison of FTS-RR and ATRC: Avarage Energy for a Robot . . . . . . 128

# Chapter 1

# Introduction

Over the past decade, the field of distributed robotics has been investigated actively, involving multiple, rather than single, robots. The field has grown dramatically, with a much wider variety of topics being addressed. Several new areas of applications of robotics, such as underwater and space exploration, hazardous environments, service robotics in both pù public and private domains, the entertainment field, and so forth, can benefit from the use of multi-robot systems. In these challenging application domains, multi-robot systems can often deal with tasks that are difficult, if not impossible, to be accomplished by an individual robot. A team of robots may provide redundancy and contribute cooperatively to solve the assigned task, or it may perform the assigned task in a more reliable, faster, or cheaper way beyond what is possible with single robot. However, the use of multiple robots poses new challenge; indeed the robots must communicate and coordinate in such a way that some predefined global objects can be achieved more efficiently.

An extensive amount of research has been carried out in the area of multi-robot coordination mechanisms. Within these settings, a key challenge is to find ways in which the members of the team can coordinate their decision processes in order to increase the overall performance of the collective. Moreover, such decision processes could consider multiple objectives, possibly conflicting.

Swarm robotics is a new approach to the coordination of multi-robot systems which consists of large numbers of mostly simple physical robots. It gets inspiration from Swarm Intelligence (SI) to model the behavior of the robots. Currently, swarm robotic algorithms are one of the most interesting research area in the robotics filed. The main open questions are:

- How to develop algorithms that allow cooperation among a swarm of robots with only simple and limited sensing and communication capabilities?

- How to describe the swarm robotics system in a mathematical model in order to predict the behaviors at both individual and swarm levels?
- How to efficiently share information among the swarm avoiding wastage of resources?
- How to adapt the robot's behaviors to unpredictable events that can occur especially in dynamic environments without decreasing the collective performance?

One of the most common approaches is to use biological inspiration, particularly social insects, in the development of similar behaviors in cooperative robot systems. Decentralized agents groups, typically, require complex mechanisms to accomplish coordinated tasks. In contrast, biological systems can achieve intelligent group behaviors with each agent performs simple sensing and actions. In these systems, each agent acts autonomously and interacts only with its neighbors, while the global system exhibits a coordinated and sophisticated behavior.

Biology-inspired meta-heuristic algorithms have recently become the forefront of the current research as an efficient way to deal with many NP-hard combinatorial optimization problems and non-linear optimization constrained problems in general (Yang 2008). These algorithms are based on a particular successful mechanism of a biological phenomena of mother nature in order to achieve the survival of the fittest in a dynamically changing environment. Examples of collective behaviour in nature are numerous. They are based, mainly, on direct or indirect exchange of information about the environment between the members of the swarm. Although the rules governing the interactions at the local level are usually easy to describe, the result of such behaviour is difficult to predict. However, through collaboration the swarms in nature are able to solve complex problems that are crucial for their survival.

On the basis of these considerations, the thesis presents an application of the swarm intelligence based approaches, that are strongly inspired by the biological behaviour of social insects, for the coordination of a swarm of robots involved in a search and rescue mission in hazardous environment. The approaches are completely distributed and no central control is used to coordinate the robots. Each of them utilizes only local information from its neighbors and then uses this information to make the best decisions by its point of view. The control law that each agent executes is simple, while the emerging global behavior is sophisticated and robust.

Although, the aim of this research is to develop effective coordination mechanisms for a team of mobile robots operating search and rescue in unknown and possibly hostile environments, the proposed approaches are generalized and they can be used for a wide range of applications with minor modifications.

## 1.1 Multi-robot systems

Research in autonomous robots has recently taken a new approach, namely, the multi-robot approach, in which systems are designed that distribute, to varying degrees, actuation and sensing, to perform tasks with or without some form of cooperation (C.R.Kube 1997).

Multi-robots systems (MRS) are a group of simple robots that are designed aiming to perform some collective behaviors. MRS have been proposed in the last decade in a variety of settings and frameworks, pursuing different research goals, and successfully applied in many application domains. The basic idea is that by this collective behavior, some goals that are impossible for a single robot to achieve become feasible and attainable. MRS have been more popular for their benefits compared to the single robot systems. These benefits include, but are not limited to the following (Khamis et al., 2015 ):

- *Resolving task complexity*: some tasks may be quite complex for a single robot to do or even it might be impossible. This complexity may be also due to the distributed nature of the tasks and/or the diversity of the tasks in terms of different requirements.

- *Increasing the performance*: task completion time can be dramatically decreased if many robots cooperate to perform it in parallel.

- *Increasing reliability*: increasing the system reliability through redundancy because having only one robot may work as a bottleneck for the whole system especially in critical times. Moreover, when having multiple robots doing a task and one fails, others could still do the job.

- *Simplicity in design*: having small, simple robots will be easier and cheaper to implement than having only a single powerful robot.

These kinds of systems are well suited for several application domains, which require high complexity coordinated tasks to be performed and in many applications that are too risky for humans (Gautam and Mohan, 2012) . Examples of these applications could be surveillance (Calvo and et.al., 2011), disaster relief (Gregory et al., 2016), environment monitoring, battlefield missions, urban search and rescue mission (Basilico and Amigoni, 2011) , collective transport (Rubenstein et al., 2013), mine-clearing (De Rango and Palmieri, 2012).

A critical undertaking in the design of task-directed multi-robot systems is related to the managing the complexity introduced by the coordination of multiple, interacting robots. For such systems to be effective, the robots' actions must be carried out in a coordinated fashion and directed toward task achievement. Typically, the system effectiveness strongly depends on the underlying coordination mechanisms used to mediate the interactions among the robots and between the robots and the task environment. Furthermore, when designing multi-robot systems, various aspects must be considered: whether the system's control is centralized or decentralized; the ability of one robot to make decisions that globally have benefits among the team, what kind of communication structure should be used.

As a result of the growing focus on multi-robot systems, the field of multi robot coordination has been object of considerable research efforts in the last years and many procedures and methods have been studied and proposed. Recently one of the most common approaches is to take inspiration from the behaviour of some kinds of animals in nature (Farinelli et al.), (2004).
Swarm robotics is a field of multi-robotics in which large number of robots are coordinated in a distributed and decentralized way. It is based on the use of local rules, and simple robots compared to the complexity of the task to achieve, and inspired by social insects. Researchers have demonstrated that individuals in nature do not need any representation or sophisticated knowledge to produce such complex behaviors. Especially in social insects, the individuals are not informed about the global status of the colony. There exists no leader that guides all the other individuals in order to accomplish their goals. The knowledge of the swarm is distributed throughout all the individual agents, where an individual is not able to accomplish its task without the rest of the swarm.

## 1.2 Swarm Intelligence

Swarm intelligence (SI) is an artificial intelligence technique based on the study of the collective behavior of decentralized, self-organized systems, natural or artificial. A swarm intelligence system usually consists of a group of simple autonomous individuals/agents interacting locally with one another and with their environment. While each agent can be considered as unintelligent, the whole system of multiple agents may show some self-organization behaviour and thus can behave like some sort of collective intelligence. The intelligent behaviour emerges from the collectivity and not relies on individuals capabilities.

The objective of SI is to model simple behaviors of the individuals, their local interactions with the environment and neighboring individuals, in order to obtain more complex behaviors that can be used to solve complex problems, mostly optimization problems. As a result, the swarm can complete the tasks without the need of centralized control and global model providing a great solution for large-scale complex problems (Tan and yang Zheng, 2013).

A typical swarm intelligence system has the following properties:

- The swarm is composed of many individuals.

- Generally, the individuals are homogeneous (either all identical or they belong to a few typologies).
- The interactions among the individuals are based on local information; that is, each individual can share directly or via environment (stigmergy).
- The overall behaviour of the system results from the interactions of individuals with each other and with the environment.

The inspiration comes from the nature, especially biological systems. The agents follow very simple rules, and although there is no centralized control structure dictating how individual agents should behave, local, and to a certain degree random, interactions between such agents lead to the emergence of an intelligent global behavior, unknown to the individual agents. Natural examples of SI include ant colonies, bird flocking, animal herding, bacterial growth, and fish schooling. The key factor is the self-organization, that leads to a set of dynamical mechanisms whereby structures appear at the global level of the systems from interactions among its lower-level components (Bonabeau et al., 1999), (Jevtić and de la Fuente, 2007). Generally, the self-organization relies on the following characteristics:

- *Positive feedback* (amplification): the examples are recruitment and reinforcement. For instance, recruitment to a food source is a positive feedback that relies in trail-laying and trail-following in some ant species, or bees.
- *Negative feedback*: counterbalances positive feedback helps to stabilize the collective pattern: it may take the form of saturation, exhaustion, or competition. In the example of foraging, negative feedback stems from the limited number of available foragers, satiation, food source exhaustion, crowding at the food source, or competition between food sources.
- *Amplification of fluctuations* (random walks, errors, random task-switching, and so on). Not only do structures emerge despite randomness, but randomness is often crucial, since it enables the discovery of new solutions, and fluctuations can act as seeds from which structures nucleate and grow. For example, foragers may get lost in an ant colony, because they follow trails with some level of error; although such a phenomenon may seem inefficient, lost foragers can find new, unexploited food sources, and recruit nest mates to these food sources.
- *Multiple interactions*: a minimal density of mutually tolerant individuals is required to generate a self-organized structure.

Many algorithms have been developed by drawing inspiration from swarm-intelligence systems in nature. All SI-based algorithms use multi-agents, inspired by the collective behaviour of social insects, like ants, termites, bees, and wasps, as well as from other animal societies like flocks of birds or fish. It is possible to cite the classical particle swarm optimization (PSO) (Kennedy and Heberhart, 1995) that uses the swarming behaviour of fish and birds while the Firefly algorithm (FA) (Yang, 2009) uses the flashing behaviour of swarming fireflies. Cuckoo search (CS) (Yang and Deb, 2009 ) is based on the brooding parasitism of some cuckoo species, while bat algorithm uses the echolocation of foraging bats. Ant colony optimization (Dorigo and Stutzle, 2003 ) uses the interaction of social insects (e.g., ants), while the class of bee algorithms (Karaboga and Akay, 2009) are all based on the foraging behaviour of honey bees.

SI-based algorithms are among the most popular and widely used. There are many reasons for such popularity, one of the reasons is that SI-based algorithms usually sharing information among multiple agents, so that self organization, co-evolution and learning during iterations may help to provide the high efficiency of most SI-based algorithms. Another reason is that multiple agent can be parallelized easily so that large-scale optimization becomes more practical from the implementation point of view (Fister Jr. et al., 2013), (Jevtić and de la Fuente, 2007 ).

## 1.3 Swarm Robotics

When Swarm Intelligence is applied to the robotic systems, it is referred to Swarm Robotics. Swarm robotics (SR) has been defined as “a novel approach to the coordination of large numbers of robots” and as “the study of how large numbers of relatively simple physically embodied agents can be designed such that a desired collective behavior emerges from the local interactions among agents and between the agents and the environment” (Şahin, 2005 ).

Swarm robotics systems typically exhibit interesting properties such as high degrees of parallelism and redundancy. Building on these properties, these systems can be engineered to be highly adaptive to changes in the environment, to be robust to unexpected events and failures, and to show good scalability to increased problem and/or swarm size. On the other hand, they may be less resource efficient than traditional systems, and may not guarantee optimal solutions to given problems. This approach, essentially, emerges in the field of swarm intelligence and it is strongly inspired from the nature where a swarm behavior can occur. The main characteristics of a swarm robotics system are as follow (Brambilla et al., 2012):

- Robots are autonomous although they are able to interact with the environment and with the other members.
- Robots are situated in the environment and can act to modify it.
- Robots are in a large number in order that the cooperative behavior may occur. The minimum number is hard to define and justify and it depends mostly on the domain and applications.
- Robots' sensing and communication capabilities are local and limited. This means that the robots should be relatively incapable or inefficient on their own with respect to the task at hand.
- Robots do not have access to centralized control and/or to global knowledge thus the coordination between the robots is distributed.
- Robots cooperate to tackle a given task.

An interesting problem in swarm robotics is how one can instruct robots in the swarm to make collective and optimal decisions in order to cooperate for executing some assigned tasks. Decision making is a complex problem for a collective robotic system, due to the necessity to reach a global consensus among the robots, which contrast with the system's inherent decentralization (Trianni et al., 2007 ). Usually, SI techniques can be used as control algorithm for distributed robot swarms. In these applications, a distributed control architecture generally is preferred compared to a centralized architecture to prevent single point failures.

### 1.3.1 Nature Inspired Robotics

The design of robot swarm systems, motivated by emergent phenomena observed in biological systems, has attracted much interest in recent times. A key feature in this context is self-organization in social insect groups such as ant colonies that, operating in a collective way, exhibit remarkable adaptability also in dynamic and uncertain environments to form coherent patterns even in the absence of leadership within the group. Indeed, the ability to replicate such behavior is an idea that offers immense promise for the design and potential applications of a robot swarm (Schroeder et al., 2017).

The idea of designing multi-robot systems, that emulate features and behaviour of animals in nature, has a long history. Biological organisms have evolved to perform many complex tasks and survive in a world characterized by rapid changes, high uncertainty, indefinite richness, and limited availability of information (Pfeifer et al., 2007 ).

Bio-inspiration has driven research and applications on robot locomotion (crawling, walking, running, climbing, jumping, swimming, and flying), navigation and orientation behaviors, spatial memory formation, exploration, environmental monitoring, manipulation, imitation, and cooperation, among others (Delcomyn, 2007).

The most common source of inspiration is done in the use of the simple local control rules of various biological societies, particularly social insects such as ants, bees, birds, fireflies, to the development of similar behaviors in cooperative robot systems. These kind of animals provide one of the best-known examples of biological self organized behaviour. They show very poor abilities, and through local and limited communication, they are highly organized colonies demonstrating impressive intelligent group behaviors allowing them to perform complex tasks.

Surprisingly, the complexity of these collective behaviors and structures does not reflect all the relative simplicity of the individual behaviors of an insect. Of course, insects are elaborated "machines", with the ability to modulate their behavior on the basis of the processing of many sensory inputs (Giurfa and Menzel, 2003), (Detrain and Deneubourg, 2006 ). Nevertheless, as pointed out by Seeley, 2002, the complexity of an individual insect in terms of cognitive or communicational abilities may be high in an absolute sense, while remaining not sufficient to effectively supervise a large system and to explain the complexity of all the behaviors at the colony scale. In most cases, a single insect is not able to find by itself an efficient solution to a colony problem, while the society to which it belongs finds "as a whole" a solution very easily (Camazine et al., 2001). Behind this "organization without an organizer" there are several hidden mechanisms which enable insect societies, whose members only deal with partial and noisy information about their environment, to cope with uncertain situations and to find solutions to complex problems.

Organization emerges at the colony level from the interactions that take place among individuals exhibiting these simple behaviors. These interactions ensure the propagation of information through the colony and they also organize the activity of each individual. Thanks to these sophisticated interaction networks, social insects can solve a whole range of problems and respond to external challenges in a very flexible and robust way (Garnier et al., 2007).

In social insects, the individuals are not informed about the global status of the colony. There exists no leader that guides all the other individuals in order to accomplish their goals. The knowledge of the swarm is distributed throughout all the agents, where an individual is not able to accomplish its task without the rest of the swarm. Social insects are able to exchange information, and for instance, communicate the location of a food source, favor a foraging zone or inform the presence of danger to their mates. The interaction between the individuals is based on the concept of locality, where there is no knowledge about the overall situation In social insects, the individuals are not informed about the global status of the colony.

Organization emerges from the interactions between the individuals and between individuals and the environment and they are propagated throughout the colony and therefore the colony can solve tasks that could not be solved by a sole individual. The collective behaviour can be indeed as self-organizing behaviour. Self-organization theories, borrowed from physics and chemistry domains, can be used to explain how social insects exhibit complex collective behaviour that emerges from interactions of individuals behaving simply. Self-organisation relies on the combination of the following four basic rules: positive feedback, negative feedback, randomness, and multiple interactions. Sahin (Şahin, 2005) lists some properties seen in social insects as desirable in multi-robotic systems:

- *robustness*: the robots swarm must be able to work even if some of the individuals fail, or there are disturbances in the environment;
- *flexibility*: the swarm must be able to create different solutions for different tasks, and be able to change each robot role depending on the needs of the moment;
- *scalability*: the swarm should be able to work in different group sizes, from few individuals to many of them.

An example of the behavior of social insects is done by the colony of ants in nature. The interactions between the ants, mostly, are based on an implicit communication that causes modification in the environment where they live in what is called stigmergy. The principle is easy: ants deposit a pheromone trail on the path that take during travel. Using this trail, they are able to navigate toward their nest or food and communicate with their peers. Organization emerges from the interactions between the individuals and between the individual and the environment. These interactions, then, are propagated among the colony

Swarm optimization algorithms, like ant colony optimization (ACO) (Dorigo and Stutzle, 2003), rely on pheromone trails to mediate communication between the individuals. Such insect-inspired technique is highly used in robotic systems for a wide range of applications such as surveillance (Calvo and et.al., 2011); exploration (Ravankar et al., 2016), (Chen et al., 2013), (Palmieri et al., 2017b); recruiting (De Rango and Palmieri, 2012); pattern formation for citing something.

### 1.3.2 Advantages to Classical Approaches

Some advantages of swarm-robotic systems make them more appealing then classical robotics. The main characteristics can be summarized as follows (Yogeswaran and Ponnambalam, 2010 ), (Şahin, 2005 ):

1. *Parallelism*: in complex applications robots can accomplish a given task more quickly than a single robot by dividing the task in sub-tasks and executing them concurrently.

2. *Robustness*: the system no has point of failures, which means that the task can be successful completed even if a single robot stops working or disturbance in the environment. The robustness can be attributed to several factors. Firstly is the redundancy in the system, that means any loss or malfunction of an individual can be compensated by the others. Secondly, using a decentralized coordination and a decentralized approach can allow to avoid any central control and obtain a computationally cheap and useful solution. Thirdly, the simplicity of the robots in term of capabilities that make them less prone to failure.

3. *Scalability*: the interactions in the swarm are local, allowing the robots to join or quit the task at any time without interrupting the whole swarm. The swarm can adapt to the changes without the need of any external operations. This also indicates that the system is adaptable for different sizes of robots without any modification of the software or hardware which is very useful for real-life application.

4. *Flexibility*: the approaches are generalized and can be used for a wide range of applications with minor modifications.

5. *Complex tasks*: tasks may be too complex or impossible to be accomplished by a single robot and the performance benefits can be gained using a swarm of robots.

6. *Low cost*: building and using several simple robots can be easier, cheaper, more flexible than having single powerful robot.

### 1.3.3 Application of swarm robotics

According to recent literature reviews, swarm robotics has been studied in the context of the following tasks:

- *Aggregation* - The goal of aggregation is to group many robots of a swarm in a certain region of the considered environment (Gauci et al., 2014). It is used as starting point for performing some additional cooperative tasks such as manipulation of complex targets (Palmieri et al., 2017b ), collective transportation (Rubenstein et al., 2013). Aggregation is very common behaviour in nature. It is can be observed in bacteria, cockroaches, bees, fish, penguins, ants.

- *Pattern formation* - The goal of patter formation is to deploy the robots in the environment forming a regular geometric pattern such as a circle, a square, a line, a star. Robots usually need to respect many constraints such as keep a specific distance between each other in order to create a desired patter (Bahceci et al., 2003 ).

- *Self-Assembly* - The goal is to connect the robots to form a particular structure in order to facilitate a given task. Self assembly can be observed in several species of ants, that are able to physically connect for performing different tasks (Groß et al., 2006 ).

- *Object clustering and assembling* - The goal is to group objects in specific regions of the environments. In the clustering task there is no connection among the objects, while in the assembling task the objects could be physically linked together. These kind of behaviors are displayed by social insects such as ants and termites that are able to build complex nests.

- *Swarm Navigation* - The goal is to coordinate the movements of the robots with simple rules. Robots are not aware, generally, of their positions and they are guided by the others. The applications regard collective exploration of an environment, area coverage, coordinated motion also knows as flocking, in which group of autonomous robots navigate into the environment avoiding collision and improve the abilities of the swarm (Ducatelle et al., 2014 ). An other application is collective transport also known as group prey retrieval.

- *Mapping* - Is the problem of obtaining a map of the environment using a robot swarm. The goals are two. Firstly, it allows to construct a map of the environment in order to map unknown area. Secondly, assisting the navigation of the swarm reducing the need for beacons and swarm guided navigation techniques (Rothermich et al., 2005).

- *Collective decision making* - It deals with how robots influence each other in making decisions. It can be used to consensus achievement, that means the swarm converges towards a single decision among possible alternatives in order to maximize the performance of the system. In nature ants use pheromone to find the shortest path between the nest and the source of food, the bees have mechanisms to collectively decide which is the best nest locations (Trianni et al., 2007 ). Moreover, task allocation is another example of collective decision, in which the robots distribute themselves among different tasks in order to maximize the performance of the system (Parker, 2012 ).

The aforementioned applications have been studied together or separately and are also being studied to real applications. The number of possible applications is really promising, but still the technology must be developed both in algorithmic and modeling part, and also in the miniaturization technologies (Navarro and Matía, 2010).

Moreover many projects confirm the enormous interest in the field of Swarm Robotics. It is possible to cite:

- Project SI (Zhu, 2011) consists of a swarm of mobile robots, controlled by swarm inspired algorithm.
- Swarm-bots project (Dorigo and et.al, 2005) for exploring the design, implementation and simulation of self-organizing and self-assembling artifacts using simple and cheap components. The inspiration is the collective behavior of social insects.
- Symbrion project (Guy and et.al, 2009) addresses both hardware, middleware and software issues.
- Swarmanoid project (Dorigo and et.al, 2013) is the extension of Swarm-bots project to three dimensional environment.
- Replicator project (Kernbach and et.al, 2008 ) deals with issues as reconfigurability of sensors and actuators, adaptive control and learning strategies of robots working, potentially, in real world.

### 1.3.4 Aspect of communication

In order to effectively perform a task, the robots need to cooperate. Cooperation among robots is often obtained by a communication mechanism that allows the robots to exchange messages. Basically, there are three ways of information sharing in the swarm: direct communication, indirect communication and sensing (Tan and yang Zheng, 2013).

Direct communication consists of exchange of messages and information between the swarm, generally obtained through using a wireless communication. This kind of communication allows to the robots to have potentially global knowledge and accurate information of the environment, but requires frequently interactions among the robots and a wastage of resources such as energy, memory and so on. Some forms of direct communication within insect societies have been studied, a well-known example being the waggle dance of honey bees. A bee is able to indicate to the unemployed workers the direction and distance from the hive of a patch of flowers, using a "dance" that gives also information on the quality and the richness of the food source.

On the contrary, indirect communication, generally, occurs through interactions between the robots and the environment. The robots can leave their trace of pheromone (such as in ants) in the environment after an action so that the others can sense this modification without any communication among them and then make eventually decisions. The object is to develop simple algorithms that can utilize the concept of the pheromone to achieve a complex collective behavior. This kind of communication has been observed especially in the foraging behaviour of many ants species, which lay a trail of pheromone, thus modifying the environment in a way that can inform other individuals of the colony about the path to follow to reach a profitable foraging area. This form of communication that, takes place through the environment, has as result of the actions performed by some individuals, which indirectly influence someone else's behaviour. Communication between robots can multiply their capabilities and increase the efficiency.

Sensing is a mechanism where the robots can sense the environments using on-board sensors in order to distinguish the robots and other objects in the environments for performing tasks. The aim is to integrate the sensors efficiently for cooperation.

As described above, all these forms of communication can be observed in biological systems, and in particular in social insects. Research in swarm robotics focuses on the application of these simple forms of communication to artificial, autonomous systems. Even though there is no clear conclusion on what type of communication is better for robot swarms, most of the current research is aiming towards a form of communication that can guarantee robust characteristics . Indeed, the domains in question do not permit reliance on complex communication but can be made significantly more manageable through the use of minimalist communication schemes. In many cases, simple forms of communication are enough to obtain the coordination of the activities of the group (Trianni et al., 2004), (Mataric, 1998).

### 1.3.5 Features of Swarm Robotic Algorithms

A variety of algorithms has been implemented to be run on swarms of robots. Some provide basic functionality, such as dispersion, while others demonstrate seemingly complex teamwork, such as chain formation. Although the algorithms produce different emergent behavior, they all have many features in common. Many research works show that swarm robotics algorithms are scalable, fault tolerant, robust and efficient. The algorithms are based on the idea that complex macro-level behaviors can emerge from simple and local interactions between agents.

These features derive from the basic goals of swarm robotics discussed earlier. Firstly, the algorithms should be simple and elegant, which means the robot controller that dictates the behavior of the individual robots is very simple. Secondly, the algorithms should be designed to be scalable so that they work for any number of robots. Also, they are expected to scale well as new robots are added. Moreover, they should be decentralized in order to avoid any central controller exterior commands, since the robots do not depend on one another. If a single module fails, the rest of the swarm can continue performing its actions as if that module never existed. Meanwhile, an individual robot system may become worthless if there is a failure in a critical component. Moreover, majority of the algorithms use local interactions over broadcasting messages and these broadcasts are used as message hopping protocols (Miner, 2007).

## 1.4 Main Contributions

The thesis addresses the problem of coordination of a multi-robot system in the context of search and rescue scenarios. The work is divided into two major phases that are the main contributions of the dissertation. The first is the exploration stage that aims to explore an unknown area in order to detect many targets, potentially hazardous, and it is mainly implemented through a modified version of Ant-based strategy which uses only a indirect communication among the swarm of robots. Secondly, the problem to form coalitions at the target's locations in order to manipulate cooperatively is treated. For this latter purpose, nature-inspired approaches are presented and compared in order to analyze the performance. Moreover, an analytical mathematical model for this problem is presented.

Therefore, the research reported makes some new contributions in this area:

- The exploration and handling problem is described using an optimization model. Both the search and handling of the targets are considered together.
- A repulsive mechanism based on Ant Colony Optimization is applied as an indirect coordination mechanism (stigmergy) for the exploration task.
- The recruitment problem to form coalition at the target's locations is treated applying some bio-inspired techniques and considering a direct communication among the robots. The problem is treated using local communication between the swarm and a local and selfish decision mechanism. In the first stage of the work, it is used only

a one hop communication mechanism and some nature-inspired algorithms such as Firefly Algorithm, Particle Swarm Optimization and Bee Algorithm are proposed and modified to fix to the considered problem. Comparisons among different algorithms is done in order to assess which algorithm potentially is the best considering many dynamic scenarios where failure and dangerous events suddenly can occur. Then a multi-hop communication mechanism is proposed and an Ant-inspired recruiting protocol is designed to resolve the coordination problem. In this case the data exchange are balanced with stigmergy in order to assure scalability in the robots communication and in order to scale well in the problem complexity.

Publications (published and submitted) related with the topics in this thesis are:

- *Conferences*

  - N.Palmieri, X.S. Yang, F. De Rango "Self-Adaptive Mechanism for Coalitions Formation in a Heterogeneous Robot Network" in *The 21st International Symposium on Distributed Simulation and Real Time Applications (DS-RT 2017)*, Rome, Italy.
  - F. De Rango, N. Palmieri, M. Tropea, G. Potrino “UAVs team and its application in Agriculture: A Simulation Environment”, in *International Conference on Simulation and Modeling Methodologies, Technologies and Applications (SIMULTECH 2017)*, Madrid, Spain.
  - De Rango Floriano, Palmieri Nunzia, Amilcare Francesco Santamaria, Potrino Giuseppe. “A Simulator for UAVs Management in Agriculture Domain”, in International Symposium on Performance Evaluation of Computer and Telecommuncation Systems (SPECTS 2017), Seattle, Wa, USA.
  - N. Palmieri, X.S. Yang, S. Marano. “Coordination Techniques of Mobile Robots with Energy Constraints”, in *International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS 2016)*, Montreal, Quebec, Canada.
  - F. De Rango and N. Palmieri. “ Ant-based distributed protocol for coordination of a swarm of robots in demining mission”, in *Proc. Of SPIE*, Baltimore, USA, 2016
  - F. De Rango, N. Palmieri, X.S. Yang, S. Marano. “Bio-inspired Exploring and Recruiting Tasks in a Team of Distributed Robots over Mined Regions”, in *International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS 2015)*, Chicago, USA.
  - F. De Rango, N. Palmieri, S. Ranieri. “Spatial Correlation based Low Energy Aware Clustering (LEACH) in a Wireless Sensor Networks”, in *Knowledge in Telecommunication Technologies and Optics 2015 (KKTO 2015)*, Ostravice, Czech Republic.
  - N.Palmieri, F. De Rango, X.S. Yang, S. Marano. “Multi-Robot Cooperative Tasks Using Combined Nature-Inspired Techniques”, in *7th International Conference on Evolutionary Computation Theory and Applications*, ECTA 2015, Lisbon (Portugal).

- F. De Rango, N. Palmieri: “A Swarm-based robot team coordination protocol for mine detection and unknown space discovery”, *Wireless Communications and Mobile Computing Conference (IWCMC)*, 2012 8th International, Limassol (Cyprus).

- *Book Chapters*

  - N. Palmieri, F. De Rango, X.S. Yang, S. Marano, "Bio-inspired Strategies for the Coordination of a Swarm of Robots in an Unknown Area". In: Merelo J. et al. (eds) *Computational Intelligence*. IJCCI 2015. Studies in Computational Intelligence, vol 669. Springer (2017)
  - N. Palmieri, S. Marano. “Discrete Firefly Algorithm for Recruiting Task in a Swarm of Robots”, *Nature-Inspired Computation in Engineering*, Volume 637 of the series Studies in Computational Intelligence pp 133-150, Springer International Publishing Switzerland (2016)

- *Journals*

  - F. De Rango, N. Palmieri, X.S. Yang, S. Marano " Swarm robotics in wireless distributed protocol design for coordinating robots involved in cooperative tasks", *Soft Computing (2017)* , DOI 10.1007/s00500-017-2819-9.
  - N. Palmieri, Xin-She Yang, F. De Rango, S. Marano, “Comparison of bio-inspired algorithms applied to the coordination of mobile robots considering the energy consumption", *Neural Computing & Applications (2017)*, DOI: 10.1007/s00521-017-2998-4.
  - F. De Rango, N. Palmieri, S. Ranieri: "Spatial Correlation Based low Energy Aware Clustering (Leach) In A Wireless Sensor Networks” in *Advances in Electrical and Electronic Engineering*, Vol.13, Issue 4, (2015), pp-350-358.

## 1.5 Structure of the thesis

Chapter 1 (this chapter) outlines the motivations of the research, highlights the main contributions of this thesis as well as the summary and structure of the thesis.

Chapter 2 gives a more detailed account about current works regarding the application of the robotics in search and rescue task. It focuses on the description of many approaches and algorithms regarding mainly the exploration problem and the problem of coordination of the robots in many domains.

Chapter 3 describes in greater detail the problem statement, the mathematical model, the characteristics of the robots in terms of capabilities and resources and other constraints regarding the communication, sensing and energy. Moreover, it also describes the environment where the robots operate.

Chapter 4 introduces an inverse Ant-based strategy as exploration mechanism. A set of experiments are conducted to test the performance of the algorithm considering different parameters of the problem. Moreover, a comparison between other algorithms is done.

Chapter 5 highlights the recruiting task and it is divided into two main parts. The first describes different modified meta heuristics such as Firefly Algorithm, Particle Swarm Algorithm and Distributed Bee Algorithms as recruiting mechanism to form coalitions in targets's location with the assumption of one hop communication with minimal interactions between the swarm of robots. A computational study and extensive simulations have been carried out to assess the behaviour of the proposed approaches and to analyze their performance considering various scenarios, parameters and metrics. The second part introduces a multi hop communication and an Ant-based protocol to recruit the robots is implemented. The protocol has been tested and analyzed considering different experimental scenarios to suitably evaluate the efficiency of the protocol in recruiting task by varying several parameters of the problem and network conditions.

Finally, Chapter 6 concludes this thesis and outlines some directions for future research.

# Chapter 2

# Literature Review

Research related to this dissertation includes the topics of multi robot exploration and multi-robot coordination. The chapter provides a review of some of the most relevant works. First, exploration algorithms are reviewed in Section 2.1. Then in Section 2.2 multi-robot coordination approaches are presented.

## 2.1 Multi-robot Exploration

Multi-robot exploration has received much attention in the research community. The unknown area exploration should not lead to an overlapping in robots movements and ideally, the robots should complete the exploration of the area with the minimum amount of the time. The overlapped area can occur when a location has been visited by one of the robots and it is visited again by the same or different robots of the team. Many approaches have been proposed for exploring unknown environments with a team of mobile robots.

Some exploration plans in the context of mapping are usually constructed without using environmental boundary information. One of the well-known techniques is the frontier based exploration, which was proposed by (Yamauchi, 1999). In this approach, these robots act independently and make probabilistic judgments regarding frontiers areas of unexplored space in an environment. The environment is decomposed into cells with each cell being represented by a probability value, and can be classified as either free, occupied or unknown. Using this representation, a robot can reach an unexplored zone by means of navigating to the frontier cells that separate the free cells from the unknown cells. Evaluating candidate locations, different criteria can be used. A simple one is the distance from the current position of the robot (Yamauchi, 1999) according to which the best observation location is the nearest one. However, most works combine different criteria in more complex utility functions. For example, (Burgard et al.2005) coordinated the robots in order to explore as much area as possible. A decision-theoretic approach trades off the utility and the cost of visiting targets. The cost of a target is defined as the length of the optimal path from a robot to it, whereas the utility of a target is defined as the area expected to be found when the robot arrives at it.

On the other hand, some researchers are focusing on the exploration by using knowledge about environmental boundary information, as described in (Choset et al., 2000), (Wattanavekin et al., 2013 ). The authors assume that the robots already have the information of all obstacles. Therefore, when a robot encounters an obstacle, it can immediately grasp the obstacle. However, this is not practical in real-world applications considering the unknown area.

Other approaches, proposed by (Solanas and Garcia, 2004,) and (Gifford et al., 2010), coordinate the robots by means of dividing the environment into as many disjoint regions as available robots and assigning a different region to each robot.

Tree-cover algorithms, instead, use a pre-calculated spanning-tree to direct the exploration effort and distribute it among the agents. These algorithms require a priori knowledge of the environment. A typical example is the Multi-Robot Forest Coverage (MFC) algorithm, described in (Zheng et al., 2010 ) and Multirobot Spanning Tree Coverage (MSTC) algorithm proposed by (Hazon and Kaminka,2008 ).

However, in the real scenario, especially in search and rescue mission, the considered area could have some uncertainty, thus build accurate maps may be problematic. Bio-inspired techniques have recently gained importance in computing due to the need for flexible, adaptable ways of solving engineering problems. Within the context of swarm robotics, most works on cooperative exploration are based on biologically behaviour and indirect stigmergic communication (rather than on local information, which can be applied to systems related to GPS, maps, wireless communications). This approach is typically inspired by the behaviour of certain types of animals and insects, like the ants, that use chemical substances known as pheromone to induce behavioural changes in other members of the same species.

One of the well known is inspired by the collective behaviour of insect colonies such as ants and fireflies (Dorigo and Stutzle, 2003), (Yang, 2008 ). These algorithms emphasis on decentralized local control, local communication and on the emergence of global behaviour as the result of self-organization. Ant and other social animals are known to produce chemical substances called pheromone and use them to mark the paths in the environment that is used as a medium for sharing information. Pheromone trails provide a type of distributed information that artificial agents may use to make decisions. Many works can be found in the literature using this kind of biology metaphor. Wagner et al. (Wagner et al., 1999) were the first who invested stigmergic multi-robot coordination for covering/patrolling the environment. In their approach a group of robots is assumed able to deposit chemical odor traces and evaluate the strength of smell at every point they reach. Based on these assumptions, they used robots to model an un-mapped environment as a graph and they proposed basic graph search algorithms for solving mainly robotic coverage problems. Kuyucu et al. (Kuyucu et al., 2015) used a guided probabilistic exploration of an unknown environment achieved via combining random movement with pheromone-based stigmergic guidance.

Chen et al. (Chen et al., 2013 ) proposed a fast two-stage ACO algorithm which overcomes the inherent problems of traditional ACO algorithms. The basic idea is to split the heuristic search into two stages: preprocess stage and path planning stage. In the preprocess stage, the scent information is broadcasted to the whole map and then ants do path planning under the direction of scent information.

Ducatelle et al. (Ducatelle et al., 2011 ) uses a swarm of wheeled robots, called foot-bots, and a swarm of flying robots that can attach to the ceiling, called eye-bots that serve as stigmergic markers for foot-bot navigation. However, in the exploration task, researchers use the concept of anti-pheromone so as to try to maximize the distance between the robots and to enforce a dispersion mechanism in different sites of the region of interest, with the aim to accomplish the mission as quickly as possible. Some examples of this approach can be found in (Calvo and et.al., 2011) and (Doi, 2013 ) for surveillance mission, in (Pearce et al., 2006), (Palmieri et al., 2017b) for guide the robots in search and rescue in a disaster site, in (Osherovich et al., 2008) and (Ranjbar-Sahraei et al., 2012 ) in multi-robot coverage. Ravankar et al. (Ravankar et al., 2016) uses a hybrid communication framework that incorporates the repelling behaviour of the anti-pheromone and attractive behaviour of pheromone for efficient map exploration.

The use of physical substances for pheromone-based communication within robots is problematic and poorly understood. However, there is undergoing work in improving their use with promising results, and it is predicted that with improvements in sensing technology it may be possible that a robot could carry a lifetime supply of chemicals (Purnamadjaja and Russell, 2010 ).

Other authors like (Payton et al., 2001 ) described techniques for coordinating the actions of large numbers of small-scale robots to achieve useful large-scale results in surveillance, reconnaissance, hazard detection, and path finding. using the notion of a "virtual pheromone," implemented using simple transceivers mounted to each robot. Unlike the chemical markers used by insect colonies for communication and coordination, our virtual pheromone is a symbolic message tied to the robots themselves rather than to fixed locations in the environment.

Chemical trail-following strategies have been implemented with real robots. For example, ethanol trails were deposited and followed by the robots: high concentrations of the pheromone yield high signal strength but the signal duration is short, while low pheromone concentrations yield low signal strength but a long signal duration in Fujisawa et al. (Fujisawa et al., 2008), but the use of decaying chemical trails by real robots can be problematic.

Other robotic implementations of insect-style pheromone trail following have instead used non-chemical substitutes for the trail chemicals. For example, Garnier et al. (Garnier et al., 2007) used a downward-pointing LCD projector mounted above their robots' arena to project light trails onto the floor. Other works that apply this similar approach were presented in (Masar, 2013) and (Sugawara et al., 2004). This latter work used virtual pheromone system in which chemical signals are simulated with the graphics projected on the floor, and in which the robots decide their action depending on the color information of the graphics.

Nevertheless, with recent developments in communication technology, electrical devices such as Radio Frequency Identification Devices (RFIDs) have gained much interest for such applications. Johansson and Saffiotti (Johansson and Saffiotti, 2009 ) and Herianto et al. (Herianto et al., 2007 ) used RFIDs for mapping and exploring an unknown environment. Moreover, Ziparo et al. (Ziparo et al., 2007 ) proposed a coordinated exploration and multi-robot SLAM for large teams of rescue robots by using RFIDs as environment features.

In essence, most of the nature-inspired approaches use a combination of stochastic components or moves with some deterministic moves so as to form a multi-agent system with evolving states. Such a swarming system evolves and potentially self-organizes into a self-organized state some emergent characteristics.

## 2.2 Multi-robot Allocation and Coordination

Coordination in multi-robot systems has been extensively studied in the scientific literature due its real-world applications including aggregation, patter formation, cooperative mapping and transport, foraging. All of these problems consist of multiple robots making decisions.

Decision-making can be regarded as a cognitive process resulting in the selection of a course of actions among several alternative scenarios. Every decision-making process produces a final choice. In MRS, the decision making guided by planning can be centralized or decentralized in accordance with the group architecture of the robots (Yan et al., 2012 ).

A solution is said centralized when a single element in the system is responsible for managing all the available resources. The strong point is that it can be used the best known algorithms and usually this kind of approach has more information available than distributed or local algorithms. This strength is in return burdened by the risk of losing contact with the controlling element, introducing a single point of failure (Mosteo and Montano, 2010 ).

Certain studies, belonging to the centralized architecture approach include (Luna and Bekris, 2011), (Yamashita et al., 2003), (Yan et al., 2012).

On the other hand, there is no central control agent in distributed architectures, such that all the robots are equal with respect to control and are completely autonomous in the decision-making process. Moreover, a decentralized architecture can better respond to unknown or changing environments, and usually has better reliability, flexibility, adaptability and robustness (Yan et al., 2013 ).

One of the most commonly used swarm-based approaches is the response threshold, where each robot has a stimulus associated with each task it has to execute. It continuously perceives the stimulus for each task; this stimulus reflects the urgency or importance of performing that task. When a robot perceives that stimulus for a particular task exceeds its threshold, it begins completing the task. When the stimulus falls below this threshold (e.g., when the task is completed), the agent stops executing those behaviors. This response can be deterministic or probabilistic (Kalra and Martinoli, 2006). Some response threshold systems use such stimuli and the threshold value for calculating the probability of executing a task (de Lopea et al., 2015) (Yang et al., 2009), (Palmieri et al., 2017a).

In recent years, market-based approaches have become popular to coordinate multi-robot systems. These methods have attempted to present a distributed solution for the task allocation problem (Triguia and et al., 2014). Essentially, robots act as self-interested agents in pursuit of individual profit. They are paid in virtual money for tasks they complete and must pay in virtual money the value of the resources they consume. Tasks typically are distributed through auctions held by an auctioneer; this auctioneer is either a supervisor agent or one of the robots. Robots compete through bidding to win those tasks that they can complete inexpensively and thus maximize their profit. This price-driven redistribution simultaneously results in better team solutions. Jones et al. (Jones et al., 2007) described a market based approach to task allocation for the fire fighting in a disaster response domain. Zhao (Zhao and Wang, 2013) used this approach to collect and transport objects in an unknown environment.

Recently, bio-inspired algorithms inspired by a variety of biological systems, have been proposed for self-organized robots. The self-organizing properties of animal swarms have been studied for better understanding the underlying concept of decentralized decision-making in nature, but it also gives a new approach in applications to multi-agent system engineering and robotics.

Ant colony Optimization (Dorigo and et.al, 2005) and its variant have been used as coordination techniques in coordinating robots. Hoff (Hoff et al., 2010) presented two ant-inspired robot foraging algorithms which allow coordination between robots. This approach uses direct communication between the robots instead of using environmental markers. They assume that the robots have limited sensing and communication capabilities and no explicit global positioning. De Rango (De Rango and Palmieri, 2012) used combined bio inspired approaches based on Ant Colony Optimization to guide the robots in an unknown mined area. Palmieri (Palmieri et al., 2015) used an hybrid approach that combines repellent and attractive pheromones to explore an area and recruit other robots respectively.

Another well known bio-inspired approach takes inspiration from the behaviour of the birds, called Particle Swarm Optimization (PSO) (Kennedy and Heberhart, 1995). PSO-inspired methods and their extended versions have received much attention and have been applied for the coordination of mobile robots. Some examples can be found such as guiding robots for targets searching in complex and noisy environment as described by Derr in (Derr and Manic, 2009). Pugh and Martinoli (Pugh et al., 2005) applied an adapted version of PSO learning algorithm to carry out unsupervised robotic learning in groups of robots with only local information. Hereford and Siebold (Hereford and Siebold, 2012 ) presented a version of PSO for finding targets in the environment. Modified versions of the PSO are proposed to balance searching and selecting in a collective clean-up task (Li and et al., 2014 ) for path planning in clutter environment (Das and et al., 2016 ) and for mimicking natural selection emulated using the principles of social exclusion and inclusion (Couceiroa et al., 2014 ).

Another nature-inspired algorithm called Bees Algorithm (BA), that mimics the food foraging behaviour of swarms of honey bees and its modified versions, has also been applied to robotic systems, demonstrating aggregation (Kernbach et al., 2009 ) and collective decision making (Jevtić et al., 2012 ), (Contreras-Cruz et al., 2015). Other studies that take inspiration from the bees and ants have also been applied to robotic systems such as task allocation (Momen, 2013), finding targets and avoiding obstacles (Banharnsakun et al., 2012 ), for solving on line path planning (Liang and Lee, 2015 ), (Garciaa and et. al., 2009); decision-making to aggregate robots around a zone (De Rango and Palmieri, 2016), (Hsieh et al., 2008), (Arvin and et. al., 2014). A Hybrid approach can be found in (De Rango and Palmieri, 2012).

Other studies take inspiration from the chemotactic behaviour of bacteria such as the Escherichia coli, called Bacterial Foraging Optimization (*BFO*). Bacteria movements mainly consist of two mobile behaviours: run in a particular direction and tumble to change its direction (Liu and Passino, 2002). Such behaviour depends on the nutrient information around them. Yang et al. (Yang et al., 2015 ) applied this method for a target search and trapping problem. An extensive review of research related to the bio-inspired techniques and the most behaviour of the robots can be found in, (Senanayake et al., 2016), (Bayındır, 2016)

Other approaches use a direct communication among the member of the swarm developing the protocol ad hoc. Direct communication refers to a process where robots exchange information directly between each other, often (but not necessarily) explicitly transmitting data to signal a particular status. Usually, according to the principle of local communication, information can be exchanged between nearby robots, which can then act upon received information modifying their behavior to improve the foraging performance. For example, Ants based routing is gaining more popularity because of its adaptive and dynamic nature and these algorithms consist in the continual acquisition of routing information through path sampling and discovery using small control packets called artificial ants. Some examples are: *AntHocNet* proposed by Di Caro et al (Di Caro et al., 2005), Ant-Colony Based Routing Algorithm (*ARA*) described by Bouazizi et al. (Bouazizi, 2002). The probabilistic emergent routing algorithm (*PERA*) (Baras and Mehta, 2003) has been proposed in which the routing table stores the probability distribution for the neighboring nodes. Singh et al. (Singh et al., 2012) presents a detail analysis of protocols based on ant-like mobile agents. Moreover, authors proposed bio-inspired routing strategies able to minimize the number of hops, the energy wastage, see (De Rango and Tropea, 2009) or able to combine more bio-inspired techniques (De Rango and Palmieri, 2016), (De Rango et al., 2017).

# Chapter 3

# Preliminaries

This study set out to investigate the ability of a swarm of robots to explore and unknown area in order to detect and manipulate many disseminated, eventually hazardous, targets basing on a self- adaptive decision mechanism.
Let us consider the following scenario. There are a number of targets scattered in an unknown area, according to a uniform distribution. A swarm of mobile robots is deployed in this area with the goal to explore the area for searching the targets and then removing/dismantling them cooperatively. Since it is either impossible or too expensive for a single robot to handle a target individually, it is necessary that, when a robot detects a target, a coalition of some robots must be formed to perform the removal task jointly. A coalition can handle a target only if the necessary robots are in the target's location. Moreover, it is assumed that there is no prior knowledge about the targets such as their total number and locations. Therefore, the only way to ensure the detection and the fulfillment of all targets is to explore the overall area.
Targets in such rescue missions can be fire, mines, human victims, or dangerous material that the robots have to handle. Since, the targets locations are detected gradually through searching, the recruitment task must start in real-time as the targets are found. The challenge is to complete the mission without any centralized control and using only minimal communication among the swarm of robots trying to minimize the total energy consumed by the team or the time to complete the mission.

Broadly speaking, the mission is divides into two phases: exploration and recruiting. During the exploration stage, it would be more efficient deploying, in a distributed manner, the robots in different regions of the area at the same time. In this phase, the robots do not use wireless communication, and the decisions are made by the robots on the basis of partial available knowledge about the environment. At each step every robot, from the current location (cell), starts to sense its neighbor cells through some sensors in order to make the decision where to go next.

When a robot detects a target, since it lacks the capabilities to carry out the rest of the task itself, it starts a recruiting process using, in this case, a wireless communication. The aim of the overall mission is, on one hand, to scatter the robots into different regions of the area and, on the other hand, to allocate a needful number of robots in the target's locations, while avoiding redundancy in both sub-tasks.

## 3.1 Assumptions of the Model

First of all, the characteristics of the unknown area and the capabilities of the robots are introduced. Then the problem is presented as an optimization problem subject to constraints. Moreover, it is considered both static and dynamic scenarios. In a static scenario is supposed that the robots have enough resources to explore the area and disarm all disseminated targets. The dynamic scenario implies likely dangerous events in the sense that the targets could explode at any time and in an unpredictable manner, producing the destruction of some robots and the damage of the nearby zones. Moreover, the robots are considered with a limited quantity of energy without the possibility of recharge or replacement. In such scenarios, the team works under more demanding time constraints.

### 3.1.1 Environment modeling

Let $A$ be the 2-D working field or grid, where $A \subset \mathbb{R}^2$. As a symbolic representation, the proposed method uses a grid map with $m$ and $n$ cells. Let us establish a Cartesian coordinate system taking the upper left corner of $A$ as the origin, each cell $c \in A$ of the area has its own definite coordinate $(x, y)$, with $x \in \{1, 2, \ldots, m\}$ and $y \in \{1, 2, \ldots, n\}$ elements. The universal set $C$ contains all possible states of a cell on the grid map. The subsets $C_1, C_2, C_3, C_4 \subset C$ (where $C_i \cap C_j = \emptyset$, $i \neq j$) represent possible states as follows:

- $C_1$:{explored by the robots},
- $C_2$:{accessible and not explored by the robots},
- $C_3$:{occupied by an obstacle},
- $C_4$:{not explored and inaccessible after hazardous events (e.g., the mine's explosion)}.

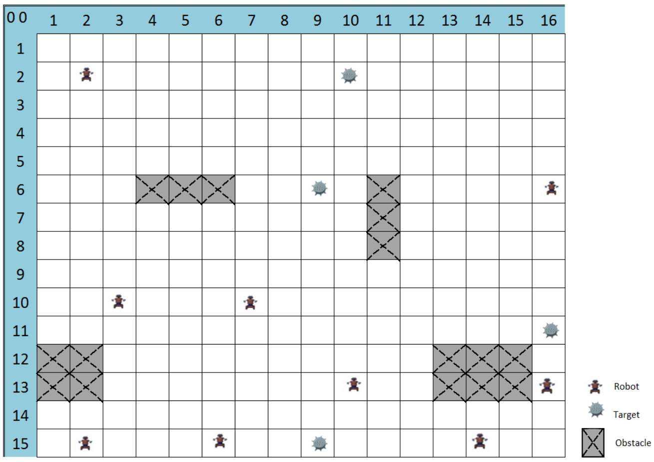



Fig. 3.1 A representation of the considered environment.

Obstacle cells are inaccessible to the robots and impenetrable to the sensors. A cell occupied, at time step $t$, by any of the robots can be considered as an obstacle, thus no other robot can take the place (see Fig. 3.1).

#### 3.1.1.1 ss

The state $C_4$ is used in the dynamic scenario, as descried above, where the found targets such as mines could explode or dangerous chemicals may leak at any time, making the nearby cells inaccessible.
While the robots explore the area, the cells transit to subset $C_1$. Each cell $c = (x,y)$ has a maximum of eight adjacent neighbors $N(c)$: $(x-1,y-1),(x-1,y),(x+1,y),(x,y+1),(x,y-1)(x+1,y+1),(x-1,y+1),(x+1,y-1)$ as shown in Fig. 3.2a.

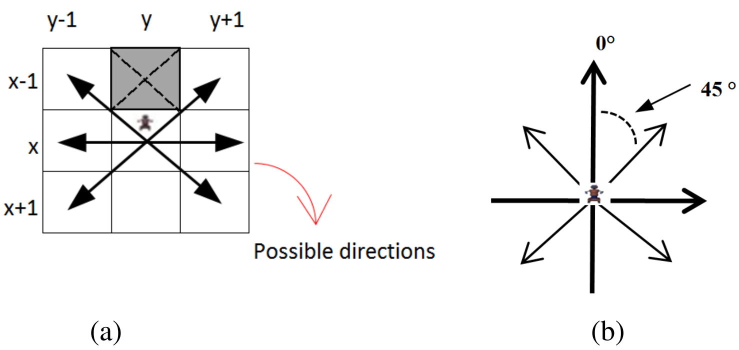



Fig. 3.2 (a) Possible directions of a robot's move (b) Possible angles of a robot's turn.

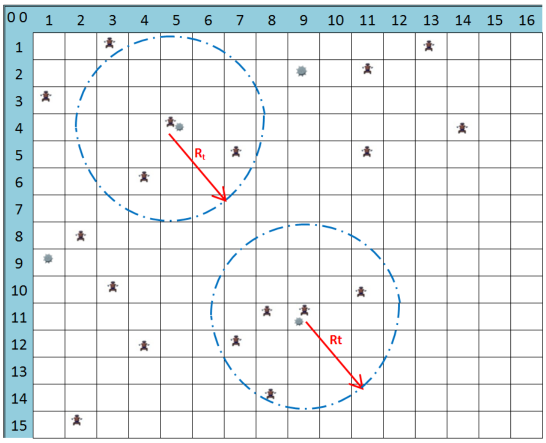



Fig. 3.3 The robots in the cells with coordinates (4,5) and (11,9) have detected a target. They start a recruitment process by sending packets that will be received by the robots within their wireless range $R_t$.

### 3.1.2 Robots Properties

A set $R$ of homogeneous robots is deployed in the area, where $R = \{ k \mid k \in \{1,2,\ldots,N^R\}\}$. At each step $t$, the current state of a robot $k$ can be represented by its coordinates $(x_k^t, y_k^t)$. The robots are modeled as rational collaborative autonomous agents that move autonomously in the environment. We assume that these robots are identical (executing the same algorithms) and follow simple local rules to communicate with neighbors and with the environment in order to provide a scalable strategy. However, for the sake of the simplicity, the robots are equipped with advanced devices such as sensors, global positioning capabilities (for instance they are equipped with a Wi-fi module) camera, radar, and an on-board automatic target recognition system, with which the robots identify the targets and obstacles or other robots in proximity. Sensor's information is assumed to be perfect, and the robots have perfect knowledge of their locations expressed in terms of their coordinates.
They are able to communicate with others using wireless communication and the communication range $R_t$ is limited compared to the size of area, so two robots can exchange information only if they are close enough; i.e., the distance between them is smaller than $R_t$. We define a local neighborhood of robot $k$ at time $t$, denoted by $LN_k^t$ , as the set of robots that are within the $R_t$ of the robot $k$ (Fig. 3.3).

In addition, it is also assumed that the communication network is perfect (no packets loss, no transmission time or delay), so robots within the same wireless range have identical information at the same time.

Furthermore, for simplicity without loss of generality, a robot uses 45° as the unit for turning (Fig.3.2b), and robot's speed is set to be one cell per time step. Moreover, the robots

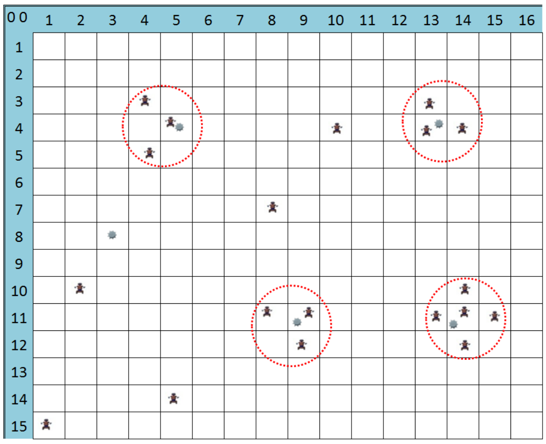



Fig. 3.4 Local coalitions of robots formed through the recruitment's processes

are considered to be synchronous; that is, each visited cell costs precisely 1 unit of time (and all internal computations are performed in zero time). Also, all robots start the search simultaneously at the same time. These assumptions can easily be removed , but are done to simplify the model since the main focus of the work consists in analyzing the proposed self-adaptive decision-making mechanisms.

The robots must explore the area for searching and dealing with a set T of $N^T$ targets such as mines disseminated in the area, i.e., T = $\{ z \mid z \in \{1, 2, \dots, N^T\}\}$. It is assumed that there is no prior knowledge about these targets such as the total number and locations. The targets can be located in any position of the area with the same probability.
Each target, is represented by its coordinates $(x_z, y_z)$. A target $z$ is detected by a robot $k$ when the target's coordinates coincides with the robot's coordinates. Once a robot finds a target, it starts a recruitment process since a target requires some amount of robots to be handled. $R_{\min}$ is a non-negative integer that represents the number of robots needed to treat safely a target. For this purpose, it is supposed that the robots that find targets communicate directly by sending out help requests through packets (that contains mainly the coordinates of the found target) to the robots within their wireless range $R_t$ (Fig.3.3). $RR_k$ is defined as a set that keeps track of the help requests that the robot $k$ receives from the others, expressed in terms of found targets, thus $RR_k \subset T$.

Figure 3.4 shows an example of local coalitions that are formed through the recruitment's processes. Since the robot's decisions can dynamically be changed in terms of robot's movements, new requests, failures, etc. the final configurations in the target's locations could change anytime.

### 3.1.3 Robot's actions

Robot's actions belong to three main classes:

I. Sensing actions that affect changes in robot's knowledge of the environment.

II. Moving actions in the cells which imply rotations to choose the right directions and obstacles avoidance.

III. Communication actions when the targets are found.

Each robot adapts its position in three different ways. The first is in the direction of minimum amount of pheromone (to indicate good feasible regions unexplored). The second is to move away from other robots or obstacles (to avoid collision). The third is in the direction of the selected target (to perform it cooperatively). The first two are based on interactions accumulated over time between the robots and the environment. The third is a reactive behavior triggered by help requests from other robots. In the following, these behavior characteristics are described.

To behave as a collective robotic organism, the robots need to be able to achieve different behavioral states. They are able to reconfigure themselves so as to achieve a transition between the states.
More specifically, at the beginning, when no target is detected, each robot collects information from its immediate surrounding cells perceiving chemical substance (pheromone) by on-board sensors and uses this information to identify the direction where to move. Each robot calculates its best move in terms of next position locally according to an Ant Colony-based approach as explained below. The goal is that the robots should explore the undetected sub-areas as much as possible in order to speed up the task. This state is named the *Forager State* and it is the initial state for each robot.

Once a robot discovers a target by itself, it will switch to a *Coordinator State*. Each coordinator robot is responsible for handling the disarmament process of the discovered target and for the recruitment of the others. The recruiting process starts by broadcasting packets to the robots in its wireless range (see Fig. 3.3), and it ends when the predefined number of necessary robots ($R_{min}$) is arrived to the target's location to form a coalition team. Then, the accumulated robots work together as a group, performing the disarmament task. Essentially a coordinator robot performs the following steps:

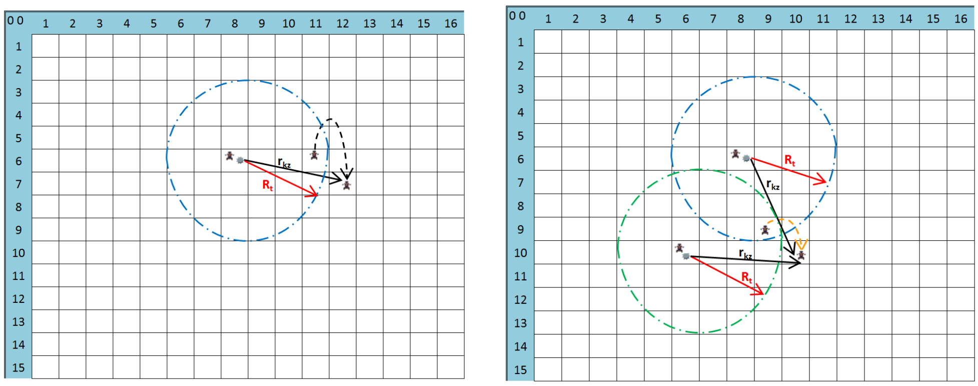
1. Check if there is a sufficient number of robots to form a coalition to handle a target.

2. If there is no a coalition that satisfies the constraint, then continue to send packets.

3. Repeat step 2 until all necessary robots are arrived.

4. Otherwise, stop the communication and start to disarm the target properly.

Once the target is disarmed, the involved robots return to continue to explore the area.




(a) (b)

Fig. 3.5 (a) The robot in the cell (6,11) that is recruited by the robot in cell (6,8) moves into a cell that is too far from the target, thus it changes its state by becoming an explorer. (b) The robot in cell (9,9) that is recruited by two robots in both cells (6,8) and (10,6), respectively. After, it moves into the cell that is too far from both targets, thus it changes its state by becoming a forager.

When a robot $k$ receives one or more request packets by coordinator robots it switches in *Recruited State*. Then, the robot will make the decision about where to move and what target to perform. A key aspect of this state is that the robots react to events that occur. Unlike common approaches, they could change the decisions taken previously during the iterations. For example, for a certain type of mission, it is possible to meet a target or receive different requests, while reaching another target in response to a recruitment process, thus reconsidering the choice of the target to be handled. Moreover, the decision can be to restart to explore the area since the movements are too far from the target's location see Fig. 3.5.

When a recruited robot, once it reaches the target's location, it will wait until the other needed robots have arrived and thus enter into the waiting mode. This state is called the *Waiting State*.

Finally, once the required robots reach the target's location, the group as a whole is involved in the disarming process and they will perform, for a fixed amount of time, some actions to deal with the targets properly. This state is the *Execution State*.

To summarize the above actions and states, Fig. (3.6) shows the state transition logic of a robot at each time step.

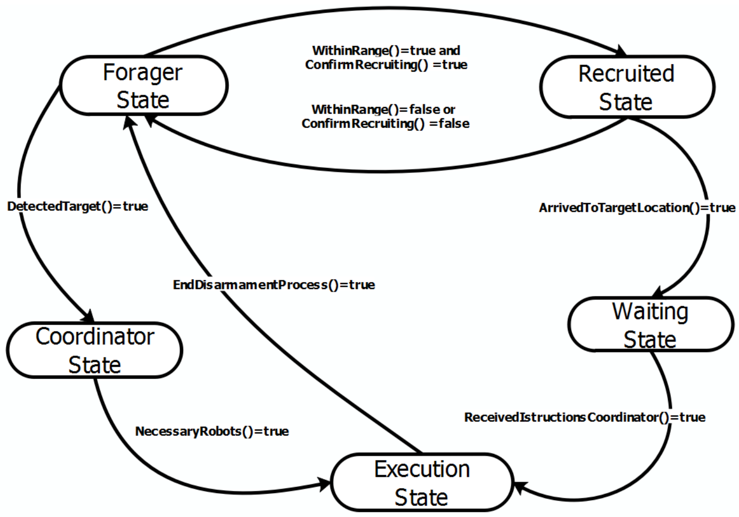



Fig. 3.6 State transition logic for a robot at each time step. WithinRange() is a function returning True when the robot is within the communication range of a coordinator robot. Confirm Recruiting() is a function returning true if the robot decides to get involved in the disarmament process of a target. ArrivedToTargetLocation() and ReceivedIstructionsCoordinator() are two functions returning true if the robot is arrived to a target's location and received the command to start the disarmament process by the coordinator, respectively. NecessaryRobot() returns true if all needed robots ($R_{min}$) have arrived at the target's location.

## 3.2 Decision Making as Optimization in multi-robot teams

In a broad sense, robot control software could be considered equivalent to decision making in multi-robot teams. In some sense, every action that a robot takes is indeed based on a decision that it has made.

A key challenge in multi-robot teaming research is determining how to properly enable robots to make decisions on actions they should take to contribute to the overall system objective. Interestingly, many forms of decision making in cooperative multi-robot systems can be formulated as optimization problems. Generally, globally optimal solutions in multi robot systems involved in multiple tasks, sometimes in conflict, are not possible, as it is well-known that such problems are intractable. A common approach is to use approximate techniques that, generally, work well in practice.

Typically, these are formulated as combinatorial optimization (Nemhauser and Wolsey, 1988) or as convex optimization problems (Boyd and Vandenberghe, 2004), in order to take advantage of the many tools available for these type of optimization.

Importantly, however, these problems are typically not treated as global optimization problems for multi-robot applications, since such problems are known to be NP-complete. Since most robotic applications require real-time robot response, there is insufficient time to cal-

culate globally optimal solutions for most applications; such solutions are only possible for very small-scale problems.
Instead, typical solutions use distributed methods that incorporate only local cost/utility metrics. While such approaches can only achieve approximations to the global solution, they often are sufficient for practical applications (Parker, 2012 ).

The paper presents the problem of search and rescue mission as a constrained bi-objective optimization problem in which mobile robots must perform two specific tasks of exploration and at same time cooperation and coordination for disarming the hazardous targets. These objectives are opposed goals, in which one may be favored, but only at the expense of the other. Therefore, a good trade-off should be found.

### 3.2.1 Mathematical Model

In order to describe the proposed system as proper mathematical model, it is useful to introduce the following notations and definitions:

- $A$: operational area, discretized as a grid map and $A \subset \mathbb{R}^2$
- $R$ : set of robots
- $N^R$ : number of robots $N^R$ = |R|
- $R_{min}$ = number of robots needed to deal with a target
- $S$: set of recruited robots $S \subset R$
- $T$: set of targets
- $N^T$: number of targets, $N^T$ = |T|
- $F$: set of the found targets, $F \subset T$
- $RR_k$: set of help requests expressed in terms of found targets received by the robot $k$ where $RR_k \subset F \subset T$.

Two main decisions have to be modelled properly. On the one hand, the cell expressed in terms of coordinates where each robot $k \in R$ should be located at each step. On the other hand, given a robot $k$ and a found target $z \in RR_k$, it has to decide if it is to get involved in the manipulation process of the found target $z$.

The first decision is mathematically represented by the decision variables:

$$v_{xy}^k = \begin{cases} 1 & \text{if the robot } k \text{ visits the cell of coordi nates } (x,y), \\ 0 & \text{otherwise.} \end{cases} \tag{3.1}$$

It is assumed that the time to visit a cell, denoted by $T_e$, is the same for all robots. Then the goal af an exploration task is to cover the whole area in the minimum amount of time, and thus the first objective becomes:

$$\text{minimize} \sum_{k=1}^{N^R} \sum_{x=1}^{m} \sum_{y=1}^{n} T_e \, v_{xy}^k. \tag{3.2}$$

Similarly, the following decision variables allow to model if a robot $k$ is involved in the recruitment process of the target $z$:

$$u_z^k = \begin{cases} 1 & \text{if robot } k \text{ is involved with target } z, \\ 0 & \text{otherwise.} \end{cases} \tag{3.3}$$

When a robot has eventually detected a target, it should act as an attractor, trying to recruit the required number of robots so as to disarm the discovered target safely and properly.

Let $T_{Start,z}^k$ be the time step at which the robot $k$ receive a help request for disarming the target $z$ and $T_{End,z}^k$ the time step at which the robot $k$ has reached the target $z$, then ($T_{End,z}^k$ - $T_{Start,z}^k$) is the coordination time for that robot $k$. Thus, the objective is the minimization of the coordination time for each found target, in order to speed up the disarming process and continue the mission effectively. Therefore, the second objective is

$$\text{minimize} \sum_{k=1}^{N^R} \sum_{z=1}^{N^T} (T_{End,z}^k - T_{Start,z}^k) \, u_z^k. \tag{3.4}$$

### The Bi-Objective Optimization Problem

The considered objective function is thus related to the minimization of the time needed to perform the overall mission. The problem, accounting both the exploration time and the coordination time, can be mathematically stated as follows:

$$\text{minimize} \sum_{k=1}^{N^R} \sum_{x=1}^{m} \sum_{y=1}^{n} T_e v_{xy}^k, \quad + \sum_{k=1}^{N^R} \sum_{z=1}^{N^T} (T_{End,z}^k - T_{Start,z}^k) u_z^k, \tag{3.5}$$

subject to

$$\sum_{k=1}^{N^R} v_{xy}^k \geq 1, \quad \forall\, (x,y) \in A, \tag{3.6}$$

$$\sum_{k=1}^{N^R} u_z^k = R_{min}, \quad \forall\, z \in T, \tag{3.7}$$

$$v_{xy}^k \in \{0,1\}, \quad \forall\, (x,y) \in A,\ k \in R, \tag{3.8}$$

$$u_z^k \in \{0,1\}, \quad \forall\, z \in T,\ k \in R. \tag{3.9}$$

$$T_e,\ T_{End,z}^k,\ T_{Start,z}^k \in \mathbb{R}, \quad \forall\, z \in T,\ k \in R. \tag{3.10}$$

The objective function in (3.5) to be minimized represents the total time consumed by the swarm of robots to accomplish the mission. It depends on the time for the exploration of the area and the time for coordinating the robots involved in the disarming process of the targets. Constraint (3.6) ensures that each cell is visited at least once. Constraint (3.7) defines that each target $z$ must be disarmed safely by $R_{min}$ robots. The constraints (3.8)-(3.10) specify the domain of the decision variables.
It is worth pointing out that the optimization problem here is intrinsically multi-objective, but it has been formulated as a single objective optimization problem. The main reason is that, in the first step, the main focus is the application and the comparison of different bio-inspired approaches in solving this challenging problem. Then an extension of the current approach to multi-objective optimization is taken into account (See Section 5.4).

## 3.2.2 Energy Model

For each activity executed by a robot $k$, a certain amount of energy is consumed. In this study, the energy model reflects mostly two activities: energy for communication and energy for mobility. The mobility energy depends on several factors. For simplicity, the mobility cost for a robot $k$ can be calculated by considering the distance traversed and it is expressed as follows:

$$E_m^k = \sum_{x=1}^{m} \sum_{y=1}^{n} C_m\, v_{xy}^k, \tag{3.11}$$

where $\sum_{x=1}^{m} \sum_{y=1}^{n} v_{xy}^k$ is the total number of visited cells for each robot $k$ while moving in the exploration phase and recruiting phase; $C_m$ is the cost given to move to one cell to another and takes into account both the cost for moving and turning.

When a target is detected, the energy consumed is instead related to the communication and to the cost for performing the planned task on the target. Since a wireless communication system is used in this phase, the energy consumed depends on the transmission and reception of the packets to communicate the position of the targets. In this case, the energy consumed by robot $k$ to transmit $E_{tx}^k$ and receive $E_{rx}^k$ a packet [87] is related to the maximum transmission range $R_t$ and to the packet size (l) as follows:

$$E_{tx}^k = l\ (R_t^{\psi}\ e_{tx} + e_{cct}), \tag{3.12}$$

where $e_{tx}$ is the energy required by the power amplifier of transceiver to transmit one bit data over the distance of one meter, and $e_{cct}$ is the energy consumed in the electronic circuits of the transceiver to transmit or receive one bit. Here, $\psi$ is called the path loss exponent of the transmission medium where $\psi \in [2,6]$.

On the other hand, the energy consumption for receiving a packet is independent of the distance between communication nodes and it is defined as:

$$E_{rx}^k = l\ e_{cct}, \tag{3.13}$$

The energy consumed to deal with a target is:

$$E_d^k = C_d, \tag{3.14}$$

where $C_d$ is the cost given to the working task for handling a target properly, and it is the same for each robot and it is related, for simplicity, to the mechanical movement.
Essentially, the energy consumed for the coordination task by the robot $k$ that is involved in the targets issue is:

$$E_{coord}^k = \sum_{z=1}^{N^T} (E_{tx}^k + E_{rx}^k + E_d^k)\ u_z^k, \tag{3.15}$$

Based on the previous considerations and models, it is now introduced a performance index, called Total-Energy-Swarm-Consumption (TESC), and it is defined as follows:

$$TESC = \sum_{k=1}^{N^R} E_m^k + \sum_{k=1}^{N^R} E_{coord}^k, \tag{3.16}$$

That is, the total energy consumed by the overall robots system in performing the mission and it is the sum of two contributions: the energy consumed for moving into the area and energy consumed for the wireless communication when the robots are involved in the manipulation of the targets.

# Chapter 4

# Searching Task

This chapter starts the discussion about one of the topics of the work that is the collective search task. The problem of collective search is a trade-off between searching thoroughly and covering as much area as possible. Solutions to the problem of collective search are currently of much interest in robotics especially the study of distributed algorithms applied to this problem. The objective is to design ways in which, without central control, robots can use local information to perform the search and rescue operations (Countryman et al.,2015 ). The problem of coordinating a team of robots for exploration is a challenging problem, particularly in unstructured areas, as for example post-disaster and hazardous scenarios where direct communication is limited.

Here, it is described an algorithm for exploring the area inspired by ant's foraging model. The approach emphasizes the role played by individual robot and stresses some crucial aspects such as the lack of a governing hierarchy, self organization of the robots, indirect communication.

Ants in nature, indeed, have evolved over a long period of time and display remarkable behaviors that are highly suitable for addressing complex tasks. In social insects, pheromone communication serves a number of social functions such as recognizing, aggregating, gathering food, mating and alarm propagation for the colony members.
Swarm optimization algorithms, such as Ant Colony Optimization (Dorigo and Stutzle, 2003), rely on pheromone trails to mediate (indirect) communication between the agents. In this kind of coordination, the environment is used as a medium to transfer information among the robots: each robot deposits traces in the environment in order to send different types of signals, depending on what it wants to indirectly communicate. The accumulation of traces in the environment provides a shared memory, which allows memoryless simple robots to coordinate easily, while robots might not have any self-awareness of other agents. These algorithms are fully decentralized and rely on memoryless agents with very simple

individual behaviors. Agents can only communicate through environment marking and, as they only mark and move according to their local perceptions.

In robotics field, with the availability of various sensors, a range of environmental markers (such as chemicals, metals, heat sources, electronic tags) can be used as a way of encoding information in the environment (Kuyucu and et.al., 2012).

## 4.1 Ant-inspired techniques

Ant colonies provide some of the richest examples for the study of collective phenomena such as collective exploration. Exploration is a very important task in nature since it allows animals to discover resources, detect the presence of potential risks, forage for food and scout for new home. Ant colonies operate without central control, coordinating their behavior through local interactions with each other. Ants perceive only local, mostly chemical and tactile cues. In a colony to monitor its environment, to detect both resources and threats, ants must move around so that if something happens, or a food source appears, some ant is likely to be near enough to find it (Gordon, 2010), (Countryman et al., 2015).

Ant colonies, despite the simplicity of single ants, demonstrate surprisingly good results in global problem solving. Consequently, ideas borrowed from insects and especially from ants behaviour are increasingly popular in robotics and distributed system.
Ant Colony Optimization have been developed by Dorigo (Dorigo and Stutzle,2003) inspired from the natural behaviour of trail laying and following by ants. They live in colonies and their behavior is governed by the goal of colony survival rather than being focused on the survival of individuals.
The behavior, that provided the inspiration for ACO, is the ant's foraging behavior, and in particular, how ants can find shortest paths between food sources and their nest. When searching for food, ants initially explore the area surrounding their nest in a random manner. While moving, ants can leave and smell a chemical pheromone trail on the ground. When choosing their way, they tend to choose, in probability, paths marked by strong pheromone concentrations. As soon as an ant finds a food source, it evaluates the quantity and the quality of the food and carries some of it back to the nest. During the return trip, the quantity of pheromone that an ant leaves on the ground may depend on the quantity and quality of the food. The pheromone trails will guide other ants to the food source. The central component of an ACO algorithm is a parametrized probabilistic model, which is called the pheromone model (Dorigo and Blum, 2005).
Over the last decay many variants of Dorigo's method have been proposed and applied in many robotics fields.

## 4.2 Ant-Based Team Strategy for Robot Exploration (ATS-RE)

In this section, it is addressed the exploration problem in the context of search and rescue operations, in which the mobile and autonomous robots must be able to decide the sequence of movements needed to explore the whole environment.

The mainstream approaches for developing exploration strategies are mostly based on the idea of incrementally exploring environment by evaluating a number of candidate observation locations, in this specific case neighbor cells, according to a criterion and by selecting, at each step the next best location. However, here it is not considered the problem to build a map of the environment, since the main object consists of locating the largest number of targets in the minimum amount of time.
Differently from map building, search and rescue settings are strongly constrained by both time and battery limitations and generally require the amount of explored regions over the map quality. Since the robots should be required to be capable of various functionalities other than area exploration, it is desirable that both the integration to a swarm and the ability to explore are seamless and these actions should not consume a large amount of the robot's resources. Moreover, to be effective, a search strategy must attract robots towards unobserved areas so as to avoid the undesirable scenario where some areas are frequently revisited while others remain unexplored.

Broadly speaking, the robots operate according to the following steps:

(a) The robots perceive the surrounding cells using on-board sensors.

(b) The robots compute the perceived information, in this case the concentration of pheromone, in neighbors cells.

(c) The robots decide where to go next,

(d) The robots move in their best local cell and start again from (a)

The basic intention behind the work described here is to design a motion policy which enables a group of robots, each equipped only with simple sensors, to efficiently explore environment eventually complex. As in biology, the basic idea pursued is to utilize the principle of pheromone-based coordination and to let each robot deposits pheromones on visited cells in order to inform, indirectly, the others about the already explored region. According to this approach, the robots need not communicate directly, but deposit pheromones on the borders of their territory for instructing other robots to not enter it. When the interior sensors detect pheromone, it should indicate to a robot that it is about entering to potentially explored

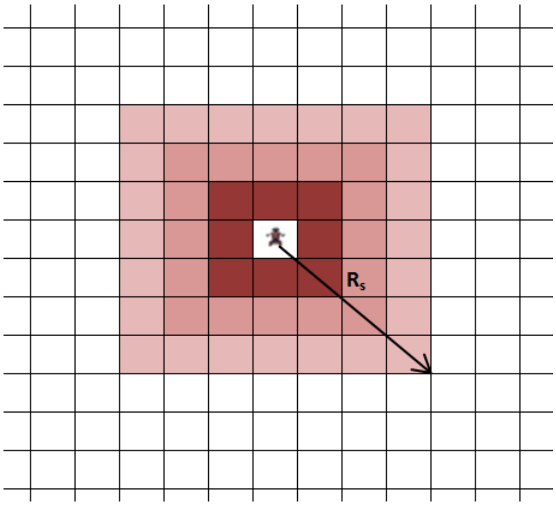



Fig. 4.1 Example of pheromone diffusion.

territory, and therefore the robot could preferentially change direction. The objective is to develop a simple algorithm that can utilize pheromones to benefit from the existing physical properties of a real environment in achieving complex collective behaviour within a large group of simple homogeneous robots. It does not need to keep a topology of the map in memory. Decision making is done probabilistically based on local pheromone information. It should be emphasized that the problem of sensors, leaving pheromone or movements do not take into account, since the main focus of the work is to design self-adaptive decision-making mechanism for performing the assigned task (Palmieri et al., 2017b ).

Broadly speaking, when the robots are exploring the area, they lay pheromone on the traversed cells and each robot uses the distribution of pheromone in its immediate vicinity to decide where to move. Like in nature, the pheromone trails change in both space and time. The pheromone deposited by a robot on a cell diffuses outwards cell-by-cell until a certain distance $R_s$ such that $R_s \subset A \subset \mathbb{R}^2$ and the amount of the pheromone decreases as the distance from the robot increases (see Fig. 4.1).

Mathematically, the pheromone diffusion is defined as follows: consider that robot $k$ at iteration $t$ is located in a cell of coordinates $(x_k^t, y_k^t) \in A$, then the amount of pheromone that the robot deposits at the cell $c$ of coordinates $(x, y)$ is given by:

$$\Delta\tau_c^{k,t} = \begin{cases} \Delta\tau_0 \, e^{\frac{-r_{kc}}{a_1}} - \frac{\varepsilon}{a_2} & \text{if } r_{kc} \leq R_s, \\ 0 & \text{otherwise,} \end{cases} \tag{4.1}$$

where $r_{kc}$ is the distance between the robot $k$ and the cell $c$ and it is defined as:

$$r_{kc} = \sqrt{(x_k^t - x)^2 + (y_k^t - y)^2}. \tag{4.2}$$

This means that pheromone spreads up to a certain distance, as in the real world, after which it is no perceivable by other robots. In addition, $\Delta\tau_o$ is the quantity of pheromone sprayed in the cell where the robot $k$ is placed and it is the maximum amount of pheromone, $\varepsilon$ is an heuristic value (noise) and $\varepsilon \in (0,1)$. Furthermore, $a_1$ and $a_2$ are two constants to reduce or increase the effect of the noise and pheromone. It should be noted that multiple robots can deposit pheromone in the environment at same time, then the total amount of pheromone that can be sensed in a cell $c$ depends on the contribution of many robots.

Furthermore, the deposited pheromone concentration is not fixed and evaporates with the time. The rate of evaporation of pheromone is given by $\rho$ ($0 \leq \rho \leq 1$ ), and the total amount of pheromone evaporated in the cell $c$ at step $t$ is given by the following function:

$$\xi_c^t = \rho \ \tau_c^t, \tag{4.3}$$

where $\tau_c^t$ is the total amount of the pheromone on the cell $c$ at iteration $t$.

Considering the evaporation of the pheromone and the diffusion according to the distance, the total amount of pheromone in the cell $c$ at iteration $t$ is given by

$$\tau_c^t = \tau_c^{(t-1)} - \xi_c^{(t-1)} + \sum_{k=1}^{N^R} \Delta\tau_c^{k,t}. \tag{4.4}$$

### 4.2.1 Probabilistic Decision Making

At each time step, the algorithm selects the most appropriate cell for each robot, among a set of neighbor cells without the knowledge of the entire area. This happens because the robots have not global information about the environment. The aim is to avoid any overlapping and redundancy efforts, therefore, the robots must be highly dispersed in the area in order to complete the mission as quickly as possible, avoiding at the same time any wastage of the robot's resources such as energy.

Each robot $k$, at each time step $t$, is placed on a particular cell $c_k^t$ that is surrounded by a set of accessible neighbor cells $N(c_k^t)$. Essentially, each robot perceives the pheromone deposited into the nearby cells, and then it chooses which cell to move to at the next step. The probability at each step $t$ for a robot $k$ of moving from cell $c_k^t$ to cell $c \in N(c_k^t)$ can be calculated by

$$p(c|c_k^t) = \frac{(\tau_c^t)^{\varphi} \ (\eta_c^t)^{\lambda}}{\sum_{b \in N(c_k^t)} (\tau_b^t)^{\varphi} \ (\eta_b^t)^{\lambda}}, \quad \forall \ c \in N(c_k^t), \tag{4.5}$$

where $(\tau_c^t)^{\varphi}$ is the quantity of pheromone in the cell $c$ at iteration $t$, and $(\eta_c^t)^{\lambda}$ is the heuristic variable to avoid the robots being trapped in a local minimum. In addition, $\varphi$ and $\lambda$ are two

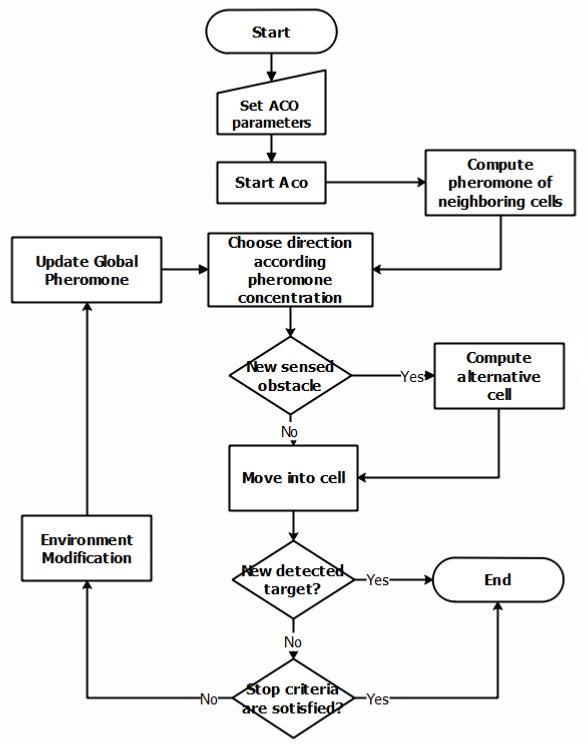



Fig. 4.2 The flow chart of Exploration task for a robot

constant parameters which balance the weight to be given to pheromone values and heuristic values, respectively. The robot $k$ moves into the cell that satisfies the following condition:

$$c = \arg\min[p(c|c_k^t)]. \tag{4.6}$$

In this way, the robots will prefer less frequently visited regions and more likely they will direct towards unexplored regions. Fig. 4.2 illustrates a simplified flowchart of the ACO-based strategy applied by each robot in *Forager State*.

### 4.2.2 ATS-RE Algorithm

The exploration strategy is detailed in Algorithm 1, that provides the pseudo-code for the pheromone-based control, which is executed periodically. At the first iteration of Algorithm 1, all cells are initialized with the same value of the pheromone trail, set to be zero that represents that the cells have not yet been visited by any of the robots, so that the initial probabilities that a cell would be chosen is almost random. Then the robots move from a cell to another based on the cell transition rule in Eq.(4.5). Unvisited cells become more attractive to the robots in the subsequent iterations. Using this approach, the robots explore the area by following the flow of the minimum pheromone. Then the pheromone trails on the visited cells by ants are updated as in Eq. (4.4).

Algorithm 1 stops executing for a robot when it becomes a coordinator or it is recruited or if the mission is completed (that is all cells have been visited at least once).

**Algorithm 1:** ATS-RE Algorithm

```
begin
    Step 1 : Initialization.
             Set t: {t is the time step}. Define φ, λ, a_1, a_2, ε, Δτ_0, ρ, R_s
    Step 2 : Generation coordination system. For the whole swarm, set the
             initial locations in terms of coordinates in x and y directions.
    Step 3 : Procedure
    while the stop criteria are not satisfied do
        foreach robot k in Forager State do
            evaluate the current position c_k^t;
            evaluate neighboorhood N(c_k^t);
            compute c according Eq. (4.6);
            if (c.hasObstacle() or c.isOccupated() or c.isInaccessible()) then
                choose a random cell c* ∈ N(c_k^t);
                move robot k towards c*;
            else
                move robot k towards c;
                deposit pheromone according to Eq. (4.1);
            end if
        end foreach
        foreach cell c ∈ A do
            update pheromone according Eq.(4.4);
        end foreach
        update t;
    end while
end
```

## 4.3 Computational Experiments

A set of experiments have been performed in order to show and analyze the effectiveness of the proposed exploration algorithm.

At this purpose a hand-designed simulator has been implemented in Java. This simulator,

used throughout the dissertation, was built from the start as a multi robot simulator. It is capable of modeling motion, targets, obstacles and local communication in a discrete world, and it can be easily extended to simulate other scenarios and domains since it is generalized. Screenshots of the simulator's graphical output option could be seen in Fig. 4.3 in which the parameters, regarding both exploration and recruiting tasks are represented.
Several experiments have been conducted in order to test the proposed exploration algorithm. Moreover the algorithm is compared with the Random Walk, Vertex-Ant-Walk (VAW) (Yanovski et al., 2001) and Inverse Ant System Based Survillance System (IAS-SS) (Calvo and et.al., 2011). Furthermore, for each set of experiments, the simulations have been repeated 100 times, thus the presented results are the mean values of those iterations.

It should be noted that, this section focuses only on exploration/navigation algorithm. All experiments consider an environment without targets, since here the main objective is to study the properties and the ability of the proposed algorithm.

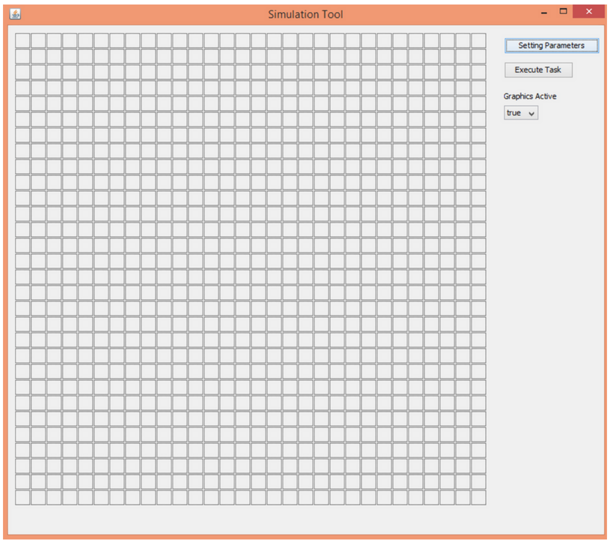
### 4.3.1 Metrics

An important aspect is the definition of effective metrics which broadly capture the performance of the algorithm and give insights into their performance and relative strengths and weaknesses. The considered performance metrics are:

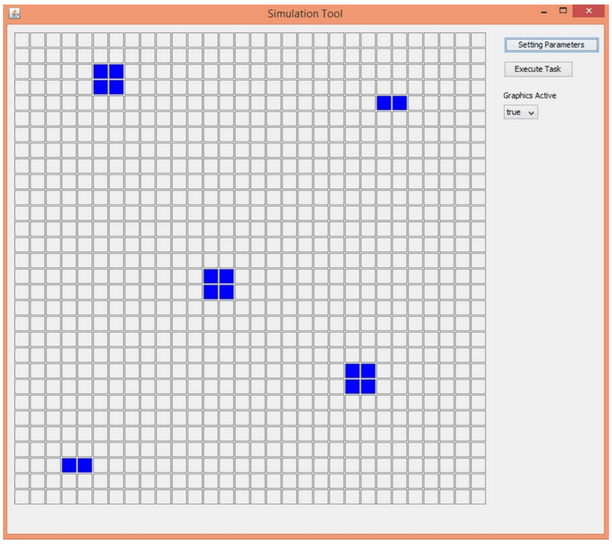
- Total time steps that is the total time to explore the overall area.
- Consumed Energy, that is the the total cumulated energy consumed by the robots (TESC). At each step of the simulation, a robot will consume an amount of energy that depends on its state and thus on the action that it performs. For example, a robot will consume more energy when handling a target than when wandering in the search area. A robot consumes 1 unit of energy for traveling from one cell to another. One stop takes an extra energy of 0.5 units. A turn of 45° takes 0.4 units of energy. Turns of 90°, 135°, 180°, take 0.6, 0.8 and 1 units of energy, respectively. These numbers are approximately derived from energy measurements for Pioneer 3-DX robot [81].
- Number of accesses for a cell that gives an information about the number of multiple visits for each cell of the environment. This metric estimates the powerful of the algorithm to spread among different regions of the environment the robots.

### 4.3.2 Case study 1: Influence of the parameters on the performance

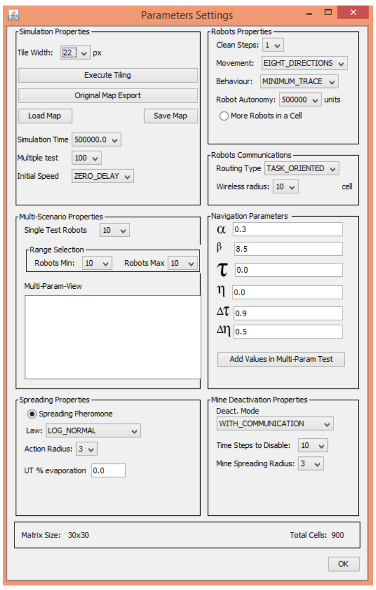
The first set of experiments are done in order to evaluate how the parameters of the model can affect the performance in terms of the considered metrics. The parameters taken into




(a)




(b)




(c)

Fig. 4.3 Simulator Front-end (a) Environment without obstacles (b) Environment with obstacles (c) Parameters Setting

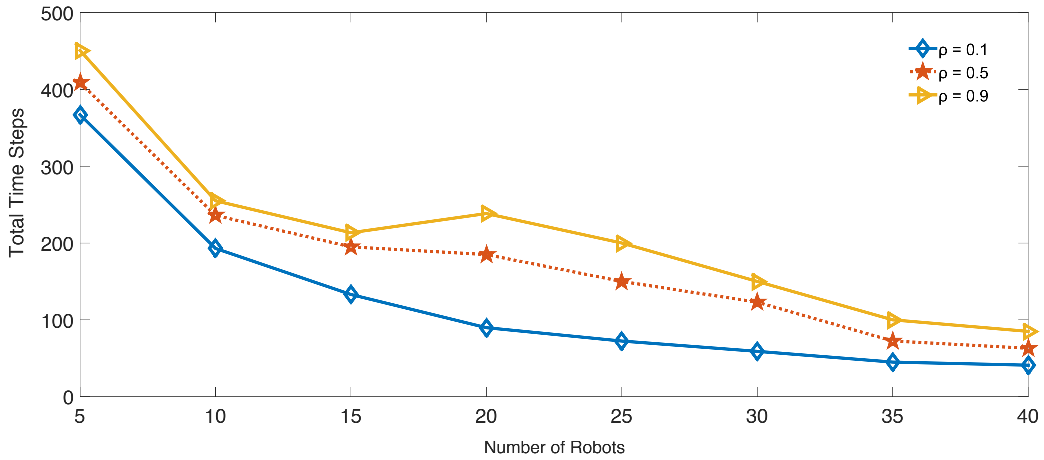



Fig. 4.4 Impact of $\rho$ on the total time steps

account are: the coefficients $a_1$, $a_2$; the sensing range $R_s$ of the Eq. 4.1 and the evaporation rate $\rho$ in Eq. 4.3. The environment size is fixed to 30x30 cells.

Fig. 4.4 shows the impact of the evaporation rate on the total time steps. A small value of $\rho$ means that the pheromone evaporates slowly and thus potentially it can be sensed by more robots, leading to a fast exploration of the robots. This is confirmed if the size of swarm operating in the area is small compared to the dimension of the area. Increasing the swarm size the time to complete the mission seem comparable.

Regarding the coefficients $a_1$ and $a_2$, a trade off among the possible value should be find. The parameters could be considered as a noise in the diffusion of the pheromone. Noise is helpful since it can help to drive a robot to move through a region that has been covered to reach another region that, potentially, needs to be explored. Without noise, a robots would not move through this already-explored cells, and could in fact become trapped. However, too much noise also has a negative impact because it marginalizes the effect of the pheromone. It is evident how too much or too little noise can negatively impact performance. A hight value of $a_1$ means that the pheromone is more perceived and the impact of the distance decreases, while a hight value of $a_2$, lead to a minor importance of the heuristic component. Fig. 4.5 depicts the impact of the coefficients on the total time steps considering an environment with and without obstacles. In both scenario the performance, mainly, depends on the size of the swarm, but usually, balance the two coefficient allow the swarm to effective execute exploration task. In the rest of the experiments $\rho$ is set to 0.2; $a_1$=0.5 and $a_2$ = 0.5.

The third set of simulations analyze the performance by varying the sensing range $R_s$ (4, 8, 10 units of cells) considering a grid area 30x30, 50x50 and 100x100 and by varying the number of robots (10, 20, 30, 40). This can play an important role in spread the robots among multiple and different regions of the area, since for a higher sensing range, potentially

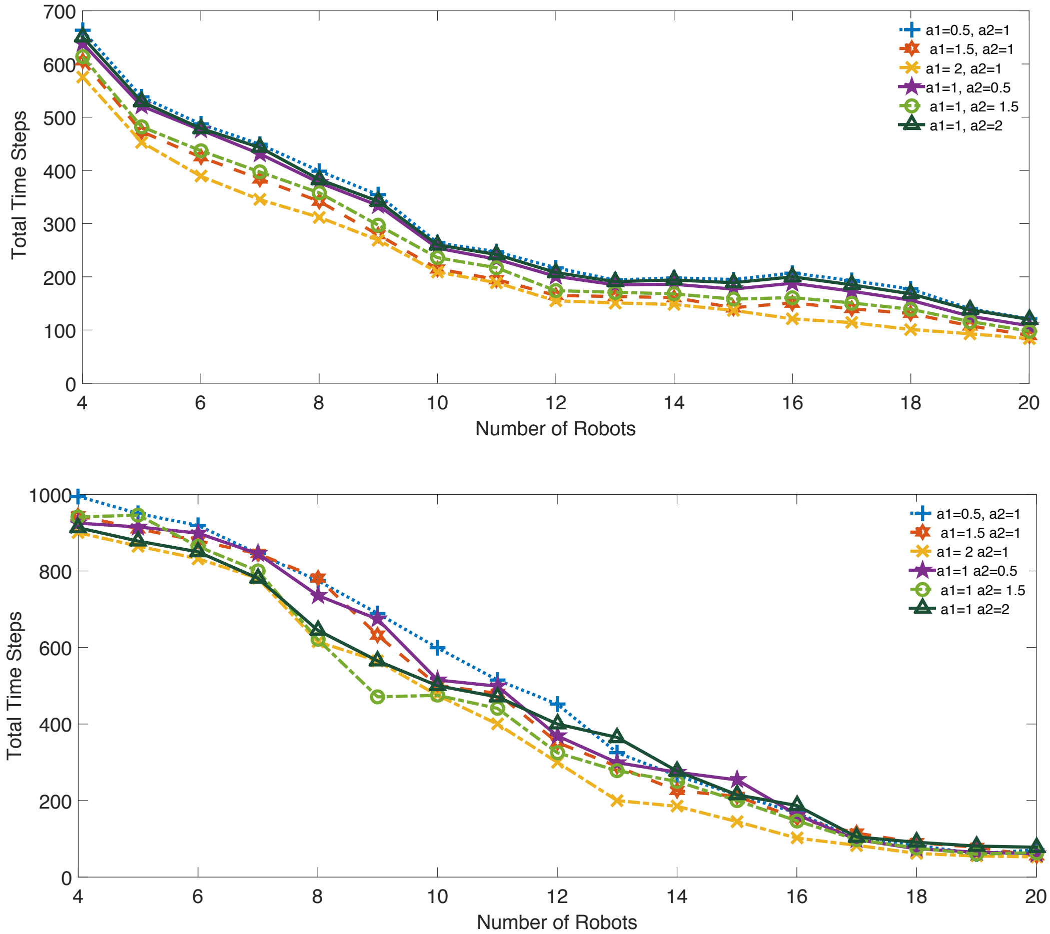


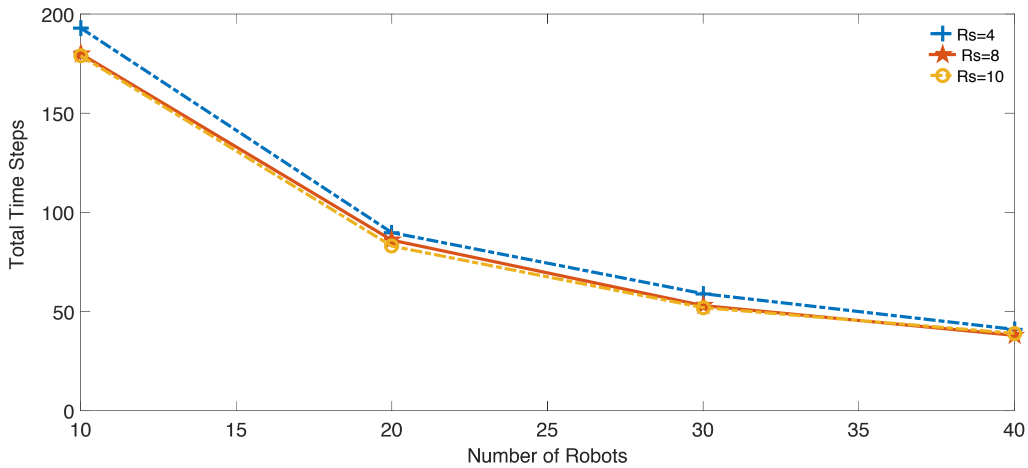

Fig. 4.5 Impact of the dispersion coefficient on the total time steps (a) Environment without obstacles (b) Environment with obstacles

the probability that more robots perceive the pheromone increases.
Figure 4.6 shows the total time steps under different conditions in terms of dispersed robots and dimension of the area.
It can be observed that the increase of the sensing range does not always imply the increase in performance in terms of time steps. One possible reason can be that if the sensing range increases, more robots potentially can sense at reasonable distance the presence of the others nearby. This may be lead to go towards opposite directions respect of the pheromone, but this not guarantee the others robot are, really, closely. This happens since the pheromone not evaporate instantaneously but it depends on the time and on $\rho$, moreover in a big area such as 50x50 and 100x100 a small robots team works under undemanding constraints and they can lead to make wrong decision and degrade the performance. As expected, if the number of the robots increases the global performance increases and the influence of the sensing range decreases. Regarding the energy consumption. Fig. 4.7 confirms, essentially, previous results. In small area size and small size of the robots, using a high sensing range could be benefit, increasing the complexity of the area in terms of cells, this not imply, always, a benefit for the previous motivations. In the rest of experiments, $R_s$ is fixed to 4 units of cell, in order to make more realistic the spread of pheromone. Indeed, in real scenarios a scent not spread a lot in the neighborhood, so its perceptions is limited.

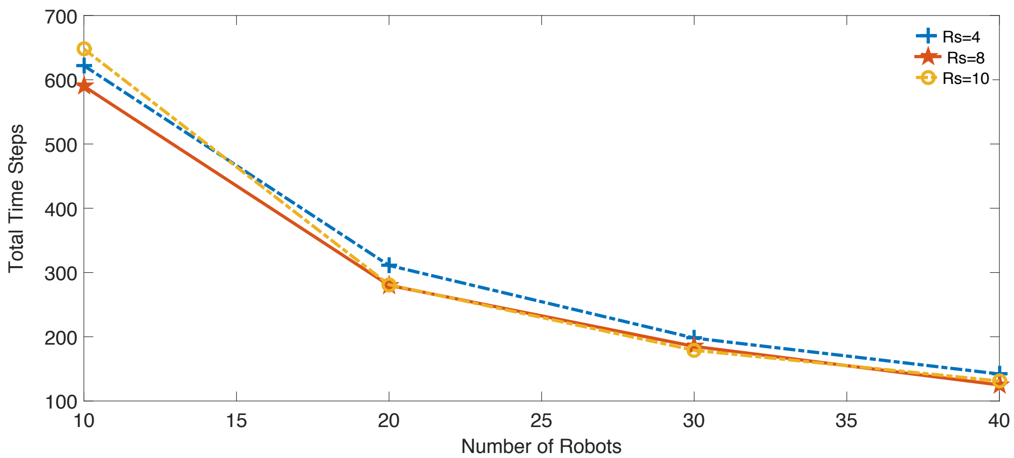
### 4.3.3 Case study 2: Influence of the number of robots on the performance

In this set of experiments, the influence of the size of the swarm on the performance is taken into account. It is considered an area of 30x30 square cells with and without obstacles, varying the team size. Fig.4.8 shows the influence of the number of robots evaluating respectively the total time steps to complete the mission, the total energy consumed by the swarm and the average number of accesses for a cell. The experiments highlight that, generally, the considered metrics decrease as the size of the swarm increases. However, after 30 robots the curves do not fluctuate a lot and especially the total time steps is almost similar.

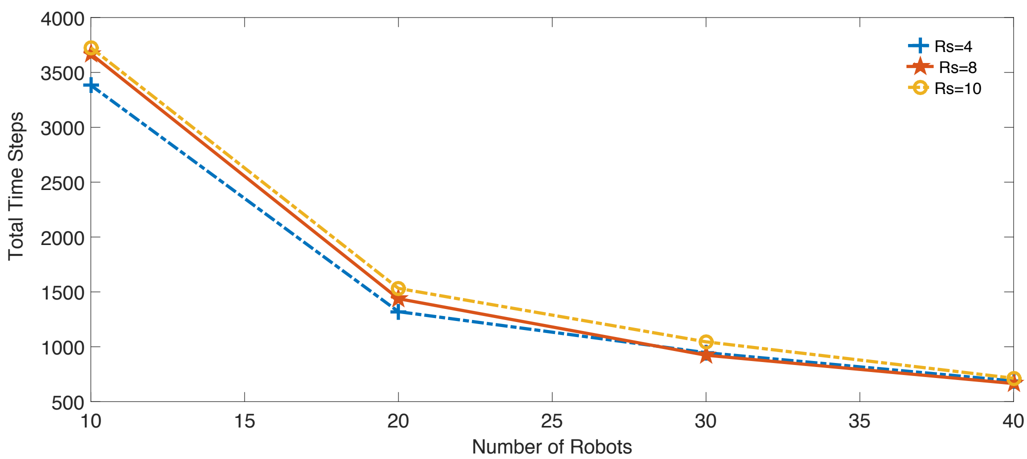
Regarding the energy consumption, it can be seen a high wastage of resource considering a small robots size. The same considerations can be done for the average of accesses for a cell. A small robots team is not able to explore efficiently the area, and this leads to revisit the same regions. Instead, increasing the number of robots, as in the many cases in swarm intelligence approach, the performance greatly improves and they are able to complete in efficient manner the assigned tasks.




(a)

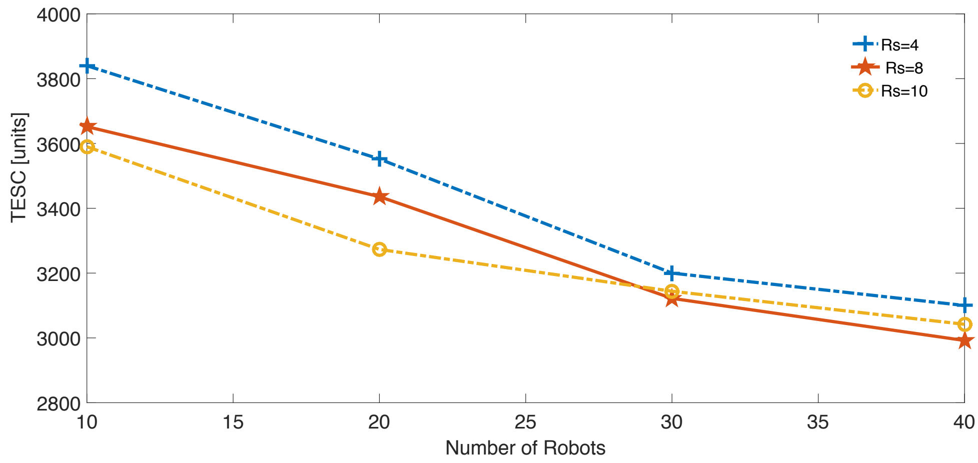



(b)

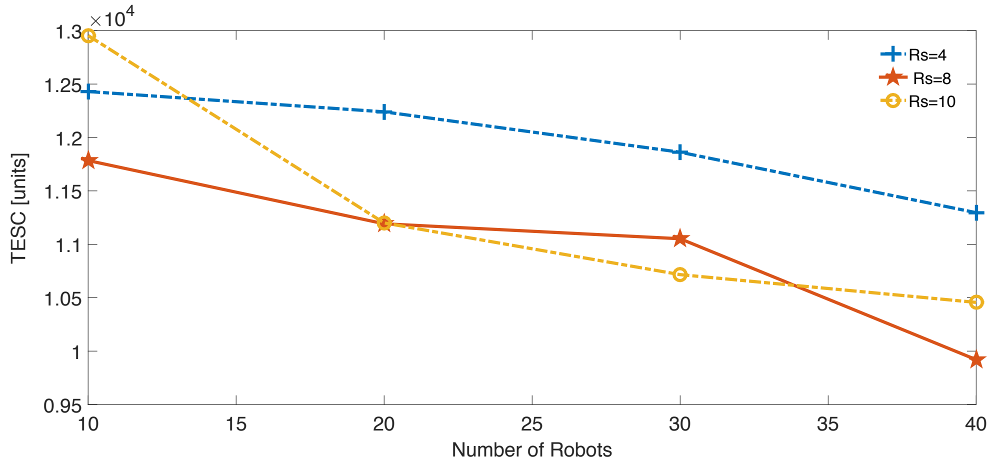



(c)

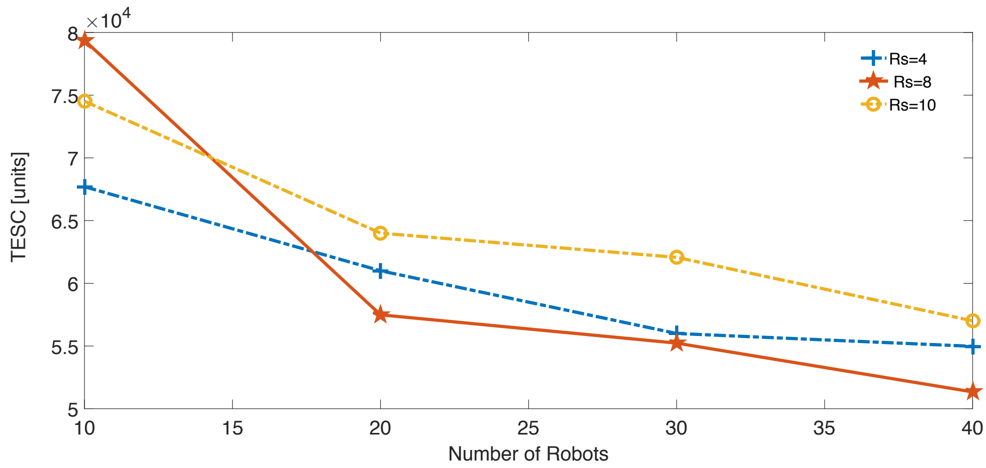
Fig. 4.6 Total time steps need to explore the area (a) 30x30 grid cells (b) 50x50 grid cells (c) 100x100 grid cells




(a)

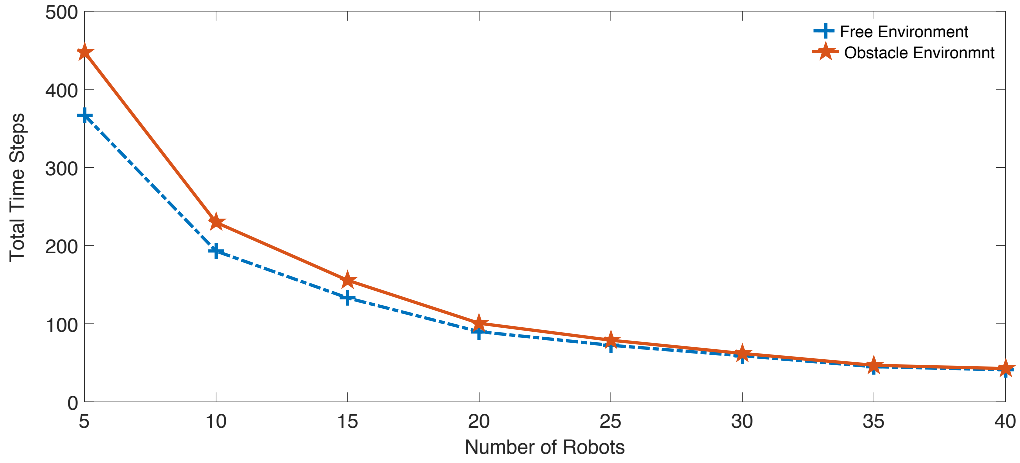



(b)

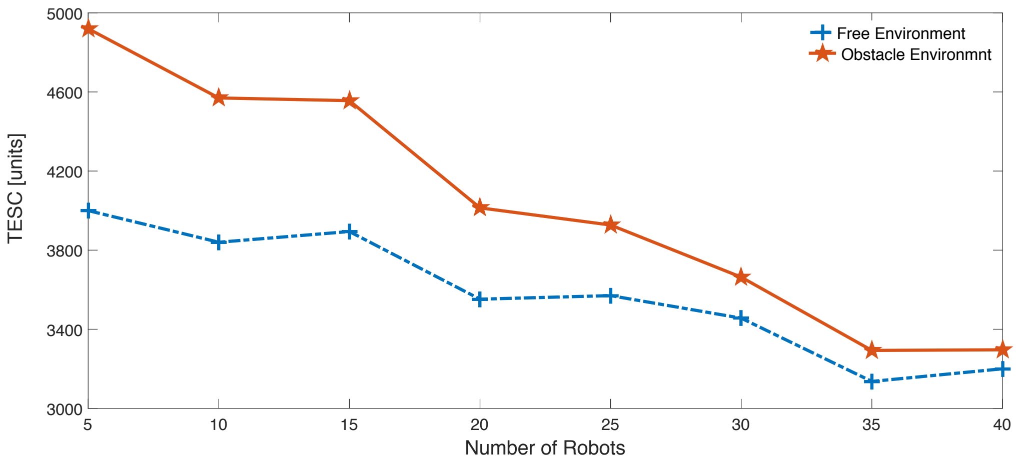



(c)

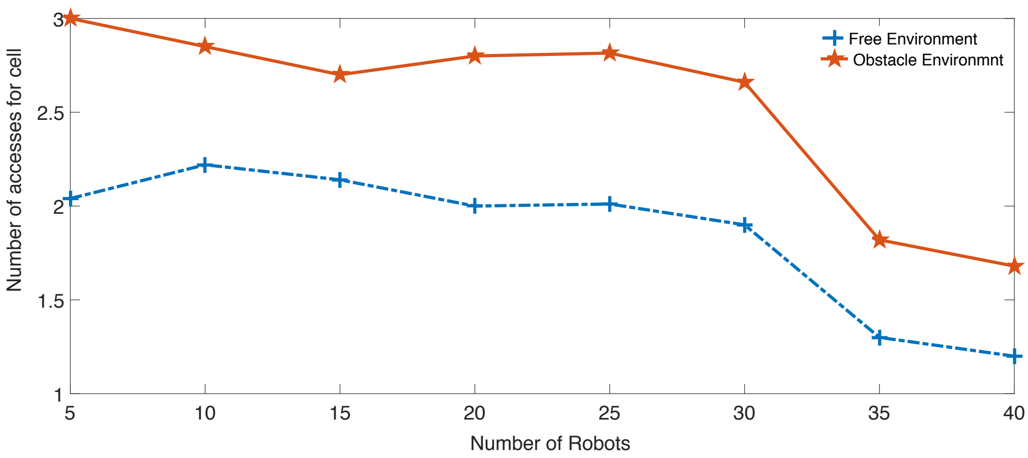
Fig. 4.7 Total energy consumed by the system to explore the area (a) 30x30 grid cells (b) 50x50 grid cells (c) 100x100 grid cells




(a)




(b)




(c)

Fig. 4.8 Performance of the ATS-RR considering a grid area 30x30 with and without obstacles (a) Total Time Steps (b) TESC (c) Average number of accesses for a cell

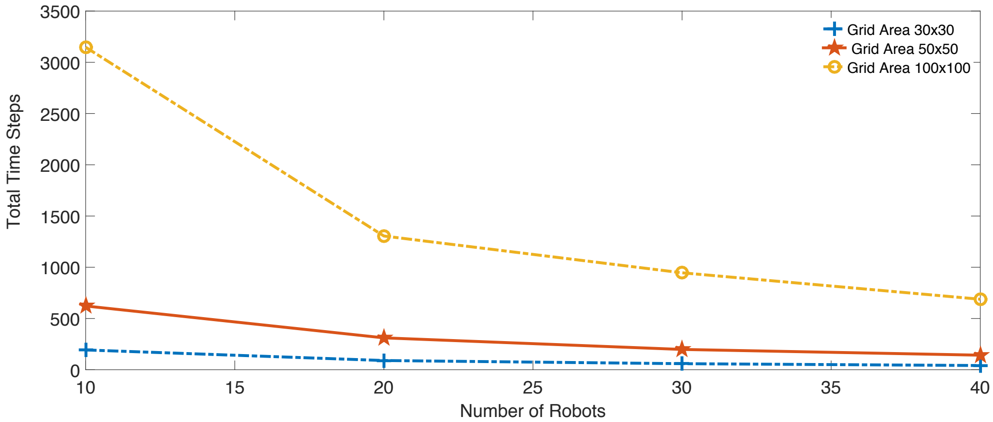



Fig. 4.9 Influence of the dimension of the area on total time steps

### 4.3.4 Case study 3: Influence of the dimension of the area on the performance

This section shows how the size of the environment affects the performance of the algorithm. Here an obstacle-free environment is used considering the size of the environment 30x30 cells, 50x50 cells 100x100 cells. Fig. 4.9 shows the influence of the dimension of the area evaluating the total time steps to complete the mission. It can be observed that the time steps increases as the area size increases, especially, when the size of the swarm is small. On the other hand, a team with a larger number of robots generally increases the performance improvements. The curves do not fluctuate a lot and the total time steps is almost similar for small grid area. This implies that the influence of the swarm on the performance in general decreases, considering an adequate swarm size.

### 4.3.5 Comparing ATS-RE to other algorithms

In this section the performance of three different algorithms like Vertex Ant Walk (Yanovski et al., 2001 ), ATS-RE (De Rango and Palmieri, 2012) and Random Walk are taken into account considering a grid area 30x30 with and without obstacles. Moreover, a comparison between ATS-RE and IAS-SS [13] is done. This last strategy tries inspiration by the inverse ant-colony optimization and it can be considered as a special case of our proposal changing in appropriate way the $a_1$ and $a_2$ values ( Eq. 4.1). Fig. 4.10 highlights the comparison between the two algorithms evaluating the total time steps.

Fig. 4.11, depicts the comparison among the three algorithm. It is obvious that ATS-RE outperforms the other two methods (De Rango and Palmieri, 2012), where the Random Walk is the worst. When the swarm size is small, ATS-RE becomes more efficient in a significant

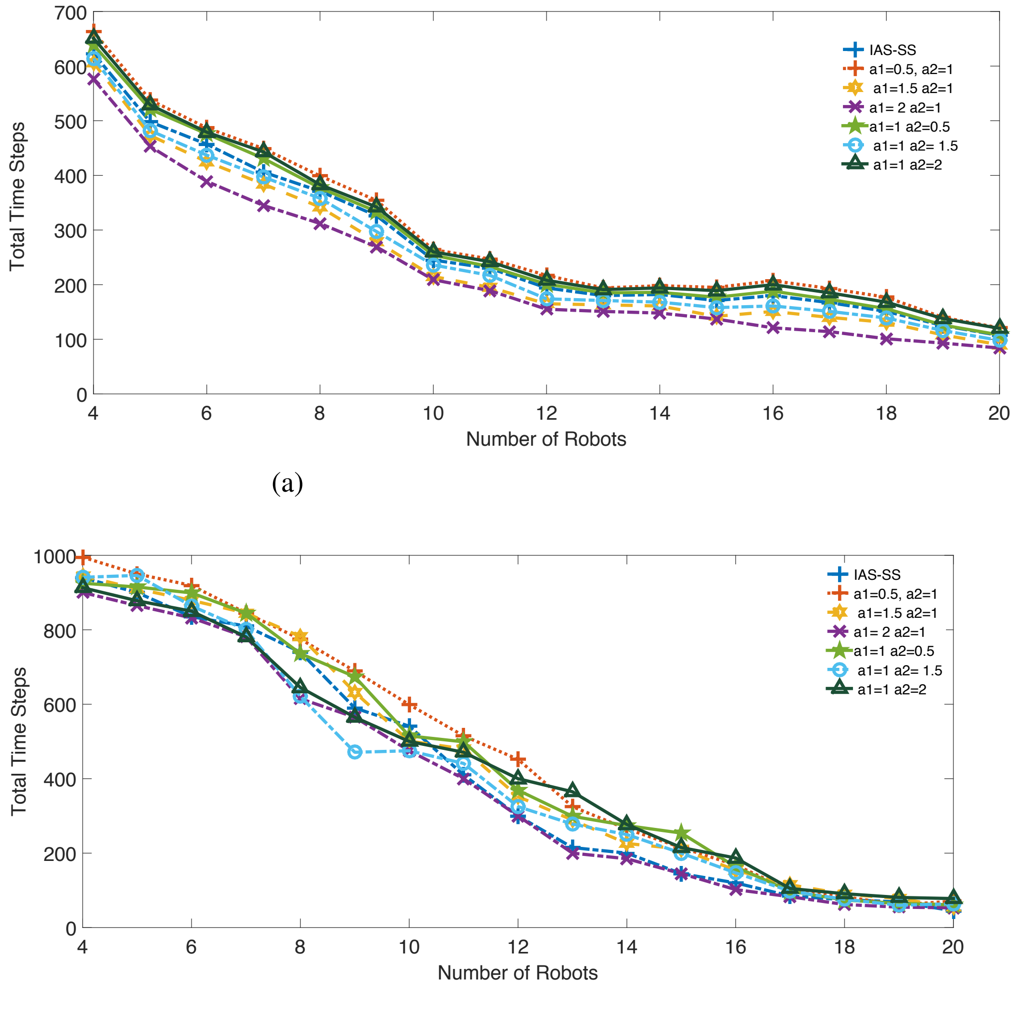



Fig. 4.10 ATR-RE vs IAS-SS (a) Free Environment (b) Obstacle Environment

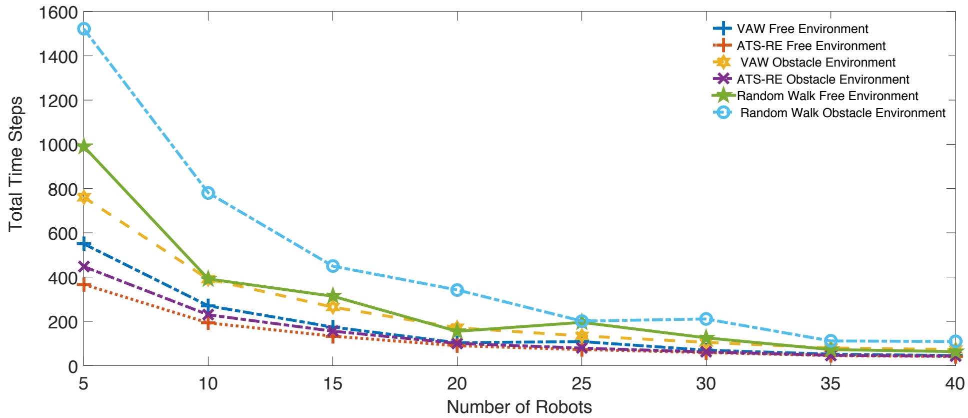



Fig. 4.11 Comparison among exploration algorithms in terms of total time steps

way, and it is able to distribute better the robots among different regions of the area. The reason could be that the robots, using ATR-RE, are greedy and would like, generally, to choose their best cell, and globally reach an efficient behaviour. However, as expected, an higher number of robots reduce the cells discovery time for all algorithms.

## 4.4 Summary

In this chapter, an ant-based algorithm for the coordination of multiple robots to achieve emergent exploration of an unknown environment for searching targets is considered. The algorithm utilizes stigmergic markers for inter-robot communication, which allows the implementation of the algorithm on simple robots, yet achieve complex organized behavior. Some terms, definitions and assumptions have been defined and used in the proposal.
Key factors include that the trace of leaved pheromone have repulsion characteristics instead of attraction in classic stigmergic coordination in an ant system and the evaporation and the diffusion of pheromone into the environment. The complete decentralized nature of the algorithm provides a fault tolerant way of coordinating multiple robots, and the use of stigmergic communication relieves the system from the processing and memory burden involved with other forms of communication.

More specifically, the pheromone based control provides a simple means of decentralized coordination of multiple-robots, giving rise to a fault tolerant system. By using the environment, the communication in a multi-robot system can be easily expanded to more than just a few robots. Moreover, the environment provides the ability to store information without any inter-robot communication overhead, and without any worries about limitations on communication ranges or limitations on the scalability of the system due to communication overhead. Thus, intelligent groups can arise from extremely simple individuals.

To demonstrate the validity of the proposed algorithm and effectiveness of the communication method with the pheromone trail, a simulator was constructed.
The experiments demonstrated that the proposed pheromone-based exploration algorithm scales well. Furthermore, it is verified that it performs well in a relatively large environment with and without obstacles. Moreover, the addition of a new robot not only reduces the amount ot time to explore the area, but also the amount of revisits cells and the consumed energy, hence providing a faster than linear speedup.

Finally, a comparison between the proposal and other approaches has been performed. Experimental results demonstrate that ATS-RE usually gives superior performance especially considering an environment with obstacles and a small robots team. In this case the algorithm is more able to coordinate through only indirect communication the robots, providing the spread among different sites reducing the time to complete the the mission in terms of completely explored area.

# Chapter 5

# Recruitment Task

The recruitment task aims to design a low-cost coordination mechanism that is able to form groups of robots at given sites where the targets are found. Once a robot detects a target, since it may not have sufficient resource capabilities to handle it, it acts as a strong attractor to the other robots to form a coalition that cooperatively works for the disarmament process of the target. The detection of a target may happen at any time during the exploration of the area, so the recruitment process is real time and it can take place in different regions of the area.

For this purpose, wireless communication is used to share the information about the found targets, since direct communication may be beneficial when a fast reaction is expected and countermeasures must be taken. In this case, each robot is assumed to have transmitters and receivers, using which it can send packets to other robots within its wireless range $R_t$.

A key issue in this problem is how to avoid deadlock; that is, the situation where robots are waiting for a long time for the others to proceed to disarming process. These issues are particularly relevant in strictly collaborative tasks since the robots need to work collectively and adaptively for the disarmament of the hazardous targets, and each robot has only locally and partially information about the environment.

The most common approach is in a greedy fashion in which a found target is instantaneously assigned to the robots without taking into account future events. Here, it is proposed a flexible strategy in which the robots can react to future new events changing, eventually, the taken decisions. However, each robot must make individual decisions that could lead to retract itself from help requests. For example, for such kind of mission, it is possible to detect a target, while reaching another, or to receive another request, and thus may change decisions to move in a more convenient way from the robot's point of view. So at each time step, the robots will make the best selfish decision based on their positions and conditions, in response to the received help requests, trying at the same time to balance the two tasks. In

order to tackle the problem, two communication mechanisms are proposed .
The first approach considers an one hop communication mechanism meaning that the coordinator robots send the packets only to the direct neighborhood (robots within the communication range) and no forwarding of information can be done. At this purpose different bio-inspired algorithms are proposed and compared.
The other approach is based on a multi hop communication, that allows the spreading of the information among the team and an Ant-based protocol is designed and developed.
It is worth mentioning that all proposed methods share the exploration algorithm, as described in Chapter 4.
The main focus in this chapter is to evaluate which could be the best recruitment bio-inspired algorithm and the best approach that may be used to form coalitions at certain locations, considering the same exploration strategy.

## 5.1 One-hop Communication

This section treats the problem of recruiting the needed robots in targets locations using only local spreading of the information about the detected targets. Essentially the information are sent using packets that contain mostly the coordinates of the detected targets. Therefore, the volume of information that is communicated among the robots is small, but it implies the robots still lack global knowledge of the environment. In this kind of approach, strongly inspired by the biological behaviour of social insects, the decisions made by the robots are independent, and the other robots and the coordinators do not know the taken decisions; therefore, the coordinators robots will continue to send packets until the needed robots have actually arrived. At this purpose, three bio-inspired techniques such as firefly algorithm, particle swarm optimization and distributed bee algorithm are proposed as coordination mechanisms to form coalitions of robots and compared.

### 5.1.1 Firefly Algorithm

Firefly Algorithm (FA) is a nature-inspired stochastic global optimization method that was developed by Yang (Yang, 2009). It tries to mimic the flashing behaviour of a swarm of fireflies. A firefly in the search space communicates with the neighboring fireflies through its brightness which influences the selection.
Fireflies swarm in nature exhibit social behaviour that use collective intelligence to perform their essential activities like species recognition, foraging, defensive mechanism and mating. A firefly has a special mode of communication with its light intensity that signals to the

swarm about its information concerning its species, location, attractiveness and so on. The two important properties of the firefly's flashing light are defined as follows:

- brightness of the firefly is proportional to its attractiveness.
- Brightness and attractiveness of pair of fireflies is inversely proportional to the distance between two.

These properties are responsible for visibility of fireflies which pave way to communicate with each other.

The distance $r(i, j)$ between any two fireflies $i$ and $j$, at positions $x_i$ and $x_j$, respectively, can be defined as the Euclidean distance as follows:

$$r_{ij} = ||x_i - x_j|| = \sqrt{\sum_{d=1}^{D} (x_{i,d} - x_{j,d})^2}, \tag{5.1}$$

where $x_{i,d}$ is the $d$th component of the spatial coordinate $x_i$ of the $i$th firefly and $D$ is the number of dimensions. In 2-D case, $r(i, j) : \mathbb{R}^2 \to \mathbb{R}$

$$r_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}. \tag{5.2}$$

In the firefly algorithm, as the attractiveness function of a firefly $j$ varies with distance, one should select any monotonically decreasing function of the distance to the chosen firefly defined as:

$$\beta = \beta_0 \, e^{-\gamma r_{ij}^2}, \tag{5.3}$$

where $r_{ij}$ is the distance defined as in Eq. (5.1), $\beta_0$ is the initial attractiveness at the distance $r_{ij} = 0$, and $\gamma$ is an absorption coefficient at the source which controls the decrease of the light intensity. The movement of a firefly $i$ which is attracted by a more attractive (i.e., brighter) firefly $j$ is governed by the following evolution equation:

$$x_i^{t+1} = x_i^t + \beta_0 \, e^{-\gamma r_{ij}^2} (x_j^t - x_i^t) + \alpha(\sigma - \frac{1}{2}), \tag{5.4}$$

where the first term on the right-hand side is the current position of the firefly $i$, the second term is used for modelling the attractiveness of the firefly as the light intensity seen by adjacent fireflies, and the third term is randomization with $\alpha$ being the randomization parameter and it is determined by the problem of interest. Here, $\sigma$ is a scaling factor that controls the distance of visibility and in most case we can use $\sigma \in [0, 1]$.

### 5.1.2 Firefly based Team Strategy for Robots Recruitment (FTS-RR)

Concerning the considered problem, each coordinator robot $k^*$, that has detected a target, starts to behave like a firefly sending out help requests to its neighborhood $LN^t_{k^*}$. When a robot $k$ receives this request and it decides to contribute in the disarming process, it stores the request in its list $RR_k$. If the list contains more requests, it must choose which target it will disarm. Using the relative position information of the found targets, the robot derives the distance between it and the coordinators and then uses this metric to choose the best target, that is usually the closer. The same information also allows to derive the next movement of the robots. The approach provides a flexible way to decide when it is necessary to reconsider decisions and how to choose among different targets.

It should be noticed that the recruited robots do not respond to the received requests, since they can change their decision at any time, so the coordinators robots do not know which robots are arriving and continue to broadcast packets until the needed robots have arrived. This has some implications. First, not all recruited robots will go towards the target's locations balancing the two task. Second, the order on which the requests are received is not as important as the allocation is not instantaneous. This allows an effective approach to reach solutions that the greedy strategy would miss. Third, the reduction of the impact on communications, so that bandwidth used will increase slowly with the team size.

Then the robots move towards target's location according to a modified version of the firefly algorithm. The aim of this strategy is to increase the flexibility of the system that let the robots be able to form groups effectively and efficiently in order to enhance the parallelism of the handling of the found targets, and at the same time move towards the targets location's avoiding overlapping regions and any redundancy (Fig. 5.1). Moreover, the algorithm allows to dynamically adjust the coordination task since it enables for each robot to make the best choice from its own point of view.

### 5.1.3 Implementation of Robot Decision Mechanism

The original version of FA is applied in the continuous space, and cannot be applied directly to tackle discrete problems, so the original algorithm has been modified properly.
In the considered scenario, a robot can move in a 2-D discrete space and it can go just in the adjacent cells. This means that when a robot $k$, at iteration $t$, in the cell $c^t_k$ with coordinates $(x^t_k, y^t_k)$ receives a packet by a coordinator robot that has found a target, the robot $k$ will move in the next step $(t+1)$ to a new position $(x^{t+1}_k, y^{t+1}_k)$, according to the FA attraction rules

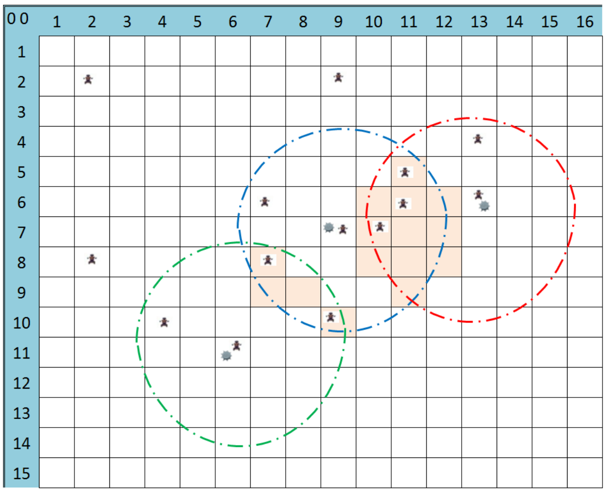


Fig. 5.1 Example of an overlap region in which some robots are in the wireless ranges of different coordinator robots and thus they must decide towards which target to move.

such as expressed below:

$$\begin{cases} x_k^{t+1} = x_k^t + \beta_0\, e^{-\gamma r_{kz}^2}(x_z - x_k^t) + \alpha(\sigma - \frac{1}{2}), \\ \\ y_k^{t+1} = y_k^t + \beta_0\, e^{-\gamma r_{kz}^2}(y_z - y_k^t) + \alpha(\sigma - \frac{1}{2}), \end{cases} \tag{5.5}$$

where $x_z$ and $y_z$ represent the coordinates of the selected target translated in terms of row and column of the matrix area, $r_{kz}$ is the Euclidean distance between the target $z$ and the recruited robot. It should be noticed that a robot can receive more than one request. In the latter case, it will choose to move towards the brighter target within the minimum distance from the target as expressed in Eq. (5.3). A robot's movement is conditioned by the target's position and by a random component that it is useful to avoid the situation that more recruited robots go towards the same target if more targets have found. This last condition enables to the algorithm to potentially jump out of any local optimum (Fig. 5.1).

A key aspect occurs when a robot $k$, moves too far from the target's position. Given a robot $k$ located at the step $t$ in the cell of coordinates $(x_k^t, y_k^t)$ and the target $z$ with coordinates $(x_z, y_z)$, the distance between the robot $k$ and the target $z$ is the Euclidean distance $r_{kz}$ as defined in Eq.5.2.

If $r_{kz} \geq (R_t + \Delta)\ \forall\, z \in RR_k$ means that the robot $k$ moves too far from the target's locations and in this case, if it has not got other requests, it switches its role into Forager State (see Fig. 3.5).

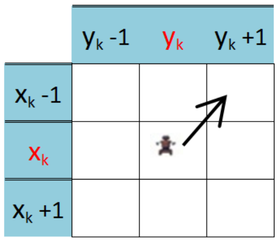



Fig. 5.2 A possible selected cell.

In order to modify the FA to a discrete version, the robot movements have been modeled by three kinds of possible value updates for coordinates { -1, 0, 1 }, according to the following conditions:

$$\begin{cases} x_k^{t+1} = x_k^t + 1 & \text{if } [\beta_0 e^{-\gamma r_{kz}^2}(x_z - x_k^t) + \alpha(\sigma - \frac{1}{2}) > 0\,], \\ x_k^{t+1} = x_k^t - 1 & \text{if } [\beta_0 e^{-\gamma r_{kz}^2}(x_z - x_k^t) + \alpha(\sigma - \frac{1}{2}) < 0\,], \\ x_k^{t+1} = x_k^t & \text{if } [\beta_0 e^{-\gamma r_{kz}^2}(x_z - x_k^t) + \alpha(\sigma - \frac{1}{2}) = 0\,], \end{cases} \tag{5.6}$$

and

$$\begin{cases} y_k^{t+1} = y_k^t + 1 & \text{if } [\beta_0 e^{-\gamma r_{kz}^2}(y_z - y_k^t) + \alpha(\sigma - \frac{1}{2}) > 0\,], \\ y_k^{t+1} = y_k^t - 1 & \text{if } [\beta_0 e^{-\gamma r_{kz}^2}(y_z - y_k^t) + \alpha(\sigma - \frac{1}{2}) < 0\,], \\ y_k^{t+1} = y_k^t & \text{if } [\beta_0 e^{-\gamma r_{kz}^2}(y_z - y_k^t) + \alpha(\sigma - \frac{1}{2}) = 0\,]. \end{cases} \tag{5.7}$$

A robot (e.g., robot $k$) that is in the cell with coordinates $(x_k^t, y_k^t)$ as depicted in Fig. 5.2 can move, potentially, into eight possible cells according to the three possible values attributed to $x_k$ and $y_k$. For example, if the result of Eqs. (5.6)-(5.7) is (-1, 1), the robot will move into the cell $(x_k^t - 1, y_k^t + 1)$.

### 5.1.4 FTS-RR Algorithm

The steps to be executed for FTS-RR are shown in Algorithm 2.
The Algorithm 2 is executed when one or more targets are found and some robots are recruited by others. If no target are detected or all targets are removed or handled, the robots perform the exploration task according to Algorithm 1.

More specifically, each recruited robot has the list of the requests in terms of target's locations and evaluates the brightness of each of them encoded as fireflies taking into account their distances. At each step, the robots select the best from their list which has the maximum brightness. Next they move to the target's location according to Firefly-based rules.

The proposed firefly-based approach is computationally simple. It requires only a few simple calculations (e.g., additions/subtractions) to update the positions of the robots. Moreover, the volume of information that is communicated among the robots is small, since only the position of the target's is sent. For this reason, FTS-RR has the benefit of the scalability. In addition, the algorithm tries to form a coalition with the minimum size of involved robots, so the remaining robots are able, potentially, to conduct other search or disarmament tasks, allowing multiple actions at a time.

---

**Algorithm 2:** FTS-RR Algorithm

---

**begin**
  **Step 1 : Initialization.**
    Set $t$ {t is the time step};
    Set the detected targets;
    Set the robots in Recruited State;
    Define the light absorption coefficient $\gamma$;
    Set the randomization parameter $\alpha$;
    Set the random number $\sigma$;
    Set the attractiveness $\beta_0$;
  **Step 2 : Generation coordination system.**
    For the detected targets and the recruited robots, set the initial locations in terms of coordinates in $x$ and $y$ directions;
  **Step 3 : Procedure.**
  **while** *The stop criteria are not satisfied* **do**
    **foreach** *robot k in Recruited State* **do**
      set $RR_k$;
      evaluate the current position $c_k^t$;
      **foreach** *target* $z \in RR_k$ **do**
        evaluate $\beta$ according to Eq. (5.3);
        choose the best target $z$ ;
      **end foreach**
      evaluate $N(c_k^t)$;
      compute the cell $c_k^{t+1}$ according to Eqs.(5.6)-(5.7);
      **if** *($c_k^{t+1}$.hasObstacle() or $c_k^{t+1}$.isOccupated() or $c_k^{t+1}$.isInaccessible())* **then**
        choose a random cell $c^* \in N(c_k^t)$;
        move robot $k$ towards $c^*$;
      **else**
        move robot $k$ towards $c_k^{t+1}$;
      **end if**
    **end foreach**
    update $t$;
  **end while**
**end**

---

### 5.1.5 Particle Swarm Optimization

Particle Swarm Optimization (PSO) is an optimization technique which uses a population of multiple agents (Kennedy and Heberhart, 1995). This technique is inspired by the movement of flocking birds and their interactions with their neighbours in the swarm. Each particle $i$ moves in the search space and has a velocity $v_i^t$ and a position vector $x_i^t$. A particle updates its velocity according to the best previous positions and the global best position achieved by its neighbours:

$$v_i^{t+1} = \omega v_i^t + r_1 c_1 (g_{best} - x_i^t) + r_2 c_2 (p_{best} - x_i^t), \tag{5.8}$$

where the individual best value is the best solution has been achieved by each particle so far that is called $p_{best}$. The overall best value is the best value (best position with the highest fitness function) that is found among the swarm, which is called $g_{best}$. Here, $r_j$ $(j = 1, 2)$ are the uniformly generated random numbers between 0 and 1, while $\omega$ is the inertial weight and $c_j$ $(j = 1, 2)$ are the acceleration coefficients. In addition, Eq.(5.8) is used to calculate the new velocity $v_i^{t+1}$ of a particle using its previous velocity $v_i^t$ and the distances between its current position and its own best found position; that is, its own best experience $p_{best}$ and the swarm global best $g_{best}$. The new position of particle $i$ are calculated by

$$x_i^{t+1} = x_i^t + v_i^{t+1}. \tag{5.9}$$

### 5.1.6 Particle Swarm Optimization for Robot Recruitment (PSO-RR)

Similarly, Firefly Algorithm, directly using this PSO-based decision strategy in the considered recruiting task would be problematic. Firstly on the two-dimensional map, there are only a limited number of possible directions for the robots to move and since we assumed that the robots can only move one cell at a time, the next position of the particles (robots) is limited to the neighbor cells as shown in Fig. (3.2a).
Moreover, in the recruiting phase, the object is to each the target location (that is $g_{best}$) and $p_{best}$ does not take into account.

Therefore, a modified PSO version is proposed and this means that for each robot $k$ at iteration $t$ in a cell with coordinates $(x_k^t, y_k^t)$, Eqs. (5.8)- (5.9) can be written as the follows:

$$\begin{cases} v_{x_k}^{t+1} = \omega v_{x_k}^t + r_1 c_1 (x_z - x_k^t), \\ \\ v_{y_k}^{t+1} = \omega v_{y_k}^t + r_1 c_1 (y_z - y_k^t), \end{cases} \tag{5.10}$$

$$\begin{cases} x_k^{t+1} = x_k^t + v_{x_k}^{t+1}, \\ \\ y_k^{t+1} = y_k^t + v_{y_k}^{t+1}, \end{cases} \tag{5.11}$$

where $(x_z, y_z)$ represent the coordinates of the detected target translated in terms of row and column of the matrix area.
In order to modify the PSO to a discrete version, similar to case of the FA, the robot movements have been considered as three possible value updates for each coordinates:$\{$ -1, 0, 1 $\}$ according to the following conditions:

$$\begin{cases} x_k^{t+1} = x_k^t + 1 & \text{if } [v_{x_k}^{t+1} > 0\ ], \\ \\ x_k^{t+1} = x_k^t - 1 & \text{if } [v_{x_k}^{t+1} < 0\ ], \\ \\ x_k^{t+1} = x_k^t & \text{if } [v_{x_k}^{t+1} = 0\ ], \end{cases} \tag{5.12}$$

and

$$\begin{cases} y_k^{t+1} = y_k^t + 1 & \text{if } [v_{y_k}^{t+1} > 0\ ], \\ \\ y_k^{t+1} = y_k^t - 1 & \text{if } [v_{y_k}^{t+1} < 0\ ], \\ \\ y_k^{t+1} = y_k^t & \text{if } [v_{y_k}^{t+1} = 0\ ]. \end{cases} \tag{5.13}$$

In this case, the PSO considers as metric the distance, thus when a robot receives more requests, it will choose to move toward the target at the minimum distance.

### 5.1.7 PSO-RR Algorithm

In the described problem, the Particle Swarm Algorithm is shown in Algorithm 3. Like FA, the steps are executed when the robots are recruited by others, but in the case when no targets are detected or all targets are handled, the robots continue to explore the area.

**Algorithm 3:** Particle Swarm based strategy algorithm

```
begin
    Step 1 : Initialization.
            Set t {t is the time step};
            Set the detected targets;
            Set the robots in Recruited State;
            Define the inertia weight ω;
            Set randomization parameter r_1;
            Set the acceleration coefficient c_1
    Step 2 : Generation coordination system.
            For the detected targets and the recruited robots, set the initial locations in
            terms of coordinates in x and y directions;
    Step 3 : Procedure.
    while The stop criteria are not satisfied do
        foreach robot k in Recruited State do
            set RR_k;
            evaluate the current position c_k^t;
            foreach target z ∈ RR_k do
                choose the best target z ;
            end foreach
            evaluate N(c_k^t);
            compute the cell c_k^{t+1} according Eqs.(5.12)-(5.13);
            if (c_k^{t+1}.hasObstacle() or c_k^{t+1}.isOccupated() or c_k^{t+1}.isInaccessible()) then
                choose a random cell c* ∈ N(c_k^t);
                move robot k towards c*;
            else
                move robot k towards c_k^{t+1};
            end if
        end foreach
        update t;
    end while
end
```

### 5.1.8 Artificial Bee Colony Algorithm

Another evolutionary approach is the Artificial Bee Colony (ABC) algorithm (Karaboga and Akay, 2009). This algorithm is inspired by the foraging behaviour of honey bees when seeking a quality food source. In the ABC algorithm, there is a population of food positions and the artificial bees modify these food positions along time. The algorithm uses a set of computational agents called honeybees to find the optimal solution. The honey bees in ABC can be categorized into three groups: employed bees, onlooker bees and scout bees. The employed bees exploit the food positions, while the onlooker bees are waiting for information from the employed bees about nectar amount of the food positions. The onlooker bees select food positions using the employed bee information and they exploit the selected food positions. Finally, the scout bees find new random food positions. Each solution, in the search space, consists of a set of optimization parameters which represent a food source position. The number of employed bees is equal to the number of food sources. The quality of food source is called its "fitness value" and it is associated with its position.

In the algorithm, the employed bees will be responsible for investigating their food sources (using fitness values) and sharing the information to recruit the onlooker bees. The number of the employed bees or the onlooker bees is equal to the number of solutions in the population ($SN$). Each solution (food source) $x_i (i = 1, 2, \ldots, SN)$ is a $D$-dimensional vector. The onlooker bees will make a decision to choose a food source based on this information. A food source with a higher quality will have a larger probability of being selected by onlooker bees. This process of a bee swarm seeking, advertising, and eventually selecting the best known food source is the process used to find the optimal solution. An onlooker bee chooses a food source depending on the probability value associated with that food source $p_i$ calculated by the following expression:

$$p_i = \frac{fit_i}{\sum_{q=1}^{SN} fit_q}, \tag{5.14}$$

where $fit_i$ is the fitness value of the solution $i$ evaluated by its employed bee, which is proportional to the nectar amount of the food source in the position $i$ and $SN$ is the number of food sources which is equal to the number of employed bees ($BN$). In this way, the employed bees exchange their information with the onlookers. In order to produce a candidate food position from the old one, the ABC uses the following expression:

$$x^*_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{lj}), \tag{5.15}$$

where $x^*_{ij}$ is the new feasible food source, which is selected by comparing the previous food source $x_{ij}$ and the randomly selected food source, $l \in \{1,2,\ldots, \mathrm{SN}\}$ and $j \in \{1,2,\ldots,\mathrm{D}\}$ are randomly chosen indexes. $\phi_{ij}$ is a random number between [-1,1] which is used to adjust the old food source to become the new food source in the next iteration.

### 5.1.9 Artificial Bee Colony Algorithm for Robot Recruitment (ABC-RR)

Similarly to the other two algorithms the ABC algorithm have modified to fit with our specific domain of interested as follows:

$$\begin{cases} x_k^{t+1} = x_k^t + \phi(x_k^t - x_z), \\ \\ y_k^{t+1} = y_k^t + \phi(y_k^t - y_z), \end{cases} \tag{5.16}$$

where ($x_z$ $y_z$) represent the coordinates of selected target translated in terms of row and column of the matrix area. Here, ($x_k^t$, $y_k^t$) is the current position of a robot $k$ and the ($x_k^{t+1}$, $y_k^{t+1}$) is the new position of the recruited robot. In order to modify the ABC to a discrete version, like the FA and PSO, the robot movements have been limited to three possible value updates for each coordinates: { -1, 0, 1 } according to the following conditions:

$$\begin{cases} x_k^{t+1} = x_k^t + 1 & \text{if } [\ \phi(x_k^t - x_z) > 0\ ], \\ \\ x_k^{t+1} = x_k^t - 1 & \text{if } [\ \phi(x_k^t - x_z) < 0\ ], \\ \\ x_k^{t+1} = x_k^t & \text{if } [\ \phi(x_k^t - x_z) = 0\ ], \end{cases} \tag{5.17}$$

and

$$\begin{cases} y_k^{t+1} = y_k^t + 1 & \text{if } [\ \phi(y_k^t - y_z) > 0\ ], \\ \\ y_k^{t+1} = y_k^t - 1 & \text{if } [\ \phi(y_k^t - y_z) < 0\ ], \\ \\ y_k^{t+1} = y_k^t & \text{if } [\ \phi(y_k^t - y_z) = 0\ ]. \end{cases} \tag{5.18}$$

Essentially two case could be happened. The first is when a robot receives only one recruitment request and in this case, it will move towards the target location according to the Eqs. (5.17)-(5.18). If a robot receives more than one request, it needs to decide which target it will move to. In this case, a concept according to the Distributed Bee Algorithm presented

in (Jevtić et al., 2012 ) has been used.
Basically, when a robot $k$ in the cell $c_k^t$ receives a packet from a coordinator in the cell $c_z^t$, the cost of the target $z$ for the robot $k$ at step $t$ is calculated as the Euclidean distance between the robot and the target in the 2-D area:

$$r_{kz} = \sqrt{(x_k^t - x_z)^2 + (y_k^t - y_z)^2}, \quad \forall z \in RR_k \tag{5.19}$$

Firstly, it is introduced the concept of the *utility* of a target $z$ for the robot $k$ the reciprocal value of the distance as:

$$\mu_z^k = \frac{1}{r_{kz}}. \tag{5.20}$$

Then, a probability that the robot $k$ chooses the target $z$ can be calculated by

$$p_z^k = \frac{\mu_z^k}{\sum_{b=1}^{RR_k} \mu_b^k}, \tag{5.21}$$

where $RR_k \subset F \subset T$. From the Eq. (5.21), it is easy to show that

$$\sum_{z=1}^{RR_k} p_z^k = 1 \tag{5.22}$$

The underlying decision-making mechanism adopts the roulette rule, also Known as the wheel-selection rule. That is, each target has been associated with a probability which it is chosen from a set of detected targets. Once all the probabilities are calculated according to Eq. (5.21), the robot will choose the target by spinning the wheel. Next the robot will move according to Eqs. (5.17)-(5.18). Such a coordination technique is well-suited, like the FA, to avoid that several robots approach the same target and spreading the robots over different target's locations (Fig. 5.1).

### 5.1.10 ABS-RR Algorithm

In the described problem, the algorithm for the bees based strategy is shown in Algorithm 4.

Like FTS-RR and PSO-RR, these steps are executed when the robots are recruited by others. In case when no targets are detected or all the tasks about the targets are performed, the robots continue to explore the area until the mission ends.

It is worth pointing out that for all strategies, the decision mechanism is done at each step; this implies that if a recruited robot at step $t$ chooses a target $z$, at the step $t+1$ takes again the decision and it could then choose another better target.

**Algorithm 4:** ABC-RR strategy

```
begin
    Step 1 : Initialization.
            Set t {t is the time step};
            Set the detected targets;
            Set the robots in Recruited State;
            Define randomization parameter φ
    Step 2 : Generation coordination system.
            For the detected targets and the recruited robots, set the initial locations in
            terms of coordinates in x and y directions;
    Step 3 : Procedure.
    while The stop criteria are not satisfied do
        foreach robot k in Recruited State do
            set RR_k;
            evaluate the current position c_k^t;
            foreach target z ∈ RR_k do
                evaluate p_z^k according to Eq. (5.21);
                choose the best target z according to the wheel-selection rule;
            end foreach
            evaluate N(c_k^t);
            compute the cell c_k^{t+1} according to (5.17)-(5.18);
            if (c_k^{t+1}.hasObstacle() or c_k^{t+1}.isOccupated() or c_k^{t+1}.isInaccessible()) then
                choose a random cell c* ∈ N(c_k^t);
                move robot k towards c*;
            else
                move robot k towards c_k^{t+1};
            end if
        end foreach
        update t;
    end while
end
```

## 5.2 Simulation Experiments

A computational study and extensive simulations have been carried out to assess the behavior of the proposed approaches and to analyze their performance by varying the parameters of the problem.

## 5.3 Evaluation of the Performance in Static Conditions

In the first stage, it is considered a static scenario, where it is assumed that the robots have enough resources to explore the area and disarm all disseminated targets. Moreover, the targets are static without possibility for example of explosion or causing damage.

### 5.3.1 Test Parameters and Metrics

To measure the performance, two metrics, mainly are used: the total time steps to complete the mission and the total energy consumed by the robots.

At the start of the simulations, all robots are in the *Forager State*. Robots and targets are initially deployed in the operative area according to a uniform distribution. At each step of the simulation, a robot will consume an amount of energy depending on its state and thus on the actions that it is performing (Fig. 3.6).

For the exploration task, the parameters used in the experiments are shown in Table 5.1 according to previous studies. Regarding the wireless communication, the value of the parameters are modelled empirically according to a study presented in (Ooi and Schindelhauer, 2009 ) and shown in Table 5.2. In the considered model, $e_{cc}$, $e_{tx}$ and $e_{rc}$ have been recalculated to express them in terms of the unit of energy. Regarding the values of the parameters of the Firefly Algorithm, please refer to (Palmieri et al., 2015). For PSO and ABC techniques, we have used the values of previous studies (Clerc and Kennedy, 2002), (Zhang et al., 2016 ), respectively. To summarize, Table 5.3 shows the parameters used in the coordination strategies.

To evaluate the proposed techniques, it was considered the environment with different levels of complexity depending on the following factors: the dimension of grid, the size of the swarm of robots and the number of targets to be treated. It is worth pointing out that the simulations were done by applying the same exploration strategy explained in Chapter 3, since the main focus of the work is to analyze the performance of the coordination techniques applied to the recruiting task.

### 5.3.2 Simulation Experiments I: Evaluation of the Time to complete the mission

In this set of experiments, the performance of the proposed algorithms in terms of the time to explore overall area and handle all targets is considered. Moreover, the average number of accesses for a cell, in order to see the effectiveness of the joint exploration task (space distribution) and disarming task (space concentration), is taken into account. This last metric

Table 5.1 Parameters used in the exploration algorithm.

| Parameters | Value |
|---|---|
| Sensing range $R_s$ | 4 |
| $\rho$ | 0.2 |
| $\Delta\tau_0$ | 2 |
| $\varphi$ | 1 |
| $\lambda$ | 1 |
| $\eta$ | 0.9 |
| $a_1$ | 0.5 |
| $a_2$ | 0.5 |
| $\varepsilon$ | Uniform [0 1] |

Table 5.2 Cost related to the wireless communication.

| Parameters | Value |
|---|---|
| Bit Rate (B) | 3 |
| Energy consumed by a transceiver circuitry to transmit o receive a bit, $e_{cc}$ (Joule) | $10^{-7}$ |
| Energy consumed by a transceiver amplifier to transmit one bit data over one meter, $e_{tx}$ (Joule) | $10^{-12}$ |
| Energy to receive a bit, $e_{rc}$ (Joule) | $10^{-7}$ |
| Path loss Exponent, $\psi$ | [2,6] |
| Wireless Range $R_t$ | $6, 8, 10$ |
| Energy consumed by a robot to handle a target $c_d$ | 5 |

gives a measure of how the recruiting strategy, using the same exploration strategy, is able to distribute among multiple target's locations the robots and thus the capability of strategy of avoiding redundancy.

Different scenarios are considered by varying the minimum number of robots necessary to disarm a target, the total number of robots in the rescue area, the dimension of grid and the number of disseminated targets.

In Fig. 5.3 and Fig. 5.4 the number of targets and the grid area have been fixed, respectively, to 3 and 20x20 cells. In particular, it is shown the time to complete both tasks measured as the total number of steps and the number of accesses in the cells varying the number of

Table 5.3 Parameters used in the coordination algorithms.

| Parameters | Value |
|---|---|
| $\alpha$ | 0.2 |
| $\beta_0$ | 0.5 |
| $\gamma$ | $\frac{1}{L}$ ($L$=max{m,n}) |
| $\sigma$ | Uniform [0,1] |
| $\omega$ | 0.729 |
| $r_1$ | Uniform [0,1] |
| $c_1$ | 2 |
| $\phi$ | Uniform [-1,1] |

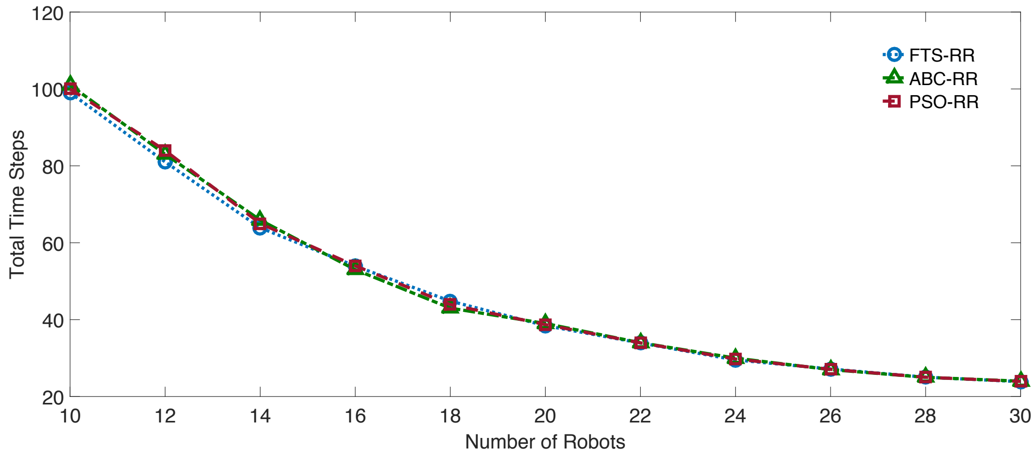
robots that need to be involved for handling a target. The convergence time and number of accesses in the cells was averaged over 100 independent simulation runs.
Fig. 5.3 depicts the total time steps for complete the missions. It is possible to see that no significant difference between the three strategies when the task is not particularity complex that is 2 or 3 robots needed to disarm properly a target. When the number of robots involved in disarming process increases, 4 robots, the difference of the strategies is more evident especially when the team size is low ( 10, 12 robots). In this case, FTS-RR performs slightly better. This is due to the better robots recruitment strategy when more targets are discovered that is able to balance the robots coordination and movements among all targets. On the other hand, when the number of robots increases no significant difference between the strategies is so evident.

The average number of accesses in a cell is plotted in Fig. 5.4. The results show that FTS-RR and ABC-RR are able to balance better, trying to reduce redundancy, the robots in the recruitment phase considering that the exploration phase is common to all algorithms. This determines that a lower average number of accesses in the cells can be obtained. Increasing the number of robots, no significant difference among the algorithms.

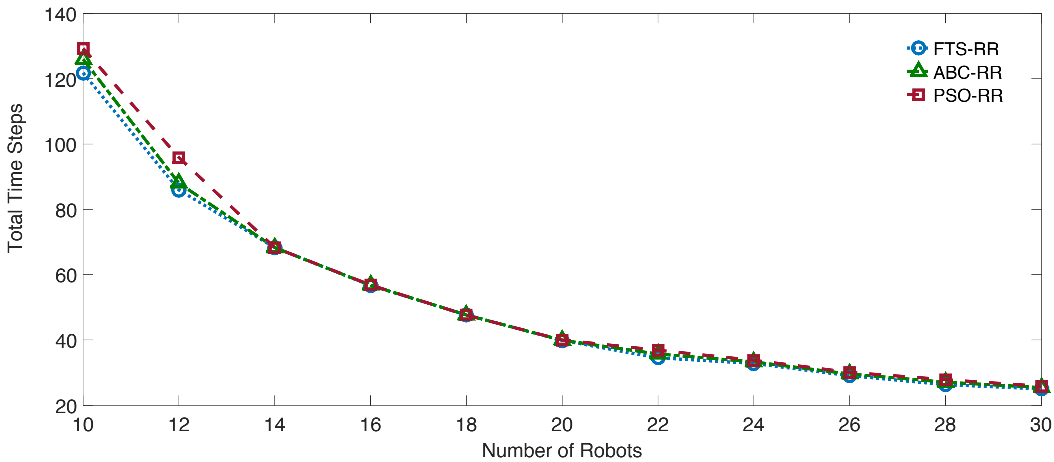
Interesting results are highlighted in Fig. 5.5 and Fig. 5.6, where more targets are introduced in the scenario. In these cases, FTS-RR and ABC-RR perform better for both low and high numbers of robots in the convergence time especially in comparison with the PSO-RR. This is due to the most effective recruitment strategy that is able to better distribute robots when, in the overlapping area, more recruiters can engage robots for disarming. In this case, the firefly algorithm allows robots to spread over different targets avoiding going towards the same targets to disarm. The overall effect is a reduction in the task execution time. Increasing the complexity of the task the difference between the three different algorithms, in terms of overall time to complete the tasks, is greater. This means that the best performing of recruiting task can affect indirectly the discovery task leading to a better distribution of robots among targets to disarm and consequently to explore the novel un-explored cells (De Rango et al., 2015 ), (Palmieri et al., 2015 ).

### 5.3.3 Simulation Experiments II: Evaluation of the total energy consumed by the system

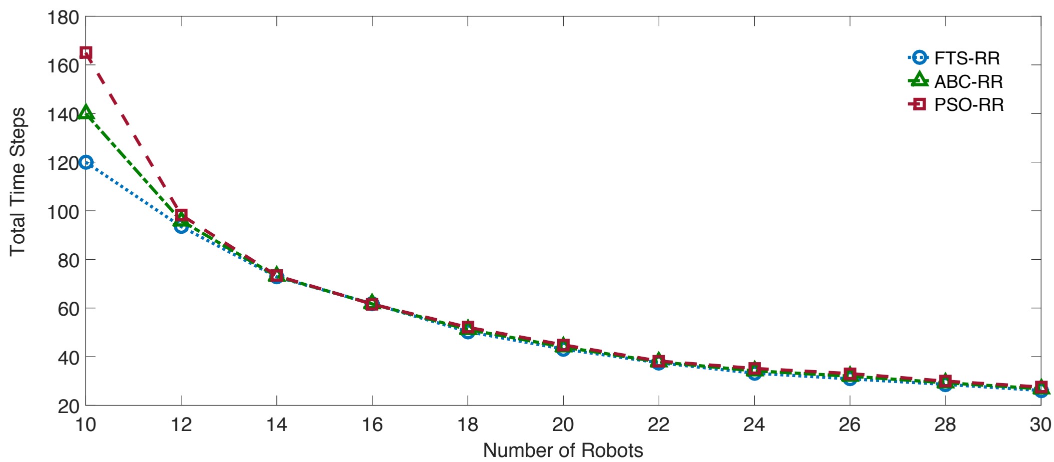
This section analyze the performance of the strategies considering the energy consumption. Energy limitation is one of the most important challenges for mobile robots. A robot is usually comprised of multiple components such as motors, sensors, controllers and embedded computers. The energy consumption is related to the physical and mechanical structure of




(a)




(b)




(c)

Fig. 5.3 Evaluation of Total Time Steps for performing 3 targets varying the number of robots needed to handle a target in an area 20x20 cells (a) 2 robots (b) 3 robots (c) 4 robots.

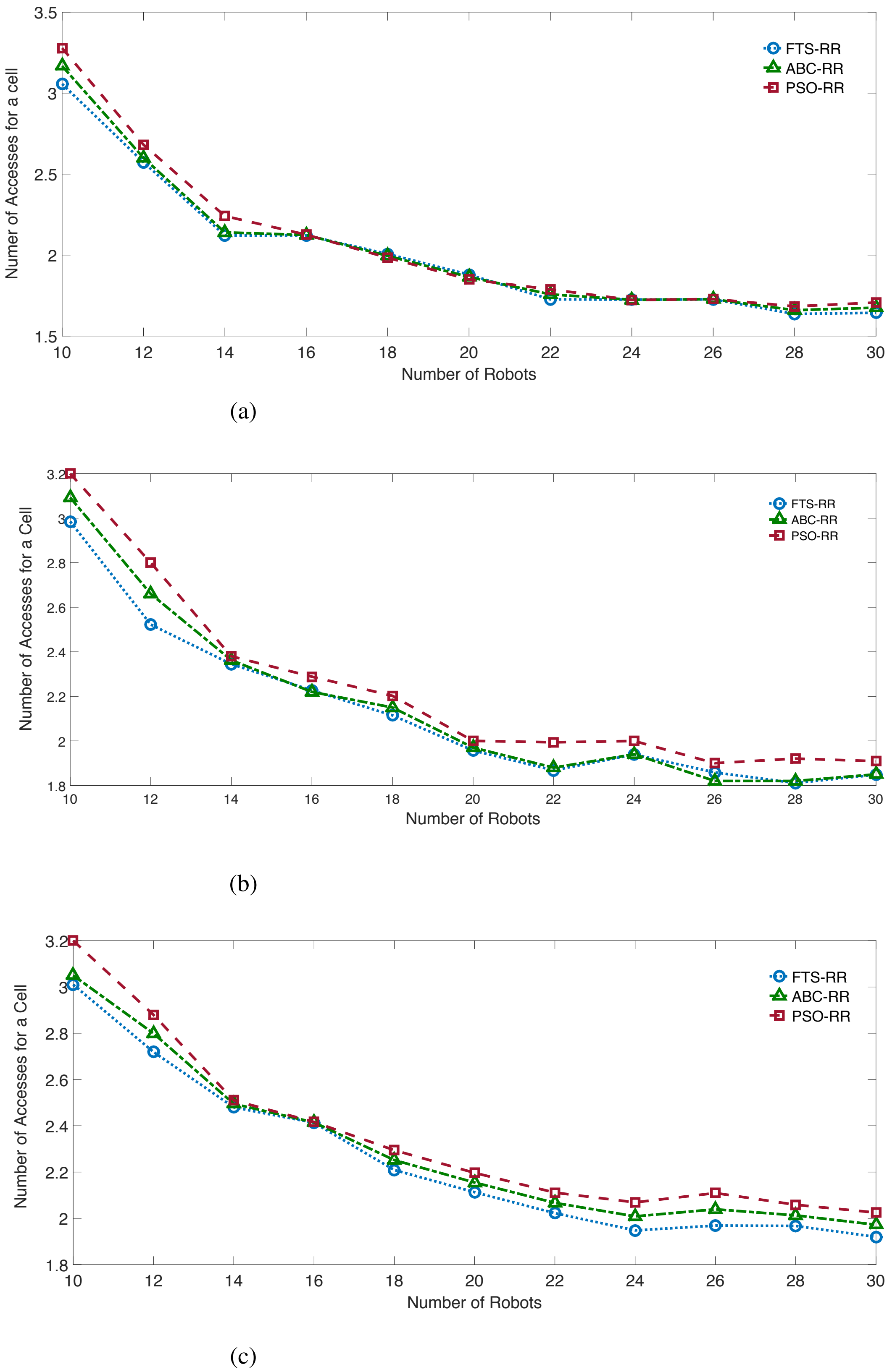



Fig. 5.4 Evaluation of the number of accesses for a cell for performing 3 targets in a grid area 20x20 varying the number of robots involved in the mission and varying the number of robots needed to handle a target in a area 20x20 cells (a) 2 robots (b) 3 robots (c) 4 robots.

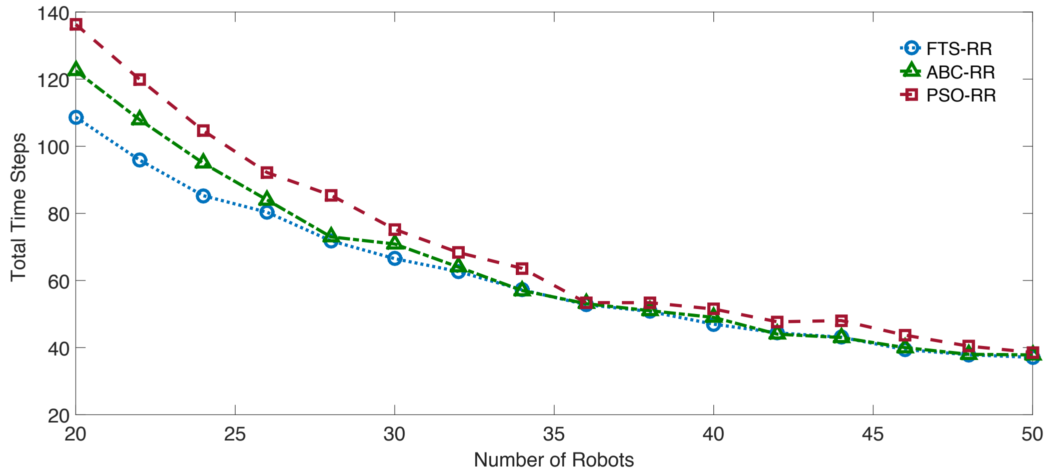



Fig. 5.5 Evaluation of the Total Time Steps considering 30x30 grid area, 5 targets and 4 robots needed to handle a target.

the robots and their abilities for moving, rotating and sensing. The power consumption of a robot can be divided into motion, power, sensing power, control power and computation power accordingly. Batteries are often used to provide power in mobile robots; however, they are heavy to carry and have a limited energy capacity.

### 5.3.4 Case study 1: Influence of the size of the swarm and the dimension of the area on the energy consumption

These experiments are designed to analyze the performance of the coordination strategies by varying the number of the robots in the area $k$={10, 15, 20, 25, 30, 35, 40, 45, 50, 60} and the grid area with different numbers of cells in $x$ and $y$ dimension {40x40, 50x50, 60x60}, keeping a constant number of targets and the number of robots needed to perform a target. The behavior of the approache,s when a few or many robots are used in the area of different sizes, is evaluated. It is considered, also, that for dealing with a target, it is required that 3 robots work together.

The simulation results are summarized in Fig. 5.7 where each point is the average of running the proposed algorithms 100 times and it summarizes the cumulated total energy consumed by the robots (TESC), collected by each algorithm. Results show that, as the size of the robots increases, the average energy of the system decreases and as the size of the operative grid increases the energy consumed increases. It is reasonable to expect that by increasing the number of robots, the efficiency of the swarm improves in terms of energy. Regarding the three strategies, the results of Fig 5.7(a) show that the performance gap is

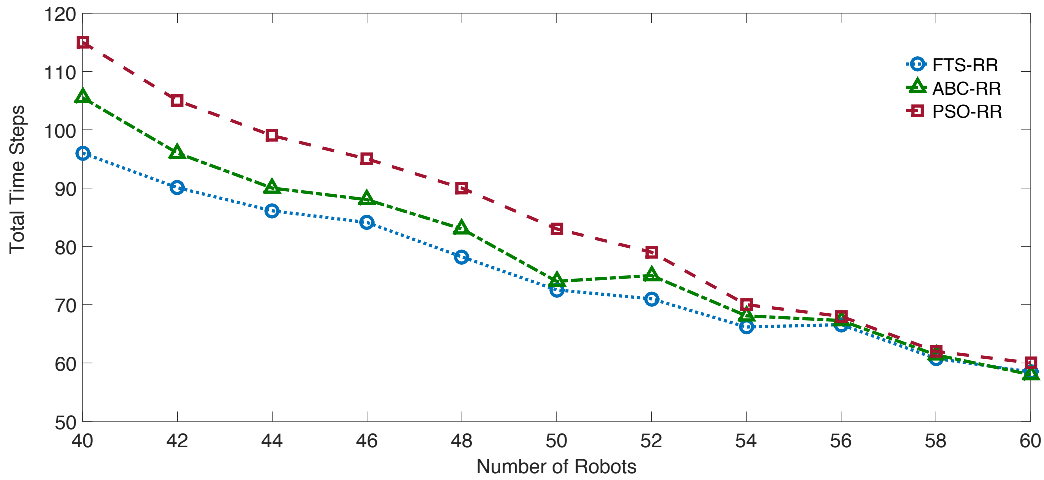



Fig. 5.6 Evaluation of the Total Time Steps considering 40x40 grid area, 10 targets and 4 robots needed to handle a target.

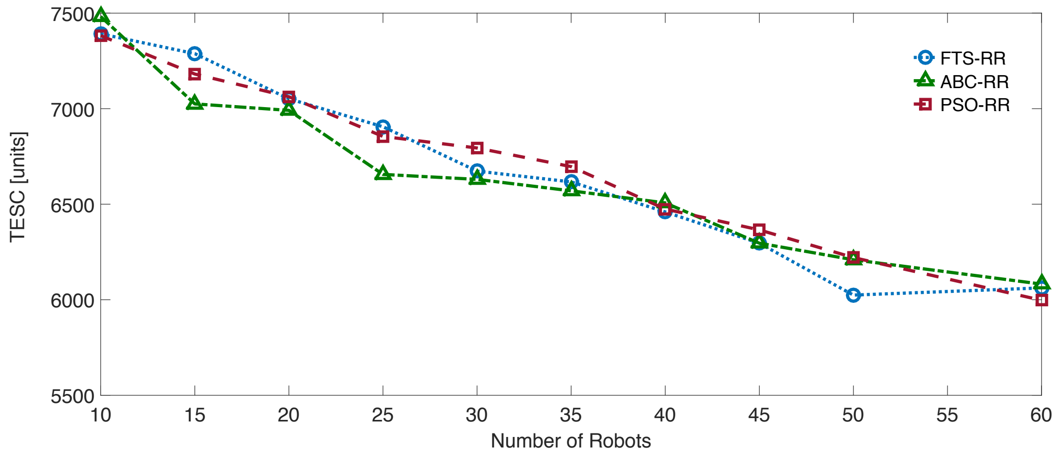
small for a grid area with 40x40 cells, but is higher with the increase of the complexity of the mission as shown in Fig. 5.7(b) and Fig. 5.7(c).

This difference is greater, comparing the PSO-RR with the others. No significant difference between the FTS-RR and ABC-RR. One possible explanation is that the decision mechanisms in FTS-RR and ABC-RR take into account different criteria. PSO-RR takes into account, only, the distance between the positions of the robots and the targets. Instead, FTS-RR considers both distance and random metrics and ATS-RR adopts the roulette rules. Therefore, both approaches, typically, allow to distribute better the robots among the targets.

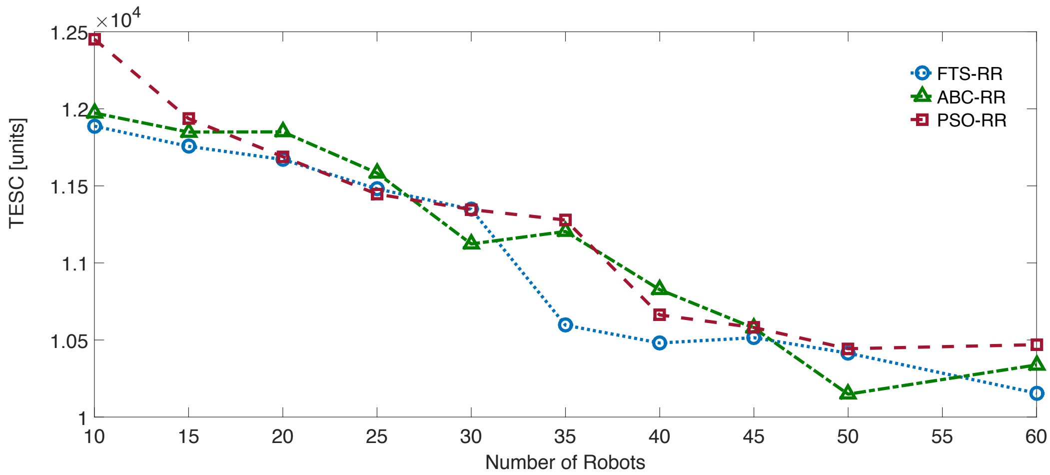
### 5.3.5 Case study 2: Influence of the number of targets on the energy consumption

Now it is evaluated the energy consumed by the system applying the strategies, when few or many targets exist, varying the terrain size and the number of involved robots. We considered $z$={3, 5, 7, 10}, the dimension of the swarm of robots $k$={20, 30, 40} and the grid area with different number of cells in $x$ and $y$ dimension {40x40, 50x50, 60x60 }. Some interesting features can be observed from Fig. 5.8. The ABC-RR and FTS-RR techniques perform better and help to allocate reasonable robots to different targets saving the energy, especially when the number of robots is small. However, a larger robot team obtains more benefit and there is no significant difference between the three strategies.

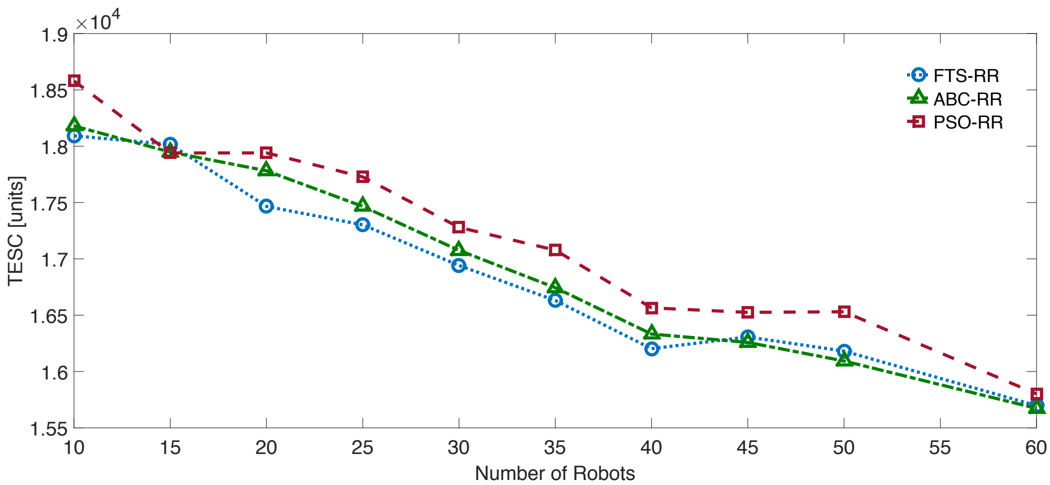
However, a team with a larger number of robots generally increase the performance, saving the total consumed energy. Obviously, the more targets are introduced, the more energy is consumed. Nevertheless, increasing the number of targets, the recruiting tasks




(a)




(b)




(c)

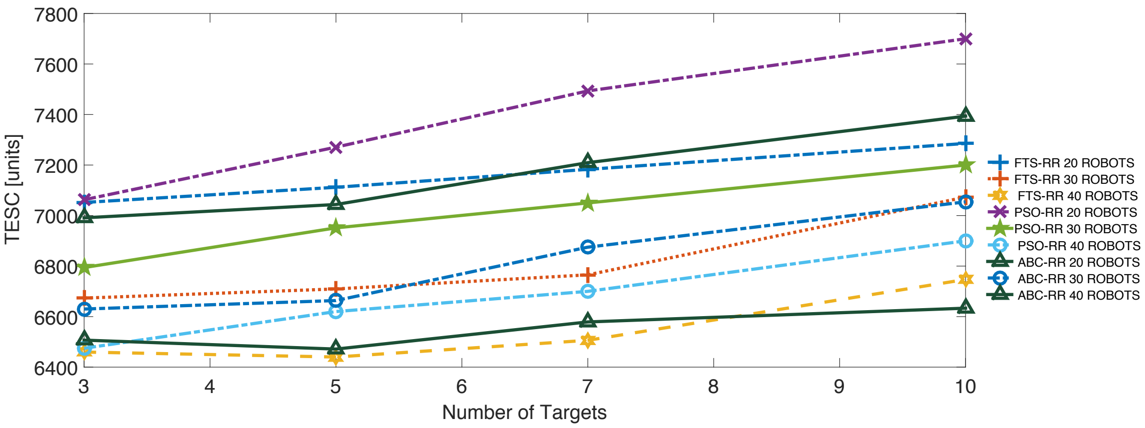
Fig. 5.7 Evaluation of Total-Energy-System-Consumed (TESC) for performing 3 targets varying the number of robots involved in the mission considering 3 robots needed to deal with a single target (a) 40x40 grid (b) 50x50 grid (c) 60x60 grid.

becomes more complex and the used strategy becomes more important. The difference of the three strategies in terms of energy consumption is high, especially when the size of swarm in the operative area is low and it is comparable when the number of robots increases at the same condition of the size of the area. When the complexity of the task increases, it can be seen from Fig. 5.8(b) and Fig. 5.8(c) that it is possible that more robots in an overlapped region receive the same requests, and go towards the same targets, creating unnecessary redundancy. However, in most scenarios, FTS-RR exhibits superior performance and distributes the robots better in the area, especially in comparison with the PSO-RR. Regarding the difference between the FTS-RR and ABC-RR, the measure of the total energy is comparable and not significant difference when the task is not complex and number of robots to coordinate is high. But increasing the number of targets and using a small team ( e.g., 20), the FTS-RR exhibits superior performance in terms of energy consumed. This implies that the FTS-RR would be more promising for solving recruitment tasks in complex scenarios.

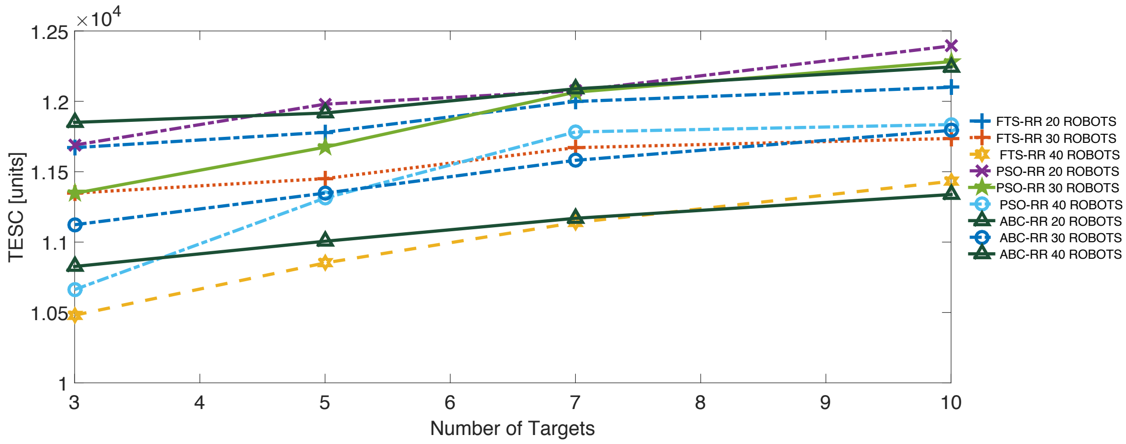
### 5.3.6 Case Study 3: Influence of the wireless range on the energy consumption

The last experiment is designed to evaluate the influence of the wireless range on the energy consumption by varying different ranges $R_t \in \{6, 8, 10\}$. Here it is considered a grid area 50x50, $z = (7, 10, 15)$ and 3 robots needed to treat a target. It is important to point out that effective communication between the robots is highly dependent on the parameters of the problem such as the size of the swarm of robots, and the number of disseminated targets in the area.

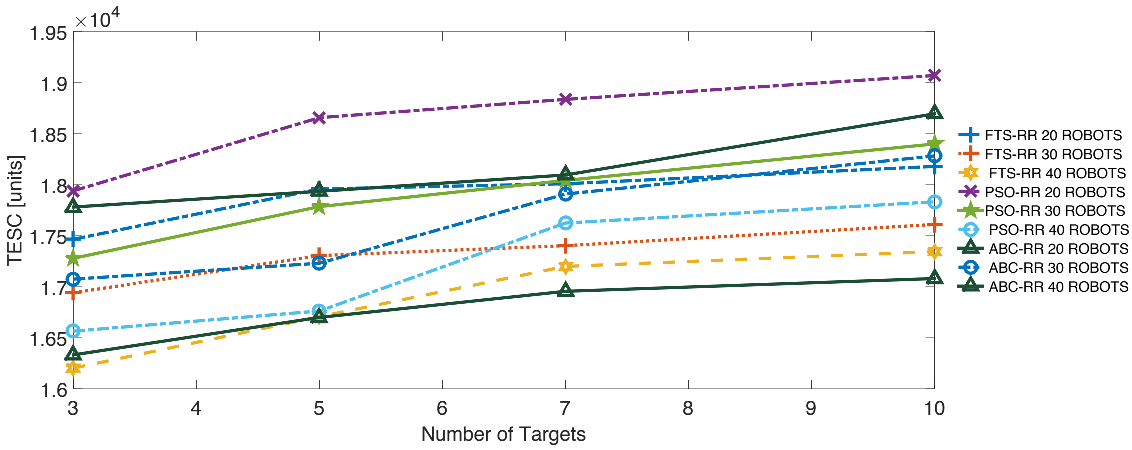
The results are summarized in Fig. 5.9 where some interesting features can be observed. A robot team with a small number of robots (e.g., 20) is mainly affected by the positive side of a high communication range, although a relatively shorter communication range means lower power consumption. The reason in that over long communication range, more robots can be recruited and they can be allocates to different targets in a shorter time. However, the results also show that, when the communication range is increased, the performance improves up to a certain point beyond which there is no change in the performance of the system and in such case the increasing of the total energy consumed. A scenario with a huge amount of robots (e.g., 40) implies a huge amount of consumed energy since the recruitment task involves multiple robots, usually unnecessary, with some consequent waste of energy. For example, Fig. 5.9(a) highlights lower consumption of energy for a larger number of robots using a short communication range than the use of the longer communication ranges (Figs 5.9(b)-(c)). Regarding the three strategies, both FTS-RR and ABC-RR perform better than




(a)




(b)




(c)

Fig. 5.8 Evaluation of Total-Energy-System- Consumed (TESC) for performing 3,5,7,10 targets and 3 robots needed to perform a target (a) 40x40 grid (b) 50x50 grid (c) 60x60 grid.

Table 5.4 Results of p value in t Test for FTS-RR and PSO-RR.

| | FTS-RR vs PSO-RR | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Number of total time steps Fig.5.3(a) | Number of total time steps Fig.5.3(b) | Number of total time steps Fig.5.3(c) | Number of accesses for a cell Fig.5.4(a) | Number of accesses for a cell Fig.5.4(b) | Number of accesses for a cell Fig.5.4(c) | Number of total time steps Fig.5.5 | Number of total time steps Fig.5.6 |
| **pvalue** | 0.0912 | 0.0524 | 0.0958 | 0.0522 | **0.0015** | **0.0045** | **0.0066** | **0.0036** |

Table 5.5 Results of p value in t Test for FTS-RR and ABC-RR.

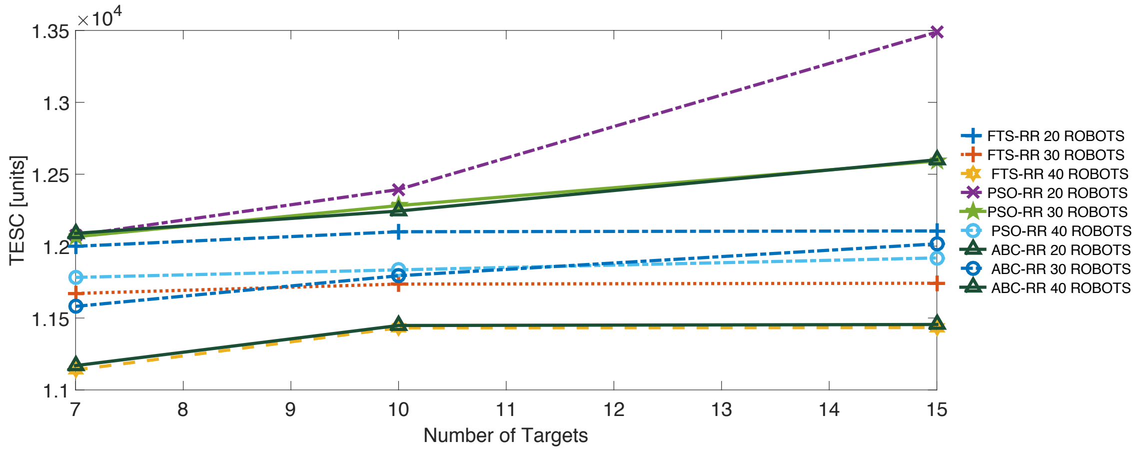
| | FTS-RR vs ABC-RR | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Number of total time steps Fig.5.3(a) | Number of total time steps Fig.5.3(b) | Number of total time steps Fig.5.3(c) | Number of accesses for a cell Fig.5.4(a) | Number of accesses for a cell Fig.5.4(b) | Number of accesses for a cell Fig.5.4(c) | Number of total time steps Fig.5.5 | Number of total time steps Fig.5.6 |
| **pvalue** | 0.1595 | 0.08428 | 0.0887 | 0.0864 | 0.0598 | **0.0075** | **0.0170** | **0.0034** |

the PSO-RR, especially in a small robot team (e.g., 20 robots) and many targets disseminated in the area (e.g., 15). Concerning the difference between the FTS-RR and ABC-RR, FTS-RR outperforms the other mainly in complex scenarios and thus allows to spread the robots in a better way over the environment, avoiding the situation that several robots approach the same target and thus saving the energy.

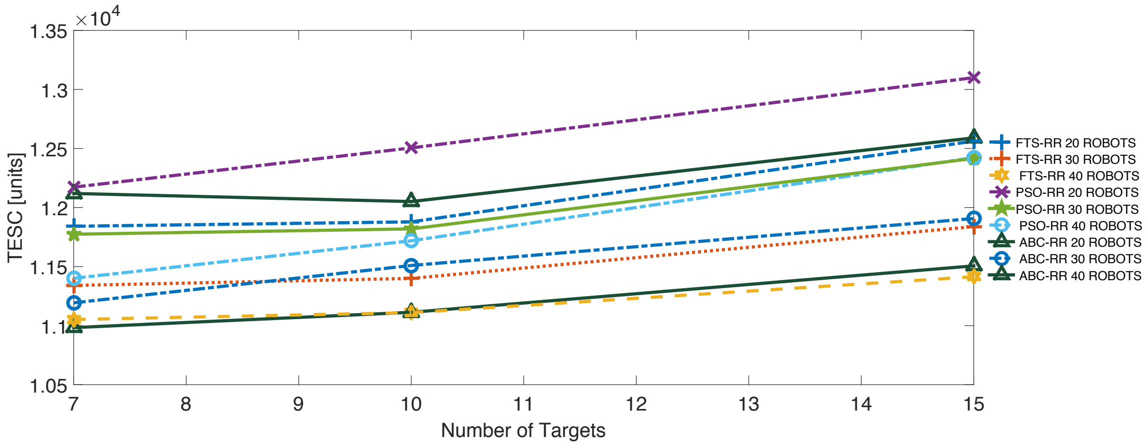
### 5.3.7 Statistical tests

To validate the quality of solutions and performance of the three meta-heuristic techniques,it is also considered the $p$-values of Student $t$-tests. The $t$-tests were used to analyze the relationships between the results obtained from the three meta-heuristics. The parameter of interest is the $p$-value. Tables (5.4)-(5.15), show the $p$-value obtained from the $t$-tests using all above simulation results for all considered scenario. If $p < 0.05$, there is a statistical evidence of the difference between the strategies.

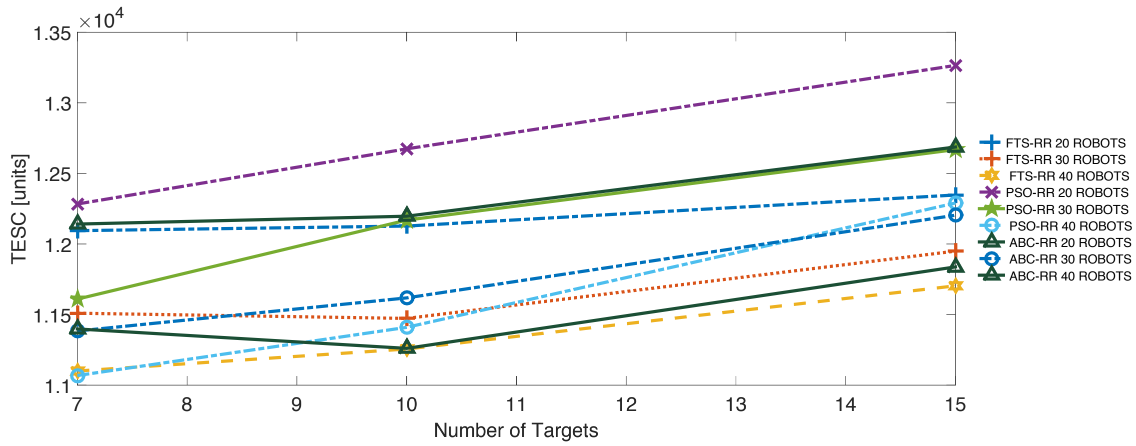
The statistical tests confirm that ABC-RR and FTS-RR perform better than the PSO-RR when the tasks to be completed is complex in terms of the terrain size and the number of targets in the area. Regarding the difference between the FTS-RR and ABC-RR, the performance of the two strategies is comparable. However, increasing the complexity of the tasks in terms of the size of area and the number of targets using a small robots team, the FRS-RR will be better with the slightly reduced energy consumption and the time [92].




(a)




(b)




(c)

Fig. 5.9 Evaluation of Total-Energy-System-Consumed (TESC) for performing 7,10,15 targets and 3 robots needed to perform a target in 50x50 grid (a) $R_t = 6$ (b) $R_t = 8$ (c) $R_t = 10$.

Table 5.6 Results of p value in t Test for PSO-RR and ABC-RR.

| | PSO-RR vs ABC-RR | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Number of total time steps Fig.5.3(a) | Number of total time steps Fig.5.3(b) | Number of total time steps Fig.5.3(c) | Number of accesses for a cell Fig.5.4(a) | Number of accesses for a cell Fig.5.4(b) | Number of accesses for a cell Fig.5.4(c) | Number of total time steps Fig.5.5 | Number of total time steps Fig.5.6 |
| **pvalue** | 0.8423 | 0.0918 | 0.2037 | **0.0403** | **0.0467** | **0.0084** | **0.0066** | **0.0046** |

Table 5.7 Results of p value Test for FTS-RR, PSO-RR

| | FTS-RR vs PSO-RR | | |
|---|---|---|---|
| | Fig.5.7(a) | Fig.5.7(b) | Fig.5.7(c) |
| **pvalue** | 0.1961 | **0.0156** | **0.0012** |

Table 5.8 Results of p value Test for FTS-RR and ABC-RR.

| | FTS-RR vs ABC-RR | | |
|---|---|---|---|
| | Fig.5.7(a) | Fig.5.7(b) | Fig.5.7(c) |
| **pvalue** | 0.2421 | 0.0879 | 0.0544 |

Table 5.9 Results of p value Test for PSO-RR and ABC-RR.

| | PSO-RR vs ABC-RR | | |
|---|---|---|---|
| | Fig.5.7(a) | Fig.5.7(b) | Fig.5.7(c) |
| **pvalue** | 0.0584 | 0.1218 | **0.0028** |

Table 5.10 Results of p value in t Test for FTS-RR and PSO-RR.

| | FTS-RR vs PSO-RR | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | 20 Robots varying the number of targets Fig.5.8(a) | 30 Robots varying the number of targets Fig.5.8(a) | 40 Robots varying the number of targets Fig.5.8(a) | 20 Robots varying the number of targets Fig.5.8(b) | 30 Robots varying the number of targets Fig.5.8(b) | 40 Robots varying the number of targets Fig.5.8(b) | 20 Robots varying the number of targets Fig.5.8(c) | 30 Robots varying the number of targets Fig.5.8(c) | 40 Robots varying the number of targets Fig.5.8(c) |
| **pvalue** | **0.0412** | **0.0158** | **0.0221** | **0.0489** | **0.0455** | **0.0103** | **0.0267** | **0.0405** | **0.0277** |

Table 5.11 Results of p value in t Test for FTS-RR and ABC-RR.

| | FTS-RR vs ABC-RR | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | 20 Robots varying the number of targets Fig.5.8(a) | 30 Robots varying the number of targets Fig.5.8(a) | 40 Robots varying the number of targets Fig.5.8(a) | 20 Robots varying the number of targets Fig.5.8(b) | 30 Robots varying the number of targets Fig.5.8(b) | 40 Robots varying the number of targets Fig.5.8(b) | 20 Robots varying the number of targets Fig.5.8(c) | 30 Robots varying the number of targets Fig.5.8(c) | 40 Robots varying the number of targets Fig.5.8(c) |
| **pvalue** | 0.4812 | 0.4921 | 0.4189 | **0.0412** | 0.1005 | 0.1675 | 0.0798 | 0.0837 | 0.1933 |

Table 5.12 Results of p value in t Test for PSO-RR and ABC-RR.

| | **PSO-RR vs ABC-RR** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | 20 Robots varying the number of targets Fig.5.8(a) | 30 Robots varying the number of targets Fig.5.8(a) | 40 Robots varying the number of targets Fig.5.8(a) | 20 Robots varying the number of targets Fig.5.8(b) | 30 Robots varying the number of targets Fig.5.8(b) | 40 Robots varying the number of targets Fig.5.8(b) | 20 Robots varying the number of targets Fig.5.8(c) | 30 Robots varying the number of targets Fig.5.8(c) | 40 Robots varying the number of targets Fig.5.8(c) |
| **pvalue** | **0.0247** | **0.0459** | 0.0663 | 0.4469 | **0.0445** | 0.0889 | **0.0192** | **0.0451** | **0.0419** |

Table 5.13 Results of p value in t Test for FTS-RR and PSO-RR related to the wireless range

| | **FTS-RR vs PSO-RR** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | 20 Robots varying the number of targets Fig.5.9(a) | 30 Robots varying the number of targets Fig.5.9(a) | 40 Robots varying the number of targets Fig.5.9(a) | 20 Robots varying the number of targets Fig.5.9(b) | 30 Robots varying the number of targets Fig.5.9(b) | 40 Robots varying the number of targets Fig.5.9(b) | 20 Robots varying the number of targets Fig.5.9(c) | 30 Robots varying the number of targets Fig.5.9(c) | 40 Robots varying the number of targets Fig.5.9(c) |
| **pvalue** | 0.1426 | **0.0469** | **0.0186** | **0.0276** | **0.0112** | **0.0413** | 0.0600 | 0.0651 | 0.1633 |

Table 5.14 Results of p value in t Test for FTS-RR and ABC-RR related to the wireless range

| | **FTS-RR vs ABC-RR** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | 20 Robots varying the number of targets Fig.5.9(a) | 30 Robots varying the number of targets Fig.5.9(a) | 40 Robots varying the number of targets Fig.5.9(a) | 20 Robots varying the number of targets Fig.5.9(b) | 30 Robots varying the number of targets Fig.5.9(b) | 40 Robots varying the number of targets Fig.5.9(b) | 20 Robots varying the number of targets Fig.5.9(c) | 30 Robots varying the number of targets Fig.5.9(c) | 40 Robots varying the number of targets Fig.5.9(c) |
| **pvalue** | 0.0978 | 0.2625 | **0.0317** | 0.0795 | 0.4542 | 0.4321 | 0.1237 | 0.2523 | 0.1142 |

Table 5.15 Results of p value in t Test for PSO-RR and ABC-RR related to the wireless range

| | **PSO-RR vs ABC-RR** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | 20 Robots varying the number of targets Fig.5.9(a) | 30 Robots varying the number of targets Fig.5.9(a) | 40 Robots varying the number of targets Fig.5.9(a) | 20 Robots varying the number of targets Fig.5.9(b) | 30 Robots varying the number of targets Fig.5.9(b) | 40 Robots varying the number of targets Fig.5.9(b) | 20 Robots varying the number of targets Fig.5.9(c) | 30 Robots varying the number of targets Fig.5.9(c) | 40 Robots varying the number of targets Fig.5.9(c) |
| **pvalue** | 0.1781 | **0.0031** | **0.0181** | 0.0712 | **0.0288** | **0.0471** | **0.0469** | 0.0518 | 0.3647 |

### 5.3.8 Robot in dynamic scenario

The above considerations provide a unified approach to consider both the complete discovery of the area and the measure of the performance needed to accomplish both exploration and disarming of the targets. This is a useful metric, but it requires that the task is completely finished, and cannot be used to evaluate partial execution of the tasks. In many case a complete exploration of the environment may not be feasible in practice, due to the time or resource constraints in large and hazardous environments.

In this section, a dynamic environment is considered, in the sense that the targets can explode at any time and in an unpredictable manner, mimicking the destruction of some robots and the damage of the nearby zones. Moreover, the robots are considered with a limited quantity of energy without the possibility of recharge or replacement. In such scenarios, the team works under more demanding time constraints.

In order to use a performance metric that is applicable to the robotic system in a dynamic scenario, several functions have been taken into account; one for each feature that must be discovered and measured from the environments. More specifically, the performance metrics are given by each function measuring the percentage/ratio of information related to the two tasks. In the case of exploration task, it is the percentage of the environment explored not covered by impassable obstacles, while in the case of the disarmament task, it is the percentage of targets successfully identified and disarmed.
The following equations summarize the region of an emergency scene as follows:

$$A_E = \sum_{x=1}^{m} \sum_{y=1}^{n} c_{(xy)} c_{xy} \in C_1, \tag{5.23}$$

$$A_{UN} = \sum_{x=1}^{m} \sum_{y=1}^{n} c_{(xy)} c_{xy} \in C_2, C_4. \tag{5.24}$$

Concerning the above regions of interest, we define the following terms:

$$F_1 = \frac{A_E}{\sum_{x=1}^{m} \sum_{y=1}^{n}, c_{(xy)}} c_{xy} \in C_1, C_2, C_4 \tag{5.25}$$

where $F_1$ is a regularized term that indicates the percentage of explored cells in the emergency scene. Thus, $F_1$ will be equal to one only in the case all cells of the area have been explored, except for the cells with obstacles $(c \in C_3)$.

Now we define the number $F_2$ of handled targets as follows:

$$F_2 = \sum_{z=1}^{N^T} f(z) = \begin{cases} 1 & \text{if target z is disarmed properly,} \\ 0 & \text{otherwise.} \end{cases} \tag{5.26}$$

In this case, the objectives essentially become the maximization of the percentage of explored area and the number of disarmed targets. In this case, the robots have a limited amount of energy and at each time step, a fixed quantity of energy is consumed (see Section 4.2) depending on what action the robot may perform and if the mines can explode. The mission can terminate for multiple reasons, including the case that all robots have used up the energy, or are damaged due to explosion.

### 5.3.9 Influence of the coordination strategies introducing energy constraints

These experiments are designed to analyze the performance of bio inspired strategies varying the minimum number of robots required to resolve a target. At the beginning, it is assumed that a robot has 1000 energy units [114], without possibility of recharging during the mission, which means that if a robot consumes its energy, it will stop to perform the task at any time. In this case, to achieve good coordination and exploration is more challenging since it is required that the robots team has to respond quickly, robustly, reliably and adaptively to unexpected events.

Each point in the graphs shown in Fig. 5.10 is the average of running the proposed algorithms 100 times. Firstly, the performance of the proposed algorithms is measured in terms of percentage of explored cells of the map. In the environment are disseminated seven targets and the number of robots needed to resolve a target is varied. This metrics could tell us how good the algorithms can achieve allocate the robots into different targets in more reasonable manner in a situation when there is a sudden stop of one or more robots for the battery consumption. Results show that as the number of robots increases the swarm is able to explore all cells in the area; if the number of robots in the area is low, using the same strategy to explore, the coordination strategy influences the distribution of the robots in the area and therefore the number of cells that they can explore. Increasing the complexity of the task in terms of number of robots to recruit, the size of the swarm and the strategy applied influence significantly the performance; using a small team to accomplish the mission, FTS-RR and ABC- RR perform better than the PSO-RR approach. If the size of swarm increases the results are comparable.

Now how the coordination strategy affects the number of completed targets is investigated. In each experiment, the number of targets is gradually increased. Fig. 5.11 shows that in a small robots team, the used strategy could have effects in the resolving the disseminated targets. However a large robots team obtains more benefit and there is no significant difference between the three strategies. Increasing the number of targets in the area as depicted in Fig. 5.11 (b) and Fig. 5.11 (c), the difference of the strategies is high especially between the FTS-RR and ABC-RR in comparison with the PSO-RR.

It should be noticed that the performance of the techniques, mostly, depends on the number of robots in the area. The difference is evident for a small robots team. Here the recruiting task is more complex since it is necessary allocated required robots in the defined regions avoiding that many robots go towards the same target and saving energy. In this case the firefly-based strategy usually gives superior performance [91].

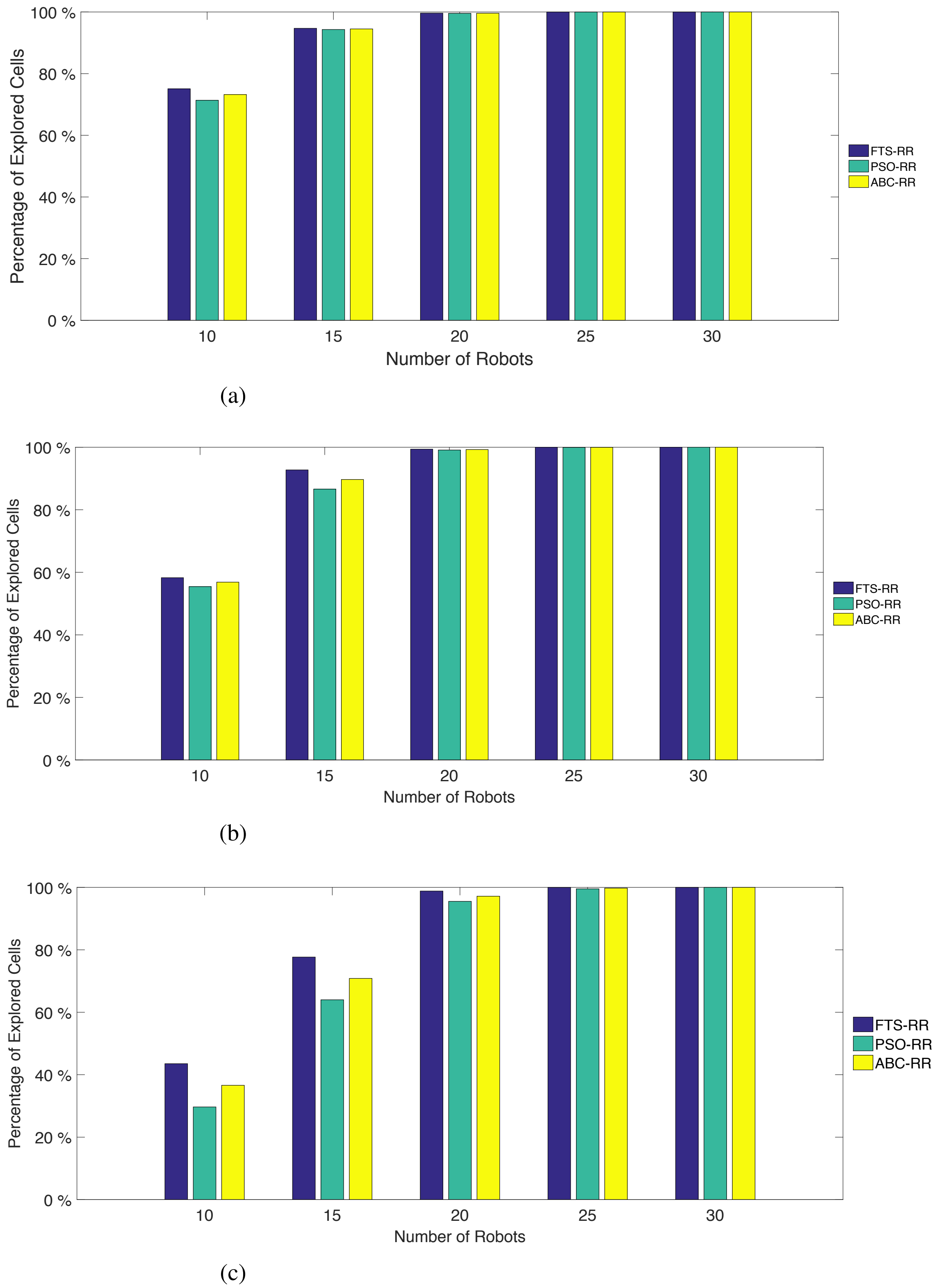
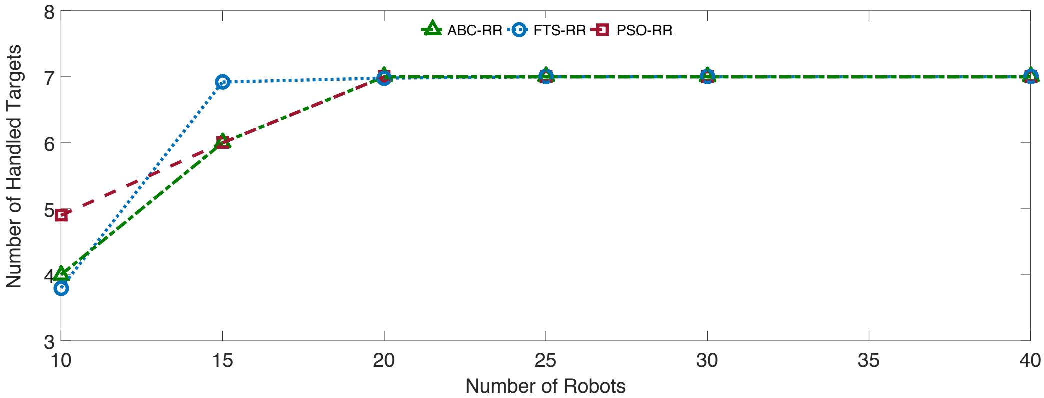
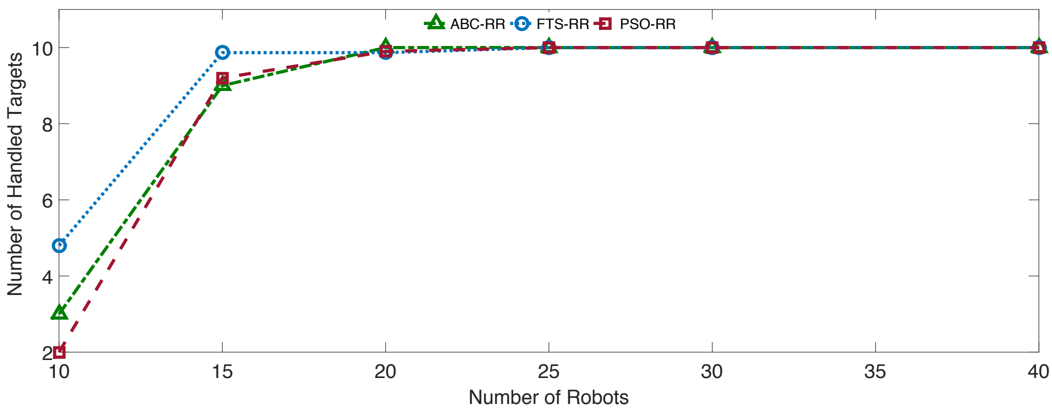


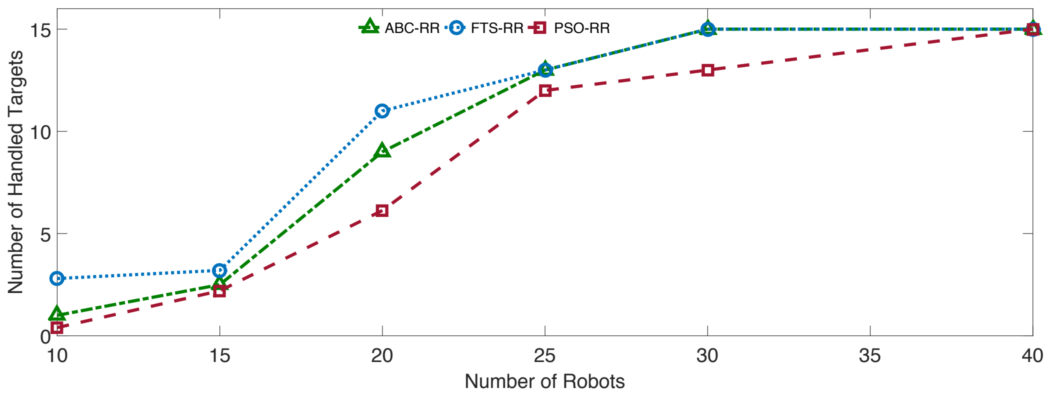

Fig. 5.10 Percentage of Explored Cells evaluation for resolving 7 targets in a grid 50x 50 (a) 2 robots needed to disarm a target (b) 3 robots needed to resolve a target (c) 4 robots needed to resolve a target.




(a)




(b)




(c)

Fig. 5.11 Number of handled targets in a grid 50x 50 and 3 robots needed to resolve a target (a) 7 targets (b) 10 targets (c) 15 targets.

## 5.4 Balance the two tasks in Pareto Sense

This section addresses the problem of considering the bi-objective problem that means, a good trade-off among the two objectives such as exploration and recruitment must be found. For this purpose, an analytical mathematical model to solve this problem considering a single equivalent weighted objective function are presented. In this case it is considered only the firefly algorithm, that has demonstrated, potentially, better performance in many scenarios.

## 5.5 Multi-Objective Optimization Formulation

Multiple conflicting objectives may arise naturally in most real-world robotic optimization problems. Several principles and strategies have been developed and proposed for over the last decades in order to solve such problems. In multi-objective optimization, as its name implies, there are multiple objective functions with a possibility of conflicting with each other. The aim is to find a set of vectors of decision variables that can satisfy constraints and optimize (minimizes or maximizes) these functions. Such solution vectors are not a unique vector, there are many such solutions vectors forming a so-called Pareto front. Each point or non-dominated solution on the Pareto front provides a preference and choice between different objectives. When the Pareto front becomes convex, weighted sum methods can aggregate different objectives into a single objective.

In general, a multi-objective optimization problem can be written mathematically as

$$\begin{aligned} \text{To find vectors} \quad & X = (x_1, x_2, \dots, x_l)^T \in \Omega \\ \text{which optimize} \quad & f(x) = (f_1(x), f_2(x), \dots, f_p(x)) \\ \text{subject to} \quad & g_j(x) \leq 0 \quad j = 1, \dots, s, \\ & h_r(x) = 0, \quad r = 1, ..., d, \end{aligned} \tag{5.27}$$

where $f_1(x), f_2(x), \dots, f_p(x)$ denote the objective functions to be optimized simultaneously, $X$ is the vector of the decision variables in the search/decision space. $\Omega$ is the set of feasible solutions and $g_j(x)$ denotes the inequality constraints, while $h_r(x)$ are equality constraints. All these functions can be linear or nonlinear [5].

As it is very difficult to effective handle with all the conflicting objective functions, several methods have been developed for this purpose. One of these methods is that the multi-objective problem is transformed into a single-objective problem by a weighted sum. In this work, in order to solve our bi-objective problem, the weighted sum method is used to deal with conflicting goals and the solutions can be obtained as a trade-off of the specific problem. The total cost of the fitness function is obtained by a linear combination of the weighted sum of two objectives in which each objective function is based on its importance or preference [78].

The problem is transformed into a single-objective optimization problem by using scalar weighting factors associated with each objective function:

$$F_{\text{weighted sum}} = \sum_{i=1}^{p} w_i f_i(x) \tag{5.28}$$

where $w_i$

$$w_1 + w_2 +, \ldots w_p = 1, \quad w_i \geq 0. \tag{5.29}$$

The weighted sum method changes weights systematically, and each different single objective optimization determines a different optimal solution. This approach gives an idea about the shape of the Pareto front and allows information to be obtained about the trade-off among the various objectives to accumulate gradually [18].

The problem was formulated as a bi-criteria model which turns out to be very challenging to solve. Indeed, the number of efficient solutions may be exponential in terms of the problem size, thus prohibiting any efficient method to determine all efficient solutions. For these reasons, following the popular approaches used to deal with multi-objective optimization problems, the model has been transformed into a single objective optimization problem using arbitrary importance factors for each criteria ( i.e. $w_1$ and $w_2$) and combining the objectives as a single function to be minimized. The resulting single objective problem with non-negative weights can be represented as follows:

$$\text{minimize} \sum_{k=1}^{N^R} \sum_{x=1}^{m} \sum_{y=1}^{n} w_1 (T_e v_{xy}^k) + \sum_{k=1}^{N^R} \sum_{z=1}^{N^T} w_2 [(T_{End,z}^k - T_{Start,z}^k)] u_z^k, \tag{5.30}$$

subject to constraints (3.6)-(3.10).

Parameters $w_1$ and $w_2$ are chosen such that the condition $w_1 + w_2 = 1$ is satisfied. In this case, the combined function is Pareto optimal [18]. The user can choose appropriate values for the parameters $w_1$ and $w_2$, depending on the preference or priority of the objectives. Indeed, by minimizing the weighted sum objective with various settings, it is possible to determine various points in the Pareto set. This approach can approximate and describe the shape of the Pareto front effectively, allowing the accumulation of information to be obtained about the trade-off among various objectives.

The proposed single objective optimization model can be solved and be relevant to many applications for robot exploration and coordination. For applications in which more relevance is given to the exploration task, more importance could be given to exploration time (thus higher value of $w_1$), whereas for applications where it is more important to reduce the disarming time, more importance could be given to $w_2$.

By minimizing the overall fitness function in regard to the assigned weights of each criterion, a suitable decision mechanism that may balance the two objectives can be obtained. The weights have been tuned through a set of simulations in order to try to find the best values.

## 5.6 Computational Experiments

Regarding the simulations, there are several test-related parameters that may influence the performance and the results and they are listed as follows:

- Area size: considering scenarios with and without obstacles.
- Robot density: This is the total number of robots in the swarm $|N^R|$.
- The number of targets $|N^T|$.
- The number of coalitions that is the minimum number of robots that can handle properly a target $R_{min}$.
- The transmission range $R_t$, which can have an effect on the recruiting task

### 5.6.1 Evaluation of the Weights under Static Conditions

In this section it is assumed that the robots have sufficient resources in terms of energy to execute the mission and the targets are static without possibility of explosion.

### 5.6.2 Case study 1

In the first set of experiments, the influence of the weights on the dimension of the area is taken into account. It is considered 50x50 square cells, 100x100 square cells, varying the team size (25, 40, 50 robots) and the number of dispersed targets. All experiments were carried out using 3 robots needed to handle a target properly.

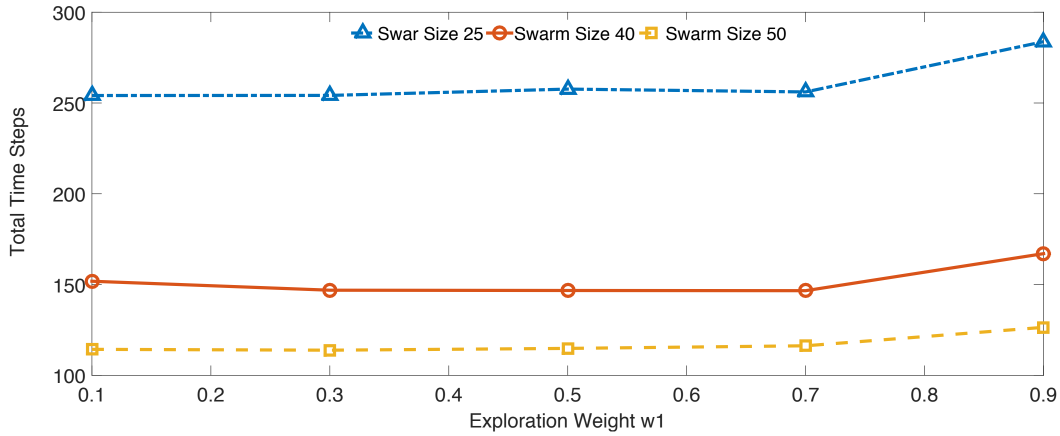
Figures 5.12 and 5.13 show the influence of the $w_1$ considering different swarm size and dimension of the area evaluating respectively the total time steps to complete a mission and the total energy consumed by the swarm. It can be observed that the time steps increase as the value $w_1$ increases when the size of the swarm is small. This behaviour can be explained, by observing the nature of the mission that implies the collaboration of more robots in target's locations. When $w_1$ increases, the robots are highly motivated to explore the area. Since, the mission is considered complete if all target are found, motivating the robots to explore the area than disarming targets, which can lead to a temporary deadlock, especially when the swarm size is small. This implies the decrease of the performance of the entire system. On the other hand, a team with a larger number of robots generally increases the performance improvements. The curves do not fluctuate a lot and the total time steps is almost similar for different $w_1$ values. This implies that the influence of $w_1$ on the performance in general decreases, considering an adequate swarm size.

Regarding the energy consumption, it is can be seen that a high wastage of resource, considering the same total time steps, when $0.7 \leq w_1 \leq 0.9$. This difference is higher in teams with a low number of robots, compared to the number of disseminated targets $\frac{N^R}{R_{min} * N^T} \ll 1$ , and in a big grid area (e.g., 30 robots operating in 100x100 cells for treating 20 targets). Considering both the total time steps and the total energy consumption, for almost all cases the best range is $0.3 \leq w_1 \leq 0.5$.

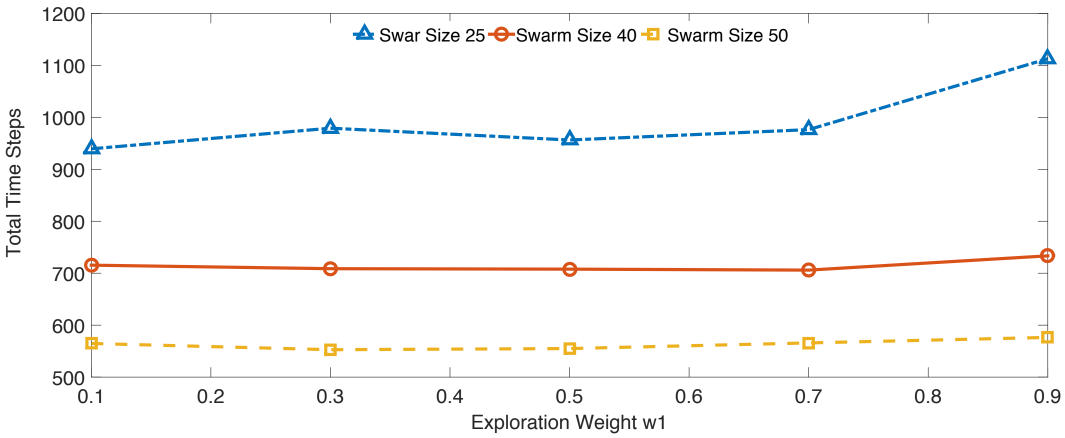
### 5.6.3 Case study 2

The second set of simulations compares the performance by varying the transmission range $R_t$ (6, 15 units of cells) considering a grid area 50x50 , a team with 40 robots and by varying the number of disseminated targets (15, 20, 35). This can play an important role in recruiting tasks, since for a higher transmission range, the probability that more robots are recruited increases. Figure 5.14 shows the total time steps under different conditions in terms of dispersed targets and the same swarm size (40 robots operating in the area).

It can be observed that the increase of the transmission range does not always imply the increase in performance in terms of time steps. The reason can be that if resources are enough in terms of robots as shown in Fig. 5.14 (a)-(b), an increase of the transmission range can cause some redundancy with the wastage of time to complete the mission. For example, considering a small team compared to the targets (40 robots and 35 targets), Fig. 5.14 (c) shows that a high transmission range with a small $w_2$ may imply a better performance. By increasing $w_2$, more robots may be involved in the recruitment task and there is no significant difference between the two ranges. As expected, if the number of the targets to be handled is small, a high transmission range deteriorates the performance since unnecessary robots could be involved in the disarmament process, depleting resources for exploration task and eventually to other targets. Although the total time steps are somewhat better, the performance in terms of energy consumed by the system strongly degrades, using a high




(a)



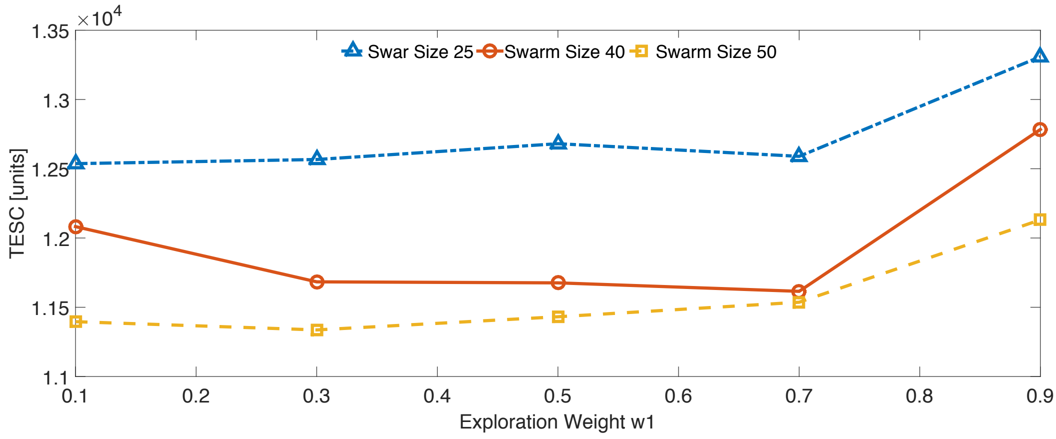

(b)

Fig. 5.12 Evaluation of the Total Time Steps to complete the mission and 3 robots needed to handle a target. (a) 50x50 grid area, 10 targets to disarm (b) 100x100 grid area, 20 targets to be disarmed.

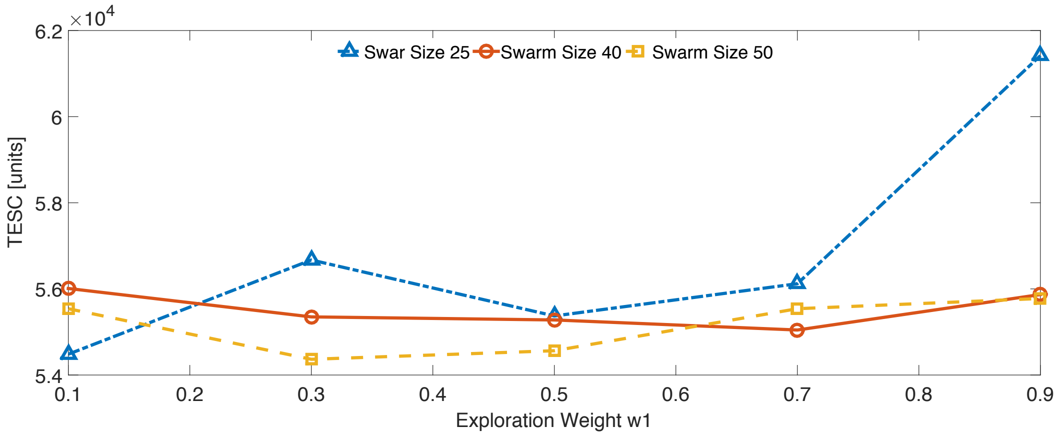
transmission range. Regarding the impact of $w_2$ on the performance considering the energy, the effect of increasing the transmission range can be quantitatively notated by looking at Fig. 5.15. The results confirm that, especially for complex missions with $\frac{N^R}{R_{min} * N^T} \ll 1$, more importance should be emphasized to the recruiting weight, thus $w_2 \geq 0.3$.

### Case study 3

The third set of simulations investigates the effect of the weights in relation to the number of disseminated targets. The performance measures have been evaluated by varying the dimension of the area, the swarm size and the number of robots that can deal with a target (2, 3, 4, 5).




(a)




(b)

Fig. 5.13 Evaluation of the Total-Energy-System- Consumed (TESC) (a) 50x50 grid area, 10 mines to disarm and 3 robots needed to handle a target. (b) 100x100 grid area, 20 targets to disarm and 3 robots needed to handle a target.

The importance of choosing the $w_2$ weight properly increases as the number of the dispersed targets increases and it depends mostly on the dimension of the swarm. More specifically, if $\frac{N^R}{R_{min} * N^T} \ll 1$ means the task is complex in terms of disarmament, more importance can be attributed to $w_2$ than $w_1$ as shown in Figures (5.16)- (5.17).

On the other hand, if $\frac{N^R}{R_{min} * N^T} \approx 1$, no significant influence in terms of total time steps is observed. Obviously, more robots are introduced, less wastage of time can be observed and $w_2$ becomes less relevant.

In another set of experiments, it is introduced an additional parameter to control the task complexity; that is, the number of robots $R_{min}$ required for treating a target. In this way, we can vary the task complexity and observe its influence on the impact of $w_2$. Fig. 5.18

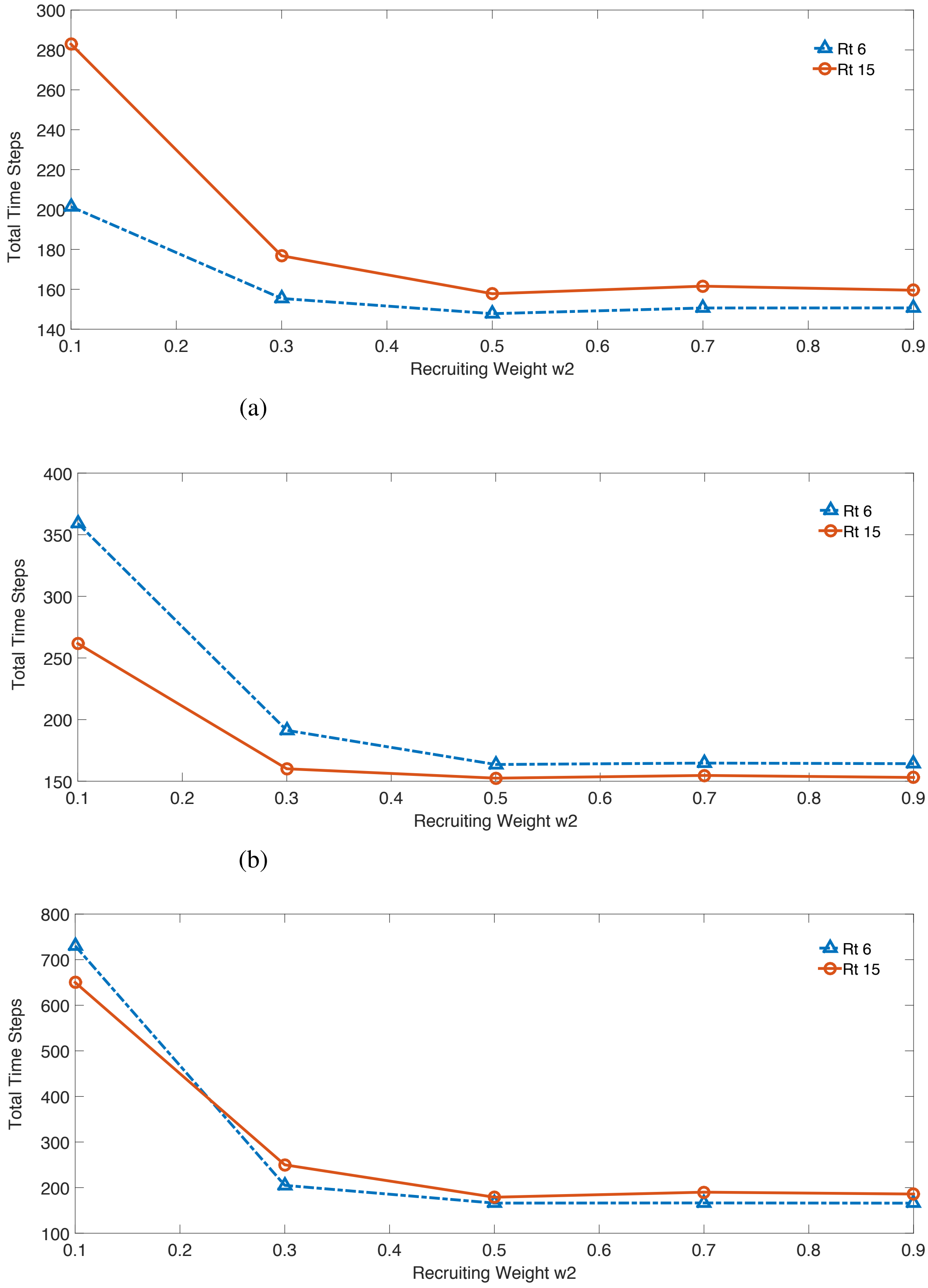



Fig. 5.14 Evaluation of the total time steps to execute the mission in 50x50 squares and 40 robots and 3 robots to handle a target. (a) 15 targets to be handled. (b) 20 targets to be handled. (c) 35 targets to be handled.

shows the impact of $R_{min}$ and $w_2$ on the performance in terms of total time steps. It can be noticed that a high number of robots necessary to handle a target (5 robots to disarm) can cause severe resource consumption in terms of total time steps, if a small value is assigned to $w_2$. This leads to weakening the ability of the robots to distribute into a target's position. As a result, the robots wondering in the area increase the time to complete the mission and the coordinators could be trapped in target's location for a long time. Thus, increasing $R_{min}$, the $w_2$ weight should be increased in order to speed up the formation of the coalition. So in this case, $w_2$ can greatly influence the performance and a proper value should be chosen ($w_2 \geq 0.5$). On the other hand, if the disarmament task is not particularly complex, the influence of the $w_2$ decreases.

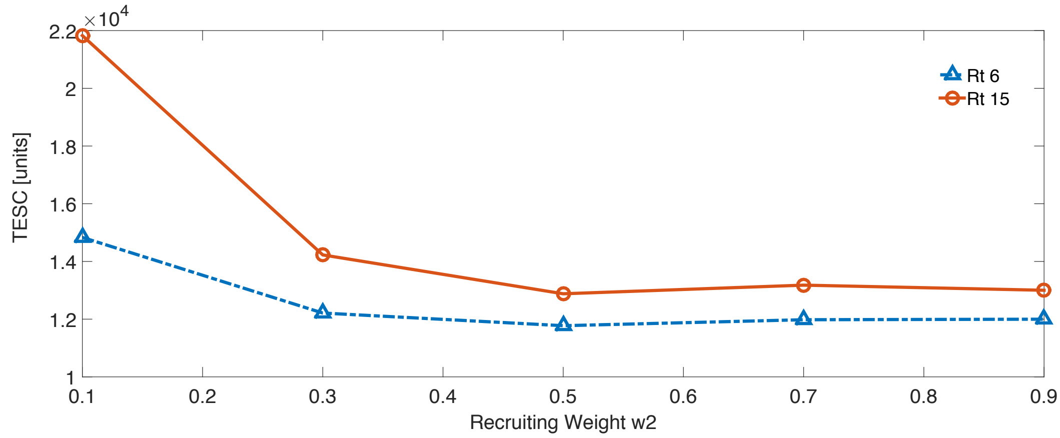
### 5.6.4 Evaluation of weights under dynamic conditions

This section investigates the effect of the weights operating in a dynamic scenario where unpredictable events can occur (such as explosion of the mines and energy constraints). It is assumed that a robot has 1000 energy units [114], without possibility of recharging during the mission, which means that if a robot consumes its energy, it will stop to perform the task at any time. In this case, to achieve good coordination and exploration is more challenging since it is required that the robot team has to respond quickly, robustly, reliably and adaptively to unexpected events.

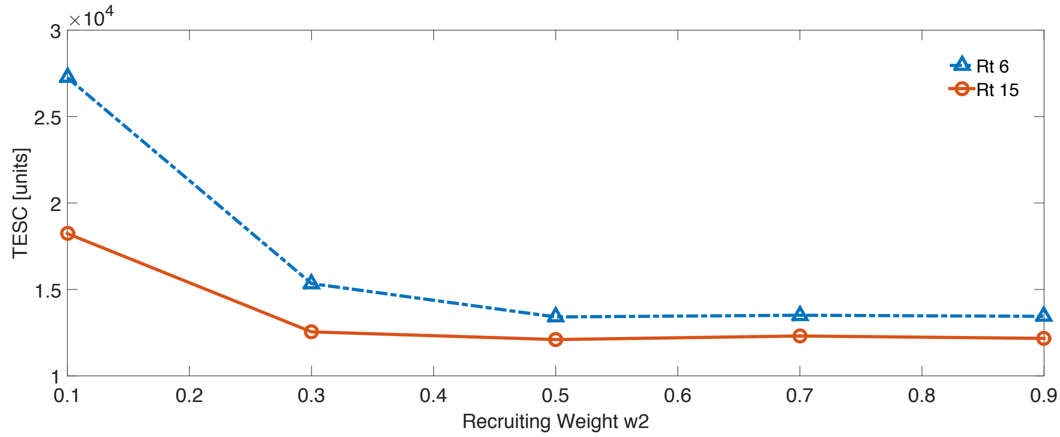
To measure the performance of such a robot team in practice, we consider a number of metrics applicable to the performance of the individual robots and the team as a whole. More specifically, we consider the percentage of unexplored cells, the number of disarmed targets and the percentage of alive robots.

Fig. 5.19 shows the impact of $w_1$ on the unexplored cells, varying both the dimension of the swarm and the number of dispersed targets, while keeping $R_{min}$ as a constant. It can be noticed that for a small robot team and a hight number of targets, the performance degrades as $w_1$ increases. This happens because the targets, for example mines, can explode at any time, causing not only the sudden stop of some robots in nearby regions, but also the damage of possible unexplored cells that become inaccessible. In these situations, the best value is about $w_1 \leq 0.5$, which allows to balance the two tasks.

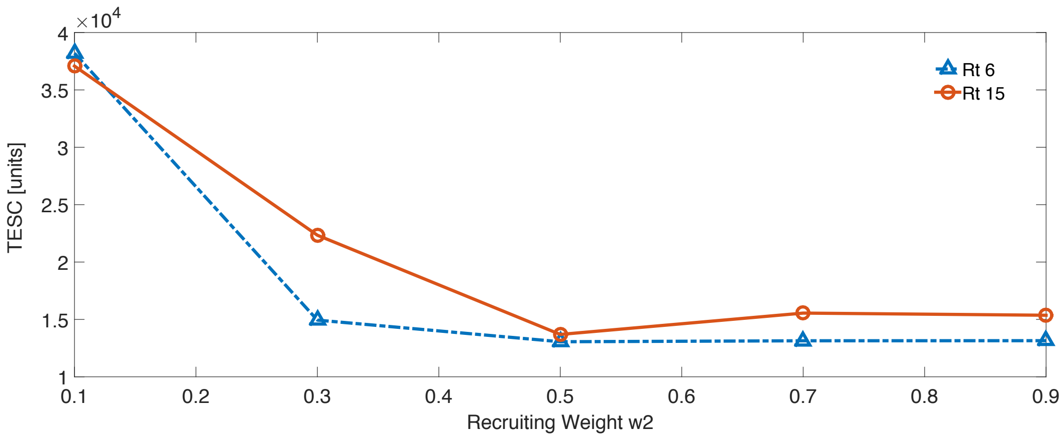
Regarding the disarmed targets, Fig. 5.20 highlights the impact of $w_2$ on the number of disarmed targets. It can be noticed that the value is particularly important for small robot teams (15 robots and 20 targets to be disarmed) and more importance can be attributed to the recruitment process ($w_2 \geq 0.3$). One possible explanation is that a robot team with higher movitation to be involved in the disarming process, can form a coalition more easily to handle a target so as to decrease the probability that it can explode.




(a)

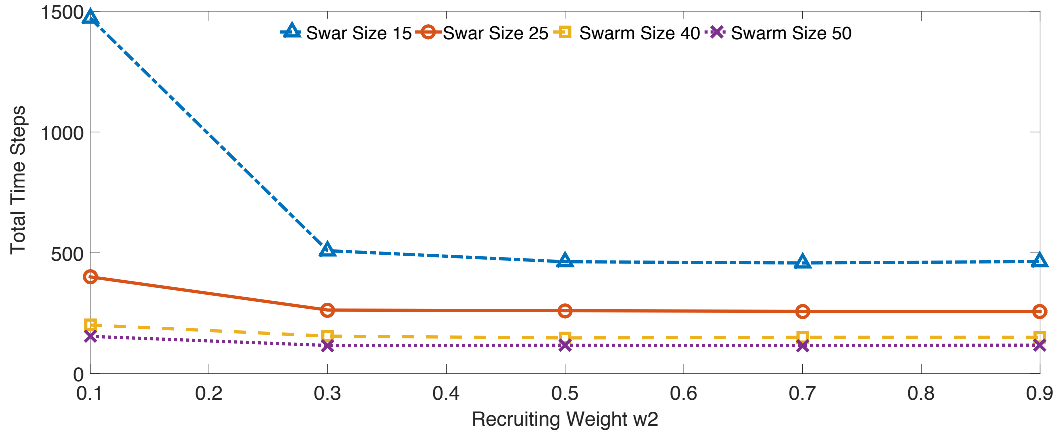



(b)

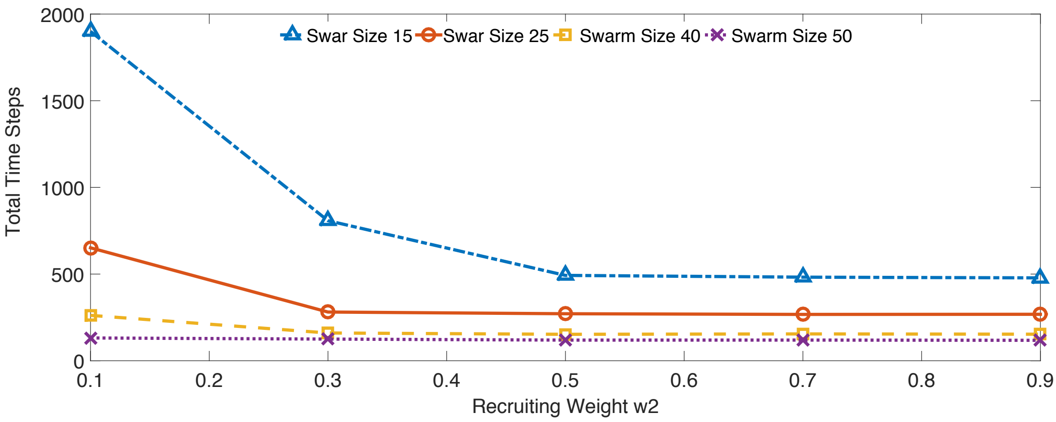



(c)

Fig. 5.15 Evaluation of the Total energy consumed by the system in 50x50 squares and 40 robots and 3 robots to treat a target. (a) 15 targets (b) 20 targets (c) 35 targets .




(a)




(b)

Fig. 5.16 Evaluation of the total time steps in a grid area 50x50. (a) 15 targets. (b) 20 targets.

The percentages of alive robots, evaluated considering different sizes of areas, swarm size and dispersed targets, are summarized in Fig. 5.21. The figure illustrates that for a small robots team, if a greater importance is given to the exploration task, some reduction of the alive robots is obtained. This behaviour seems to be influenced by the number of dispersed targets. However, increasing the swarm size leads to no significant differences. This can be justified by previous motivations; if $w_1$ is high, the robots may be less likely to respond to the help requests, thus leading to the coordinator robots be trapped into targets's locations waiting for others to arrive and consequently increasing the probability of some explosions. Therefore, unbalanced resources can cause severe resource wastage, lead to the potential failure of the robots due to both the energy limitation and potential explosion risks.

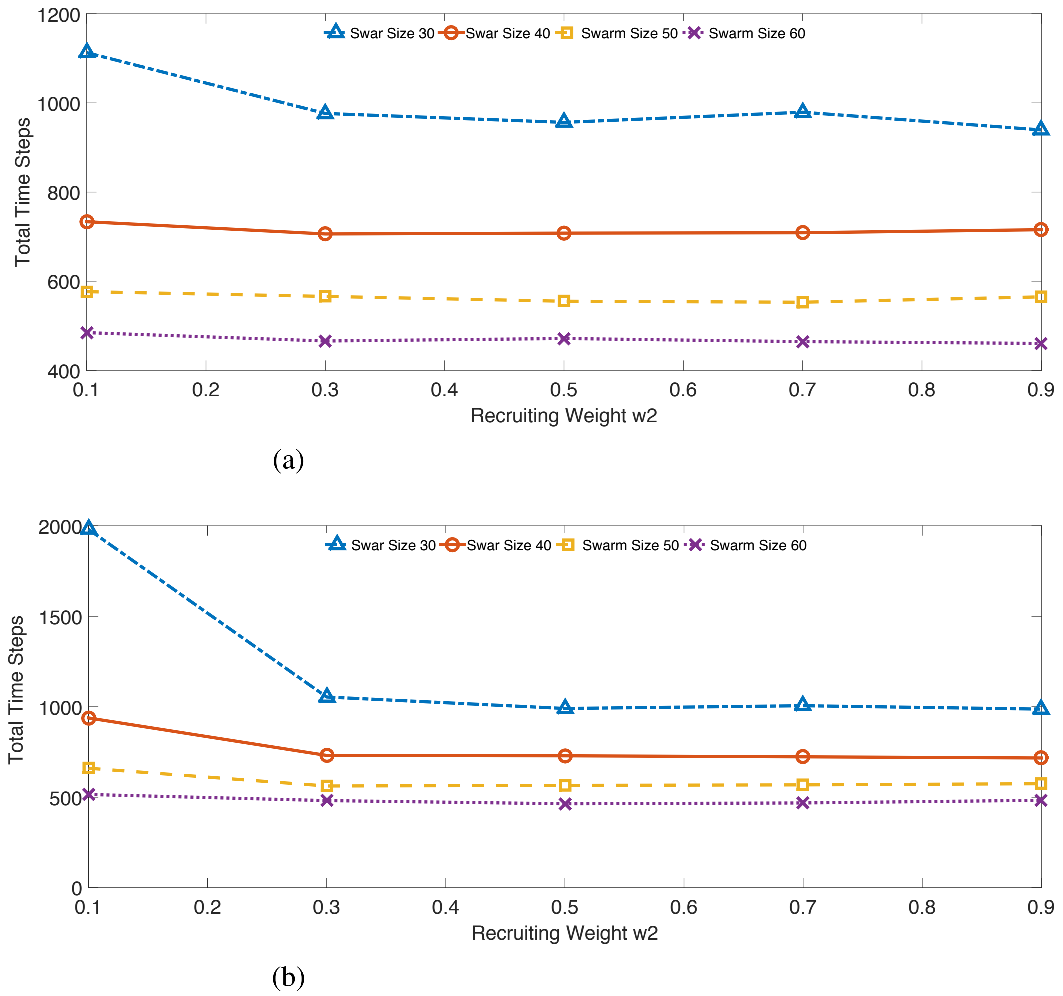



Fig. 5.17 Evaluation of the total time steps in a 100x100 grid area, varying the dimension of the swarm with (a) 20 targets and (b) 30 targets.

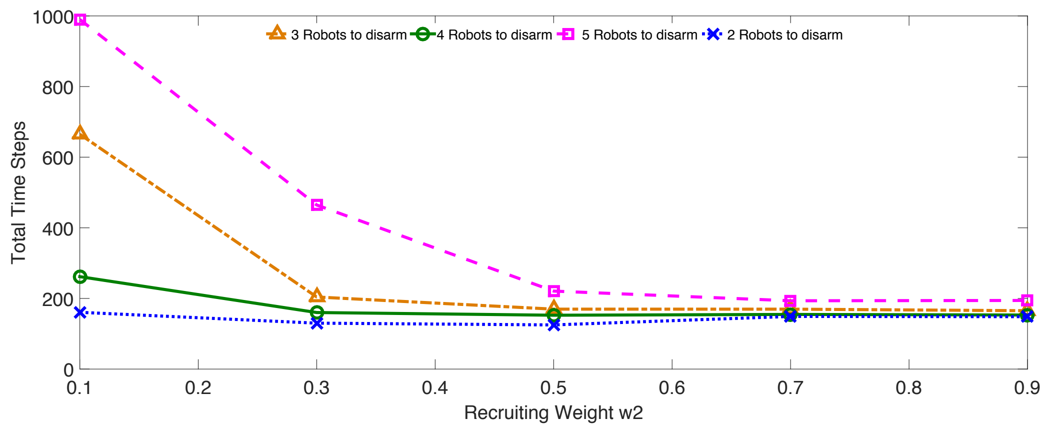
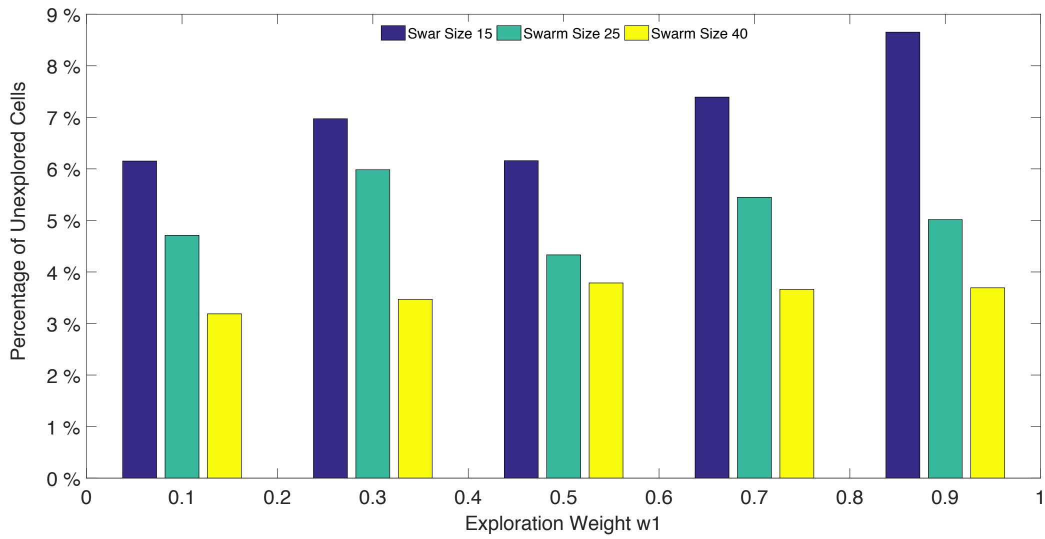
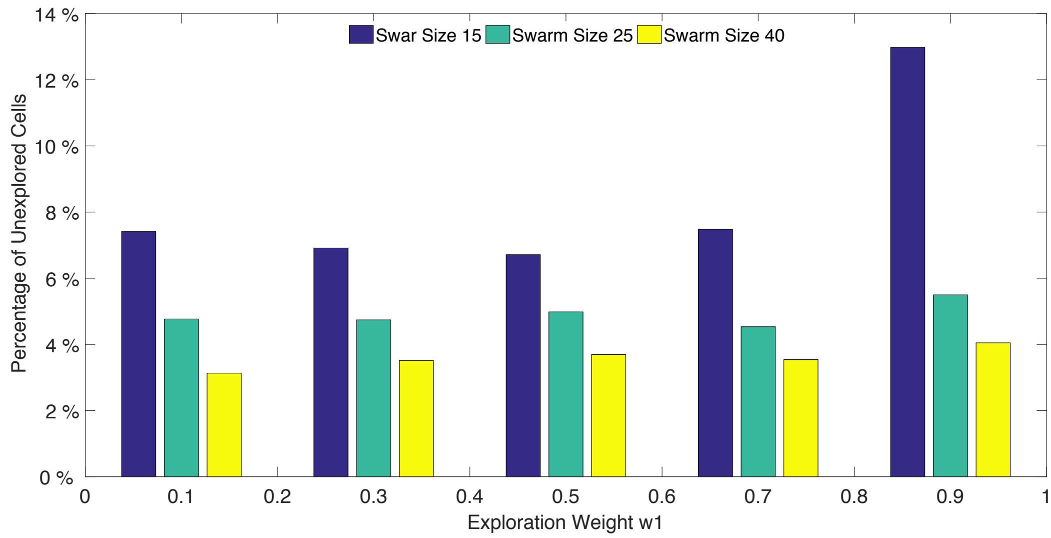



Fig. 5.18 Evaluation of the total time steps varying $R_{min}$ in an area with 50x50 squares, 40 robots and 20 targets.

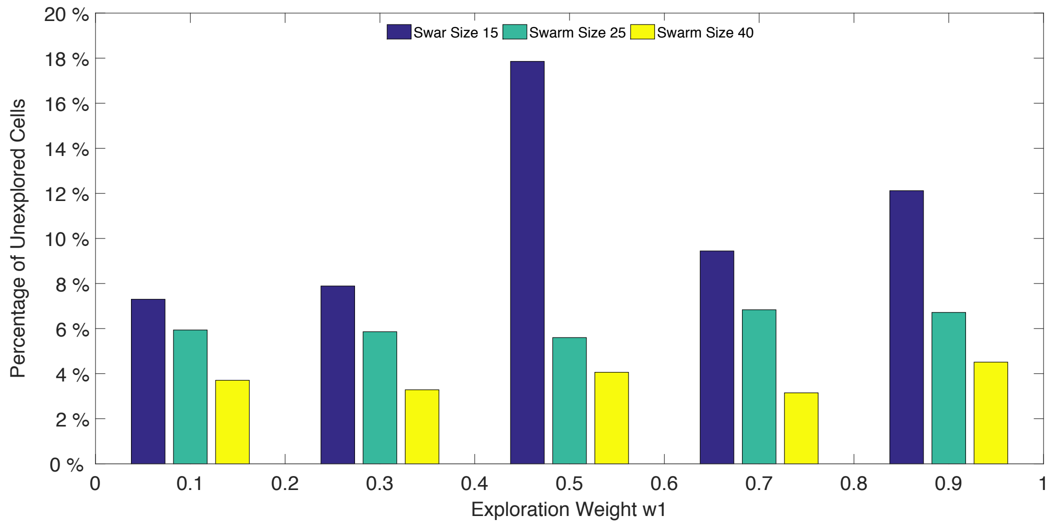
In almost all experiments, the performance fluctuates according to the number of disseminated targets. This may indicate that the solution, would be greatly influenced by the recruiting weight value.




(a)

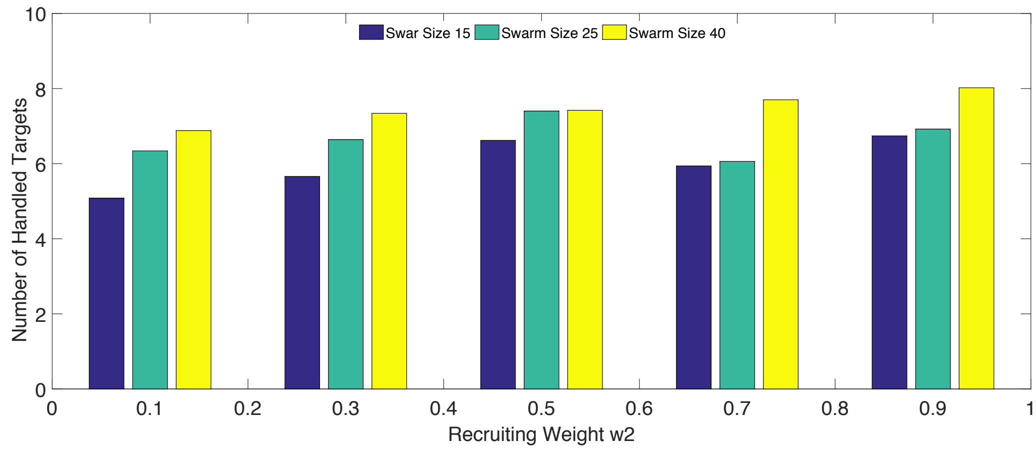



(b)

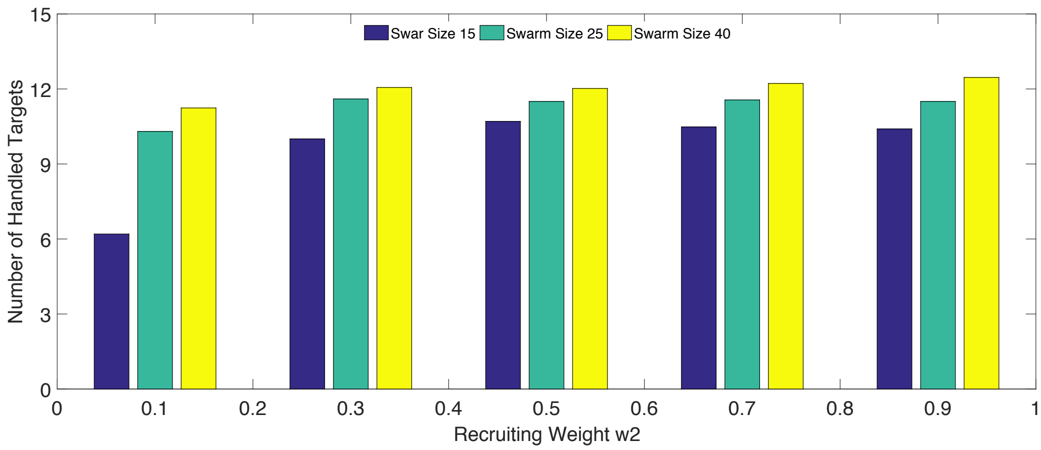



(c)

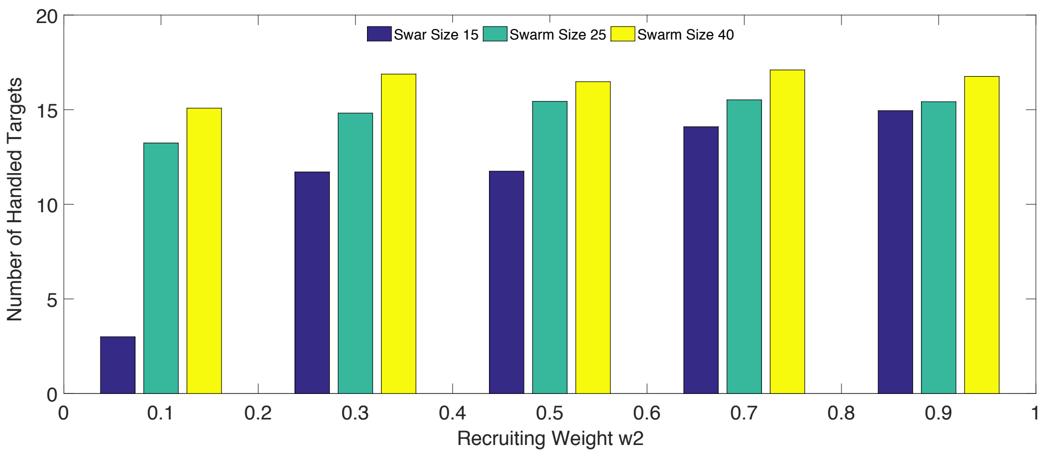
Fig. 5.19 Percentage of unexplored cells in a 50x50 grid area, varying the dimension of the swarm of robots and 3 robots needed to handle a target. (a) 10 targets (b) 15 targets (c) 20




(a)




(b)




(c)

Fig. 5.20 Percentage of disarmed targets in a 50x50 grid area, varying the dimension of the swarm of robots and 3 robots needed to handle a target. (a) 10 targets (b) 15 targets (c) 20 targets

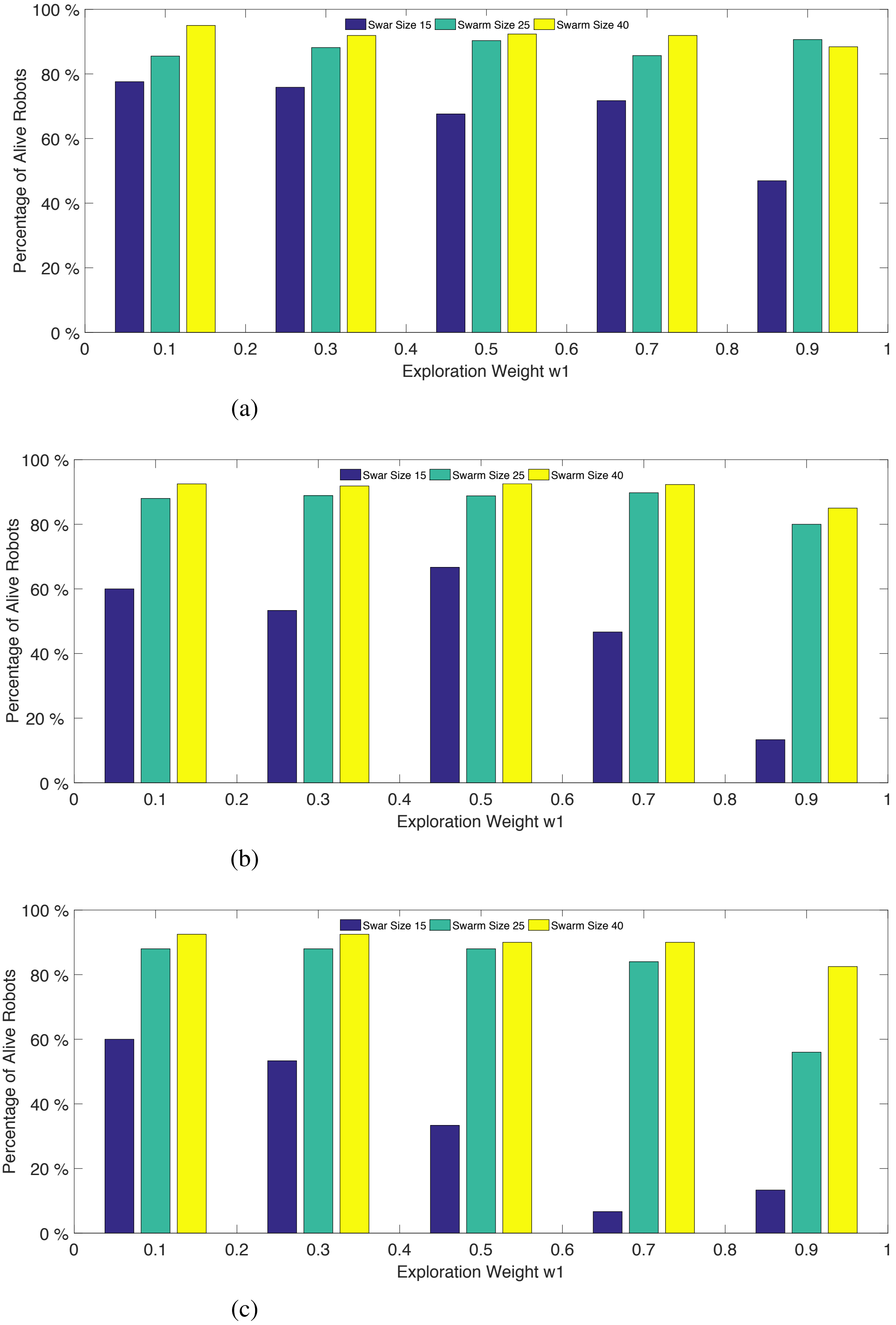



Fig. 5.21 Percentage of alive robots in a 50x50 grid area, varying the dimension of the swarm of robots and 3 robots needed to handle a target. (a) 10 targets (b) 15 targets (c) 20 targets

## 5.7 Summaries

This section describes three biologically inspired coordination strategies for robot swarm coordination under complex constraints. These techniques have been based on the firefly, particle swarm and artificial bee behaviour, and some discrete modifications have been carried out to make these algorithms suitable for the purpose. The most important features of the proposed metaheuristics are:

- flexibility: parameters can be easily tuned so that the proposed methodologies can used to carry out exploration and recruitment tasks for a system of mobile robots.
- scalability: the algorithms work well for any number of robots and targets.
- adaptability: the approaches can be used in the environment, allowing different conditions and distributions of targets and robots.
- parallelism: the algorithms are distributed and each robot performs its task in parallel and make decision individually, based on local partial information.

The experiments through simulation have showed that the performance in terms of time and energy consumed by the system is better for the Firefly Algorithm and Artificial Bee Algorithm especially when the task in not complex in terms of size of the swarm and number of disseminated targets. Increasing the complexity of tasks, considering an higher number of targets comparing to the swarm size, the firefly-based strategy usually gives better performance.

Moreover, in this sections, a bi-objective optimization problem for robot coordination and exploration tasks has been investigated. In this case it is applied as recruiting mechanism the firefly algorithm. Specifically, the problem has been modeled as a bi-objective model and the weighted sum method is used to find trade-off between the two tasks by varying different weighted values.

Different experimental scenarios have been considered to suitably evaluate the impact of the weight values on the critical parameters of the problem such as the dimension of the area, number of disseminated targets, number of robots to coordinate. The results have demonstrated that the choice of the right compromise between the two tasks is not straightforward. Generally speaking, the proper values depend on the application context. In most cases, the trade-off between the two objectives is highly correlated with the number of targets dispersed in the area, compared to the dimension of robot swarm. In these cases, more importance should be attributed to $w_2$ so that ($w_2 \geq 0.5$). However, in general case, balanced weights $w_1$ and $w_2$ (around 0.5) can offer a better trade-off.

## 5.8 Multi hop Communication

In this section an on demand mobile ad hoc network related to the problem to form coalitions in certain locations of the area is presented. The network architecture is created once a robot detects a target in the area and from this point that initiates communication with neighbor to neighbor. The idea is to use ad hoc routing protocol to report a detected target and the robots that wants to serve it over a MANET (De Rango and Palmieri, 2012), (De Rango and Palmieri, 2016).

Mobile ad-hoc networks (MANETs) consist of special kind of wireless mobile nodes which form a temporary network without using any infrastructure or centralized administration. In networks, all nodes are mobile and communicate with each other via wireless connections. Nodes can join or leave the network at any time. There is no fixed infrastructure. All nodes are equal and there is no centralized control or overview. There are no designated routers: all nodes can serve as routers for each other, and data packets are forwarded from node to node in a multi-hop fashion (Ducatelle et al., 2006).
Since in mobile ad-hoc networks there is no infrastructure support and nodes being out of range of a source node transmitting packets; a routing procedure is always needed to find a path so as to forward the packets appropriately between the source and the destination.
Moreover, due to limited resources such as power, bandwidth, processing capability, and storage space at the nodes as well as mobility, it is important to reduce routing overheads in MANETs, while ensuring a high rate of packet delivery. MANETs can be used in wide range of applications as they have the capability to establish networks at anytime, anywhere without aid of any established infrastructure. It is a challenging task to find most efficient routing due to the changing topology and the dynamic behavior of the nodes in MANET.

Due to the dynamic nature of MANETs, route maintenance is quite difficult task. Basically, routing is the process of choosing paths in a network along which the source can send data packets towards destination. Routing is an important aspect of network communication because the characteristics like throughput, reliability and congestion depend on the routing information. An ideal routing algorithm is one which is able to deliver the packet to its destination with minimum amount of delay and network overhead. The nodes update the routing tables by exchanging routing information between the other nodes in the network.

In literature exists a large family of ad hoc routing protocols. However, it has been found that bio- inspired approaches such as ant colony optimization (ACO) algorithms can give better results as they have characterization of Swarm Intelligence (SI) which is highly suitable for finding the adaptive routing for such type of volatile network.
ACO routing algorithms use simple agents called artificial ants which establish optimum

paths between source and destination that communicate indirectly with each other by means of stigmergy (Singh et al., 2012).

## 5.9 ACO routing algorithms

Routing in MANET is therefore important to design algorithms that are adaptive, robust and self-healing. Moreover, they should work in a localized way, due to the lack of central control or infrastructure in the network. Nature's self-organizing systems like insect societies show precisely these desirable properties. Making use of a number of relatively simple biological agents (e.g., ants) a variety of different organized behaviors are generated at the system-level from the local interactions among the agents and with the environment.

The basic idea behind ACO algorithms for routing is the acquisition of routing information through sampling of paths using small control packets, which are called ants. The ants are generated concurrently and independently by the nodes, with the task to test a path to an assigned destination. An ant, going from source node $s$ to destination node $d$, collects information about the quality of the path (e.g. end-to-end delay, number of hops, etc.), and uses this on its way back from $d$ to $s$ to update the routing information at the intermediate nodes (Di Caro et al., 2005).

The routing tables contain for each destination a vector of real-valued entries, one for each known neighbor node. These entries are a measure of the goodness of going over that neighbor on the way to the destination. They are termed pheromone variables, and are continually updated according to path quality values calculated by the ants. The repeated and concurrent generation of path-sampling ants results in the availability at each node of a bundle of paths, each with an estimated measure of quality. In turn, the ants use the routing tables to define which path to their destination they sample: at each node they stochastically choose a next hop, giving higher probability to links with higher pheromone values. For this reason, generally, the routing tables are also called pheromone tables. A routing table at each node is organized on a perdestination basis and is of the form (*Destination, Next hop, Probability*). It contains the goodness values for a particular neighbor to be selected as the next hop for a particular destination. Further, each node also maintains a table of statistics for each destination $d$ to which a forward ant has been previously sent. The routing tables then contain, basically, the probability (goodness value) of taking as next hop node $h$ at a node $p$ (Baras and Mehta, 2003).

In ACO routing algorithms, multiple ants, created by a node, traverse the network to search paths between two nodes. If the ant finds a path, it lays down pheromone on the path. The amount of pheromone depends on the quality of the path such as its number of hops, delay, and energy of nodes on the path. A data packet is transmitted on a link with probability based on the amount of pheromone. ACO routing algorithms exhibit interesting properties for MANETs, as they work in a fully distributed way and provide multi-path routing.

Generally, an Ant-based algorithm, applied to the routing, consists of three phases as expressed following:

- *Route Discovery phase* - In this phase, new paths are discovered. The creation of new routes requires the use of *Forward Ant* (called FANT) , which establishes the pheromone track to the source node and *Backward Ant* (called BANT), which establishes the track to the destination node. FANTs are broadcasted by the sender to all its neighbors. Each FANT has a unique sequence number to avoid duplicates. A node receiving a FANT for the first time creates a record (destination address, next hop, pheromone value) in its routing table. The node interprets the source address of the FANT as destination address, the address of the previous node as next hop, and computes the pheromone value depending on the number of hops the FANT needed to reach the node. Then the node relays the FANT to its neighbors. When the FANT reaches the destination, it is processed in a special way. The destination node extracts the information and then destroys the FANT. A Backward Ant (BANT) is created and sent towards the source node. In that way, the path is established and data packets can be sent.

- *Route Maintenance* - Once the FANTs and BANTs have established the pheromone tracks for the source and the destination nodes, subsequent data packets also increase the pheromone value. Data packets are used to maintain the path, so no overhead is introduced. Pheromone values keep on changing. When a node relays a data packet toward destination to a neighbor node, it increases the pheromone value for that entry. The same happens in the opposite direction. The evaporation process is simulated by regular decreasing of the pheromone values.

- *Route Failure* - In Route Failure phase, a node deactivates the path by reducing pheromone value to 0 in corresponding route table entry and goes to the Route Discovery phase for selecting path and sending packets to the destination over that path.

### 5.9.1 Data structures of ants

Broadly speaking, a FANT is broadcasted by the sender and will be relayed by the neighbors of the sender. A node receiving a FANT for the first time, creates a record in its routing table. A record in the routing table is a triple and consists of (destination address, next hop, pheromone value). The node interprets the source address of the FANT as destination address, the address of the previous node as the next hop, and computes the pheromone value depending on the number of hops the FANT needed to reach the node. Then the node relays the FANT to its neighbors. Duplicate FANTs are identified through the unique sequence number and destroyed by the nodes. FANT reaches the destination node, it is processed in a special way. The destination node extracts the information of the FANT and destroys it. Subsequently, it creates a BANT and sends it to the source node. The BANT has the same task as the FANT, i.e. establishing a track to this node. When the sender receives the BANT from the destination node, the path is established and data packets can be sent. Each node periodically sends FANTs to randomly choose destination nodes throughout the network. Basically, each Forward Ant packet contains the following fields:

- Source node IP address
- Destination node IP address
- Next hop IP address
- Stack Hop count

When a source node needs some information or content from an existing MANET, it first checks the cache for existing routes, when no routes are known, it broadcasts Forward request Ants with content tag and it is propagated through the network till it reaches maximum hop count. The forward ant carries the content to be searched, when a relevant content is found then Forward Ant is converted in to Backward Ant, at the same time the Forward Ant continues its travel for more relevant contents till it reaches maximum hop count. A Forward Ant at each intermediate node selects next hop using the information stored in the routing table of that node or by rebroadcast.
Then, the Backward Ant updates pheromone value as it moves on its way to source node. The content relevancy and availability ratio decides pheromone value, more relevant content increases pheromone value.

The routing tables contain for each destination a vector of real-valued entries, one for each known neighbor node. These entries are a measure of the goodness of going over that neighbor on the way to the destination. They are termed pheromone variables, and are continually updated according to path quality values calculated by the ants

The repeated and concurrent generation of path-sampling ants results in the availability at each node of a bundle of paths, each with an estimated measure of quality. In turn, the ants use the routing tables to define which path to their destination they sample: at each node they stochastically choose a next hop, giving higher probability to those links which are associated with higher pheromone values. The pheromone information is used for routing data packets, more or less in the same way as for routing ants: packets are routed stochastically, giving higher probability to links with higher pheromone values.

Furthermore, the stack of the Forward Ant is a dynamically growing data structure that contains the IP addresses of the nodes that the forward ant has traversed as well as the time at which the Forward Ant reached these nodes. The Backward Ant inherits the stack contained in the Forward Ant. The main purpose of the Backward Ants is to propagate information regarding the state of the network gathered by the forward ants. The backward ant retraces the path of the forward ant by popping the stack, making modifications in the routing tables and statistic tables at each intermediate node according to a function of some metric or a combination of metrics, e.g. delay or the number of hops.

## 5.10 Ant-Based Task Robot Coordination Protocol (ATRC)

In the following the proposed Ant-based Task Robot Coordination Protocol (ATRC) (De Rango and Palmieri, 2012), (De Rango and Palmieri 2016) , for coordinating the robots in the recruiting task is described and analyzed, trying inspiration from previous works (Di Caro and Dorigo, 1998), (Baras and Mehta, 2003), (Bouazizi, 2002).

### 5.10.1 ATRC Communication Structure

The network of robots is created when one or more robots find a target. More specifically, the robot that has detected a target sends announcement messages that are forwarded by the other robots so that the information about the target can spread among the swarm.
The messages that a robot can send or receive are:

1. **HELLO**: Hello packets are used to notify the robot presence in its transmission range to other robots. A HELLO packet contains the ID of the sending robot. When a robot receives this packet becomes aware of the presence of another robot in its range and it writes the ID in a data structure (neighbors table) which takes into account all the robots in the direct communication range. If, after a time period, it does not receive HELLO packets from other robots present in its neighbor table, it deletes the correspondent entry line. In this way, a robot will know the robots that can be reached directly (one-hop).

2. *Requiring Task Forward Ant* (**RT-FANT**): it is a packet sent by the robot that has detected a target (that is the coordinator robot) to know how many robots are available to treat the target.

3. *Requiring Task Backward Ant* (**RT-BANT**): it is a packet that a generic robot (called forager) sends as response to a RT-FANT.

4. *Recruitment Fant* (**R-FANT**): it is a packet sent by a coordinator, to the link from which came the higher number of RT-BANT responses; this link has higher recruitment probability.

5. *Recruitment Bant* (**R-BANT**): it is a packet sent by a robot in response to a positive recruitment by a coordinator.

6. Leaving position (**LP**): if a R-BANT, generated by a robot in response to the R-FANT, does not arrive to coordinator within a certain time (it is a timer), and in target's location have arrived the needed robots, the coordinator sends this message informing this robot to continue to explore the area or serve other requests.

In the following the actions, in terms of received packets are described, in order to deeply understand the functioning of the protocol and the different of packets that are sent during the mission.

For the most time, a robot is in *Forager* executing the exploration task. Its operations are essentially the following:

I. Process packet content: when a robot receives a packet it forwards the packet to another destination.

II. Exploration phase according to ATS-RE algorithm.

A coordinator robot performs these operations:

- FANT Generating and Forwarding: it creates and sends broadcast requests in the network; in this step the coordinator sends a RT-FANT to know how many robots are, eventually, available for disarming the found target. The RT-FANT, identified by the triple (ID-Coordinator, Task-ID, ID-FANT), is sent in broadcast to all robots in the transmission range.
- Set waiting timer: after sending the RT-FANT, the coordinator sets a timer to wait the RT-BANT packets sent by robots available to be recruited; after timing out it checks the number of received RT-BANT. If the coordinator does not receive enough replies, analyses the number of received replies: if it does not receive any replies it becomes a Forager robot, else it creates and sends a new Request Task FANT and forwards in broadcast on the network. If the coordinator has enough replies (RT-BANT) to perform the task, it creates and sends R-FANT on the link with higher recruitment probability.
- Wait incoming robots: the coordinator waits for the incoming recruited robots.
- Submit disarming order: When all needed robots are recruited into the interested cell, the coordinator sends a message to announce the starting of the manipulation task of the target.

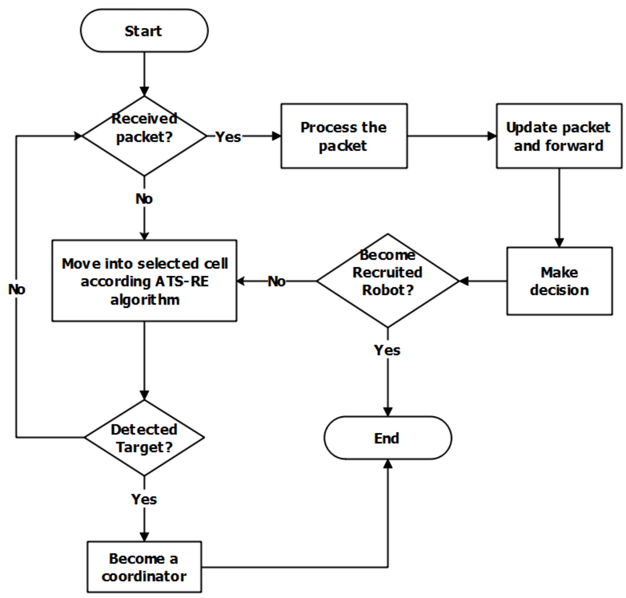



Fig. 5.22 The Flow Chart of a Forager Robot

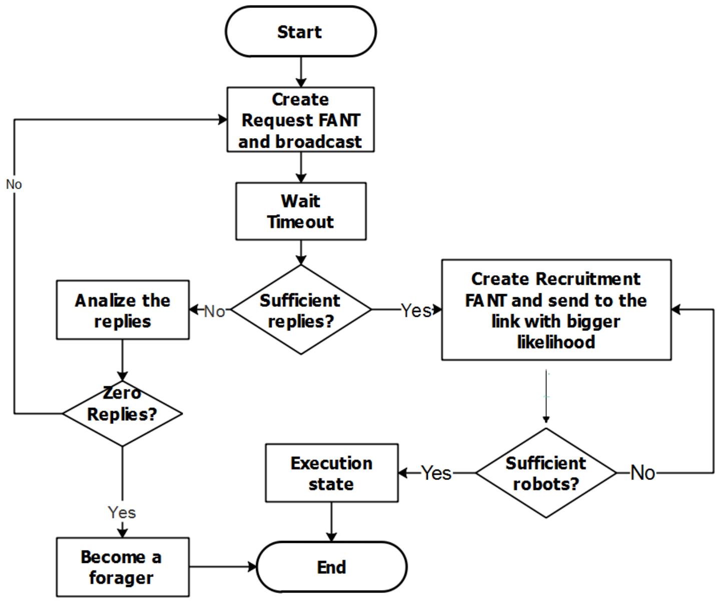



Fig. 5.23 The flow Chart of a Coordinator Robot

When a robot receives a RT-FANT packet and sends a RT-BANT to the coordinator, it becomes a *Recruited Robot*. Then, its task is to reach the destination cell. Essentially, the recruited robot moves into the area in order to reach the target's location.

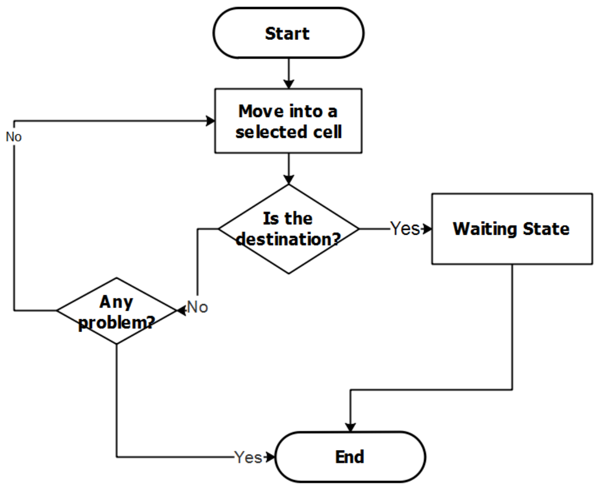



Fig. 5.24 The Flow Chart of a Recruited Robot

## 5.10.2 Forwarding mechanism of FANT and BANT

In the considered problem an Ant-based Team Robot Coordination (ATRC) protocol has been applied and it uses typically probabilistic routing tables to establish to which robots distribute

the coordination tasks. This routing table is populated and updated on the basis of the packets sent from coordinators to recruited robots (Forward ANT: R-FANT and RT-FANT) and vice versa (Backward ANT: R-BANT and RT-BANT). To ensure that for every FANT sent on the path from the coordinator to the recruited sent back a BANT on the reverse-path forwarding to the coordinator, each node crossed by the FANT enters its ID in the packet. Once it reaches its destination a Backward ANT (BANT) response is created; in this packet the ID of crossed robots and additional information for updating the routing tables are copied. BANT follows the route tracked by FANT so it reaches the destination host (coordinator). For this behavior the two considered packets are called Forward Ant (FANT) and Backward Ant (BANT).
During this discovery procedure BANT updates the entry in the routing table of the node. The law for updating the pheromone is usually based on the path length, that is the number of hops ( in terms of robots) crossed by FANT to reach the destination. The routing table in this work are not deterministic, but probabilistic. Essentially a packet has the following fields:

- ID Coordinator: ID of the coordinator robot and it is inserted in a RT-FANT;
- Task ID: it is the ID of the task requested by the coordinator.
- Task Type: in this case there are three tasks (recruiting, disarming and discovery), but this field can be useful for future purpose and extensions to multiple and more complicated tasks.
- Path Degree $p_D$: it is a weight given to a path in order to understand which route can be the best according with some specific metrics; it can affect the link selection probability for each link between the current robot and its neighbors.

ID Coordinator, Task ID and Task Type allows the unique identification of an entry. Initially, when a RT-FANT is sent on the network, each robot receives RT-FANT and creates an entry in the routing table and sets a balanced selection probability of the neighbors. These probabilities are then updated through the response RT-BANT. Each robot that receives an RT-BANT from a particular link, updates the probability associated to that link and decreases the other link probabilities through the use of two concepts:

1. Evaporation
2. Reinforcement.

The evaporation is applied to all links, while reinforcement learning is applied to the link receiving the RT-BANT. The quality of a link depends on the distance of the robot that

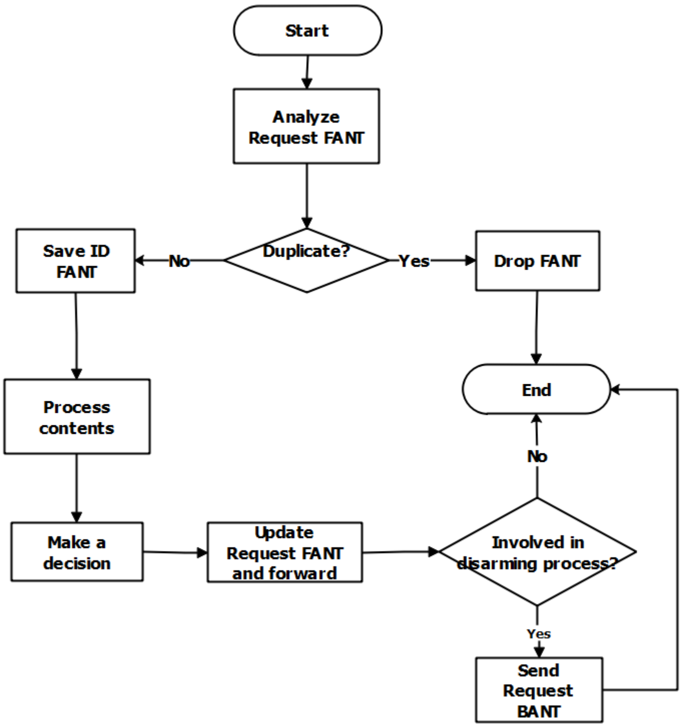



Fig. 5.25 The Flow Chart of Request FANT and BANT

creates the RT-BANT to the destination (cell where the target needs to be deactivated). In this way the probability of the link that receives the highest number of RT-BANT increases. Having to submit the R-FANT in a deterministic way, a robot is able to choose the link with the highest recruitment probability. Also, the received R-BANT contains a recruitment task during the travelling for each link, the robot only executes the process of evaporation. This is made to improve the link selection probability, indicating a high number of robots willing to perform the task requested.

### 5.10.3 Task Requesting BANT and FANT Management

When the coordinator sends RT-FANT, only foragers process this packet. If the packet is received by robots that are in other state they forward in broadcast the RT-FANT. The forager receiving RT-FANT performs the same operations below:

- Checking uniqueness of received FANT: a forager, after receiving a packet containing RT-FANT, controls if it processed this packet previously. In this case the robot drops the packets and carries on its operations, otherwise it saves the ID FANT in a data structure and processes the packet content.

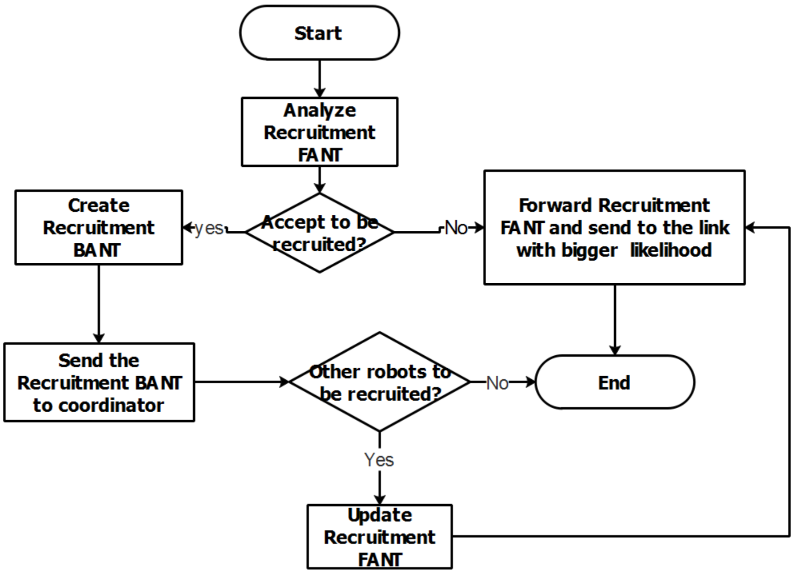



Fig. 5.26 The Flow Chart of a Recruitment FANT and BANT

- Process requirements: If the received RT-FANT is not duplicated, the forager checks the required characteristics. If it is able to perform the task, it controls the percentage of BANTs already forwarded to the coordinator, according with previously forwarded FANTs, and decides, in a probabilistic manner, whether to forward or not its answer. Next it creates and sends an RT-BANT to the coordinator. The forager, finally, sends the received RT-FANT in broadcast also if it is not able to perform the task.

## 5.10.4 Recruitment FANT and BANT Management

A coordinator, after receiving enough responses by foragers, sends R-FANT on the link that has the highest success probability. The foragers receiving this FANT execute these operations:

- Processing R-FANT: Initially, the forager checks whether the FANT has been previously processed; in this case it discards the packet. In other case it adds its identifier in the list of crossed robots by R-FANT and then processes the recruitment request.

- BANT Management: if the robot decides to participate in the disarmament of the target, it creates and sends a R-BANT to coordinator as a recruitment confirmation. The R-BANT updates the routing table of the crossed nodes.

- FANT Forwarding: independently by the response of R-BANT, a forager receiving a R-FANT creates and sends new R-FANT to other robots if there is the need to recruit other robots on the link with higher recruitment probability otherwise, if itself is the last robot, it does not forward any R-FANT.

## 5.11 Simulations

The simulations were executed to validate the proposed protocol and by varying different parameters of the problem in order to verify its robustness, convergence and scalability for increasing complexity.
Performance metrics considered for the simulation are:

- Total Time steps to complete the mission.
- Average energy consumed by a robot.
- Control Overhead: it accounts the number of packets such as R-FANT, R-BANT, RT-FANT, RT-BANT sent on the network.

The experiment were carried out considering 3 robots to disarm a target and by changing many parameters of the problem in order to understand how many variables, such as dimension of the area in terms of cells, number of disseminated targets and wireless range ($R_t = 3, 6, 9$) in terms of unit of cells, can influence the results.

The first set of experiments takes into account the influence of the wireless range used for transmitting the protocol in order to understand how the wireless communication influences the time needed to complete the mission. Fig.5.27 and Fig. 5.28 show the performance of the proposed protocol considering different grid area and varying the size of the team. It can be seen that the curves decrease as the communication range increase. The reason is that over long communication range, the coordinator robots can recruit better the others and they are able to reach the target's location in shorter time. Some interesting features are observed from the figures: a team with small number of robots is mainly affected by the positive side of an hight communication range. On the other hand, increasing the number of robots in the area, the results are comparable and no significant difference among the curves.

Regarding the traffic on the network in terms of sent packets, the results confirm that using a small wireless range ($R_t$ = 3) means that only very local robots can receive packets and thus to inform the overall team about the found targets it is necessary generate more packets. The difference is greater considering a small robots team and in more complex scenario that is a big area with more targets and with obstacles.
However, if the number of robots increases, the number of generated packets can be reduced

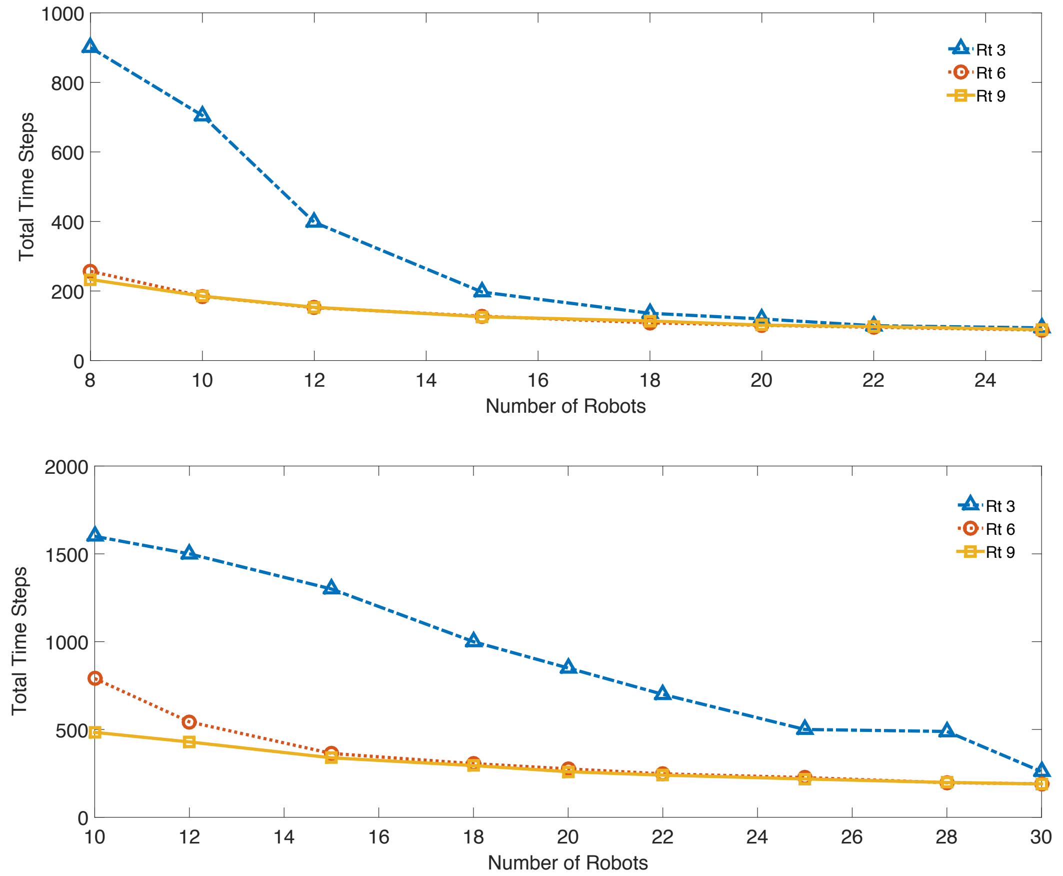



Fig. 5.27 Influence of the $R_t$ on the total time steps in an area without obstacles (a) 30x30 grid area and 3 targets (b) 50x50 grid area and 5 targets.

and, after a certain amount, having more robots do not introduce more any benefits in completing the mission.

Now it is studied how the number of targets can influence of the performance in terms of time and sent packets. Fig. 5.31 highlights, as expected, that more complex is the mission and more resources are consumed. However, the system obtains benefit in terms of time introducing more robots in the area that area able to distribute among multiple sites and targets. Regarding the number of packets it is shown that it mainly depends on the number of targets in the area. The number of robots does not affect greatly the overhead, since the proposed algorithm, such as designed, avoids an excessive increase of packets forwarding in the network. The number of packets in the network is nearly constant increasing the number of robots with a certain number of targets; instead increasing the number of targets with a certain number of robots the number of packets increases. This is due to the scalable approach of ATRC that adopts just local information to know where to send packets (highest link selection probability) and global information through the stigmergy avoiding to increase

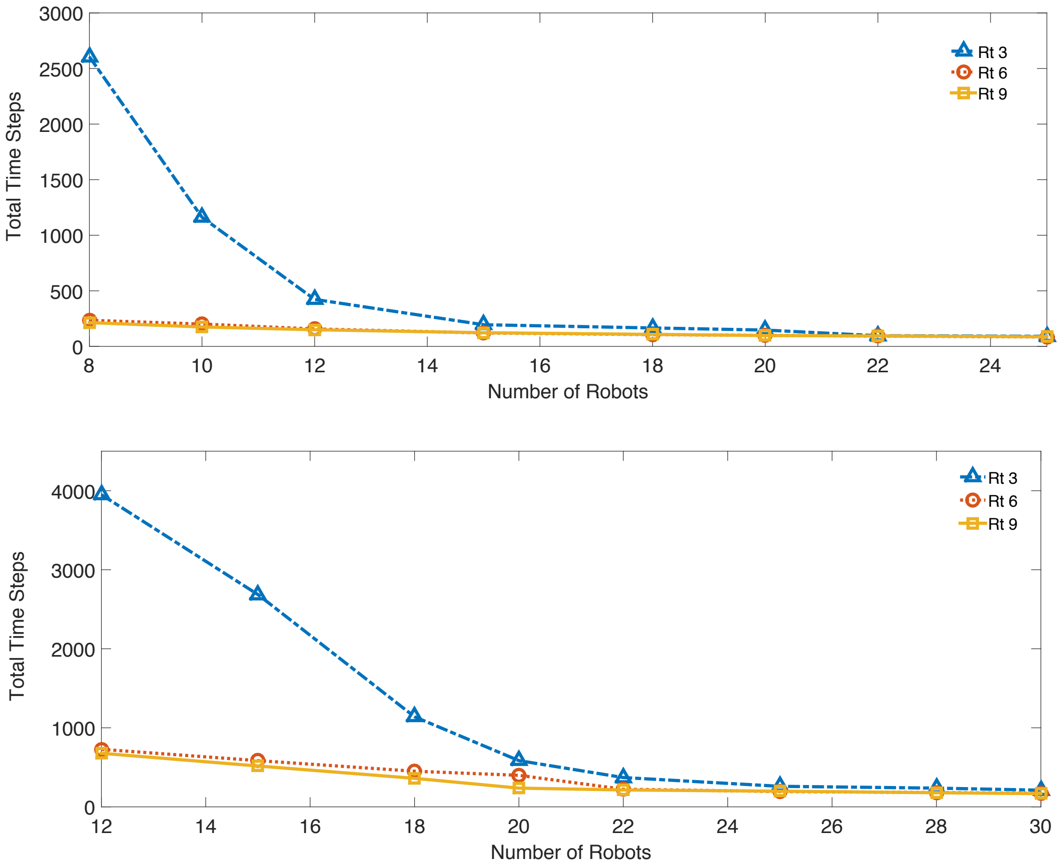



Fig. 5.28 Influence of the $R_t$ on the total time steps in an area with obstacles (a) 30x30 grid area with 3 targets (b) 50x50 grid area with 5 targets.

the control overhead to maintain the robot topology and distribute tasks. Increasing the size of swarm, the results are comparable because the higher number of robots assures a natural distribution among exploring and disarming tasks leading to a reduced overall execution time. Moreover, the number of packets decreases by increasing the number of robots until a certain point ( 20 robots) beyond which the number of packets seems to be equal. Increasing the number of robots means more forwarded packets that cause more traffic on the network for coordinate the team. Set the optimal number of robots is hard since it depends on several factors mainly on the dimension of the area and the number of disseminated targets. More complex is the scenario and more high is the number of robots that could grantee a reasonable results both in terms of time and both in terms of packets. Indeed, especially in hazardous and dynamic environment where the communication is not completely reliable, reducing the number of packets is crucial.

Regarding the influence of the dimension of the area, Fig.5.32 depicts the main obtained results. It can be observed that the size is an important parameters of the problem since

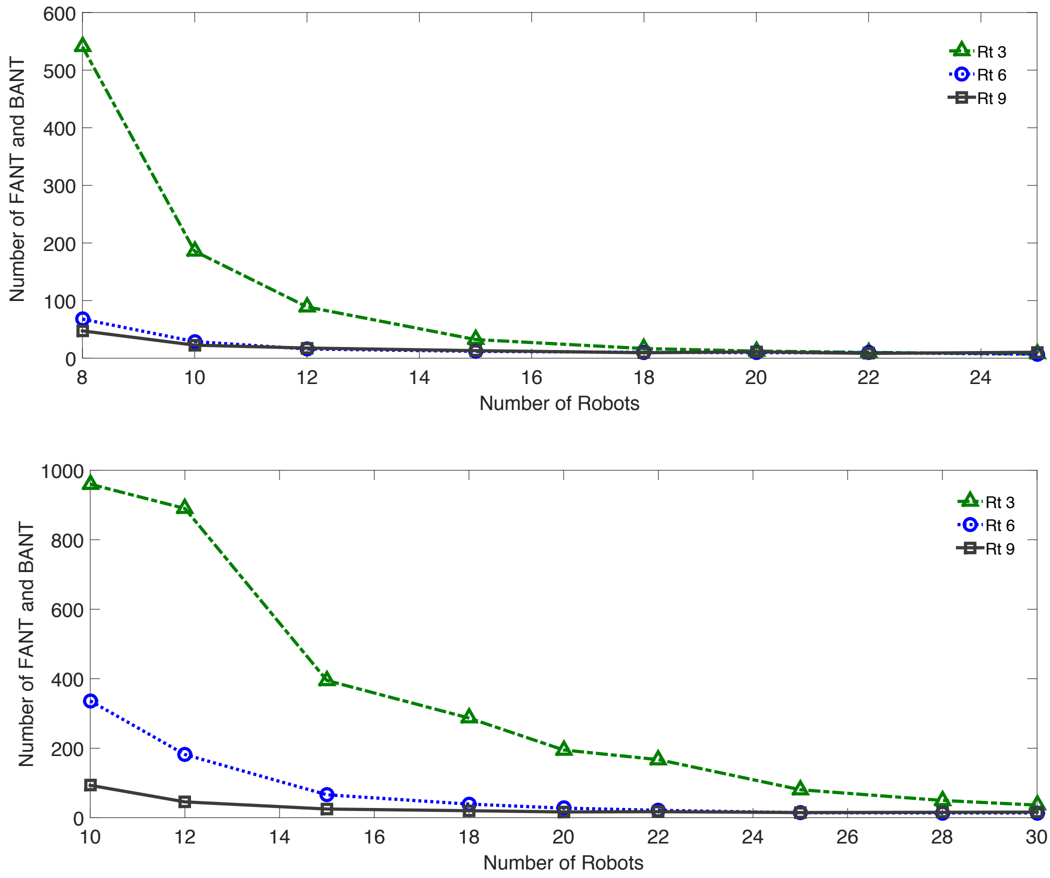



Fig. 5.29 Influence of the $R_t$ on the sent packets in an area without obstacles (a) 30x30 grid area and 3 targets (b) 50x50 grid area and 5 targets.

increasing the area in terms of cells, means more resources to be utilized. Indeed the time is affected by the size of the area; more cells needed to be explored and more time is required to complete the mission. Regarding the number of packets increase proportionally to the size of area when there are few robots since the network is instable and all tracks cannot be completed and robots are not immediately released to complete the exploration. However, the network reaches the stability increasing the number of robots and with the possibility to distribute both tasks (recruiting and exploration) in the overall area.

The last set of experiments are done to evaluate the energy consumed by a robot when a few or many targets exist, varying the terrain size and the number of involved robots. As previous results, more complex is the mission and more energy is consumed to finish the mission (Fig. 5.33).

However, a team with a larger number of robots generally increase the performance, saving the consumed energy. Obviously, the more targets are introduced, the more energy is

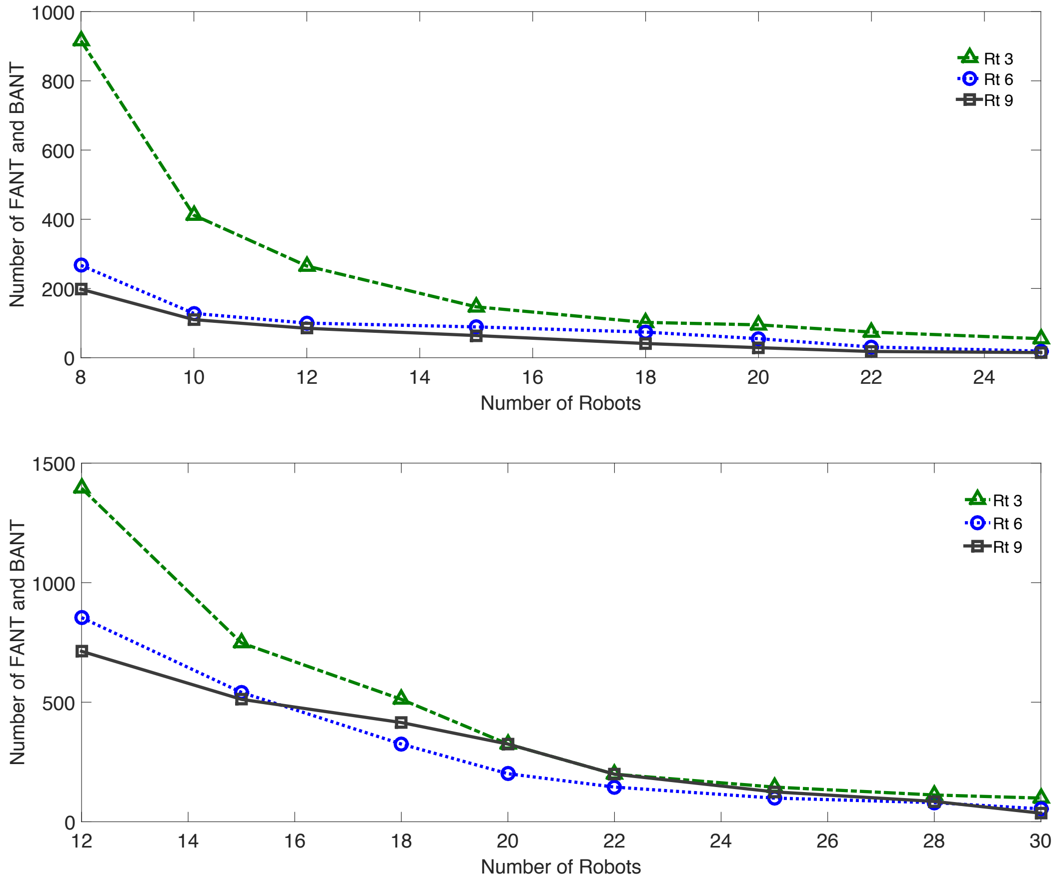
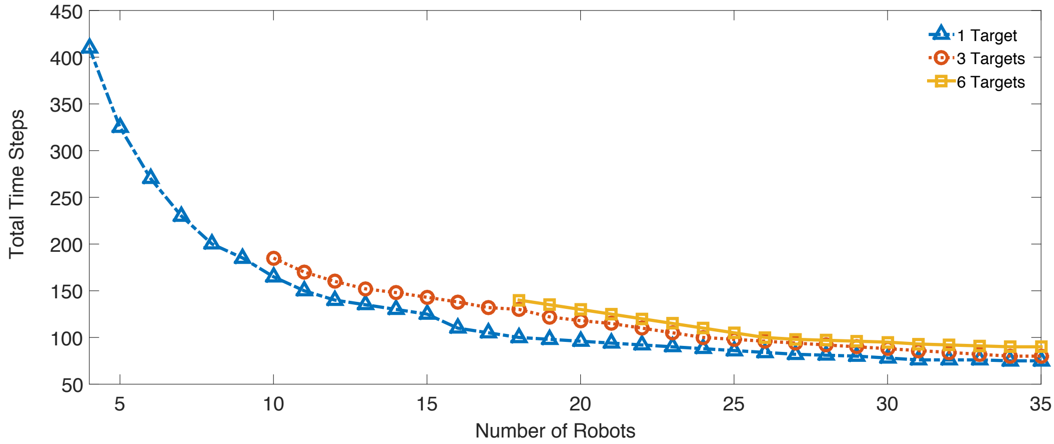



Fig. 5.30 Influence of the $R_t$ on the sent packets in an area with obstacles (a) 30x30 grid area and 3 targets (b) 50x50 grid area and 5 targets.

consumed. Nevertheless, increasing the number of targets, the recruiting tasks becomes more complex and thus require more communication and movements.



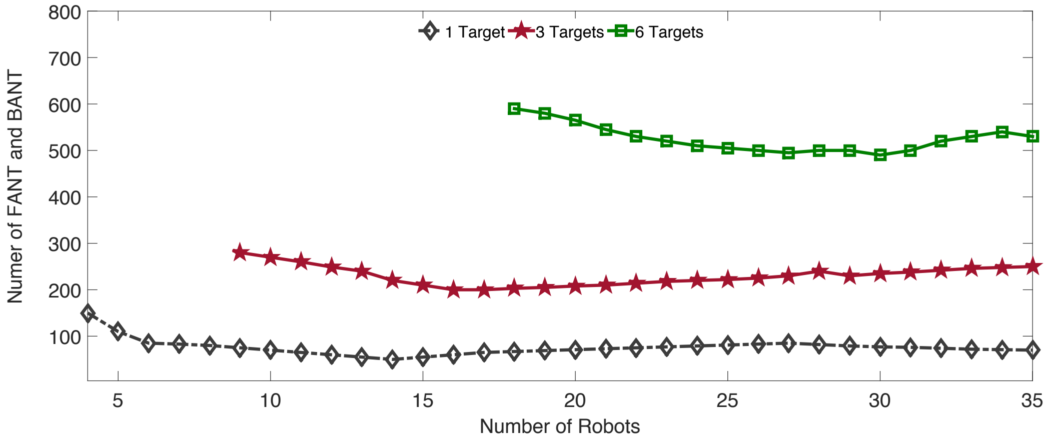




Fig. 5.31 Evaluation of the performance of ATRC in 30x30 grid area (a) Total Time Steps (b) Number of sent packets

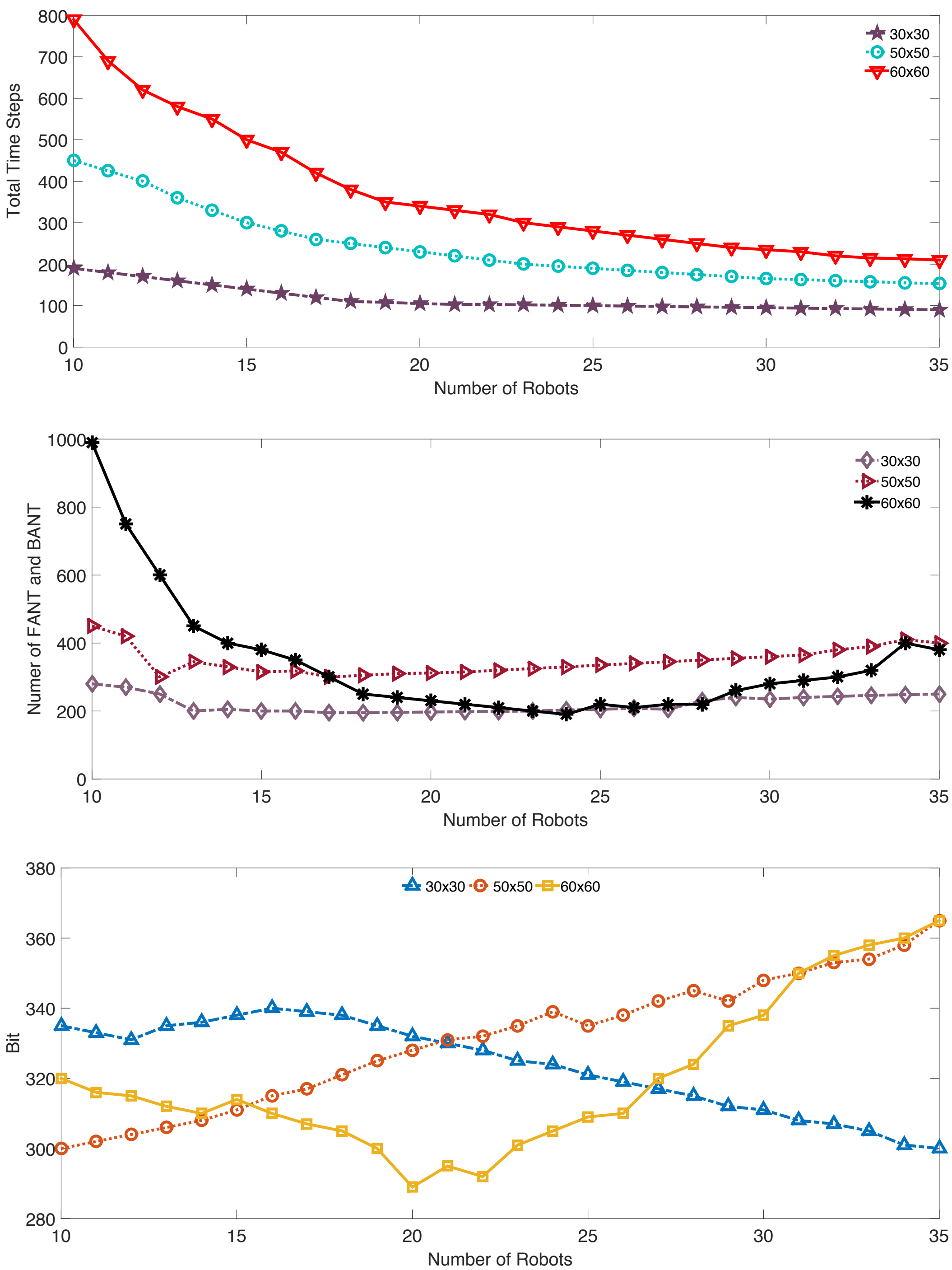



Fig. 5.32 Evaluation of the influence of the dimension of the area (a) Total Time Steps (b) Packets (c) Bit

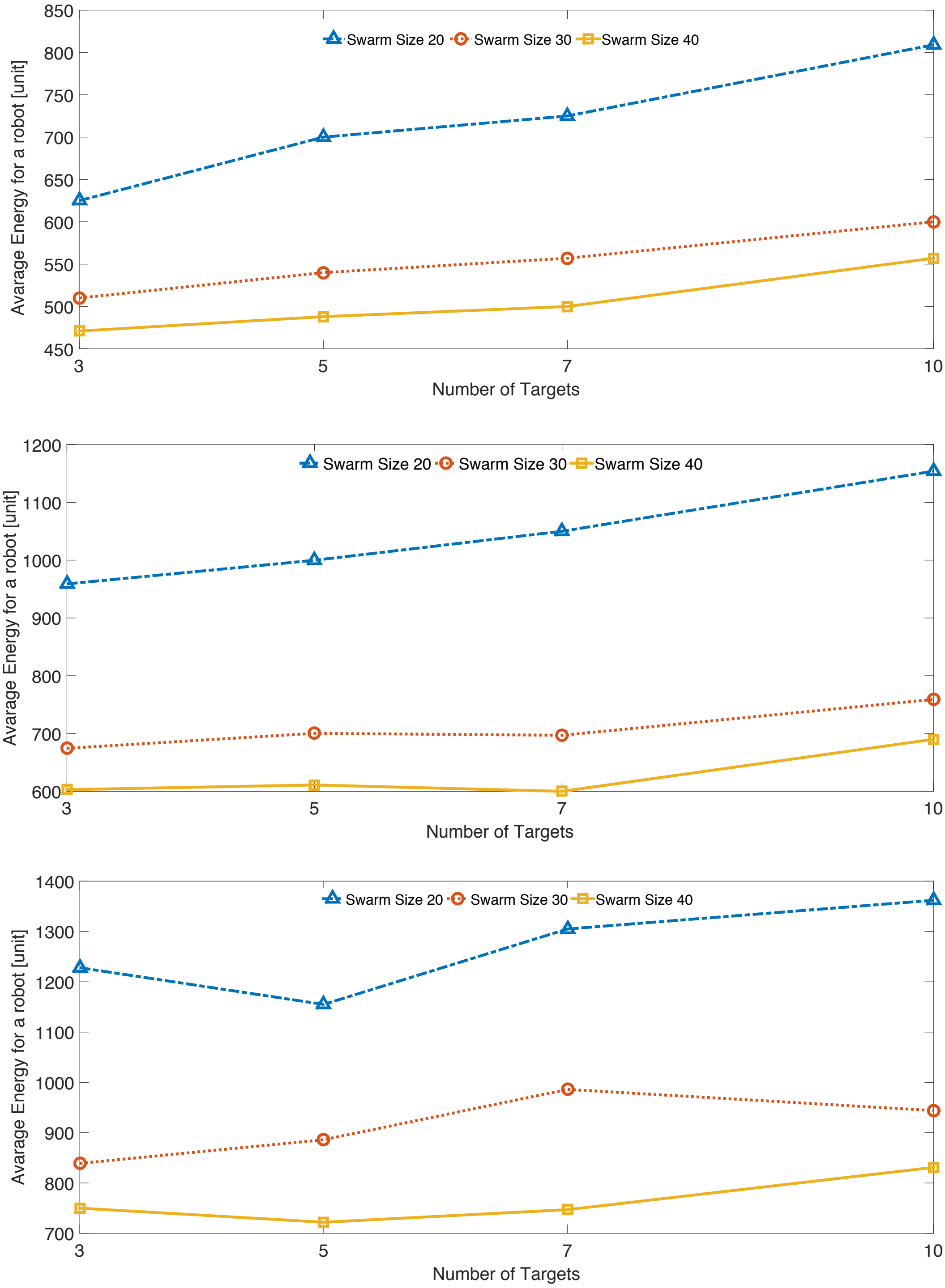



Fig. 5.33 Evaluation of average energy consumed by a robots varying the number of targets (a) 40x40 grid cell (b) 50x50 grid cell (c) 60x60 grid cell

Table 5.16 Comparison of FTS-RR and ATRC: Total Time Steps

(a)

| | FTS-RR vs ATRC | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | 30X30<br>20 robots<br>1 targets | 30X30<br>30 robots<br>1 targets | 30X30<br>40 robots<br>1 targets | 30X30<br>20 robots<br>3 targets | 30X30<br>30 robots<br>3 targets | 30X30<br>40 robots<br>3 targets | 30X30<br>20 robots<br>6 targets | 30X30<br>30 robots<br>6 targets | 30X30<br>40 robots<br>6 targets |
| **FTS-RR** | 103 | 91 | 74 | 178 | 109 | 95 | 173 | 106 | 75 |
| **ATRC** | 96 | 86 | 75 | 118 | 98 | 80 | 130 | 105 | 85 |

(b)

| | FTS-RR vs ATRC | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | 30X30<br>20 robots<br>3 targets | 30X30<br>30 robots<br>3 targets | 30X30<br>40 robots<br>3 targets | 50X50<br>20 robots<br>3 targets | 50X50<br>30 robots<br>3 targets | 50X50<br>40 robots<br>3 targets | 60X60<br>20 robots<br>3 targets | 60X60<br>30 robots<br>3 targets | 60X60<br>40 robots<br>3 targets |
| **FTS-RR** | 178 | 109 | 95 | 294 | 171 | 131 | 434 | 284 | 203 |
| **ATRC** | 105 | 99 | 85 | 230 | 190 | 153 | 340 | 280 | 210 |

## 5.12 One Hop Communication VS Multi Hop Communication

The previous sections have described deeply the problem of coordinating a team of robots to form coalitions in certain places of an unknown area. The presented approaches use both the wireless communication to announce the discovering of the targets. One approach uses only one hop communication and local interactions among the robots and no exchange of decisions among the team is considered.
The other approach has regarded the development of a protocol to coordinate the team. In this case the robots exchange simple information to avoid the redundancy in reaching the target's location. Although in the last approach is based on interactions among the robots, the protocol has been designed to minimize the number of generated packets. It is obtained considering a probabilistic mechanism of forward the packets.

Establish what it the best approach, is very hard since it depends on the context and on what is the metric most important. For the comparison it is used the firefly algorithm. Tables 5.14-5.15 show the main simulation results described above, in order to try to understand, potentially, what is the best approach to use.

Although the protocol, generally, can offer more benefits in terms of time, since it speeds up the mission, the consumed energy is grater since there is more communication among the team. The best approach to be used depends on many factors. Firstly, if the time is a

Table 5.17 Comparison of FTS-RR and ATRC: Avarage Energy for a Robot

(a)

| | **FTS-RR vs ATRC** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | 30X30<br>5 targets<br>20 robots | 30X30<br>7 targets<br>20 robots | 30X30<br>10 targets<br>20 robots | 50X50<br>5 targets<br>20 robots | 50X50<br>7 targets<br>20 robots | 50X50<br>10 targets<br>20 robots | 60X60<br>5 targets<br>20 robots | 60X60<br>7 targets<br>20 robots | 60X60<br>10 targets<br>20 robots |
| **FTS-RR** | 455 | 499 | 464 | 689 | 729 | 805 | 898 | 950 | 993 |
| **ATRC** | 625 | 700 | 725 | 958 | 999 | 1050 | 1228 | 1155 | 1305 |

(b)

| | **FTS-RR vs ATRC** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | 30X30<br>5 targets<br>30 robots | 30X30<br>7 targets<br>30 robots | 30X30<br>10 targets<br>30 robots | 50X50<br>5 targets<br>30 robots | 50X50<br>7 targets<br>30 robots | 50X50<br>10 targets<br>30 robots | 60X60<br>5 targets<br>30 robots | 60X60<br>7 targets<br>30 robots | 60X60<br>10 targets<br>30 robots |
| **FTS-RR** | 333 | 395 | 435 | 581 | 689 | 791 | 676 | 780 | 887 |
| **ATRC** | 510 | 540 | 557 | 674 | 697 | 719 | 839 | 886 | 944 |

(c)

| | **FTS-RR vs ATRC** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | 30X30<br>5 targets<br>40 robots | 30X30<br>7 targets<br>40 robots | 30X30<br>10 targets<br>40 robots | 50X50<br>5 targets<br>40 robots | 50X50<br>7 targets<br>40 robots | 50X50<br>10 targets<br>40 robots | 60X60<br>5 targets<br>40 robots | 60X60<br>7 targets<br>40 robots | 60X60<br>10 targets<br>40 robots |
| **FTS-RR** | 261 | 384 | 408 | 471 | 578 | 645 | 477 | 564 | 633 |
| **ATRC** | 471 | 488 | 501 | 602 | 611 | 690 | 750 | 722 | 811 |

critical variable thus using the protocol could be better to speed up the mission. Secondly if the resources of the system in terms of energy are crucial using only local interaction among the team may allow to minimize the consumed energy. Thirdly, it should be considered the conditions of the environment where the team operates. If the area is highly dynamic, hazardous and the conditions to maintain the network among the robots are unreliable, it could be suitable adapt an one hop communication. In uncertain area the robots may change decisions anytime. In these situation using a protocol the communication in terms of packets could increase and thus lead to an overhead of communication. However, the designed protocol is based on probabilistic mechanism to forward of the packets and make decision, so it can offers a scalable and distributed solution.

## 5.13 Summary

This chapter has described different approaches for the coordination of robots that need to form coalition at certain locations in search and rescue mission. Special attention was devoted to self adaptive behaviors. The sections analyze how the robots, make complex decisions based only from simple information forwarded to the team. The robots do not cooperate in making decisions, but each of them may optimize its own resources performance and incentive self-interested decision. In the proposal the robots collect information from their neighbors and then use this information to make decisions and performs actions based on its local constraints and a selfish perspective. Considering a small time window, a part of robots can have the same information and can make decisions in parallels that potentially can affect each other indirectly. The key issues of the problem are avoid redundancy to form the coalitions in target's location, reducing time and wastage of resources such as energy. Although, the control low that each robot executes is simple and decentralized, the emerging global behaviour is sophisticated and robust.

The proposed approaches are two. One uses a one hop communication mechanism to announce the detected targets and only local robots are informed to the targets and no propagation of information is done. At this purpose different strongly nature-inspired algorithms such as firefly algorithm, particle swarm optimization and distributed bee algorithms are proposed and modified properly fo the coordination of the swarm. The mechanisms are completely distributed and focus on self organizing and decision making mechanisms rather than routing, trying to minimize the exchange of informations among the robots. Moreover, each robot has no extra knowledge except for itself, such as resource capabilities, energy level and position. Each robots makes the decision individually without informing the others since it can potentially change the decision in next steps for some events such as new discovered targets, energy capabilities, other announcements and so on. Indeed, reconfiguration is essential for adaptive systems. If the robots work in unknown or dynamic environments, its members should be able to change their behaviour to improve the group's performance.
The algorithms were tested under different conditions and considering many parameters of the system in order to understand what is, potentially, the best algorithm that should be used. The results demonstrate that usually the firefly algorithm gives better performance especially in complex scenarios. A scenario can be considered complex when there are many disseminated targets compared to the number of the robots that operate in the area or a high number of robots that need to be involved in disarming process of a target. Under these conditions, usually, firefly algorithm and in many cases distributed bee algorithm, allows to spread among multiple target's locations the robots saving the resources of the system.

The second approach presents a network architecture for multi robots system where the information about the found targets can spread over the network of robots in a multi hop fashion. The idea is to use an ad hoc routing protocol to report the detected targets and the robots that want to help in disarming process over a MANET. Also, in this case a bio-inspired routing protocol is proposed in order to reduce the communication traffic in terms of packets and allows, at the same time, a self adaptive behaviors of the robots. More specifically, the protocol takes inspiration from the ability of certain types of ants in nature to find the shortest path between their nest and a food source through a distributed process based on stigmergic communication. In this case more importance is done to the routing mechanism to the packets that announce the targets. ACO routing algorithms boast a number of interesting properties compared to traditional routing algorithms. First of all, they are adaptive, thanks to the use of continuous path sampling and probabilistic ant forwarding, which leads to an uninterrupted exploration of the routing possibilities. Next, they are robust; this is because routing information is the result of the repeated sampling of paths. Finally, ACO routing algorithms can usually set multiple paths, over which data packets can be forwarded probabilistically like ants. This can result in throughput optimization, automatic data load balancing, and increased robustness to failures.
The proposed Ant-based task team robots coordination (ATRC) has been tested through simulations by varying different parameters of the problem in order to verify its robustness, convergence and scalability for increasing complexity. Results have shown that the proposed decentralized approach enables the swarm of robots to perform cooperative tasks intelligently without any central control.

Finally, a critical discussion among what kind of approach is the best has been treated. Establish what it the best approach is very hard since it depends on the context and on what is the metric most important. The use of a protocol and a minimal exchange of information among the team allows, relatively, to speed up the mission in terms of time. Although the robots do not exchange direct information of what they decide, through the information of the packets the robots indirectly could know, potentiality, the decisions and adapt their self to the system, showing a global desired behaviour. On the other hand, introducing communication among the swarm that is not only local, there are a wastage of resources in terms of energy, bandwidth and so on. Thus in hazardous scenario with constraints of resources such as energy or unreliable network conditions, generally, it could be useful use only local communication.

# Chapter 6

# Conclusion and future work

Multi-robot systems (MRS) have been proposed in the last decade in a variety of settings and frameworks, pursuing different research goals, and successfully applied in many application domains. Given a group of robots and a task to be performed, how to coordinate the robots in order to successfully complete the task is one of the most challenging research issues. A coordination mechanism should provide flexibility and adaptability, allowing multi-robot teams to execute tasks efficiently and robustly.

Recent works in robotic systems have been greatly inspired by the study of some species of animals in nature. The investigation of biological examples is playing a vital role in developing new robotic mechanisms, actuation techniques, and algorithms. Nature is of course a great and immense source of inspiration for solving hard and complex problems in computer science since it exhibits extremely diverse, dynamic, robust, complex and fascinating phenomenon. Nature inspired algorithms that can avoid complex, heavy computation and establish lightweight interactions and are therefore highly desirable for MRS.

In this work different techniques, able to coordinate and control groups of autonomous robots have been presented. The main feature of these techniques is to drive the group of robots to make the best decisions in order to perform search and rescue mission in unknown hazardous area. One of the key issues is how to specify the rules of behavior and interactions at the level of an individual robot in order to minimize unnecessary movements, turning, and communication that causes a wastage of the resource of the systems. The problem has been modeled as a bi-objective optimization model with two main goals: exploring the area for searching targets and targets resolving. The proposed approaches related to each phase form the main contributions of this work.

The work is based on a hybrid strategy which combines both indirect and direct communication mechanisms. It is studied how robots can accomplish the mission in a distributed and self-organized way through a stigmergic process in the exploration task, and simple information locally sent by the robots in the recruitment task. The system has unique features such as the minimal information exchange, and local interactions between simple homogeneous robots, achieving complex collective behaviour. Such solutions are in line with the general approaches used in swarm robotics, and support the desired system properties of robustness, adaptivity and scalability.

The exploration stage aims to explore the region and detect some targets distributed randomly in an unknown area and this is mainly implemented through an ant based strategy. In exploration, a swarm of mobile robots is deployed in an initially unknown environment with the goal of autonomously detect many targets disseminated in the area. A communication via environment (stigmergy) to share local knowledge on cells gained by individual robots is used. Using this approach, it is assumed that the robots do not know their positions and the positions of the others in the area, but they move according to what they can sense into the environment. Essentially, a repulsive pheromone mechanism is introduced into the swarm. This pheromone is deposited, immediately when a robot reaches a new cell in order to mark all cells that have been visited. The use of pheromone is similar to the use in Ant Colony Optimization method, but unlike ants, the robots should search for the cells without any pheromone or with the smallest pheromone value. Utilizing a stigmergic communication would be an efficient method of achieving such emergent behaviour with low overhead.

When a robot detects a target during the exploration phase, it becomes a coordinator for this target and it starts to initiate a recruitment process so as to attract other robots. This coordinator robot, together with recruited robots, will perform the handling of the found target to make it safe cooperatively. The detection of a target may happen at any time during the exploration of the area, so the recruitment process can take place in different regions of the area.

For this purpose, two different approaches, are used as coordination mechanism and wireless communication is used to share the information about the found targets, since direct communication may be beneficial when a fast reaction is expected and countermeasures must be taken.

The first uses only local simple interactions among the swarm and no exchange of information is done. These techniques have been based on the firefly, particle swarm and artificial bee behaviour, and some discrete modifications have been carried out to make these algorithms suitable for the purpose. The aim is to evaluate and then compare these techniques, which provides some insight into how a group of robots can respond to a task of demands effectively minimizing the resources of the system such as time to complete the mission and total energy consumed by the swarm. The second approach presents a network architecture that incorporates an self-regulation mechanism allowing the distribution among the targets, with minimal exchange of information.

The core idea is to use a multi-hop communication mechanism to spread the information of the detected targets, but limiting the sent packets, and the interactions among the robots. An Ant-based team robots coordination (ATRC) protocol is proposed as coordination mechanism among the team. The work is focused mainly on routing problem and how the information about the detected targets can be disseminated amount the overall swarm of robots. Special attention was devoted to the reduce the communication traffic and self adaptive behaviors. A purpose-designed simulator has been implemented in Java to test the effectiveness of the proposed algorithms and approaches. Moreover, a set of experiments has been conducted for evaluating the approaches considering different network parameters, and studying the scalability and the robustness of the proposal. The results give the evidence that the proposal can successfully solve the issue of multi robot coordination operating in complex and multi-tasking environment.

In order to properly characterize the proposal, it is worth highlighting the most important features described by the present work that can be summarized as follows:

- Flexibility since the parameters can be easily tuned so that the proposed methodology can be used to carry out exploration and recruitment tasks for a system of mobile robots.
- Scalability: the proposed approaches work well for any number of robots and targets.
- Adaptability: the proposed approaches can be used in the environment, allowing different conditions and distributions of targets and robots.
- Parallelism: the approaches are distributed and each robot performs its task in parallel and make decision individually, based on local partial information.
- The approaches have a low computation cost since it is not required to know the decisions of the other robots, but each robot acts selfishly taking the best decision from its own point of view.
- Since the algorithms for the coordination are not constructed for the specific target type, all of them are treated in the exact same manner: the same variable types are used, regardless of what the target is and what the robot is performing. Therefore, the proposal is generalized and can be used for a wide range of applications with minor modifications.

- Although each independent task is executed individually, the whole system can attempt to globally optimize the process.

The work and approaches presented in this dissertation have paved a way for exploring new bio-inspired techniques for optimizing complex tasks for swarming robots. Future work will focus on the extension of the current approaches to discrete domains to continuous domains. Extension will also explore the possibility of more complex, 2D geometrical areas with multiple obstacles or barriers and even 3D terrains with inaccessible regions such as rivers and lakes. Regarding the protocol, possible future works could include the extension of methods to dynamically adjust the number of hops to send the packets during the mission so as to be adaptive to the resource of the robots or other constraints. In addition, the proposed method can be modified to potentially deal with the unknown but mobile targets in an unknown area. Furthermore, further research can also consider the uncertainty concerning unreliable communication than can cause packets loss and inaccurate information, and thus make the overall system more reliable and robust.

# References


Arvin, F. and et. al. (2014). Cue-based aggregation with a mobile robot swarm: a novel fuzzy-based method. *Adaptive Behavior*, 22:189–206.

Bahceci, E., Soysal, O., and Sahin, E. (2003). A review: Pattern formation and adaptation in multi-robot systems. Technical Report CMU-RI-TR-03-43, Pittsburgh, PA.

Banharnsakun, J., Achalakul, T., and Batra, R. C. (2012). Target finding and obstacle avoidance algorithm for microrobot swarms. In *Systems, Man, and Cybernetics (SMC), IEEE International Conference on*.

Baras, J. S. and Mehta, H. (2003). A probabilistic emergent routing algorithm for mobile ad hoc networks.

Barichard, V., Gandibleux, V., and T'Kindt, V. (2009). *Multiobjective Programming and Goal Programming Theoretical Results and Practical Applications*. Springer-Verlag Berlin Heidelberg.

Basilico, N. and Amigoni, F. (2011). Exploration strategies based on multi-criteria decision making for searching environments in rescue operations. *Autonomous Robots*, 31(4):401.

Bayındır, L. (2016). A review of swarm robotics tasks. *Neurocomputing*, 172:292 – 321.

Bonabeau, E., Dorigo, M., and Theraulaz, G. (1999). *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press, Inc., New York, NY, USA.

Bouazizi, I. (2002). Ara - the ant-colony based routing algorithm for manets. In *Proceedings of the 2002 International Conference on Parallel Processing Workshops*, pages 79–. IEEE Computer Society.

Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press, New York, NY, USA.

Brambilla, M., Ferrante, E., Birattari, M., and Dorigo, M. (2012). Swarm robotics: a review from the swarm engineering perspective.

Burgard, W., Moors, M., Stachniss, C., and Schneider, F. E. (2005). Coordinated multi-robot exploration. *Trans. Rob.*, 21(3).

Calvo, R. and et.al. (2011). Inverse aco applied for exploration and surveillance in unknown environments. pages 131–142. in:COGNITIVE 2011 : The Third International Conference on Advanced Cognitive Technologies and Applications.

Camazine, S., Deneubourg, J. L., Franks, N. R., Sneyd, J. amd Theraulaz, G., and Bonabeau, E. (2001). *Selforganization in biological systems.* Princeton: Princeton University Press.

Chen, X., Kong, Y., Fang, X., and Wu, Q. (2013). A fast two-stage aco algorithm for robotic path planning. *Neural Computing and Applications*, 22(2):313–319.

Choset, H., Acar, E., Rizzi, A., and Luntz, J. (2000). Exact cellular decompositions in terms of critical points of morse functions. In *In: IEEE International Conference on Robotics and Automation*.

Clerc, M. and Kennedy, J. (2002). The particle swarm - explosion, stability, and convergence in a multidimensional complex space. *Trans. Evol. Comp*, 6(1):58–73.

Collette, Y. and Siarry, P. (2004). *Multiobjective Optimization Principles and Case Studies*. Springer-Verlag Berlin Heidelberg.

Contreras-Cruz, M. A., Ayala-Ramirez, V., and Hernandez-Belmonte, U. H. (2015). Mobile robot path planning using artificial bee colony and evolutionary programming. *Applied Soft Computing*, 30:319 – 328.

Couceiroa, M. S., Vargasc, P. A., Rochaa, R., and Ferreirab, N. (2014). Benchmark of swarm robotics distributed techniques in a search task. *Robotics and Autonomous Systems*, 62:200–213.

Countryman, S. M., Stumpe, M. C., Crow, S. P., Adler, F. R., Greene, M. J., Vonshak, M., and Gordon, D. M. (2015). Collective search by ants in microgravity. *Frontiers in Ecology and Evolution*, 3:25.

C.R.Kube (1997). Task modelling in collective robotics. *Autonomous Robots*, 4:53–72.

Şahin, E. (2005). Swarm robotics: From sources of inspiration to domains of application. In *Proceedings of the 2004 International Conference on Swarm Robotics*, SAB'04, pages 10–20, Berlin, Heidelberg. Springer-Verlag.

Das, P. and et al. (2016). A hybrid improved pso-dv algorithm for multi-robot path planning in a clutter environment. *Neurocomputing*, 207:735–753.

de Lopea, J., Maravalla, D., and Y.Quinonezb (2015). Self-organizing techniques to improve the decentralized multi-task distribution in multi-robot systems. *Neurocomputing*, 163:47–55.

De Rango, F. and Palmieri, N. (2012). A swarm-based robot team coordination protocol for mine detection and unknown space discovery. In *Wireless Communications and Mobile Computing Conference (IWCMC), 2012 8th International*.

De Rango, F. and Palmieri, N. (2016). Ant-based distributed protocol for coordination of a swarm of robots in demining mission. In *Proc. SPIE, Unmanned Systems Technology*.

De Rango, F., Palmieri, N., X-S., Y., and Marano, S. (2015). Bio-inspired exploring and recruiting tasks in a team of distributed robots over mined regions. In *Performance Evaluation of Computer and Telecommunication Systems (SPECTS), 2015 International Symposium on*. IEEE.

De Rango, F., Palmieri, N., Yang, X.-S., and Marano, S. (2017). Swarm robotics in wireless distributed protocol design for coordinating robots involved in cooperative tasks. *Soft Computing*.

De Rango, F. and Tropea, M. (2009). Energy saving and load balancing in wireless ad hoc networks through ant-based routing. In *Proceedings of the 12th International Conference on Symposium on Performance Evaluation of Computer & Telecommunication Systems*, SPECTS'09, pages 117–124. IEEE Press.

Delcomyn, F. (2007). *Biologically Inspired Robots*. Bioinspiration and Robotics Walking and Climbing Robots.

Derr, K. and Manic, M. (2009). Multi-robot, multi-target particle swarm optimization search in noisy wireless environments. In *IEEE, Human System Interactions, HSI '09. 2nd Conference on*.

Detrain, C. and Deneubourg, J.-L. (2006). Self-organized structures in a superorganism: do ants "behave" like molecules? *Physics of Life Reviews*, 3(3):162 – 187.

Di Caro, G. and Dorigo, M. (1998). Antnet: Distributed stigmergetic control for communications networks. *J. Artif. Int. Res.*, 9(1):317–365.

Di Caro, G., Ducatelle, F., and Gambardella, L. M. (2005). Anthocnet: An adaptive nature-inspired algorithm for routing in mobile ad hoc networks. *EUROPEAN TRANSACTIONS ON TELECOMMUNICATIONS*, 16:443–455.

Doi, S. (2013). Proposal and evaluation of a pheromone-based algorithm for the patrolling problem in dynamic environments. In *Swarm Intelligence (SIS), 2013 IEEE Symposium on*, pages 48–55.

Dorigo, M. and Blum, C. (2005). Ant colony optimization theory: A survey. *Theoretical Computer Science*, 344(2):243 – 278.

Dorigo, M. and et.al (2005). *The SWARM-BOTS Project*, pages 31–44. Springer Berlin Heidelberg, Berlin, Heidelberg.

Dorigo, M. and et.al (2013). Swarmanoid: a novel concept for the study of heterogeneous robotic swarms. *IEEE Robotics & Automation Magazine*, 20(4).

Dorigo, M. and Stutzle, T. (2003). The ant colony optimization metaheuristic:algorithm, applications, and advance. In *Handbook of Metaheuristic*, pages 250–285. Springer US.

Ducatelle, F., Di Caro, G. A., Förster, A., Bonani, M., Dorigo, M., Magnenat, S., Mondada, F., O'Grady, R., Pinciroli, C., Rétornaz, P., Trianni, V., and Gambardella, L. M. (2014). Cooperative navigation in robotic swarms. *Swarm Intelligence*, 8(1):1–33.

Ducatelle, F., Di Caro, G. A., and Gambardella, L. M. (2006). *An Analysis of the Different Components of the AntHocNet Routing Algorithm*, pages 37–48. Springer Berlin Heidelberg.

Ducatelle, F., Di Caro, G. A., Pinciroli, C., and Gambardella, L. M. (2011). Self-organized cooperation between robotic swarms. *Swarm Intelligence*, 5(2):73.

Farinelli, A., Iocchi, L., and Nardi, D. (2004). Multirobot systems: a classification focused on coordination. *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS*, 25:2015–2028.

Fister Jr., I., Yang, X.-S., Fister, I., Brest, J., and Fister, D. (2013). A brief review of nature-inspired algorithms for optimization. *Elektrotehniski vestnik*, 80(3):116–122.

Fujisawa, R., Dobata, S., Kubota, D., Imamura, H., and Matsuno, F. (2008). *Dependency by Concentration of Pheromone Trail for Multiple Robots*, pages 283–290. Springer Berlin Heidelberg, Berlin, Heidelberg.

Garciaa, M. P. and et. al. (2009). Path planning for autonomous mobile robot navigation with ant colony optimization and fuzzy cost function evaluation. *Applied Soft Computing*, 9:1102–1110.

Garnier, S., Gautrais, J., and Theraulaz, G. (2007). The biological principles of swarm intelligence. *Swarm Intelligence*, 1(1):3–31.

Gauci, M., Chen, J., Li, W., Dodd, T. J., and Groß, R. (2014). Self-organized aggregation without computation. *The International Journal of Robotics Research*, 33(8):1145–1161.

Gautam, A. and Mohan, S. (2012). A review of research in multi-robot systems. In *Industrial and Information Systems (ICIIS), 2012 7th IEEE International Conference on*.

Gifford, C. M., Webb, R., Bley, J., Leung, D., Calnon, M., Makarewicz, J., Banz, B., and Agah, A. (2010). A novel low-cost, limited-resource approach to autonomous multi-robot exploration and mapping. *Robotics and Autonomous Systems*, 58(2):186 – 202.

Giurfa, M. and Menzel, R. (2003). *Cognitive Architecture of a Mini-Brain*, pages 23–48. Springer Berlin Heidelberg, Berlin, Heidelberg.

Gordon, D. (2010). *Ant Encounters: Interaction Networks and Colony Behavior*. Princeton, NJ: Princeton University Press.

Gregory, J., Fink, J., Stump, E., Twigg, J., Rogers, J., Baran, D., Fung, N., and Young, S. (2016). *Application of Multi-Robot Systems to Disaster-Relief Scenarios with Limited Communication*, pages 639–653. Springer International Publishing.

Groß, R., Bonani, M., Mondada, F., and Dorigo, M. (2006). *Autonomous Self-assembly in a Swarm-bot*, pages 314–322. Springer Berlin Heidelberg, Berlin, Heidelberg.

Guy, B. and et.al (2009). Open-ended on-board evolutionary robotics for robot swarms. In *IEEE Congress on Evolutionary Computation*, pages 1123–1130. IEEE.

Hazon, N. and Kaminka, G. A. (2008). On redundancy, efficiency, and robustness in coverage for multiple robots. *Robotics and Autonomous Systems*, 56(12):1102 – 1114.

Hereford, J. M. and Siebold, M. A. (2012). Bio-inspired search strategies for robot swarms.

Herianto, Sakakibara, T., and Kurabayashi, D. (2007). Artificial pheromone system using rfid for navigation of autonomous robots. *Journal of Bionic Engineering*, 4(4):245 – 253.

Hoff, N., Sagoff, A., Wood, R. J., and Nagpal, R. (2010). Two foraging algorithms for robot swarms using only local communication. In *ROBIO*, pages 123–130. IEEE.

Hsieh, M. A., Halász, A., Berman, S., and Kumar, V. (2008). Biologically inspired redistribution of a swarm of robots among multiple sites. *Swarm Intelligence*, 2:121–141.

Jevtić, A. and de la Fuente, D. A. (2007). Swarm intelligence and its applications in swarm robotics. In *Proceedings of the WSEAS international conferences : 6th WSEAS International Conference on Computational Intelligence, Man-Machine Systems and Cybernetics*, [Athens]. WSEAS.

Jevtić, A., Martín, Á. G., de la Fuente, D. A., and Jamshidi, M. (2012). Distributed bees algorithm for task allocation in swarm of robots. *IEEE Systems Journal*, 6(2):296–304.

Johansson, R. and Saffiotti, A. (2009). Navigating by stigmergy: A realization on an rfid floor for minimalistic robots. In *Proceedings of the 2009 IEEE International Conference on Robotics and Automation*, ICRA'09, pages 3520–3527. IEEE Press.

Jones, E. G., Dias, M. B., and Stentz, A. (2007). Learning-enhanced market-based task allocation for oversubscribed domains. In *Intelligent Robots and Systems, IROS, IEEE/RSJ International Conference on*.

Kalra, N. and Martinoli, A. (2006). *Comparative Study of Market-Based and Threshold-Based Task Allocation*, pages 91–101. Springer Japan.

Karaboga, D. and Akay, B. (2009). A comparative study of artificial bee colony algorithm. *Applied Mathematics and Computation*, 214(1):108 – 132.

Kennedy, J. and Heberhart, R. (1995). Particle swarm optimization. In *In Proceedings of the 1995 IEEE International Conference on Neural Networks*, pages 1942–1948.

Kernbach, S. and et.al (2008). Symbiotic robot organisms: Replicator and symbrion projects. In *In Proceedings of the 8th Workshop on Performance Metrics for Intelligent Systems*, pages 62–69.

Kernbach, S., Thenius, R., Kernbach, O., and Schmickl, T. (2009). Re-embodiment of honeybee aggregation behavior in an artificial micro-robotic system. *Adaptive Behavior - Animals, Animats, Software Agents, Robots, Adaptive Systems*, 17(3):237–259.

Khamis, A., Hussein, A., and Elmogy, A. (2015). *Multi-robot Task Allocation: A Review of the State-of-the-Art*, pages 31–51. Springer International Publishing, Cham.

Kuyucu, T. and et.al. (2012). *Evolutionary Optimization of Pheromone-Based Stigmergic Communication*. Springer Berlin Heidelberg, Berlin, Heidelberg.

Kuyucu, T., Tanev, I., and Shimohara, K. (2015). Superadditive effect of multi-robot coordination in the exploration of unknown environments via stigmergy. *Neurocomputing*, 148:83 – 90.

Li, J. and et al. (2014). A modified particle swarm optimization algorithm for distributed search and collective cleanup. In *IEEE, Awareness Science and Technology and Ubi-Media Computing (iCAST-UMEDIA),International Joint Conference on*.

Liang, J. and Lee, C. (2015). Efficient collision-free path-planning of multiple mobile robots system using efficient artificial bee colony algorithm. *Advances in Engineering Software*, 79:47–56.

Liu, Y. and Passino, K. (2002). Biomimicry of social foraging bacteria for distributed optimization: Models, principles, and emergent behaviors. *Journal of Optimization Theory and Applications*, 115(3):603–628.

Luna, R. and Bekris, K. E. (2011). Efficient and complete centralized multi-robot path planning. *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3268–3275.

Marler, R. and Arora, J. (2010). The weighted sum method for multi-objective optimization: new insights. *Structural and Multidisciplinary Optimization*, 41:853–862.

Masar, M. (2013). A biologically inspired swarm robot coordination algorithm for exploration and surveillance. In *In: Proceedings of the IEEE 17th internatinal conference on intelligent engineering systems (INES 2013)*, pages 271–275.

Mataric, M. (1998). Using communication to reduce locality in distributed multiagent learning. *Journal of Experimental & Theoretical Artificial Intelligence*, 10(3):357–369.

Mei, Y., Lu, Y., Lee, C., and Hu, Y. (2006). Energy efficient mobilerobot exploration. In . *In the Proceedings of the IEEE International Conference of Robotics and Automation*, pages 505–551. IEEE.

Miner, D. (2007). Swarm robotics algorithms: A survey.

Momen, S. (2013). Ant-inspired decentralized task allocation strategy in groups of mobile agents. *Procedia Computer Science*, 20:169–176.

Mosteo, A. R. and Montano, L. (2010). A survey of multi-robot task allocation. In *Tech.Rep.*

Navarro, I. and Matía, F. (2010). An introduction to swarm robotics. *ISRN Robotics*.

Nemhauser, G. L. and Wolsey, L. A. (1988). *Integer and Combinatorial Optimization*. Wiley-Interscience, New York, NY, USA.

Ooi, C. C. and Schindelhauer, C. (2009). Minimal energy path planning for wireless robots. *Mobile Networks and Applications*, 14:309–321.

Osherovich, J., Yanovki, V., Wagner, I. A., and Bruckstein, A. M. (2008). Robust and efficient covering of unknown continuous domains with simple, ant-like a(ge)nts. *The International Journal of Robotics Research*, 27:815–831.

Palmieri, N., De Rango, F., X-S., Y., and Marano, S. (2015). Multi-robot cooperative tasks using combined nature-inspired techniques. In *Computational Intelligence (IJCCI), 2015 7th International Joint Conference on*. IEEE.

Palmieri, N., X-S., Y., and De Rango, F. (2017a). Self-adaptive mechanism for coalitions formation in a heterogeneous robot network. In *The 21st International Symposium on Distributed Simulation and Real Time Applications (DS-RT 2017)*.

Palmieri, N., X-S., Y., and Marano, S. (2016). Coordination techniques of mobile robots with energy constraints. In *Performance Evaluation of Computer and Telecommunication Systems (SPECTS), 2016 International Symposium on*. IEEE.

Palmieri, N., Yang, X.-S., De Rango, F., and Marano, S. (2017b). Comparison of bio-inspired algorithms applied to the coordination of mobile robots considering the energy consumption. *Neural Computing and Applications*, pages 1–24.

Parker, L. E. (2012). *Decision Making as Optimization in Multi-robot Teams*, pages 35–49. Springer Berlin Heidelberg, Berlin, Heidelberg.

Payton, D., Daily, M., Estowski, R., Howard, M., and Lee, C. (2001). Pheromone robotics. *Autonomous Robots*, 11(3):319–324.

Pearce, J. L., Powers, B., Hess, C., Rybski, P. E., Stoeter, S., and N.Papanikolopoulos (2006). Using virtual pheromones and cameras for dispersing a team of multiple miniature robots. *Journal of Intelligent and Robotic Systems*, 45:307–321.

Pfeifer, R., Lungarella, M., and Iida, F. (2007). Self-organization, embodiment, and biologically inspired robotics. *Science*, 318(5853):1088–1093.

Pugh, J., Martinoli, A., and Zhang, Y. (2005). Particle swarm optimization for unsupervised robotic learning. *Proceedings 2005 IEEE Swarm Intelligence Symposium, 2005. SIS 2005.*, pages 92–99.

Purnamadjaja, A. h. and Russell, R. a. (2010). Bi-directional pheromone communication between robots. *Robotica*, 28(1):69–79.

Ranjbar-Sahraei, B., Weiss, G., and Nakisaee, A. (2012). *A Multi-robot Coverage Approach Based on Stigmergic Communication*, pages 126–138. Springer Berlin Heidelberg.

Ravankar, A., Ravankar, A., Kobayashi, Y., and Emaru, T. (2016). On a bio-inspired hybrid pheromone signalling for efficient map exploration of multiple mobile service robots. *Artificial Life and Robotics*, 21:221–231.

Rothermich, J. A., Ecemiş, M. İ., and Gaudiano, P. (2005). *Distributed Localization and Mapping with a Robotic Swarm*, pages 58–69. Springer Berlin Heidelberg, Berlin, Heidelberg.

Rubenstein, M., Cabrera, A., Werfel, J., Habibi, G., McLurkin, J., and Nagpal, R. (2013). Collective transport of complex objects by simple robots: Theory and experiments. In *Proceedings of the 2013 International Conference on Autonomous Agents and Multi-agent Systems*, AAMAS '13, pages 47–54. International Foundation for Autonomous Agents and Multiagent Systems.

Schroeder, A., Ramakrishnan, S., Kumar, M., and Trease, B. (2017). Efficient spatial coverage by a robot swarm based on an ant foraging model and the lévy distribution. *Swarm Intelligence*, 11(1):39–69.

Seeley, T. D. (2002). When is self-organization used in biological systems? *The Biological Bulletin*, 202(3):314–318.

Senanayake, M., Senthooran, I., Barca, J. C., Chung, H., Kamruzzaman, J., and Murshed, M. (2016). Search and tracking algorithms for swarms of robots: A survey. *Robotics and Autonomous Systems*, 75, Part B:422 – 434.

Singh, G., Kumar, N., and Verma, A. K. (2012). Ant colony algorithms in manets: A review. *Journal of Network and Computer Applications*, 35(6):1964 – 1972.

Solanas, A. and Garcia, M. A. (2004). Coordinated multi-robot exploration through unsupervised clustering of unknown space. In *in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst*, pages 852–858.

Sugawara, K., Kazama, T., and Watanabe, T. (2004). Foraging behavior of interacting robots with virtual pheromone. *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, 3:3074–3079 vol.3.

Tan, Y. and yang Zheng, Z. (2013). Research advance in swarm robotics. *Defence Technology*, 9(1):18 – 39.

Trianni, V., Ampatzis, C., Christensen, A. L., Tuci, E., Dorigo, M., and Nolfi, S. (2007). *From Solitary to Collective Behaviours: Decision Making and Cooperation*, pages 575–584. Springer Berlin Heidelberg, Berlin, Heidelberg.

Trianni, V., Labella, T., and Dorigo, M. (2004). Evolution of direct communication for a swarm-bot performing hole avoidance,. pages 131–142. in: Proceedings of the fourth International Workshop on Ant Colony Optimization and Swarm Intelligence.

Triguia, S. and et al. (2014). A distributed market-based algorithm for the multi-robot assignment problem. *Procedia Computer Science*, 32:1108 – 1114.

Wagner, I., Lindenbaum, M., and Bruckstein, A. (1999). Distributed covering by ant-robots using evaporating traces. *IEEE Trans. Robot. Autom*, 15(5):918–933.

Wang, T. e. a. (2008). Staying-alive and energy-efficient path planning for mobile robots. In *American Control Conference, IEEE.*

Wattanavekin, T., Ogata, T., Hara, T., and Ota, J. (2013). Mobile robot exploration by using environmental boundary information.

Yamashita, A., Arai, T., and Ota, J. (2003). Motion planning of multiple mobile robots for cooperative manipulation and transportation. *IEEE Transactions on Robotics and Automation*, 19(2):223–237.

Yamauchi, B. (1999). Decentralized coordination for multirobot exploration. *Robotics and Autonomous Systems*, 29(2–3):111 – 118.

Yan, Z., Jouandeau, N., and Cherif, A. A. (2012). Multi-robot heuristic goods transportation. In *In Proceedings of IS12,*.

Yan, Z., Jouandeau, N., and Cherif, A. A. (2013). A survey and analysis of multi-robot coordination. *International Journal of Advanced Robotic Systems*, 10(12):399.

Yang, B., Ding, Y., Jin, Y., and Hao, K. (2015). Self-organized swarm robot for target search and trapping inspired by bacterial chemotaxis. *Robotics and Autonomous Systems*, 72:83 – 92.

Yang, X. (2009). Firelfly algorithms for multimodal optmization. In *Stochastic algorithms:foundations and applications. SAGA*, volume 5792, pages 169–178. Springer.

Yang, X. and Deb, S. (2009). Cuckoo search via lévy flightsn. In *Nature Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on*. IEEE.

Yang, X. S. (2008). *Nature inspired MetaHeuristic Algorithms*. Luniver Press.

Yang, Y., Zhou, C., and Tian, Y. (2009). Swarm robots task allocation based on response threshold model. In *Autonomous Robots and Agents, ICARA 4th International Conference on*.

Yanovski, V., Wagner, I. A., and Bruckstein, A. M. (2001). Vertex-ant-walk – a robust method for efficient exploration of faulty graphs. *Annals of Mathematics and Artificial Intelligence*, 31(1):99–112.

Yogeswaran, M. and Ponnambalam, S. G. (2010). *Swarm Robotics: An Extensive Research Review*.

Zhang, B., Liu, T., Zhang, C., and Wang, P. (2016). Artificial bee colony algorithm with strategy and parameter adaptation for global optimization. *Neural Computing and Applications*, pages 1–16.

Zhao, T. and Wang, Y. (2013). Market based multi-robot coordination for a cooperative collecting and transportation problem. In *Southeastcon, 2013 Proceedings of IEEE*.

Zheng, X., Koenig, S., Kempe, D., and Jain, S. (2010). Multi-robot forest coverage for weighted and un-weighted terrain. . *EEE Transactions on Robotics.*, 26(6):1018 – 1031.

Zhu, Q. (2011). *Design and implementation of swarm robots.* Shanghai: Shanghai Jiao Tong University.

Ziparo, V., Kleiner, A., Marchetti, L., Farinelli, A., and Nardi, D. (2007). Cooperative exploration for USAR robots with indirect communication. In *6th IFAC Symposium on Intelligent Autonomous Vehicles (2007), Intelligent Autonomous Vehicles, Volume 6, Part 1*.