Specialized Agents Task Allocation in Autonomous Multi-Robot Systems

By

Omar S. M. AL-Buraiki

Dissertation submitted to the University of Ottawa

in partial fulfillment of the requirements for the

Doctorate in Philosophy degree in Electrical and Computer Engineering

Ottawa-Carleton Institute of Electrical and Computer Engineering

School of Electrical Engineering and Computer Science

Faculty of Engineering

University of Ottawa

Table of Contents

List of Figures

List of Tables

Acronyms

Acronym	Meaning
ATSU	Automatic Task Selection Unit
BETA	Binary Encoding Task Allocator
CNN	Convolutional Neural Networks
FBL	Feedback Linearization
HSFL	High Specialty Fitting Level
LSFL	Low Specialty Fitting Level
MFT	Minimum Fitting Threshold
META	Modulated Encoding Task Allocator
PDF	Probability Density Function
SAR	Search-And-Rescue
UGV	Unmanned Ground Vehicle
UAV	Unmanned Aerial Vehicle
USV	Unmanned Sea Vehicle

Abstract

With the promise to shape the future of industry, multi-agent robotic technologies have the potential to change many aspects of daily life. Over the coming decade, they are expected to impact transportation systems, military applications such as reconnaissance and surveillance, search-and-rescue operations, or space missions, as well as provide support to emergency first responders.

Motivated by the latest developments in the field of robotics, this thesis contributes to the evolution of the future generation of multi-agent robotic systems as they become smarter, more accurate, and diversified in terms of applications. But in order to achieve these goals, the individual agents forming cooperative robotic systems need to be specialized in what they can accomplish, while ensuring accuracy and preserving the ability to perform diverse tasks.

This thesis addresses the problem of task allocation in swarm robotics in the specific context where specialized capabilities of the individual agents are considered. Based on the assumption that each individual agent possesses specialized functional capabilities and that the expected tasks, which are distributed in the surrounding environment, impose specific requirements, the proposed task allocation mechanisms are formulated in two different spaces. First, a rudimentary form of the team members' specialization is formulated as a cooperative control problem embedded in the agents' dynamics control space. Second, an advanced formulation of agents' specialization is defined to estimate the individual agents' task allocation probabilities in a dedicated specialization space, which represents the core contribution of this thesis to the advancement and practice in the area of swarm robotics.

The original task allocation process formulated in the specialization space evolves through four stages of development. First, a task features recognition stage is conceptually introduced to

leverage the output of a sensing layer embedded in robotic agents to drive the proposed task allocation scheme. Second, a matching scheme is developed to best match each agent's specialized capabilities with the corresponding detected tasks. At this stage, a general binary definition of agents' specialization serves as the basis for task-agent association. Third, the task-agent matching scheme is expanded to an innovative probabilistic specialty-based task-agent allocation framework to generalize the concept and exploit the potential of agents' specialization consideration. Fourth, the general framework is further refined with a modulated definition of the agents' specialization based on their mechanical, physical structure, and embedded resources. The original framework is extended and a prioritization layer is also introduced to improve the system's response to complex tasks that are characterized based on the recognition of multiple classes.

Experimental validation of the proposed specialty-based task allocation approach is conducted in simulation and on real-world experiments, and the results are presented and discussed in light of potential applications to demonstrate the effectiveness and efficiency of the proposed framework.

Acknowledgements

First, I would like to express my gratitude to my supervisor, Dr. Pierre Payeur, for guiding me in this journey, shaping me as an independent researcher, and teaching me how to think critically and how to write science. For his patience and support, I am grateful.

Second, I offer my heartfelt thanks to those who truly love me, especially my beloved parents, siblings, wife, daughter, and sons. Words cannot express how grateful I am for the endless support and love you have given me to achieve my goal. In particular, I give special thanks to my beloved mother and to my role model, my father. They have taught me the real value of progressing in this life. They have always trusted me and supported me in whatever path I chose, in any way. Finally, they gave me the conditions and education to reach this point where I am today.

Third, this work would not have been possible without financial support from the Hadhramout Foundation. To this foundation I would like to express my deep gratitude. I am especially grateful to the leader of the educational revolution in Hadhramout, Eng. Abdullah Bugshan, for his support through the challenging road of this journey since the onset of my Master's program to the completion of my Ph.D.

I would also like to thank my closest friend, Dr. Karama Al-Tamimi, for all the support and encouragement you give me. Last but not least, I am thankful to my friends and colleagues at the University of Ottawa. I would like to express my special thanks to Mohammed Bin Hariz, Hitham Jleed, Mathew Ross, Shengsong Yang, Danial Nakhaeinia, Wenbo Wu and Angle Valencia for their valuable help and cooperation.

Dedication

To

The souls of those dearest to me, my grandparents.

While completing my master and Ph.D. studies away from home, I have been deeply saddened by your passing. I dedicate my work to you.

Chapter 1 Introduction

1.1 Motivation

Robotic technologies, by providing smart and powerful solutions to support daily life activities, are promising to take over some of the roles occupied by human beings for performing complex and dangerous tasks. Swarm robots consist of multi-agent systems comprised of a group of individual robotic agents working together toward a common objective. Swarm robotics is an approach to the coordination of robots with the objective that a desired collective behaviour emerges from the interactions between the individual robots and the surrounding environment. Swarm robots will continue to play a prominent role in the manufacturing and service industry, as well as in space and security/military applications in the future. Along this trend of development, swarm robotics requires improvements in order to allow robotic agents to achieve a high level of self-determination on task allocation and completion. In the context of this research, task allocation, or task-agent assignment, refers to the process of selecting robotic agent/agents that are to be allocated to a given task and ensuring that the selected agent/agents are sufficiently qualified, in terms of the agents' functional capabilities, to fulfill the execution requirements of the task to the greatest extent.

To support such initiatives, the formation control of multi-robot cooperative systems has been studied extensively in the literature. Collaboration in multi-agent systems provides extended capabilities such as data exchange and cooperation control. Such cooperative approaches allow robots to handle more complex tasks including data collection, environment exploration, search-and-rescue, mining, harvesting, manufacturing, delivery, or damage assessment in disaster-affected areas.

Motivated by the latest developments in the field of unmanned autonomous intelligent systems and cooperative control, this research aims to contribute to the evolution of the future generation of cooperative robots for them to be smarter, more accurate, and accessible to a broader range of applications. For that matter, the individual components of cooperative robotic systems will need to be specialized while ensuring accuracy and ability to perform diverse tasks.

Therefore, the goal of this research is to evolve the swarm team to be well functioning as a cooperative team in which individual agents' specializations are defined on the basis of their heterogeneous functionalities. The resulting team of robots should be able to maintain a precise level of accuracy while focusing on the specialization that leads to the distribution of the tasks, where each task is to be mastered by an individual agent or a group of agents among the swarm. The focus in this thesis is on the specific constraints set by the specialization of the individual agents, and solutions are proposed for a swarm of specialized robots to be efficiently allocated or dispatched to corresponding tasks that impose specific requirements for their satisfactory completion. This perspective has not been extensively addressed in the literature.

Previous and current research on the coordination of multi-agent systems has mainly focused on two directions. On one hand, the literature has focused on the coordination of equal agents, or homogeneous coordination techniques, where the coordination of a group of homogeneous robotic members is extensively studied. On the other hand, heterogeneous coordination is considered to solve the coordination problems in multi-agent systems composed of individual robotic agents that are different in their physical construction. However, while research on multi-agent systems has focused on the above-mentioned directions, future development needs to go beyond the consideration of only homogeneous or heterogeneous systems. The goal is to reach formal

considerations of specialization in order to optimize the functionalities of the swarm's members based on the specific constraints imposed by the given tasks to be performed. This concept opens new directions based on the formal consideration of a swarm's individual agents possessing heterogeneous functionalities.

In this thesis, the problem of multiple robotic agents' specialization is studied from two perspectives. First, a specialty-based task allocation of the individual agents is formulated as a swarm's cooperative formation control problem. This rudimentary solution allocates the most competent individuals over a given workspace while considering some specific tasks to be performed. Second, a more advanced specialty-based task allocation solution is formulated in a probabilistic form. At this level of development, a rigorous formulation of the agents' specialization is developed to compute the respective agents' task allocation probabilities in a dedicated specialization space. This represents the core contribution of the thesis.

Ultimately, while being allocated to specific tasks based on their embedded functional capabilities that best match task-related constraints, the individual agents should still interact as a cooperative team. As a result, the significance of this study also emerges from the integration between individual agents' specialized skills and swarm effectiveness.

1.2 Objectives and Contributions

The main objective of this thesis is to address the design aspects of a cooperative coordination framework for a group of robots that consists of specialized individual members. While each member is prepared or equipped to perform specific tasks, the latter impose specific constraints. The proposed system considers different types of unmanned systems such as unmanned ground,

aerial, or sea (UGV, UAV, USV) vehicles to be heterogeneous at the functionality level, where heterogeneity is imposed by the specific constraints of the task recognized through an embedded sensory layer. This research considers the multi-robot coordination problem as a two-level coordination strategy. That is, agents' coordination control is developed in two spaces: first, the control space, which accommodates fundamental robot requirements (e.g. navigation, formation control, and localization), and second, the specialization space, which addresses the specialization required for interacting with the environment (e.g. exploration, sensing, tracking, and picking). This trend contributes to enhance the individual's particularities among a robotic team in terms of specialization, quality, precision, and efficiency of the autonomous team operation, and collectively in terms of the smartness of the swarm. The framework that emerges from this work presents a comprehensive study and experimental evaluation of the proposed cooperative coordination strategy for multi-agent systems in the two coupled spaces. The specialization space, which includes several design aspects for coordinating the specialized individual agents to the corresponding tasks, represents the main contribution of this thesis. A conceptual mechanism is developed for task recognition under uncertainty, followed by a formal probabilistic representation of task-agent specialty matching and allocation of the specialized agents to perform a task with optimization of agents' competencies while also considering the agents' availability, the agents' attendance level, and the potential influence of human-on-the-loop control. The proposed framework is experimentally validated in simulation and on real robotic platforms.

1.3 Problem Statement

This work aims to fill technological gaps from previous works that have addressed the problem of specializing individual members of robotic swarms. The outcomes of this thesis contribute to

advance the field of swarm robotics by the development of an innovative framework that leverages sensing capabilities to support the conceptualization and implementation of specialization among the agents. The focus is placed on defining the specialty of the individual agents and on their matching with tasks to be mastered by these individual agents based on particular constraints imposed by each task. To this end, the research intends to provide answers to the following research questions:

Question 1: How can a swarm of robots with a relatively low-level of intelligence perform complex tasks requiring specialization in the individual robotic agents?

Over the last two decades, cooperative formation control of multi-agent systems has received significant attention from researchers. Building upon methodologies that are considered in the literature review (Chapter 2), this thesis presents a solution to fill a research gap regarding the specialization of individual members in a robotic team. The focus here is to reinforce the swarm's capabilities by benefiting from the agents' non-homogeneity. To that end, this thesis extends the concept of cooperative swarms and proposes a rigorous process to leverage the specialized heterogeneities between individual agents.

To answer Question 1, an early investigation is first conducted to validate the concept. This process defines different zones of a robotic team's workspace as separated areas assigned to different tasks. The developed solution allows robots to smoothly and safely switch their positions and dynamically adapt the overall formation as a response to a specific task to be performed in each visited zone. It defines how the swarm's formation is managed over each zone. As a result of a rudimentary form of specialization, the framework switches the agents' leader role from one agent to another as the group transfers from one zone to another. In the first phase of research, it

is assumed that a specific agent (i.e. one member of the swarm) is the only specialized one for the leading task in each zone. This agent is assigned as the specialized leader of the group in the related zone until the task is completed. Details on this first part of the research are reported in Section 3.4.

***Question* 2: Can the initial approach be upgraded to allow specialized individuals to benefit from automatic target recognition in their environment, and can the cooperation between agents become robust and flexible enough to automatically assign the proper specialized agents to their corresponding tasks upon discovery?**

To answer this question, the original coordination system for task-based leadership assignment among specialized robots is further extended. The problem is reformulated by evolving the definition of specialized zones of influence that are distributed over the workspace for the group of robots. Inspired by earlier formation approaches, an interesting solution emerges. Starting from the assumption that automatic target recognition in the workspace is possible using on-board sensors, the process of selecting a specialized agent, which is qualified to perform a recognized task, should progressively proceed through three basic states which are respectively named *searching*, *task*, and *execution* states. This approach is also extended to account for dynamic zones of influence when a target is moving, as well as when multiple tasks share the same zone. This better reflects realistic intervention scenarios with mobile robots. The work performed in this direction is reported in Section 3.5.

***Question* 3: For improving the swarm's management, can the specialized functionality of each agent be properly defined and formulated in a probabilistic matching mechanism based on a certain level of suitability of the agents' specialization to respond to a given task?**

To address this question, the framework design used for specializing the role of individual agents can be refined by leveraging probability-based modelling. Building such a probabilistic

representation of the behaviour of the individual agents leads to a scalable mechanism. The latter supports an automated process that is able to handle uncertainty in tasks and specialization definitions, as well as cope with an imperfect match between task constraints and specialized agents' capabilities. The proposed solution for specialized agents' selection is formulated in terms of the confidence level in task recognition and on the probability of a given agent to be qualified to meet the task's specific requirements. The proposed specialty-based task allocation scheme aims to calculate the matching suitability, or task-agent specialty fitting probability, of the individual agents among the swarm based on the recognized task constraints. The development of this framework is detailed in Sections 4.3 and 4.4.

Question 4: Can the proposed framework be applied in real systems?

To demonstrate the validity of the proposed approach and its potential for real-world applications, it is tested in two ways. First, the proposed framework is validated in simulation to demonstrate its effectiveness. Extensive simulation experiments are presented in Chapter 5. Then, a potential application of the proposed approach is presented as a case study, while the implementation of the proposed framework on real robotic platforms is detailed to validate and examine the performance of the specialty-based task allocation scheme to achieve the expected level of coordination. This work is presented in Chapter 6.

1.4 Thesis Organization

The chapters of this thesis are organized as follows:

Chapter 2 provides an extensive survey of previous works related to coordination in swarm robotics. The literature review covers six main directions to identify the research gaps and

evaluates existing solutions. The first part presents a generic review of the state-of-the-art efforts on swarm's formation and coordination control. In the second part, a more detailed survey is conducted on the methodologies used in the literature where homogeneous agents or equal agents are considered. In part three, a detailed investigation is performed on previous approaches and solutions that consider heterogeneous robotic systems. Parts two and three shed light on the non-homogeneous behaviour and coordination of individual members of a group of agents and emphasize the existing research challenges. The fourth part of the literature review presents the division of labour in social colonies such as social insects, which is what many researchers aim to achieve with swarm robotics. Part five covers probabilistic modelling to support the formulation of the proposed specialty-based task allocation approach. Finally, part six sheds light on the literature that discusses possible cooperation between humans and robots.

Chapter 3 details the design aspects of specialty-based task-agent allocation as a cooperative control approach. The research work that is reported in this chapter addresses the non-homogeneity of individuals among a swarm of robots as specialized constraints that govern the automatic process of selecting agents associated to specific tasks.

In Chapter 4, the original probabilistic approach for specialty-based task-agent allocation is presented. In this chapter, the basis for the proposed probabilistic formulation of the tasks' characteristics and the approach for their related agents' assignment are detailed. The original formulation of the individuals' specialization is defined and a novel task-agent probabilistic matching process is implemented. In addition, the original framework is extended to tackle more complex tasks. The developments in this chapter represent the core contribution of this research.

In Chapter 5, the developments of the proposed framework that are introduced in Chapter 4 are investigated under various conditions. A simulator is implemented to confirm the validity of all components of the proposed framework that are introduced in Chapter 4. In addition, the efficiency of the proposed framework is also investigated under different operational conditions. These conditions include when the system deals with the detection of one task at a time, as well as when it deals with the simultaneous detection of multiple tasks which must then be simultaneously distributed to different robotic agents. Results of extensive simulation experiments are presented, discussed, and analyzed in this chapter to demonstrate the reliability of the proposed framework in pursuit of the mission goals. As well, the framework's adaptability to lead specialized teams over small, medium, and wider workspaces is demonstrated.

In Chapter 6, the proposed approach is analyzed in terms of its potential applications. A case study considering a simulated search-and-rescue indoor application is presented. In addition, results from experiments conducted on physical mobile robots are presented in this chapter to demonstrate the effectiveness and evaluate the performance of the proposed task allocation mechanism on a multi-agent robotic team.

Chapter 7 concludes this thesis. It emphasizes the achieved original contributions, as well as proposes future extensions of the proposed framework.

Chapter 2 Literature Review

2.1 Introduction

A multi-agent system is composed of multiple interacting agents that operate with each other within a bounded environment whereas the system behaviour is collectively produced from these interactions.

A well-known approach for the coordination of multi-agent systems that consist of a large number of robotic units is called “swarm robotics”. In other words, the concept of “swarms” from the robotic perspective is a system that creates patterns or some kind of ordered structures. This coordination approach is inspired from the behaviours of social insect colonies such as ants and bees, which stand as examples of how collectively intelligent systems are created because of the interaction between a large number of simple individuals. The concept of swarms upgrades the capability of multi-agent systems to carry out useful and intelligent tasks [1]. The research found in the literature is still improving upon existing multi-robot systems coordination capabilities with the objective to reach the capabilities of social insects’ swarms.

Beni and Wang introduced the term "Swarm Intelligence" to describe the collective behaviour of decentralized, self-organized systems, in the context of cellular robotic systems [2]. The term “swarm robotics” is defined in [3] as the study of:

> “How a large number of relatively simple physically embodied agents can be designed such that a desired collective behaviour emerges from the local interactions among agents and between the agents and the environment”.

The early research interests in multi-agent robotic systems that have a cooperative behaviour were to emphasize various theoretical problems of cooperative robotics, which aimed to study many issues associated with multiple robots such as group architecture, origin of cooperation, and geometric problems. These ideas have been reported in survey papers [4] and [5]. Another essential principle of multi-agent or multi-robot systems coordination is its control architecture. As the robustness of a practical system requires a proper controller, the work in [6] pays a particular attention to control.

While multi-agent systems are capable to solve problems that are difficult or impossible for individual agent systems, as a prominent discipline, swarm robotics or cooperative multi-agent systems have attracted a significant number of researchers to propose solutions to problems and address specific demands from the industry as well. Multi-agent systems have been successfully applied in the real world and many examples have been reported, such as mapping and exploring of the environment, self-organization, distributing antennas and sensors, surveying areas, goal searching, foraging, searching in risky areas, cooperative transport, mining, etc. The following sections present an extensive literature survey on the topic. The focus is on previous studies that analyze sophisticated algorithms for swarms' coordination.

2.2 Coordination of Systems with Multiple Robots

This section focuses on previous art that analyzes general algorithms for multi-agents coordination. Many problems related to multi-agent systems such as cooperative formation and control, path and motion planning, as well as localization and mapping are considered.

Multiple robot systems involve a group of robots that can work cooperatively to accomplish a desired task. The overall performance of a group of robots depends on the level of cooperation, coordination and communication among the robots in the group. These three concepts are essential in multi-robot systems. A *cooperative* robotic system is defined as a group of robotic agents that have a joint operation to perform. A *collaborative* robot is a robot intended to physically interact with humans or with other robotic agents in a shared workspace. *Coordination* is a form of cooperation in which the actions performed by each robotic agent takes into account the actions executed by the other ones. *Communication* is the process that allows the robots to communicate and exchange the data that is required to ensure the interaction between the agents in a proper way to accomplish tasks [7].

2.2.1 Cooperative Coordination of Multi-Robot Systems

The significant benefits emerging from cooperation between the agents of multi-robot systems attracted the interest of many researchers. This subsection presents some of the notable systematic examples from the literature on the cooperation of multiple robots.

An environment boundary estimation algorithm is introduced in [8] to deploy a sensor network based on a group of jointly operated mobile robots to obtain an optimal approximation polygon. Each sensing agent exchanges the information with its neighbours along the boundary and provides information about the tangent and curvature of the boundary. Multiple robots behavioural coordination based on null space approach [9] is proposed in [10] to escort a moving target with a team of mobile robots. Multi-robot cooperation can be taken advantage of to solve the problem of localization and mapping. A solution proposed in [11] combines the localization technology of wireless sensor network (WSN) with multi-robot cooperative localization to ensure a satisfactory

accuracy. Multi-robot systems can also improve the efficiency of task allocation. As such, multi-robot system algorithms to accomplish tasks are addressed in [12] and optimal task allocation patterns are presented in [13]. A complex self-organization approach for a system with multiple robots and based on task partitioning and allocation is proposed in [14].

2.2.2 Coordination Strategies

Multi-robot coordination is controlled either as a centralized system, a decentralized system, or alternatively as a hybrid system, which is a combination of the two prior models. In this section, centralized and decentralized coordination control strategies are considered.

A. Centralized Architectures

In centralized architectures, a central unit controls the coordination of the individual robots in the team. Each robot can also communicate with the robots in the team via the same central unit. This central unit makes decisions regarding the response to faults among the individual agents and monitors task accomplishments. The central unit can be one of the robots or a computer. The control unit of centralized architectures usually faces limitations when controlling large teams of robots. The potential failure of the central unit makes for a highly vulnerable system. At the same time, centralized controlled systems offer many advantages such as the reduction of effort duplication as well as cost savings and reduced time consumption [15]. Many studies discuss centralized architectures, including tasks assignments [16] [17] [18], and motion planning [19] [20]. The details of a central planner to run a multi-robot system for outdoor applications is discussed in [21]. In general, centralized architectures of multi-robot systems are robust to failure of individual agents.

B. Decentralized Architectures

A decentralized multi-robot system involves no coordination by any centralized control unit. The team members can then be controlled individually, and the robots among the team can communicate and talk to each other directly. Contrary to the limitations observed for centralized architectures to control large teams of robots, decentralized architectures offer robustness and scalability. The failure of any agent in a decentralized multi-robot system does not affect the entire system.

A decentralized algorithm for a multi-robot system has been implemented in [22] for tasks allocation. This decentralized approach allows the robots to exchange messages and operate autonomously. Two decentralized task-allocation algorithms are presented in [23] to coordinate a fleet of autonomous robots. Other works based on decentralized approaches for multi-robot systems are proposed that include applications in exploration [24], traffic control [25] and navigation [26].

2.2.3 Cooperative Control for Coordinating Multi-Agent Systems

From a control point of view, different coordination control techniques for multi-robot systems are proposed in the literature, such as potential fields, leader-follower, virtual structure, behaviour-based, and graph theory-based approaches. This section summarizes some of these techniques.

A generic framework is proposed in [27] to control a distribution and group formation behaviour of a multi-robot system considering potential fields. In [28], force equilibrium is analyzed to show how potential fields can be used for different shape generation and formation control. A decentralized control approach based on potential fields for a holonomic multi-robot

system is presented in [29] that allows not only to control robots shape formation, but also to determine their positions.

The formation problem is addressed in [30] based on the principle of leader-follower formation. Leader-follower formation is further investigated in [31], where it is assumed that each robot is equipped with a local sensor and its sensor-based information can be used to control the leader-follower formation. The control problem of leader-follower formation is considered in [32] as a model switching control system. In [33], a framework is implemented for leader-follower formation. A reactive tracking controller is proposed to make each follower maintain its desired position relative to its leader.

A formation approach is presented in [34]. Using generalized coordinates, the agent's location, ordination, and the group's shape are characterized with respect to a reference point. In [35] and [36], the shape formation is also expressed in terms of generalized coordinates. Another formation approach for multi-robot systems is introduced in [37], where the group of robots should maintain a rigid formation based on virtual structure control. The formation control of multi-agent systems is derived in [38] from general non-linear dynamics (servomechanism method). Alternatively, a switching algorithm is proposed in [39] that combines equal agents' formation and reinforcement learning-based obstacle avoidance.

The idea of aggregate motion is modelled in [40] as a computer animation model achieved by a combination of many separate agents that move together. The simulated animation results from the dense combination of each individual agent behaviour. A non-colliding aggregate motion of each agent is simulated and acts as an independent contributor to the total animation. The agents

navigate based on their individual behaviour and on their interaction with the surrounding environment and their groupmates. The approach addresses the problem of multi-robot group formation and collision avoidance.

Multi-robots formation control using a behaviour-based approach is also discussed in [41], which means that the behaviour of each agent in the group is designed such that the proper position of each robot is computed with respect to the other robots location. The positions of the robots in the desired formation are referenced based on one of these techniques: unit centre reference, leader reference and neighbour reference. The desirable group behaviour emerges as a result. In [42], various behaviours are analyzed where the sensory input is used to choose an appropriate behaviour for each robot. Behaviour-based formation approach is also adopted in [43] and a genetic algorithm is used to determine the behaviour control parameters. A dynamic approach to behaviour formation is addressed in [44], where formation control is modeled into nonlinear attractor dynamics. Graph theory is also adopted to control the coordination of multi-robot systems. Control of a group of mobile robots navigating while maintaining a desired formation using graph theory is addressed in [45]. A leader-follower formation control based on graph theory approach is also proposed in [46].

2.2.4 Applications of Multi-Robot Systems

Multi-robot systems have the flexibility, robustness and scalability to deal with tasks that are difficult to be accomplished by a single robot. A team of robots may contribute cooperatively to solve or perform several difficult tasks in an easier way beyond what is possible with an individual robot. Several application areas can benefit from the use of multiple robots, such as exploration,

monitoring, surveillance, hazardous environments, military, agricultural and industrial applications.

Some applications of multi-robot systems have been studied more extensively, for example, for the military sector [47-48]. The main parts of a multi-robot based architecture for landmines detection are developed and described in [49] to support the demining process. The problem of environment monitoring is also targeted by multiple robot systems. An autonomous distributed multi-robot solution for monitoring and confrontation of oil spills is proposed in [50]. Alternatively, monitoring physical phenomena can be of extreme importance when the safety of people is concerned. Maglogiannis [51] presents a multi-robot system forming a robotic sensor network designed to monitor electromagnetic fields.

A cooperative formation of multi-agents is also proposed in [52], to transport an overweight object based on a pusher-puller formation [53]. The problem of cooperation between multiple robots and a human operator is addressed in [54] to carry out an inspection task. Moving target tracking with a group of mobile robots is developed in [55]. Several tasks that can be performed using multi-robot systems such as cooperative localization, mapping, exploration, and object transportation are reported in [56]. Multi-agent systems are also used to execute complicated tasks. In [57], a multi-robot system is developed and equipped with advanced telecommunication technology, to share and distribute knowledge, and to improve the efficiency of manufacturing operation. Similarly, a framework is proposed in [58] for a large class of industrial task scheduling problems. Multi-robot systems are also found in disaster relief applications. In [59], a system is presented as a solution for disaster relief tasks when communication reliability is limited. Multi-agent systems are also applicable in the healthcare field, as exemplified in [60] and [61].

2.3 Multi-Agent Systems Coordination Methodologies

The coordination of a group of mobile robots is widely studied in the literature through rules that govern the local interaction between the robots. The coordination of a fleet of robots can be defined as a collective behaviour [62] that results from interactions between robots. From a control engineering point of view, the behaviour of a group of robots is usually designed to tackle a real-world task or to form desired shapes or behaviours such as aggregations, patterns and chains formation, clustering objects, group navigation and path planning, etc.

2.3.1 Robotic Teams Formed of Individuals with Identical Functionalities

Soysal and Sahin [63] present a gathering model of swarm robots, which are initially distributed into a bounded space. The proposed gathering behaviour is a result of switching between three basic steps, namely random walk, wait and approach. Each step is activated and controls the robots behaviour for a certain duration. Then it expires to allow the second step to start and control the robots for another certain duration. By the end, the final group gathering creates one robot aggregate. Single robot transitions happen among the aggregations based on the assumption that the robots have equivalent transition properties, which depend on the transition probability. The transition probability is constrained by the parameters of the swarm system, which include the sensing characteristics of the robots, the number of robots and the area of the workspace. Their results show that they achieve a significant gathering behaviour, but for equal robotic agents and for missions consisting of a single task objective, that is gathering between the individuals to create one robot aggregate. The authors also mention that their model excludes the spatial positions information.

Shucker and Bennett [64] simulate the coordination of a scalable physical system that deploys a large number of robots as a distributed remote sensing array. The proposed algorithm treats the robots as equal particles to localize them in a specific pattern, creating a hexagonal lattice. The proposed system can be tasked to behave as one system, with robotic members having equal functioning properties, for exploring unknown environments or tracking targets.

In order to address the connectivity problem of cooperative swarm robotics as an autonomous control problem, Hettiarachchi et al. [65] consider local interactions between the robots to propose a physics-based framework. The latter allows the robots to act as a distributed mobile sensor network and maintain a high connectivity during their movement and self-organization. The physical forces drive the robots to contribute equally to reach the desired formation of the mobile sensor network. Each of such connected robots has duplicated task to go through a field of obstacles, or to create chains of sensors.

O'Grady et al. [66] propose a decentralized technique, called SWARMORPH, for controlling a group of robots. The proposed approach is implemented on a real-world multi-robot system. The robots can be physically connected to create stable morphological structures to navigate easily through rugged terrains. However, the proposed approach shows that the autonomous robots are individually operated and demonstrate an equal ability to be physically connected.

A multi-robot system is adopted by Beckers et al. [67] to cluster a large set of randomly distributed objects (pucks) into a single cluster. The authors address three behaviours for the robots based on the assumption that the system activates only one robot at any time. The robot activates its default behaviour to move in a straight line to search for an object. On detecting an obstacle,

the obstacle avoiding behaviour is activated (pucks are not considered as obstacles). When the robot finds an object, a pick-dropping behaviour is triggered to pick it up, then the robot reverses its movement's direction to back to the clustering point to deposit it. Every individual robot is randomly exploring the surrounding environment, finding an object that it might face, picking it up, dropping it off and avoiding the obstacles. All objects are picked-up and deposited equally with a probability proportional to the observed number of objects, and each individual robot behaves as a copy of each other.

Wawerla et al. [68] provide a multi-robot based solution to the construction industry. They address a cooperative construction problem using a behavioural control approach. This approach allows letting the robots build a 2D structure from colored blocks. The robots move within a bounded environment to search for blocks and collect them. The robots pick-up the blocks and put them into place in order to build up a wall. While each robot in the team repeats the same job, each one also checks if any other team member has the same status, meaning that another robot is already placing a block, using inter-robot communication. After that, the block is placed. The proposed system assists multiple robots to search for the blocks and pick them up at random, but the main focus of the work is on improving the task of cooperative construction by using inter-agent communication.

Inspired by the concept of “Virtual Pheromone” where a scent produced by mammals or insects is released to the environment to be used for their communications, Payton et al. [69] propose “Pheromone robotics” to coordinate a large number of robots to navigate towards a common goal or locate a certain resource. Transceivers mounted on each robot corresponds to the idea of a virtual pheromone. The robots are randomly distributed on the field. When the first robot detects the resource of interest, it emits a “virtual pheromone” signal to trigger the swarm that the source is discovered. The emitted signal is distributed by the transceivers between the robots. To establish a path that leads to the source using a line-of-sight communication, each robot directs itself towards the closest robot that received a message, which then serves as a guidepost. The proposed scheme accomplishes coordination activities without centralized control, but one message is diffused to all robots. At the same time, all robots are reacting equally to transmit the message to their neighbours and navigate towards the same source (Figure 2.1).

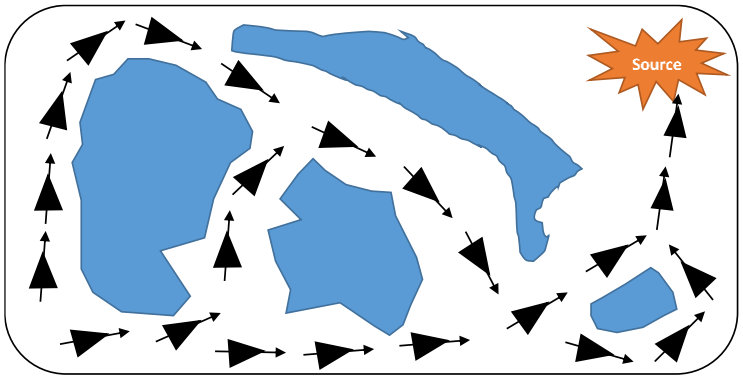


Figure 2.1 Robots distributed throughout field to find the shortest path to a source.

Motion coordination or flocking of multi-robot systems is inspired from animals flocking, which can be a useful approach to improve the abilities of the swarm. A classical potential fields motion coordination approach, for multi-robot exploration, is proposed by Howard et al. [70]. Each robot in such a system serves as a sensor node and is subject to virtual potential forces. These

potential forces are divided into two types: attractive potential forces, which are used to attract the robots toward each others and toward goals or targets; and repulsive potential forces, which are used to repeal the robots from colliding and to avoid obstacles (Figure 2.2). Sabattini et al. [71] also present a decentralized approach based on potential fields to control multi-robot formation. Their proposed strategy is not only to handle robots formation, but also to determine the robots' positions. The proposed potential fields approaches mentioned in [70] and [71] provide a distributed scalable solution to the swarm deployment problem, to control the position of each robot, and to serve as sensor nodes. However, each single agent does the same exploration task and copies the function of each other. A similar approach of using potential functions to control homogeneous multi-robot teams is presented in [72].

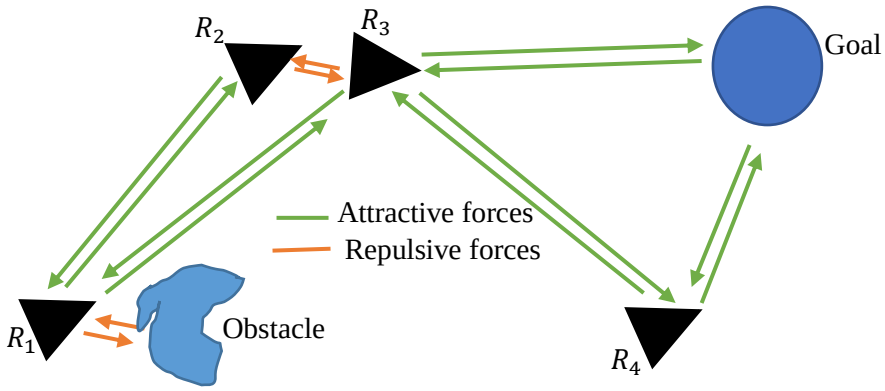


Figure 2.2 Attractive and repulsive virtual potential forces.

Baldassarre et al. [73] present a set of experiments to implement a swarm's flocking technique for a group of robots equipped with various types of sensors, that is infrared sensors to detect obstacles and other robots, microphones to determine the relative position of other robots, and light sensors to identify the targets. The group of robots presents different flocking behaviours. The authors evolve the individual robots' behaviour, which is an important factor that affects the

swarm behaviour, and they display different patterns. However, the group of robots that creates each pattern acts as a single unit.

Turgut et al. [74] report a flocking behaviour method of a swarm of robots. The proposed method uses a digital compass, infrared sensors and wireless communication to estimate the orientation and the distance between neighbours. The digital compass readings can be broadcasted through the communication channel, and the relative headings can be obtained among the group of robots based on the assumption that the North direction is fixed over the neighbouring robots. With the estimated information on robots orientation, which allows each robot to know the heading of each other, and the distance from each other (using IR sensors), the behaviour control of the swarm is designed to create a desired flocking. A motion coordination of a group of mobile robots to efficiently navigate between two targets, without the need for knowledge about their neighbours orientation, is introduced by Stranieri et al. [75]. Ferrante et al. [76] extend this idea by proposing that instead of a control approach based on neighbours' information, motion control can consider only relative distance and bearing angle to the neighbours to execute the attraction–repulsion dynamics to achieve a flocking behaviour. The methods that are presented in [74], [75] and [76] report that a motion controller leads to self-organization or flocking of robotic swarms, but it only provides solutions to multi-robot systems from a flocking coordination point of view, while the coordination of the individual agents creating the desired flocking task is controlled with equal coordination properties for all agents.

A collective transport method is proposed by Ferrante et al. [77] for a group of three robots to collect an object to be transported between two locations. The robots are equipped with different types of sensors to perceive the surrounding environment. The proposed method manipulates the

heading angle regarding the perception of all members of the group towards achieving the collective transport of the object. However, the resulting collective behaviour moves the entire group of three robots together from the start to the goal position, and the main task of the swarm is to transport the target object by the whole group of robots.

Wessnitzer and Melhuish [78] propose a multi-robot behaviour mechanism for a group of robots, which possess minimal sensing capability or short communication range, to enable a swarm of robots, and to decide which one of two targets to be captured first. Two mechanisms are proposed. First, a majority voting approach is adopted to cooperatively decide which target to follow, then the other one is pursued. In the second mechanism, the robots measure the distance towards the target, then they follow the robot that is closest to the target. Finally, the surrounding decision is made based on the spatial distribution of the robots around the target. An emergent cooperative behaviour is achieved, but at the end, the distributed robots converge into one large group to hunt one target, while the distributed individuals are initialized with equal coordination characteristics.

A resource selection mechanism is proposed in [79] to choose the best resource, which can fit the group's needs, or accommodate a group of robots. The individuals make their switching decision, to select the best resource, based on the density probability function of the detected agents that have already reached the resource of interest. The collective group’s behaviour leads the available agents to enter and stay at the same resource. The discrimination is applied to select the best resources, but the proposed mechanism deals with robots as similar agents.

Chamoun [80] proposes a multiple robots system to push an object considering that the object's orientation and its alignment have to be preserved while it is moving along the path. The robots are divided into sub-groups, each of which assigned to a different task such as pushing the object and aligning it with the desired path. The proposed approach associates specific tasks to different sub-groups of robots but it is assumed that the system has a global controller, which has a full knowledge of the object and keeps monitoring its path during the course of the operation. The global controller provides the robots with the required data such as the target point that the robots must move toward, the object orientation and the task to be performed by each robot. As a result, the sub-group of robots perform specific behaviours towards the desired tasks to be achieved, such as moving the object or correcting its orientation. But they do not adapt their allocation to the corresponding tasks based on an allocation scheme or matching cost function.

Ma and Koenig [81] introduce an optimal target assignment and path finding (TAPF) approach for swarms of robots to partition the swarms into groups and to assign each agent among each group to a target, given that each group has the same number of targets. The proposed TAPF approach also plans collision-free paths for the agents between their current locations and corresponding targets, and minimizes the targets' make span. However, it addresses the problem of assigning equal agents to equal targets. Similarly, a software application for task assignment purposes in a heterogeneous swarm is designed in [82]. The proposed algorithm divides the tasks into smaller sub-tasks which are then assigned to an optimal number of agents.

Recently, Scheidler et al. [83] proposed a collective behaviour decision-making method for multi-robot systems or swarms. The proposed method enables the robots among the swarm to select action priority, based on identifying the fastest mean execution time and using the observed

information of the robots state. Initially the swarm is supposed to execute many actions in equal time. The developed procedure breaks this equality and chooses the fastest action based on a stochastic process of the observations. The executed action is the action that owns the fastest execution time out of a set of possible actions, and the proposed procedure increases the accuracy of the decision-making process. However, the authors mention that the collective behaviour leads all of the robots to execute the same action.

Alitappeh et al. [84] also recently addressed multi-agent deployment and redeployment problems in a dynamic environment as an optimization problem. They assume that a fleet of robots is deployed within a partitioned area where the size and location of the regions that should be covered by the robots might change, thus the robots must be redistributed over time. The number and position of objects to be served in each partition are also varying in time; therefore, robots must be redeployed periodically. Each object has a sensor to send its current position to the main system. After updating location information of the objects after each period of time, t, the densities are changed and redeployment can be triggered. A solution to the deployment in the environment is proposed as an optimization problem by considering two objectives to form the optimal cost function: 1) the estimation of final position of the robots after the deployment, and 2) finding the shortest path from the robots initial location to their final position. To perform the optimization they apply an "Elimination and Choice Translating Reality (ELECTRE I)" method [85]. The start position of the robots in the environment is initialized and the configuration of each individual robot in the final deploying positions is given. The classical graph theory concept and optimization algorithms are then used to solve the optimization problem. The proposed algorithms deal with the robots as equal agents, to be assigned to perform repeated tasks in different places.

2.3.2 Robotic Teams with Specialized Individuals

As shown in Section 2.3.1, consistent research has focused on the cooperative coordination and applications of equal agents but, so far, few studies explicitly addressed the problem of specialized agents' missions. This part of the literature review examines recent research concerned with specialization or different individual agents' behaviours in robotic swarms.

O'Hara and Balch present the coordination of a heterogeneous multi-robotic system in [86], which is composed from distributed embedded sensors and a group of larger mobile robots, to explore the environment with the purpose of finding a route, and to lead the group of robots towards a specific target. The group of sensors work together as a wave front to provide the necessary information to plan the path, but the robots are coordinated similarly along the path to perform one task in each mission.

Mathews et al. [87] use a communication link to enable the cooperation between flying and ground robots. The necessary number of ground robots, that can perform an upcoming task, is estimated by a flying robot based on its spatial information. A communication link is established and the flying robots send instructions to ground robots on how to form the morphology or chain, that is appropriate to perform the task. Globally, the proposed system is a heterogeneous robotic system, as it consists of two different types of robots, respectively ground agents and quadrotors. However, the ground robots have equivalent capabilities to create the desired monograph or chain. Similarly, the flying robots are tasked equally to estimate the task type or to identify the morphology of the ground robots.

Sadeghi and Smith [88] propose an optimal task allocation method for a team of heterogeneous mobile robots. The objective of their research is to find an optimal sequence of task-agent allocation to minimize the task time or energy consumption for performing the tasks. In their work, the tasks are defined as points to be visited by robots in the workspace and the robots are heterogeneous with differential motion constraints. The proposed approach addresses the nearest path, between the tasks and the robots, and an optimal method of energy consumption is implemented for the robots to perform the tasks. But all tasks are treated as equal events.

Ferrante et al. [89] propose a novel tool to synthesize a collective behaviour of swarm robots based on the individuals' behaviour, which is called Grammatical Evolution for Swarm Robotics (GESwarm). In this method, a swarm's collective behaviour depends on analyzing the internal states and behaviours of the individual agents. Each robot, R, in the swarm executes a composed set of rules, $Rule_i$, where:

$$Rule_i = \{P_i(preconditions) \times B_i(behaviors) \times A_i(actions)\}.$$

The rule $Rule_i$ can be presented as a conditional statement that is satisfied if all the preconditions of the environment (P_i) are true, and if the robot is executing any of its low-level behaviours contained in (B_i) such as moving in the direction of the light source that has the highest intensity, or performing a directed random walk over the swarm workspace, and if all actions 'control events' contained in A_i can be executed. The GESwarm method takes into account the individuals' behaviour to generate a collective behaviour of the swarm, but it assumes that all individual agents in the swarm execute equal behaviours. This work is extended in [90] to specialize two robots in collecting two objects from their sources and bring them back to specific

nests. The specialty of agents is formulated based on the difference in the environmental conditions. The first robot is specialized to transport an object across a slope environment, whereas the second one is specialized to transport an object across a flat environment, but each one can handle the two behaviours in different times.

The stick-pulling problem [91] is used by Hsieh et al. [92] to investigate the advantages of specialization in the case where two of the cooperative mobile robots are required to execute a cooperative task. This work is extended by Halász et al. [93] to find the collective behaviour optimal configurations, for any number of robots, based on individual adjustment of the agents. The robots are initialized to search for randomly distributed sticks in their vicinity. The stick that can be discovered by the robot might be in one of two statuses: 1) free stick, or 2) held by another robot. The robot deals with the discovered stick based on its status. The behaviour of the robots is summarized in Figure 2.3. The specialization level of the robots does not address any absolute specialty of the individual robots. The provided specialty is defined by assigning two of the robots among the swarm to pick up any discovered stick, hold it, and then release it after a designated period. The robot that discovers the free stick should hold it for a while, while the robot that discovers the held stick should assist the robot that holds it until it releases it. Any one of the robots can hold the free stick or assist the robot that is already holding it. The proposed system is optimized considering the stick holding and releasing times.

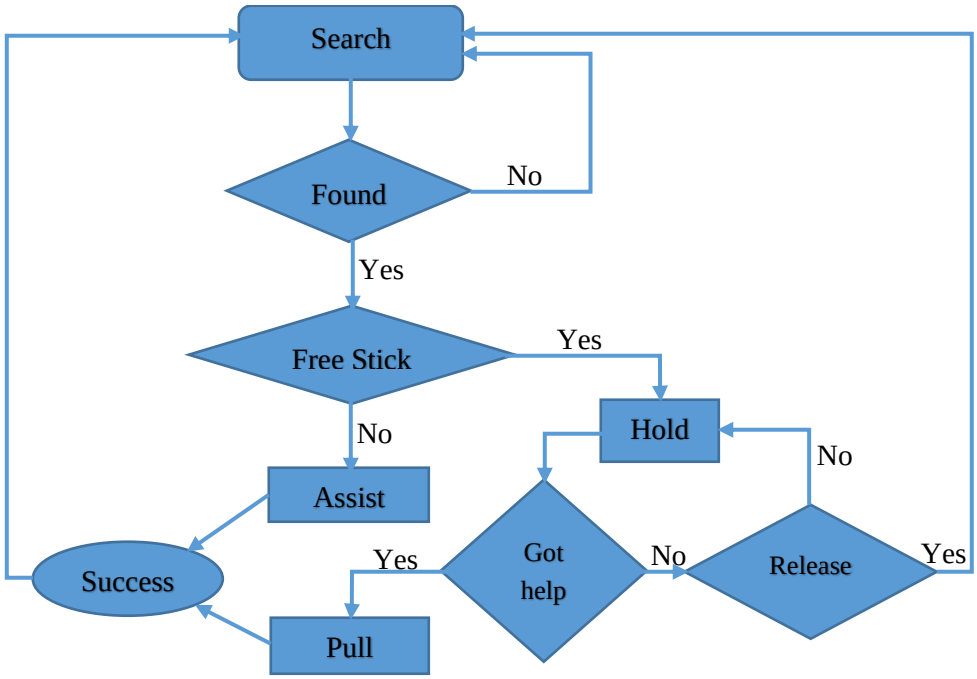


Figure 2.3 Robot behaviour of stick-pulling problem.

Quinn [94] uses a team of two robots to present two approaches, called clonal and aclonal, to evolve motion controllers for coordinating a group of robots. In clonal, all individuals among the group share a same physical construction (homogeneous team). In the aclonal approach, the group consists of different types of robots (heterogeneous team). Simple task allocation performance behaviour of the individuals in the aclonal approach is computed separately based on the fitness of each robot and its task. In the clonal approach, the individuals' homogeneous controller is implemented based on the average fitness of the group. The individual's fitness evaluation is not considered in the clonal approach to find the fitness of each single robot. Based on the assumptions made for the system modelling of these two approaches, the author argues that the aclonal team performs better than the clonal team. The homogeneous controller in the clonal approach [94] is extended by Tuci and Trianni [95] to consider an individual's oriented evaluation function. They simulated two Khepera mini-robots to test a task allocation problem in a simple scenario of nest

patrolling and foraging. The robots remain individually in the nest or move between the nest and foraging source. The results show that the clonal-based design successfully solves the proposed task-allocation scenario and outperforms the aclonal design approach.

Pini et al. [96] propose a task partitioning strategy to split the task of object retrieval between the source and nest into sub-tasks. Initially the robots explore the surrounding space to find the objects. When a robot finds the object, it captures it, and then the robot navigates towards the nest. The distance between the source and the nest segments is based on distance thresholds, each of which is called a partition length. When the robot is holding the object and heading to the nest, it stops at the end of the segment and waits for another robot to hand over the task and pass the object, as shown in Figure 2.4. In order to improve the proposed system, Buchanan et al. [97] overcome the effect of the errors that are induced by sensors or actuators. They propose a dynamical mechanism to let the robots divide the distance between the source and the nest into a varying number of partitions based on the success and failure probabilities of the robot's object exploration. The proposed mechanisms [96-97] partition the tasks among a robotic swarm but the tasks are partitioned between the individuals as a consequence of repeated distance-based subtasks.

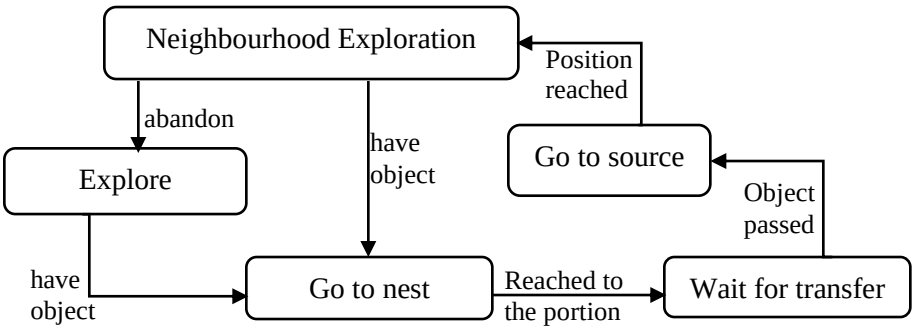


Figure 2.4 Robot's behaviour based on employing a partitioning strategy.

Behavioural specialization [98] in a multi-robot system can emerge as the result of interactions among the robots and with the environment. In the domain of multi-robot systems, Montanier et al. [99] focus on behaviour specialization as behaviourally heterogeneous systems that show potential in a number of tasks. Their experiments introduce two constrained resources. The first set of robots can only feed from one specific resource while the other set feeds from the other resource. The members of each group are selected based on a designated cost function, which emerges from the energy function of the agents and the availability of the resources. This work addresses the specialization as involving the robots into similar groups, each group having a specific specialty and employing many agents that share an equal behaviour.

Gigliotta et al. [100] propose a communicative interaction-based technique to address the problem of specialization among the individual agents as a dynamic role allocation in the case where each single robot can take one of two different roles. The individuals are evolved to distinguish themselves from each other using different communication signals. The proposed system evolves the groups of equal agents to become different or specialized groups, whereas each individual agent differentiates itself and identifies its role through the differentiation of the signal that they can send. Then, each agent can be categorized to which group it belongs to. Each group has its specialized role. For example, some of the robots maximize the power level of their communicative signals, while the others should minimize the power level of their signals. As a result, the team of robots is categorized into two specialized groups based on the power level of the output communicative signal. This work is extended twice. First, Gigliotta [101] evolves the ability of the communicative system. The author focuses on fostering a successful role allocation strategy. The efficiency of communication based signaling differentiation is investigated in this

work with respect to how this technique is essential to differentiate the tasks in a dual patrolling or role allocation system. Later on in [102], this work is further extended to address the efficiency of the originally proposed system [100] using a neural network controller with lower computational power. This work categorizes the individuals into specialized groups and the fitness function of specialization has varying values, which depend on the environmental conditions.

Zavlanos et al. [103] propose an auction algorithm that allows every agent to independently determine a task to be assigned to. The agents can bid for the tasks to which they wish to be assigned to. However, the task-agent assignment cost is a function of the distance that the agents need to travel to the tasks, the time and power that are required to accomplish that task. The proposed actuation algorithm treats the agents equally and assigns the individuals based on the local information without considering any specialized constraints, neither in the agents nor in the tasks. An auction method of task-agent allocation for multi-agent systems is also proposed in [104] based on the well-known Contract Net Protocol. The auctioneer agent explores the workspace and publishes announcements for the tasks. The announcement message contains the individual tasks details. These messages are circulated among the group of agents and evaluated by each individual to assign the capable agent to perform the corresponding task. This approach assigns the applicable robots based on minimized distance cost to perform the current task but the cooperation is at the level of the task-agent assignment only, while the individuals cannot operate as a cooperative swarm at the task execution level. This approach only enables cooperation between the auctioneer agent and the capable agent to respond to the current task, and then the agent undertakes the execution process as a single agent system.

2.4 Labour Division

The division (specialization) of labour is the separation of tasks in any cooperative system, or the participant's specialization in narrow tasks. In manufacturing processes, the specialization of labour is meant to assign different tasks to different specialized people. Division of labour or task partitioning is one of the essential features of sociality; the division of labour has been extensively studied in biological science as observations of social insects' activities [105]. Dividing the tasks benefits the social insects in several ways and facilitates the team of insects to become either specialized or heterogeneous workers [106]. As a result of task partitioning between the individuals, the swarm can be more efficient [105]. Many studies quantify the division of labour as a statistical pattern across social groups [107].

Biological research studies found that the division of labour in social insects, based on different response thresholds such as spatial location and individuals' matching sizes, allows the individuals to segregate themselves amongst different tasks. With increased task specialization, it often leads to increase the group efficiency [108-109]. The division of labour increases the organism efficiency because it can perform many physiological processes simultaneously [109]. Goldsby et al. [108] study the division of labour by evolving digital organisms; the efficiency of division of labour is analyzed as a cost function of task-partitioning behaviours of the individuals or specialized groups. They found that the groups of organisms are increasingly evolving the division of labour strategies as the cost function of the task partitioning is raised. Rueffler et al. [110] predict that specialization for different tasks by different organs or cells can be beneficial if the effectiveness of the task performance can be accelerated by employing different specialized units.

2.5 Representation of Task-Agent Matching

The concept of allocating the individual agents of a multi-agent system to given tasks is addressed extensively in the literature. Previous research works introduced many robot-task assignment or matching solutions in many ways and matching algorithms have been proposed such as maximum matching algorithms [111] that match equal number of vehicles and tasks. These algorithms simply assign one vehicle to one task. Alternatively, the perfect matching problem [112] involves having a convention to map a set of n robots to a set of m tasks. Task-agent assignment is addressed in [111] and [112] without considering task-agent specialty.

The rest of this section covers the literature that focuses on a stochastic or probability-based representation of task-agent assignment. Jones and Mataric [113] build a state transition probabilistic model to respond to changed tasks. Two agents perform a foraging task with equal probabilities. These agents keep their current foraging state and observe the vicinity environment. The current foraging state is re-evaluated based on new observations, and the probability of the foraging state priority of the robot changes with respect to the observed foraging task. Then, based on the current probability, the robot can change its foraging state. For example, if the current foraging state of a robot is RED and an observed Puck (current task) is RED, then the robot can grasp this Puck. However, if the current foraging state of the robot is RED and the observed Puck (current task) is Green, then the robot should change its current foraging state to Green. After that, it can grasp the observed Puck.

Smith and Bullo [114] propose a task-agent assignment probabilistic algorithm called “grid assignment algorithm” to partition the targets environment to a grid of cells. Then the available

robots in each cell are assigned to the targets that occupy the same cell. They analyze the time complexity bounds of assigning uniformly distributed agents and targets. The completion time upper bound with higher probability is addressed based on the diameter of the environment. They found that the completion time of task-agent matching reached its highest probability when all of the available agents are assigned to the available targets. The task-agent assignments are formulated in their simpler manner, without considering any specialized matching features.

Claes et al. [115] address the spatial task assignment problem as a multi-agent planning problem using the Markov decision process. The proposed model aggregates the effect of the other team members into a probabilistic model to control the individual agents that are trying to perform spatially distributed tasks. From the multi-agent planning perspective, the individual robots need to know the performed tasks, which are already served by the other robots, to plan the individuals' response. The main objective of the assignment process is to plan the behaviour of a swarm or a multi-agent system in order to ensure that all spatial tasks can be serviced. However, this proposed solution is not applicable when a distinct agent must serve a specific task, and it is impossible to coordinate particular agents to specific spatial tasks.

The architecture of a constrained tasks model for autonomous robots, to be acting in a human environment, is presented in [116] for a single household service robot. This model couples the probabilistic knowledge of task parameters with the robot controllers. The task's parameters are represented based on people heuristics such as their activity in the kitchen, the positions of the individual seats around the table, as well as meals time. The robot controller is implemented as an integration of a high-level reasoner, which manages the probabilistic information, to control a

simple path planning of the household robot. The proposed models are applicable only for a single household service robot with an accurate knowledge of the surrounding environment.

Lang and Toussaint [117] introduce a probabilistic model to define object-action relevance. The proposed model involves only a subset of objects that are relevant for specific planning purposes but it is not reliable enough to estimate the features of the objects for a convenient matching process. The model that is presented in [117] is used in [118] to compute a sequence of actions and apply approximate inference to control the robot planning, grasping and reasoning for the arrangement of table-top objects.

Yasuda et al. [119] propose a response threshold probabilistic model to control the individuals that perform a food foraging process. As a result, the robots that have a probability that exceeds a specific threshold can leave the nest. Then they search for food. This paper investigates the effect of the individual's response threshold, which allows the robots to divide the labours among the swarm. However, the individuals can only perform two different tasks, nesting and searching.

Recently, Wu et al. [120] introduced a task assignment probabilistic model based on an environmental stimulus and the agent's response threshold. The environmental stimulus is modeled based on a specific increment at each time, t, and decrement with every active agent. The proposed model is implemented to allow the agents to react individually with categories of tasks. Then the individual agents can carry out different types of performance based on the threshold level. This threshold level is increased when the total number of similar agents is decreased. The system is modeled for battle field attacks and the transition probability can only transfer the individual agents among the swarm between two task state probabilities for the entire targets,

which are the inactive and reconnaissance/attack states. As such, the swarm members are specialized to be reconnaissance agents or attack agents. The stimulus from the environment is used to categorize the available agents based on their specialities, but it neither matches nor takes the decision to make the specialized agent interact with the current tasks.

2.6 Human on the Loop

Based on real-world experiences with automated swarms, many studies suggest that a human operator should stay in the operational loop and supervise the automated process to avoid causing problems to its components [121]. As well, human supervision should remain involved on the control loop of automated systems to evaluate, interpret the data, reason about it and coordinate the automated process at the decision stage, and to improve the process situational awareness as general solutions move toward smarter autonomous systems [48].

The literature divides the decision making process into three types, whether decisions are taken by human, by an intelligent system, or both. Systems that have a high-end decision component, which involves a human operator and an intelligent layer to share the decision making process, are called mixed-initiative systems. The flexibility of mixed-initiative systems depends on the intervention level of the operator, the available cognitive resources, and critical conditions of the work environment [122].

Valero-Gomez et al. [123] found that the operational effectiveness of an adjustable automated process, in which a human operator can intervene to reconfigure critical events, is improved over that of fixed automated processes. It is also demonstrated that high workload missions present an improved performance with adjustable approaches compared to fixed adaptive systems [124-125].

2.7 Summary and Research Gaps

This chapter presented a literature study of previous works related to swarms' coordination from the perspective of task allocation. The first part (section 2.2) offered a generic overview of swarms' coordination by reviewing general algorithms and summarizing the main concepts of multi-agent coordination. Emphasis was placed on the coordination problems of multi-agent systems such as cooperative formation and control. In addition, multi-robot coordination strategies such as centralized and decentralized coordination approaches were reviewed. As well, several applications of swarm robotics were presented.

To identify research gaps, the second part of the literature (section 2.3) focused on homogeneous and heterogeneous swarm agents' coordination methodologies, which represent the vast majority of the literature on swarm robotics. The study introduced details and provided greater depth into the coordination methodologies. A homogeneous system is defined as a group of agents that have equal kinematics, sensing, and actuation capabilities to achieve certain mission goals. As such, the reviewed methodologies show that the proposed approaches do not pay attention to which agent performs a given action. While in some applications the system values how many agents perform the task, for optimization purposes, the individual agents are initialized with equal coordination probabilities. On the other hand, heterogeneous systems are defined as a group of agents that have different kinematics and/or sensing and/or actuation capabilities; that are meant to achieve different goals, or that are designed to perform different actions.

More attention was given, in section 2.3.2, to studies that have explicitly addressed the problem of specialized agents' behaviours among robotic swarms, which represents the core research

direction of this thesis. Existing solutions often consider the individual agents specialized interactions with the surrounding environments. However, the considered specialty level does not generally achieve a specialization level that allows for agents to divide the labour among the individuals based on the specialized functionalities embedded in each individual agent or the specific characteristics of the given tasks. In spite of the specialization factor that is considered in this part of the literature, robots are usually coordinated similarly [84], have equal capabilities [86], intervene with the tasks as equal events [87], have equal behaviours to allocate robots to the tasks [88-89] [98], or the cooperation between agents remains at a limited level [103].

The third part of the literature study examined specialization among social insects. In section 2.4, biological studies show that the division of labour facilitates the members of a team of insects to become specialized workers. It is observed that partitioning the work between the individuals based on their specialized functionalities results in adaptive swarms that lead to more efficiency. Many studies have quantified the significant impact of the division of labour among cooperative swarms as a statistical pattern across social groups. This sheds light on a research gap, which is the labour division within multiple-agent systems. That is, one important objective that must be achieved in swarm robotics. For this to happen with some form of optimality, the agents need to be matched with tasks such that a certain form of specialized labour division emerges in robotic swarms.

Finally, to support the development of solutions to the problem considered, part four of the literature review (section 2.5) focused on relevant representations for task-agent assignment using probabilistic formulations. This opens the door to probabilistic modelling directions to perform task-agent matching, with the goal to also take advantage of the collaboration between human and

intelligent robots to strengthen the decision-making process. For the matter, the last section of this literature survey (section 2.6) reviewed several studies that shed light on the role of human beings involved on the control loop of automated processes.

As a conclusion, a significant technological gap among existing solutions for swarm robotics is identified, which is the consideration of formal specialization of the individual agents among a group of robots based on their respective embedded functionalities. In order to take advantage of cooperation in a robotic swarm consisting of agents that possess specialized functionalities, a formal probabilistic mechanism for task-agent matching is designed in this research. The proposed framework encodes robots specialization and achieves efficient strategic allocation of specialized agents to specific tasks as a form of collaboration in the swarm.

The research that is discussed in the remaining chapters is conducted to implement a framework that addresses a specialized task-agent assignment approach for a swarm of robots working as a cooperative team. The proposed framework is presented following a progressive research and design methodology, providing a comprehensive study that covers the swarm's cooperative formation and the agents' dynamic control. Primarily, the focus is given to specialized task exploration, estimation, and matching of the specialized robotic individuals with tasks that are dynamically detected and recognized in the environment of the robotic swarm. The approach matches the tasks' specific requirements with the agents' specialized predefined capabilities. The original task-agent specialty-matching scheme is proposed and developed in a probabilistic form with different matching paradigms, which forms the core contribution of this thesis.

Chapter 3 Specialty-based Task-Agent Allocation – Cooperative Control Approach

3.1 Introduction

In this research, the individual agents are considered specialized with respect to each individual's functionalities. In other words, the agents' specialized capabilities form a level of sophistication to be emphasized and directly leveraged in this study. In this chapter, the findings from the early phase of research are presented. They contribute to fill research gaps inferred from the analysis of the literature in Chapter 2, namely the fact that specialization of individual agents, as a definition of non-homogeneity of a robotic swarm, has not yet been extensively investigated. The proposed approach addresses the allocation of specialized individuals in response to corresponding tasks, as a cooperative formation control problem. The research examines the general design of multi-agent systems involving specialized agents, each possessing different and specific capabilities. It extends the concept of cooperative formation found in the literature and proposes a rigorous process taking into account such specialized agents to dispatch the robotic resources in a given operational scenario.

In this chapter, the proposed approach is presented in two phases of development. In the first phase, the concept of specialization is addressed as a zone-based specialization assignment. In response to a new specific task, the swarm of robots changes its current formation and adopts a new one. The swarm's reformation is controlled by subdividing the plane of movement into different zones based on the assumption that each zone involves a specialized task to be performed by a specific robotic agent. As such, the robots adapt their new formation in accordance with the swarm's leading task as soon as they hit a new zone. The leader robot is assigned as the agent that

best responds to the new task requirements, as discussed in Section 3.4. The approach defines how the formation is managed in each zone and how the formation transition happens from one zone to the other. The first research problem that is addressed studies the transition strategy of the swarm formation for a group of cooperative mobile robots. The key idea that is formulated makes the swarm switch to a new formation in order to assign the specialized agent to handle a corresponding specific task.

Next, the proposed approach for the swarm's coordination and task-agent specialty-based assignment is extended by evolving the definition of specialization to match with target recognition, such as detecting specific landmarks via embedded sensors. In this phase of the research, the specialization zones are refined as circular areas, called zones of influence, around each detected target corresponding to a specific task meant to be assigned to a specialized agent. A specialized task-agent assignment approach is also formulated in an algorithmic manner that evolves the agents' specialization to match with target recognition. In addition, an Automatic Task Selection Unit (ATSU) is introduced and implemented to operate in two possible modes: either under close supervision of a human operator or in a more automated way. Both modes perform each specialized task in three phases that ensure target search, target detection and recognition via virtual embedded sensors, and finally the execution of a given task that is compatible with the resources available on a selected agent. The selected agent also serves as the specialized swarm's leader for the duration of a specialized assignment, as explained in Section 3.5.

For the sake of system's validation, the proposed approach was modeled for a swarm of ground robotic agents and simulation experiments were conducted. The proposed system is modeled using nonholonomic mobile robot dynamics. A kinematic and torque controller for nonholonomic

mobile robots is adapted to control path following and swarm formation. Collision avoidance is also introduced using repulsive potential forces to prevent the agents from colliding during formation transition and to control inter-agents separation. Simulation results are presented which show that the group of robots effectively coordinate themselves within each zone of influence in a desired formation. Coordination involves selecting a new robot to be the specialized leader while ensuring smooth switching of the swarm formation from zone to zone without collisions.

In Chapter 4, the concept of task-agent specialty assignment will be further developed in the specialization space. The framework will be refined with an expanded formulation of the proposed task-agent assignment approach from a probabilistic perspective, which represents the core contribution of this thesis.

3.2 System Modelling and Assumptions

In general, the proposed approach solves the problem of specialized task-agent assignment as a cooperative formation problem upon the recognition of task specific constraints. The specialization of each agent can be related to the type of unmanned system considered (UGV, UAV, USV), or take the form of particular embedded sensors, actuators, computational power, or communication links. The main concept is to make each robot best prepared or equipped to respond to a certain range of tasks, such as exploring, collecting, holding, tracking, etc. For example, if search-and-rescue scenarios are considered, the task could be to search for a person who disappeared in a forest or to search for a specific object that might be placed there, whereas the forest contains lakes and different land terrains. In this case, the swarm will consist of specialized agents that have different capabilities (e.g. agents able to function on specific terrains

and others on water-covered areas). The agents' functionality is also specialized toward agents that have specific capabilities to rescue humans and others with different capabilities to grasp objects of specific types.

For validation of the proposed approach in simulation, the agents are assumed to be homogeneous at the level of the agents' construction and heterogeneous at the level of the agents' functionality. In this chapter, a model of nonholonomic mobile robot with an n dimensional configuration space subjected to m constraints and a two-dimensional coordinates workspace, $D = 2$, is considered. The coordinates configuration of the robot is described in the global coordinates configuration by q, and the coordinates configuration of a swarm consists of a number, a, of agents defined by $\{q_1, q_2, \dots, q_a\}$. The Euler Lagrange form [126] of the robot model can be expressed as:

$$\mathrm{M}(q)\ddot{q} + V_m(q,\dot{q})\dot{q} + F(\dot{q}) + G(q) + \tau_d = B(q)\tau - A^T(q)\mathcal{E} \tag{3.1}$$

where $M(q) \in \mathcal{R}^{n\times n}$ is a symmetric positive definite inertia matrix; $V_m(q,\dot{q}) \in \mathcal{R}^{n\times n}$ is a centripetal and coriolies matrix; $F(\dot{q}) \in \mathcal{R}^n$ denotes the surface friction; $G(q) \in \mathcal{R}^n$ refers to the gravitational vector, and $\tau_d \in \mathcal{R}^n$ denotes unknown disturbance. $B(q) \in \mathcal{R}^{n\times r}$ is the input transformation matrix (r is the number of control signals); $\tau \in \mathcal{R}^r$ is the input vector; and $A^T(q) \in \mathcal{R}^{m\times n}$ is a matrix associated with constraints. Finally, $\mathcal{E} \in \mathcal{R}^m$ is a constraint forces vector. If we consider that all kinematic equality constraints are not time-dependent [126], it follows that:

$$A(q)\dot{q} = 0 \tag{3.2}$$

Let $\varrho(q)$ be a full rank matrix of a set of smooth and linearly independent vector fields in the null space of $A(q)$, i.e:

$$\varrho^T(q)A^T(q) = 0 \tag{3.3}$$

If we consider (3.2) and (3.3), it will be possible to find a vector time function $\mathrm{v}(t) \in \mathcal{R}^{n-m}$ such that:

$$\dot{q} = \varrho(q)\mathrm{v(t)} \tag{3.4}$$

The position of the robot shown in Figure 3.1 in an inertial Cartesian frame $\{\mathrm{O}, \mathrm{X}, \mathrm{Y}\}$ can be represented by the vector $q = [x, y, \theta]^T$. The motion kinematic relation, (3.4), can be represented in terms of translational, v, and angular, ω, velocities as:

$$\varrho(q) = \begin{bmatrix} cos(\theta) & -d_r sin(\theta) \\ sin(\theta) & d_r cos(\theta) \\ 0 & 1 \end{bmatrix} \tag{3.5}$$

$$\mathrm{v} = \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{3.6}$$

$$\dot{q} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} cos(\theta) & -d_r sin(\theta) \\ sin(\theta) & d_r cos(\theta) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{3.7}$$

where d_r is the distance between the centrepoint of the robot and the back axel defined in Figure 3.1.

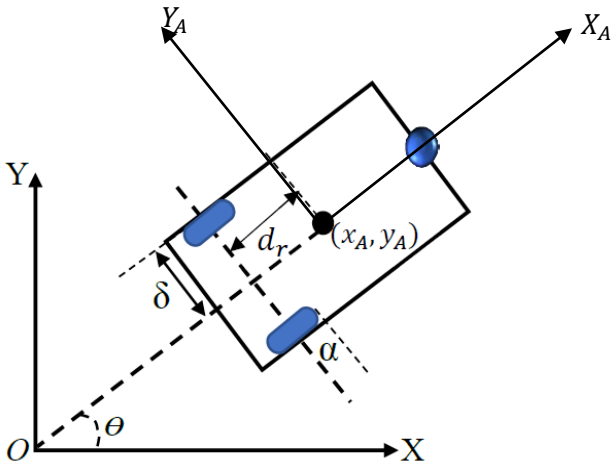


Figure 3.1 Coordinates of mobile robot in inertial global Cartesian frame $\{\mathrm{O}, X, Y\}$.

The matrices that define the model dynamics based on Figure 3.1 are expressed in equation (3.1) above where,

$$M(q) = \begin{bmatrix} m_a & 0 & m_a d_r \sin(\theta) \\ 0 & m_a & -m_a d_r \cos(\theta) \\ m_a d_r \sin(\theta) & -m_a d_r \cos(\theta) & I \end{bmatrix} \tag{3.8}$$

$$V(q, \dot{q}) = \begin{bmatrix} -m_a d_r \dot{\theta}^2 cos(\theta) \\ -m_a d_r \dot{\theta}^2 sin(\theta) \\ 0 \end{bmatrix} \tag{3.9}$$

$$G(q) = 0 \tag{3.10}$$

$$B(q) = \frac{1}{\alpha} \begin{bmatrix} cos(\theta) & cos(\theta) \\ sin(\theta) & sin(\theta) \\ \delta & -\delta \end{bmatrix} \tag{3.11}$$

$$\tau = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \tag{3.12}$$

where m_a is the mass of robotic platform. It is also assumed that the gravity impact on the robots' 2-D motion that moves on the ground equals zero, from equation (3.10). The system represented by (3.1) can be transformed to be more appropriate for control consideration. Equation (3.4) can be differentiated to obtain $\ddot{q}$ and substituted into equation (3.1). The final motion equations of the nonholonomic mobile robot is given as:

$$\varrho^T M \varrho \dot{\mathrm{v}} + \varrho^T (M \dot{\varrho} + V_m \varrho) \mathrm{v} + \bar{F} + \bar{\tau}_d = \varrho^T B \tau \tag{3.13}$$

where $\mathrm{v}(t) \in \mathcal{R}^{n-m}$ is a vector of the translational and angular velocities. Equation (3.13) can be rewritten as:

$$\bar{M}(q) \dot{\mathrm{v}} + \bar{V}_m(q, \dot{q}) \mathrm{v} + \bar{F}(\mathrm{v}) + \bar{\tau}_d = \bar{B} \tau \tag{3.14}$$

$$\bar{\tau} = \bar{B} \tau \tag{3.15}$$

where $\bar{M}(q) \in \mathcal{R}^{r\times r}$ is a symmetric positive definite inertia matrix; $\bar{V}_m(q,\dot{q}) \in \mathcal{R}^{r\times r}$ is the centripetal and coriolies matrix; $\bar{F}(v) \in \mathcal{R}^{r\times 1}$ is the surface friction; $\bar{\tau}_d$ refers to unknown disturbances and $\bar{\tau} \in \mathcal{R}^{r\times 1}$ is the input vector.

Equation (3.14) describes the system behaviour in the vehicle coordinates frame, (X_A, Y_A). This means that $\varrho(q)$ is a transformation matrix which relates the velocities 'v' that are represented in the vehicle coordinates to Cartesian coordinates velocities, $\dot{q}$. Finally, the dynamics of the i^{th} robot system should satisfy the following assumptions [126]:

1. ***Boundedness***:

   $\bar{M}_i(q)$, the norm of $\bar{V}_{m_i}(q,\dot{q})$, and $\bar{\tau}_d$ are bounded.

2. ***Skew Symmetric***:

   The matrix $\bar{M}_i(q) - 2\bar{V}_{m_i}(q,\dot{q})$ is skew symmetric such that:

$$x^T\left(\bar{M}_i(q) - 2\bar{V}_{m_i}(q,\dot{q})\right)x = 0 \tag{3.16}$$

3.3 Formation Control Design

The proposed system combines two cascaded stages. The first stage controls the robot dynamics system and the second one controls the formation. The schematic structure of the proposed system is presented in Figure 3.2.

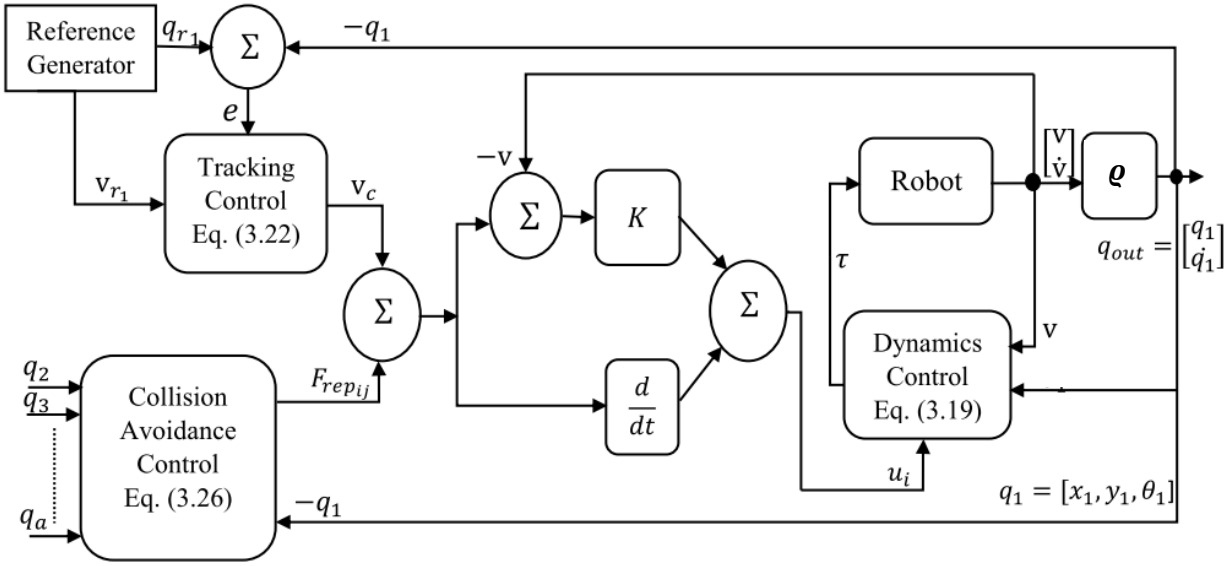


Figure 3.2 Schematic diagram of the proposed formation controller.

3.3.1 Dynamics Control

A method for controlling the dynamics nonlinearities of the nonholonomic mobile robot is implemented based on the output feedback. In particular, feedback linearization [127] is used to cancel the nonlinear part of the vehicle dynamics. The resulting system is an adequate linearized system for both navigation and group formation. In the sake of modelling simplicity it is assumed that the surface friction, $\bar{F}(\mathrm{v}) = 0$, and the unknown disturbances, $\bar{\tau}_d = 0$. The robot dynamics can be linearized as follows:

$$\dot{\mathrm{v}} = \bar{M}^{-1}(-\bar{V}_m(q, \dot{q})\mathrm{v} + \bar{B}\tau) \tag{3.17}$$

$$\dot{\mathrm{v}} = -\bar{M}^{-1}\bar{V}_m(q, \dot{q})\mathrm{v} + \bar{M}^{-1}\bar{B}\tau \tag{3.18}$$

Then using feedback linearization (FBL), one can choose the input torque, τ, to be:

$$\tau = (\bar{M}^{-1}\bar{B})^{-1}(u_i + \bar{M}^{-1}\bar{V}_m(q, \dot{q})\mathrm{v}) \tag{3.19}$$

where u_i is the proposed control signal which controls the whole system dynamics behaviour in the vehicle local frame, as shown in Figure 3.2. Substituting (3.19) in (3.18) leads to

$$\dot{\mathrm{v}} = u_i \tag{3.20}$$

This means that the translational and the angular velocity of the robot can be controlled directly by the proposed control signal u_i. The $\dot{\mathrm{v}}$ dynamic in (3.20) is in the robot's body frame representation (i.e. its local frame (X_A, Y_A) as shown in Fig. 3.1).

3.3.2 Steering Control

The resulting robot's velocity, v, which is the integration of the output signal $\dot{\mathrm{v}}$, is used to contribute in robots' torque control u_i (as shown in Fig. 3.2). More details are given below. For that purpose, (3.20) satisfies the desired behaviour such that the robot's deriving velocity, v, can be converted to a control signal (input torque).

For the system to track a reference trajectory resulting from path planning, it is assumed that a full knowledge of the system dynamics is available. Therefore (3.19) can be used to calculate the input torque, τ, from a given control signal, u_i, which means that the control signals, u_i and τ, can be derived from a tracking control signal, $\mathrm{v_c}$, in order to control the linearized steering system (3.20). In this control design, back-stepping control is used.

$$u_i = \dot{\mathrm{v}}_c + K(\mathrm{v_c} - \mathrm{v}) \tag{3.21}$$

where K is a positive definite diagonal matrix and

$$\mathrm{v}_c = \begin{bmatrix} \mathrm{v}_r \cos e_3 + k_1 e_1 \\ \omega_r + k_2 \mathrm{v}_r e_2 + k_3 \mathrm{v}_r sin\ e_3 \end{bmatrix} \tag{3.22}$$

$$\mathrm{v}_r = [v_r\ \omega_r]^T \tag{3.23}$$

$$q_r = [x_r\ y_r\ \theta_r]^T \tag{3.24}$$

with
$$\begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} cos\theta & sin\theta & 0 \\ -sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_r - x \\ y_r - y \\ \theta_r - \theta \end{bmatrix} \tag{3.25}$$

and v_r is the reference velocity, q_r is a given arbitrary configuration, and e is the tracking error.

3.3.3 Inter-Agent Collision Avoidance

In order to achieve the ability to navigate safely and switch to a new formation without any inter-agent collisions, a potential field based control technique [29] is used. A repulsive potential force controller is developed to generate internal repulsive forces between the robots to keep a minimum distance, *L*, to separate them from collision. The collision avoidance control along with the steering control and dynamics controllers work together to keep the swarm members in a collision-free formation during navigation and formation transitions preserving the predefined separation, *L*, between the robots. The repulsive potential between every two neighbouring agents i, j, can be expressed as:

$$F_{rep_{ij}} = -\nabla_{P_i} \mathrm{V}_{ij}(P_i, P_j) \tag{3.26}$$

$$\nabla_{P_i} = \frac{\partial}{\partial P_i},\ P_i = [x_i\ , y_i\ ]^T,\ P_j = \left[x_j, y_j\right]^T \tag{3.27}$$

Define: $\vec{\ell}_{ij}(t)$ as the vector relating the Euclidean distance between agents i and j with $\|\vec{\ell}_{ij}\| = \ell_{ij}$, we have $\ell_{ij} = -\ell_{ji}$. Let, $\ell_{ij}(t) = \|P_i(t) - P_j(t)\|,\ i = 1,2,\dots\dots,a;\ j = 1,2,\dots,a;$ and $i \neq j$; where a is the total number of swarm members. P_i and P_j are the coordinates of the robots $\boldsymbol{R}_i$ and $\boldsymbol{R}_j$ respectively.

$$\mathrm{V}_{ij} = \begin{cases} \frac{1}{2}K_a(\ell_{ij} - L)^2, & \ell_{ij} < L \\ 0, & Otherwise. \end{cases} \tag{3.28}$$

where L is the minimum desired separation between each two agents, and V_{ij} is continuously differentiable. If $\ell_{ij} < L$, this potential produces a repulsive force, and it will produce a null force if $\ell_{ij} \geq L$. Now suppose that:

$$\ell_{ij} = \sqrt{\left(P_i(t) - P_j(t)\right)^T \left(P_j(t) - P_i(t)\right)};\ \ell_{ji} = \sqrt{\left(P_j(t) - P_i(t)\right)^T \left(P_i(t) - P_j(t)\right)}$$

so,

$$\frac{\partial \ell_{ij}}{\partial (P_i, P_j)} = \left[\frac{1}{\ell_{ij}}\left(P_i(t) - P_j(t)\right)^T + \frac{1}{\ell_{ji}}\left(P_j(t) - P_i(t)\right)^T\right] \tag{3.29}$$

$$\frac{\partial \mathrm{V}_{ij}}{\partial \ell_{ij}} = K_a\left(\ell_{ij} - L\right) \tag{3.30}$$

using the chain rule we get,

$$\frac{\partial \mathrm{V}_{ij}}{\partial (P_i, P_j)} = \frac{\partial \mathrm{V}_{ij}}{\partial \ell_{ij}} \frac{\partial \ell_{ij}}{\partial (P_i, P_j)} \tag{3.31}$$

substituting (3.29), (3.30) in (3.31) leads to the inter-agent repulsive potential to be given by

$$F_{rep_{ij}} = -K_a(\ell_{ij} - L)\frac{1}{\ell_{ij}}\left[\left(P_i(t) - P_j(t)\right)^T - \left(P_j(t) - P_i(t)\right)^T\right] \quad (3.32)$$

Then u_i in (3.21) will change as illustrated in Figure (3.2) to

$$u_i = \dot{\mathrm{v}}_c + \dot{F}_{rep_{ij}} + K\left(\mathrm{v_c} + F_{rep_{ij}} - \mathrm{v}\right) \quad (3.33)$$

where u_i is the proposed swarm's formation control signal.

3.4 Zone-Based Specialization Assignment

In a first exploratory phase of the task allocation problem with specialized robotic agents, the proposed formation transition strategy for a multi-robot system that involves specialized agents assumes that specialized zones of operation subdivide the robots' workspace and are interconnected in a way that the robots can change their formation when transferring from zone to zone. This formulation generalizes the definition of specialized tasks associated with various parts of an operation to which specific agents, or robots, would be assigned based on their on-board sensing or tooling devices. In this context, the problem of transitioning the swarm between different formations as a response to different tasks is addressed in the following subsections.

3.4.1 Formation Transition

The swarm formation is monitored with respect to a global reference frame (X, Y) in order to create a global representation of the formation and to preserve a stable formation with all robots. The swarm starts searching the workspace in a stable formation scheme. The approach builds upon the assumptions that the workspace is divided into specialized zones, each of which involving a specific task, and that a dedicated swarm's member is supposed to be the most competent leader for the swarm while this task is performed. Then, when the swarm hits a new task zone, the system

must assign another specialized member to become the leader of the swarm. At that point, the swarm switches to a new formation around the newly assigned leader.

3.4.2 Local to Global Frame Position Conversion

The position of the robot is initially referenced in its local frame (X_A, Y_A), shown in Fig. 3.1, and then transformed from its local frame to the global formation frame (X, Y), as follows:

$$P_{global} = \Theta P_{loc} + q_r \tag{3.34}$$

where $P_{loc} = (x_A, y_A, \theta)$ is the robot current pose in its local frame, $q_r = (x_r, y_r, \theta_r)$ defines the generated reference trajectory, and P_{global} defines the resulting trajectory in a global reference frame (X, Y, O). The latter, equation (3.34), denotes the pose of the robot in the global frame, shown in Fig. 3.1, where O is the origin of the global coordinates frame, with:

$$\Theta = \begin{bmatrix} \cos(\theta_r) & -\sin(\theta_r) & 0 \\ \sin(\theta_r) & \cos(\theta_r) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.35}$$

While the robots are moving, the angles of their references are changing based on the generated trajectory, q_r. For stable reference following and proper swarm's formation control, each robot is required to update its new global heading angle based on the direction of the movement with respect to its trajectory. The updated heading angle of the robot is obtained using the differences in x and y between their pre-planned path $(x_{r_{i+1}}, y_{r_{i+1}})$ and current positions (x_{r_i}, y_{r_i})

$$\theta_{updated} = \theta + atan2\left(\left(y_{r_{i+1}} - y_{r_i}\right), \left(x_{r_{i+1}} - x_{r_i}\right)\right) \tag{3.36}$$

Based on the updated value of the heading angle of the robot in the global formation frame the corresponding translational and angular velocities, (v, ω), of the nonholonomic mobile robots defined in (3.6) must be updated. Based on (3.7) we got:

$$\dot{q} = S\mathrm{v} \tag{3.37}$$

$$\mathrm{v} = S^{-1}\dot{q} \tag{3.38}$$

$$\begin{bmatrix} v_{updated} \\ \omega_{updated} \end{bmatrix} = S^{-1}\dot{q} \tag{3.39}$$

where, $S = \begin{bmatrix} cos\theta_{updated} & -d_r sin\theta_{updated} & 0 \\ sin\theta_{updated} & d_r cos\theta_{updated} & 0 \\ 0 & 0 & 1 \end{bmatrix}$ and $\mathrm{v} = \begin{bmatrix} v \\ \omega \end{bmatrix}$

3.4.3 Switching to New Zone Formation

The procedure for switching to a new formation is implemented based on subdividing the workspace in several zones. Each individual agent is associated to one zone, such that each zone requires a specific formation. This concept can be used to rearrange the positions of the robots based on their specialty or zone's requirements to handle and execute its specified task. When the group of robots hits a new zone in the subdivided workspace, the corresponding specific task is associated with one of the robots. The latter becomes the group leader for as long as the formation remains within that zone. For example, if robot 1 is assigned to be the leader in zone 1, and robot 2 is assigned to be the leader in zone 2, the proposed control system manages the cooperative formation and determines how the group leader is changed automatically from the first zone to the second one. The proposed framework includes a number of specific procedures for managing the specialized tasks, as defined below:

A. Zone Definition

With no loss of generality, the zones are predefined as rectangular sectors with different width and length in this first phase of the research. In a sample case, zone parameters are defined as shown in Figure 3.3. The pose is the (x, y) coordinates with respect to the lower left corner of the zone. Dim_1 is the dimension length along the x axis, and Dim_2 is the dimension length along the y axis. Table 3.1 summarizes the parameters of the sample case shown in Figure 3.3.

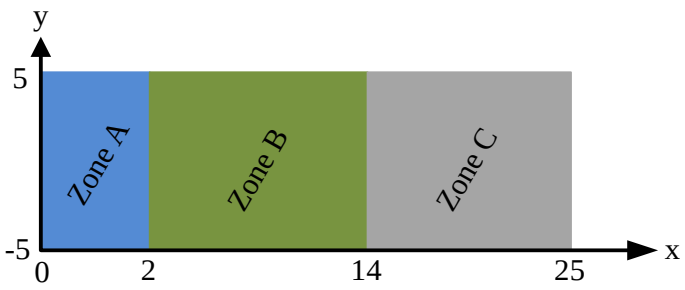


Figure 3.3 Example of zones subdivision.

Table 3.1 Structured environment zones borders parameters.

$Zone$	$(ZoneX\,, ZoneY)$	(Dim_1, Dim_2)
A	$(\,0, -5\,)$	$(2\,, 10\,)$
B	$(\,2, -5)$	$(12\,, 10)$
C	$(14, -5)$	$(11\,, 10)$

B. Zone Detection

The robots detect their zone by calculating the centrepoint of their formation. Then this point is checked against the subdivisions of the workspace to determine in which zone the group of robots is located. The central position of the formation is calculated by averaging all of the robots positions, $Position_i \in \mathcal{R}^2, i = 1, \dots, a$, where a is the total number of robots.

$$p_{center} = \frac{\sum_{i=1}^{a} Position_i}{a} \tag{3.40}$$

C. Zone Based Formation Switching

When the centrepoint, p_{center}, hits a new zone, the robots positions are updated such that the references of the previous and new leader of the swarm are exchanged and the robots are re-organized into a new formation distributed around a new leader. The latter is selected as the specialized agent for the task at hand in the zone where the formation just entered.

D. Assigning a Specialized Leader for Corresponding Zone

When the group of robots hits a new zone, the first step is to determine which robot should become the leader in that new zone. The proposed leader changing approach is summarized in Figure 3.4 for a variable number of zones and robots. The switching and re-formation algorithm for each robot is further detailed in pseudo-code in Algorithm 3.1.

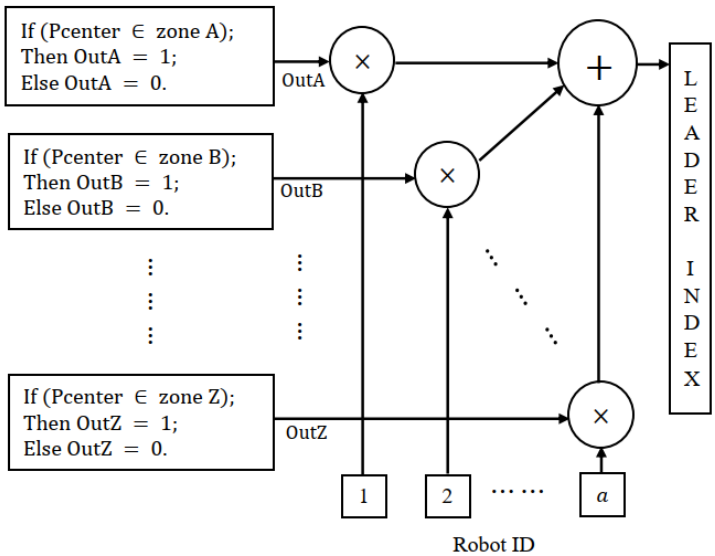


Figure 3.4 Leader index selection.

It relies on the assumption that a new leader index is selected as soon as the centre of the formation exits the current zone, that is, any of the *Out* signals defined in Algorithm 3.1 transitions

to 0. As such, the *Out* signals are mutually exclusive. For example, when the p_{center} passes from zone A to zone B, as shown in Fig. 3.3, it will lead to $OutA = 0$ and $OutB = 1$, and the new leader for zone B is identified. Then, when the centre of the formation hits the new zone, the switching process starts to take place.

Algorithm 3.1: Swarm's formation switching.

$Step1$:
$Compute\ the\ swarm\ center\ point\ position, p_{center} = (x, y);$
$Step2$:
$Lower\ left\ corner\ point\ in\ zone = (zoneX, zoneY);$
$Length\ of\ the\ zone\ on\ x\ axis = Dim_1;$
$Length\ of\ the\ zone\ on\ y\ axis = Dim_2;$
$If(x \leq zoneX \,||\, x \geq zoneX + Dim_1);$
$out = 0;$
$elseif(y \leq zoneY \,||\, y \geq zoneY + Dim_2);$
$out = 0;$
$else\ out = 1;$
$Step3$:
$If\ out = 0$
$change\ the\ formation\ to\ the\ new\ zone$
$elseif\ out = 1$
$keep\ the\ same\ formation, move\ swarm, and\ repeat\ steps\ 1\ \&\ 2;$

3.5 Task-Agent Specialization Based on Target Recognition

A specialty based task-agent allocation approach is formulated in this section by evolving the definition of specialization to match with targets recognition, such as detecting specific landmarks via embedded sensors, or the presence of an object in the environment. In this section, a supervisory layer, called Automatic Task Selection Unit (ATSU), is added to manage the decision-making process of the proposed approach. In addition, the idea of specialization zones that are mentioned in the previous section is extended and zones are generally defined as a circular area of influence around each detected target corresponding to a specialized task to be dealt with by a specific robot. Entering within an area of influence triggers the switching of the specialized leader

as well as the reconfiguration of the swarm formation. Appropriateness, smoothness and safety of the transition remain the major factors of performance considered. The framework is further refined by making the targets, and therefore the corresponding zones of influence, dynamic, which leads to the consideration of combined specialization areas. The robot control and the specialized leader switching processes are studied in relation with target recognition and adapted to accommodate real-time dynamic leadership allocation and swarm reformation.

3.5.1 Proposed Deterministic Task-Agent Allocation Framework

The proposed coordination and control structure combines two cascaded stages, as shown in Figure 3.5. The first stage is the supervisory layer, namely the Automatic Task Selection Unit (ATSU), which is the extension to the initial framework from Section 3.4 that is further developed in this section. The second stage is the proposed formation control stage that was presented in Section 3.3 to control the robot dynamics and swarm's formation. In the proposed framework, the task selection system operates in two main modes, a manual mode and an automatic mode.

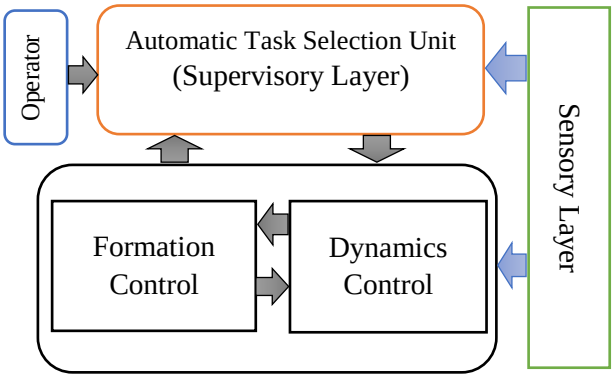


Figure 3.5 Proposed control structure.

The system's operator can switch in between these modes at the beginning of the operation by choosing whether the robots should look for a specific task, in manual mode, or run in automatic

mode where the swarm of robots will use an embedded sensory layer to explore its environment and identify tasks at hand.

3.5.2 Supervisory Layer

In this research, a centralized task allocation mechanism is adopted. The ATSU is proposed as a system's supervisory layer or central unit that assigns the detected tasks to the specialized agents. Thus, the team members have to be fully connected with this central unit to exchange the agents' local information (i.e. agent's dynamics control variables such as agent's location, velocity and energy level) and surrounding environment information (i.e. targets and navigation information). The central unit's role is to process this information and return the appropriate commands to the individual agents to be assigned and execute the estimated tasks. To support extensive data processing operations, a centralized approach is the best-suited solution. The proposed central unit can be built in one of the mobile agents or as a static central station.

A. Task Allocation Approach

The group of robots navigates over three independent and successive zones surrounding a given task. Accordingly, the system evolves through three corresponding states: a Search state, a Task state, and an Execution state. Fig. 3.6 presents a schematic diagram of these three states.

In the Search state, depicted by the gray area in Fig .3.6, the robots adopt a cooperative leader-follower formation until the sensors that are on-board the robots recognize a target. The system switches to the Task state when the centrepoint of the group of robots enters the zone of influence that surrounds the detected target, delimited by the task zone border, shown in blue in Fig. 3.6. The robots then change their formation according to the specialty that the task requires while

continuing the navigation towards the target position and until they approach the execution zone border, shown in orange in Fig. 3.6. Then the Execution state takes place, as the specialized agent, which became the swarm's leader in the Task state, approaches towards the target and executes the task associated with the detected target. When the task is completed, the mission automatically switches back to the Search state to further explore the environment for a new task.

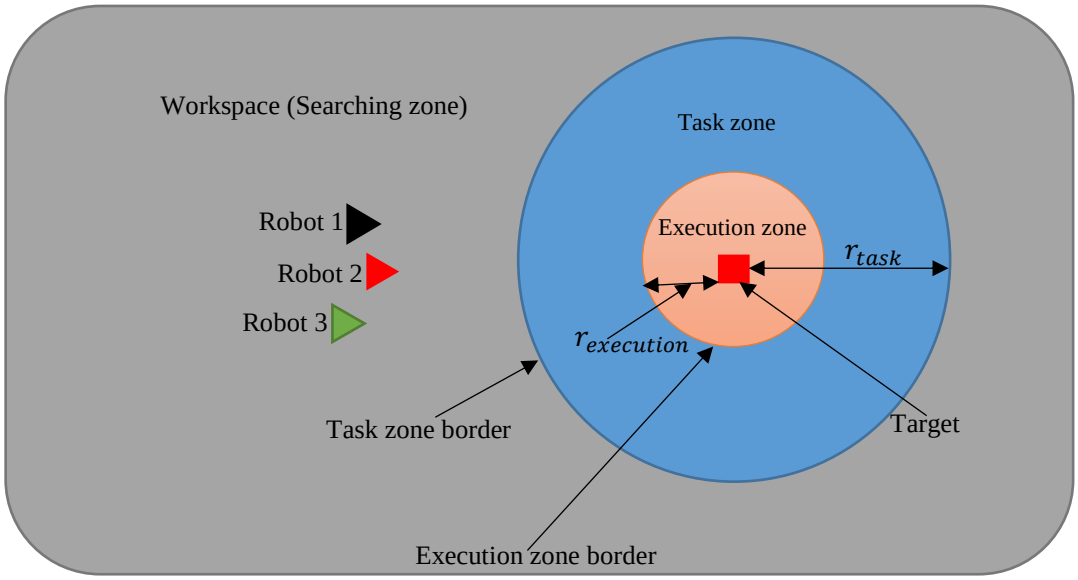


Figure 3.6 Three states of operation (Search, Task, and Execution).

B. Operation modes

As mentioned above, the task assignment problem is modeled under two different modes: the manual and automatic modes. Under manual operation, the target location is predefined by an external operator, while under automatic operation the leader robot uses embedded sensors to search for a target while patrolling the workspace, therefore generating the path for the group of robots that follow it. Once the group of robots is located within a given subspace, their initial priority is to navigate over that subspace and search for a recognizable target upon which they can perform a specific task. During this searching process, the robots navigate and keep a cooperative

formation where a by-default group leader is generating the path that the group of robots follows. Moreover, under the operation of the proposed framework, robots can perform the task while the target is static or dynamic. The specific formulation for both modes of operation and three possible states is detailed in this section.

i. Manual Mode

The manual operation mode provides the system with the capability to be semi-automatically driven, under the supervision of a human operator. This mode is suitable when there are a known number of tasks to be performed at predefined locations in the workspace. The operator identifies a specific target to reach, on which the group of robots is to execute a specific task. Then, the cooperative group of robots proceeds through the three states defined above. During the Search state, the group of robots navigate over the workspace and keep their cooperative formation while following their default leader until its on-board sensors recognize the desired target. Then the system switches to the Task state, and later on to the Execution state at the execution zone border to perform the predefined task. After the task is completed, the ATSU takes control and robots stand and wait for the operator to command them to change the mode of operation to automatic, or keep to the manual mode, in which case the operator needs to identify another specific task to accomplish.

ii. Automatic Mode

When the number, location and specificity of tasks to be executed are unknown a priori, the automatic task selection mode is preferable, as it supports automation of the task detection and recognition, allocation of the proper specialization agent, and execution processes. When the group of robots is operating in automatic mode, they aim to complete the entire set of tasks that are

available in the workspace, based on the nature of targets that can be detected using embedded sensors without consulting the system operator.

C. Operation States

To accomplish each individual task, the automated mission of the swarm can go over three states that are:

i. Search State

To make the proposed framework generic, specialized agents identified as $R_i, i \in \{1, \dots, a\}$, where a is the number of specialized agents available, are considered. Also, different tasks are assumed to be available in the environment, each identified as $T \in \{1, \dots, t\}$, where t is the number of possible tasks. A specific agent is chosen to perform each specific task in a deterministic way, as defined through a lookup table. Based on the assumption that the agents start their navigation while targeting one of the tasks existing in the workspace, for example T_1, the corresponding specialized robot, R_1, is assigned as the default group leader during the Search state. The path of the swarm is then planned to begin from the initial positions of each of the agents. The trajectory is designed to make the swarm survey the entire workspace in order to successively perform all tasks that can be detected. The position of each robot in a global reference frame (X, Y) is defined as (x_i, y_i). The path planning, swarms' formation and stabilization during the Search state are further discussed in sections 3.3 and 3.4.

ii. Task State

When the sensing system mounted on robots recognizes a given target and localizes its position, (x_T, y_T), and type T, the system measures the Euclidean distance between the target and the centre

of the swarm as the group of robots continues to approach the given task position. This process continues until the robots enter the area that surrounds the selected target, defined by the zone of influence of the target. A variable radius, r_{task}, characterizes that zone of influence surrounding any target, limited by the task zone border in Figure 3.6. The radius is predefined as a switching edge to trigger the group of robots to transit to a new formation in preparation to execute the specialized task. As soon as the swarm enters the zone of influence, considering the specialty assigned to each robotic agent, the ATSU identifies which agent is appropriate to perform this specific task based on Algorithm 3.2. In the present case, each agent is deterministically associated to a specific type of task, T. The ATSU then triggers the Task state and assigns the relevant agent to become the leader of the swarm, and switches the swarm's formation around the assigned agent. During the Task state, the swarm smoothly transitions its formation to a new distribution adequate to perform the given specialized task, with the most competent agent now leading the formation. While the transition to the new leader takes place, the robots continue approaching the target until they reach close to it.

Within the zone of influence of a target, the new leader coordinates (x_l, y_l) are assigned based on the current position of the proper specialized agent, (x_i, y_i), selected to become the leader, and from the position of the target, (x_T, y_T). The former leader and all other robots involved in the swarm become followers. The follower robots continue their progression up to the execution zone border and then hold their positions until the leader completes the task and another transition happens.

When the switching to a new leader takes place, its current state, which was reached during the Search state, is considered as the start point to further reach toward the target, that is:

$$State(Leader_{ID}) = [x_l, y_l, \theta_l, V_l, \omega_l]^T \tag{3.41}$$

where θ_l represents the current heading angle of the leader robot, and V_l, ω_l correspond to its current linear and angular velocities.

Algorithm 3.2: Task-agent specialty based assignment.

Step 1: Simulated sensing stage detects task and estimates $Task\ State = (x_{\mathrm{T}}, y_{\mathrm{T}}, T)$

Step 2a: Estimate Euclidean distance (ED) between task position ($x_{\mathrm{T}}, y_{\mathrm{T}}$) and current center point position of the swarm $p_{centre} = (x_{\mathrm{centre}}, y_{\mathrm{centre}})$; and compares it with radius of the Task Zone (r_{task})

ATSU: if $ED \leq r_{task}$
then switch to Task State.
else
remain in Search State;

Step 2b: ATSU checks Task Type (T);
Select suitable specialized robot to perform the recognized task, via lookup table
Assign a specialized robot as new specialized leader R_i ; where $i \in \{1, \dots, a\}$
Return leader agent identity, $Leader_{ID} = i$.

Step 3: Assign new leader coordinates:
ATSU: if $Leader_{ID} = i$;
new leader position $(x_l, y_l) \leftarrow (x_i, y_i)$;
Return the new leader coordinates (x_l, y_l);

Then an iterative path planning process is defined where the next desired position for the leader within the task zone is computed iteratively as the middle point between the task position and the current position of the assigned leader. Collision avoidance is dealt with at a formation control level using repulsive potential fields, as discussed in Section 3.3.3. Assuming that the current position of the leader is (x_l, y_l) and the task position is (x_T, y_T), then the next desired state for the leader in the task zone is:

$$\mathrm{State}'(\mathrm{L}) = [x_l', y_l', \theta_l', V_l', \omega_l']^T \tag{3.42}$$

where the new leader position is defined in eq. (3.43) and (3.44), and the heading angle of the robot, θ_l', is set to aim toward the target's position, as defined in eq. (3.45).

$$x_l' = x_l + \frac{x_T - x_l}{2} \tag{3.43}$$

$$y_l' = y_l + \frac{y_T - y_l}{2} \tag{3.44}$$

$$\theta_l' = \theta_l + \text{atan2}\left(\frac{y_l' - y_l}{x_l' - x_l}\right) \tag{3.45}$$

The updated velocities (V_l', ω_l') of the leader, equation (3.39), are calculated as follows:

$$\begin{bmatrix} V_l' \\ \omega_l' \end{bmatrix}^{\mathrm{T}} = \dot{q}^T S^{-1} \tag{3.46}$$

where:

$$\dot{q} = [(x_l' - x_l), (y_l' - y_l)]^T \tag{3.47}$$

$$S = \begin{bmatrix} \cos(\theta_l') & -d_r \sin(\theta_l') \\ \sin(\theta_l') & d_r \cos(\theta_l') \end{bmatrix} \tag{3.48}$$

where d_r is the distance between the centre of the robot and its back axle, as shown in Fig. 3.1.

iii. Execution State

Robot embedded sensors continue to monitor the distance separating the task position, (x_T, y_T), from the current leader coordinates, (x_l, y_l). Simultaneously, the leader continues to approach the target. From the moment that the leader hits the execution zone, of radius $r_{execution}$, the ATSU switches the swarms' formation to the last state of the process. The Execution state ensures that the selected specialized robot performs the task. The follower robots then reconfigure themselves to reach a distribution around the leader, and they stop at the execution zone border. During the Execution state, the followers provide a cooperative coordination and can be exploited as stationary sensors to support the leader robot with extra sensing capabilities while it performs the task.

D. Special Cases

i. Overlapping zones of influence

Overlapping zones of influence correspond to areas that are within reach of more than one target, as a result of targets that exhibit a large area of influence, or moving targets which make their respective area of influence to be overlapping with others for certain periods of time. Such a situation is illustrated in Figure 3.7, where the blue and red targets' respective areas of influence share a mutual portion of the workspace. The proposed ATSU is designed to tackle such situations. Whenever overlapping areas of influence are detected close to the location of one of more targets, the system switches to a semi-automated mode and the operator is consulted to choose which target is given the execution priority. Then, the system operator commands the system to give priority to one of the detected targets. After that, the ATSU drives the system to perform operations on the selected target and the robots continue their operation while concentrating on the selected target. Once that target is resolved, they go directly to the second recognized target, which is located in the same area, and so on for more targets. When all targets have been visited, the ATSU switches formation control back to the Search state, which resumes the regular operation to look for a new task in the workspace.

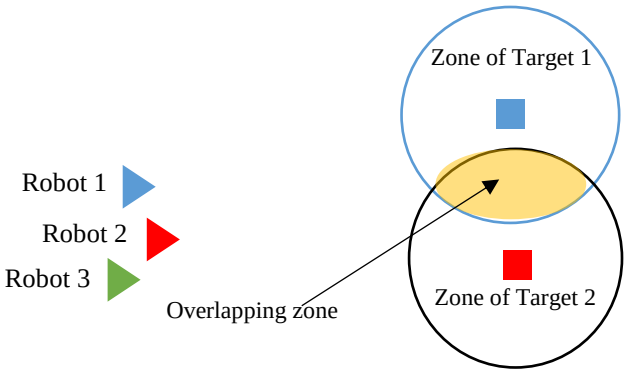


Figure 3.7 Overlapping zones of influence.

ii. Moving Targets

Another important case to be considered is when the targets are changing their position over the workspace, such as in scenarios where people, animals or motorized devices are associated with tasks to be performed (e.g. people following, assistive robotics, or vehicle interception). Such scenarios are also supported in the proposed automatic mode of operation. When a robot embedded sensor reports different positions for successive measurements on a given target, the formation control scheme exploits a specially designed *Target Following state*, which temporarily replaces the *Task State*. In the *Target Following State*, the swarm keeps following the target all together until the target becomes static. Once the target is stopped, then the usual *Execution State* is triggered and the task completion proceeds as usual. The *Target Following State* is assumed to work only when a target is moving slower than or equal to the maximum speed of the swarm. In the case where the target speed is faster than the maximum speed of the swarm, this means that the target is assumed to be outside of the sensing range and the system will return to the Search state. An overview of operational modes and full mission operation is presented in the flowchart of Figure 3.8. The color of each state in Figure 3.8 is matched with its corresponding zone in Figure 3.6.

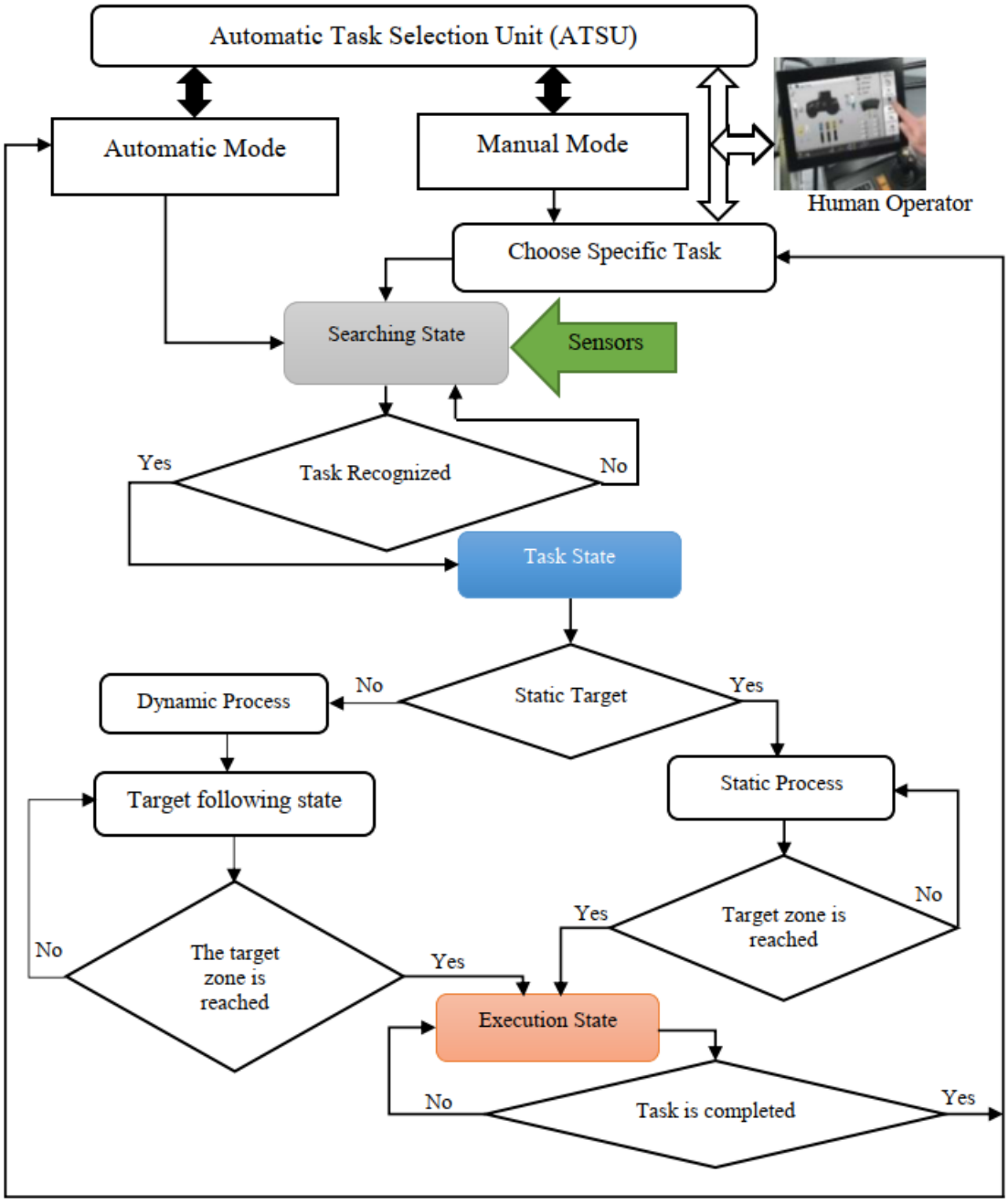


Figure 3.8 Schematic structure and sequencing for all operational modes.

3.6 Simulation Experiments

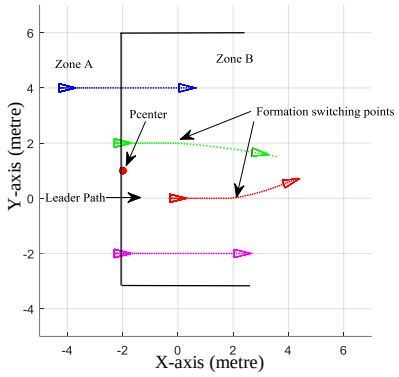
This first version of the proposed conceptual framework has been validated in simulation using Simulink. In this phase of the research, three robots are considered for acting as specialized agents

and three different types of target are also considered. Each robot is specialized to execute only one type of task. Therefore, the task allocation process is deterministic and the objective of this experimental evaluation is mainly to confirm the coherence of the proposed framework as a whole, before the task allocation scheme is refined in Chapter 4.

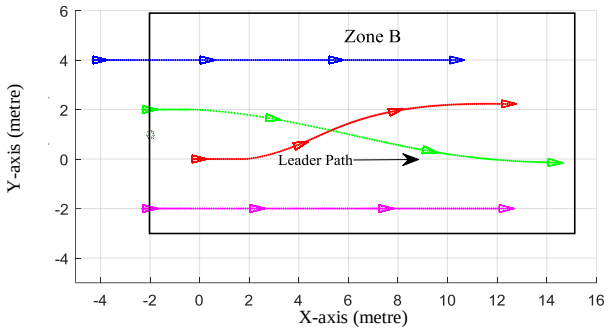
To validate the first approach that considers simplistic rectangular subdivision of the workspace to form zones that will trigger the response of specific specialized robotic agents, addressed in section 3.4, the robot parameters used in the simulation, as depicted in Figure 3.1, are $m_a = 10kg, \alpha = 0.05\ meter, \delta = 0.5\ meter, I = 5kg\text{-}m^2; d_r = 0.8$. Figure 3.9a shows the initial conditions of the group formation in a sample simulation case. Triangles represent the respective specialized agents and dotted lines show the corresponding paths over time. Colors are associated to the specialization of each agent, which should match with specific zones, whereas the leading role of the swarm in zone A is associated with the red robot, with the green robot in zone B, and with the pink robot in zone C.

It is also shown in Figure 3.9a that the moment when the centre of the formation, p_{centre}, represented by a red dot at (-2,1) enters into zone B, the group's leader (red) starts to change its path at point (2,0) and the green robot also begins, at point (0, 2), to take the leading role, as the green robot is here defined as the specialized agent appropriate to tackle the leading role in zone B. The system switches the formation automatically to make the relevant specialized robot (green) to become the group's leader while the swarm circulates in zone B. Figure 3.9b indicates that the green robot is leading in zone B as it circulates ahead of the other robots. Figure 3.9c presents the trajectories of the whole group navigation in zone C and shows how the system eventually switches to select the third robot (pink) to become the group leader in that zone, that agent being the proper

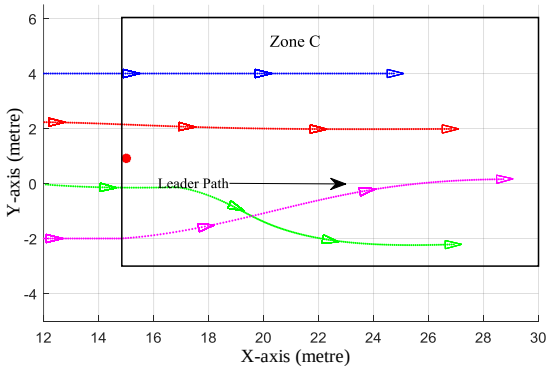
one to perform tasks in zone C. The minimum distance desired between the green and the pink robots is also achieved under the influence of repulsive forces, as detailed in Section 3.3.3, whereas the formation controller, eq. (3.33), controls the minimum distance in the generated trajectories using repulsive potential forces to achieve the required separation between the agents. In this case, the pink agent passes the cross point (19.5, -1) at a given instant, while the green agent passes a little later.



a) Formation switches when the group enters in zone B, that is when the formation's centre (red dot) first hits zone B at point (-2,1).



b) Reformation process takes place in zone B and the green robot becomes the leader.



c) Reformation takes place in zone C as the pink robot becomes the leader.

Figure 3.9 Group formation and switching control of the leading role over three different and specialized zones.

Simulation experiments are also conducted to validate the extended conceptual framework based on targets recognition, that is introduced in section 3.5, using Simulink. For system's

validation purposes, the case with three robots, $a = 3$, to serve as specialized agents associated to three different types of target, $t = 3$, is again considered. It is still assumed that every agent is specialized to perform only one type of task. For clarity and generality, the specialized agents and the corresponding tasks to be matched are marked with corresponding colors. Colored lines trailing behind triangular robotic agents in Fig. 3.10 to 3.13 depict the robots' paths, and tasks are represented by colored square dots circled by dotted lines. The latter indicate their respective area of influence.

In the automatic mode of operation, during the Search state, the agents navigate over the workspace and simultaneously keep their cooperative group formation as they follow the swarm's default leader (e.g. red agent "red triangle" in Fig. 3.10a) until the simulated on-board sensing system recognizes a first target (blue square). Thereafter the system switches first to the Task state, the blue agent becomes the leader in order to complete the blue task, and subsequently to the Execution state, as shown in Figure 3.10b and 3.10c to perform that task, which is shown as completed (black) in Figure 3.10d. Then, the system returns back to search for a new target and automatically transitions the formation to re-assign the default leader (red agent) ahead of the swarm (Figure 3.10e-f) with the other agents performing as followers again. Subsequently, the ATSU assigns the proper specialized agents to perform other available tasks (blue agent in Fig. 3.10g; green agent in Fig. 3.10i), therefore alternating in between Search, Task and Execution states (Fig. 3.10e-j), as expected. The lower green task in Figure 3.10h is beyond the sensing range when the formation passes by, and as a result it is skipped. In Fig. 3.10, the simulation was stopped after having processed the upper green, which explains why the two red targets are not visited. For simulation purposes, a task is considered fulfilled when the proper specialized robot hits the

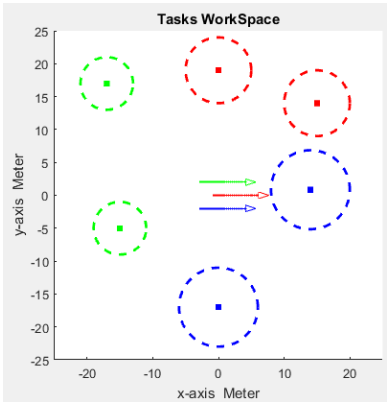
position of the matching target. Then the target symbol changes to black, which indicates that the given task has been executed, as shown in Fig. 3.10d, 3.10h and 3.10i.

When the manual mode of operation is selected, this means that one target, manually selected by the human operator, is the first one to be performed. In this case, the ATSU immediately selects the robot that is the one specialized to execute this specific task and this robot immediately becomes the leader. Figure 3.11 provides an illustration of such a situation as the group of robots skips over the blue task because the red task (red square dot) was manually selected by the operator and therefore has priority. As such, the red robot becomes the leader and initially drives the group toward the red target while the blue robot remains a follower even though the group passes close by the blue target. This scenario demonstrates the flexibility of the proposed framework to reliably implement prioritization functions in the control of the swarm. Following completion of the higher priority task, the formation can either resume to Search state in automatic mode, or be provided a second priority target to execute in manual mode.

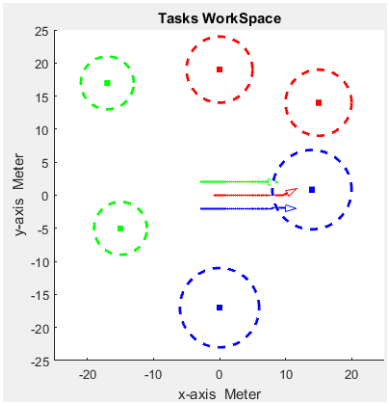
Figure 3.12 represents a condition of overlapping zones of influence, that is when more than one target's influence area share common sections of the workspace. In such a scenario, the ATSU consults the system's operator to select the target that should be executed first. In the simulation illustrated in Fig. 3.12a, the operator selects the red task to be performed in priority. Once the selected task is completed (Figure 3.12c), then the robots switch their leading task automatically to execute the blue task as it shares the same area of influence. The new leader (blue) is smoothly transferred into the leading role for the group of agents while the other agents become followers, as shown in Figure 3.12d-f. Once the conflicting tasks are resolved, the swarm switches back to the regular process outside the dual-task zone, as shown in Figure 3.12g-h. The default red leader

agent drives the swarm in exploration until the swarm switches to the green agent as the leader to perform the left-hand side green task. Finally, the other red tasks is also performed successively following the same logic with the red agent as the leader. Figure 3.13 shows a scenario where a moving target is involved. The swarm of robots initially follows a moving green target while it attempts to evade from the swarm. The temporary Target Following state is triggered until the target can be reached by the swarm (Figure 3.13a-d). At that point, the corresponding specialized green agent, which was assigned as the leader agent during the pursuit, executes the task (Figure 3.13e). Once completed, the system returns in the standard Search state, which reinstalls the default red agent in the leader position until other static or dynamic targets are detected (Figure 3.13f).

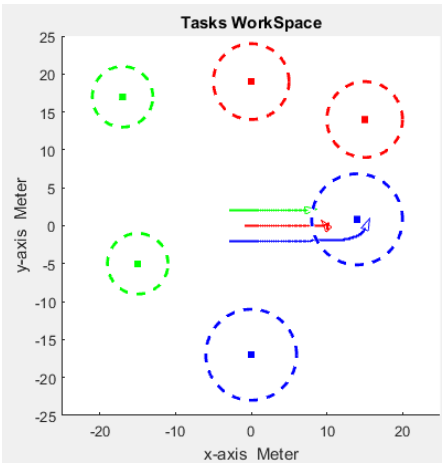
These experimental results show the conceptual validity of the proposed framework. Cooperative navigation and swarm's formation control as well as smooth switching and transition of the leadership in between the specialized individual agents is achieved. The ATSU is robust enough to deal with a varying number of targets, including targets that are sharing the same specialized zones. It also supports prioritization to take place in between targets, via an embedded supervisory layer that can be manually driven or fully automated. The three operational states of the proposed approach, from broad search to execution, provide the necessary support for the ATSU to respond to the exploratory, tracking, and task execution components typically found in swarm robotic scenarios. Considering the cases of overlapping areas or shared zones of influence as well as moving targets also contributes to bring the proposed protocol closer to realistic operational contexts.



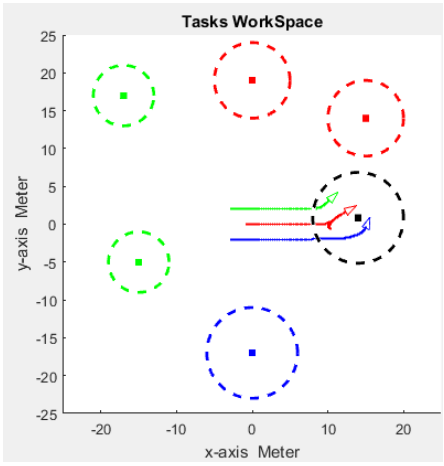
(a): Formation in Search state with default red leader.



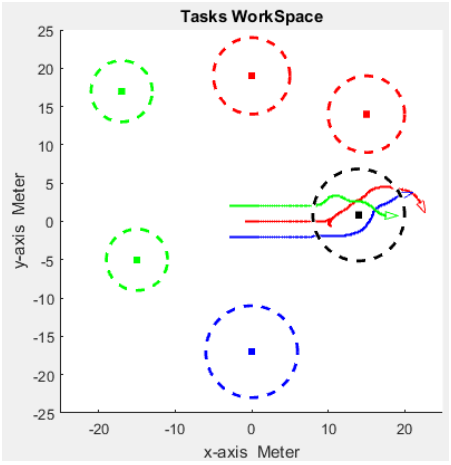
(b): Formation switched to Task state.



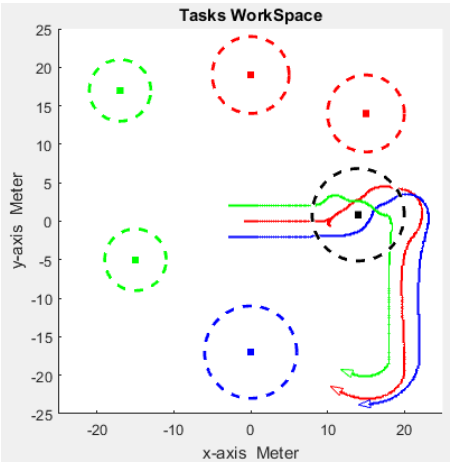
(c): Formation switched to Execution state with blue leader.



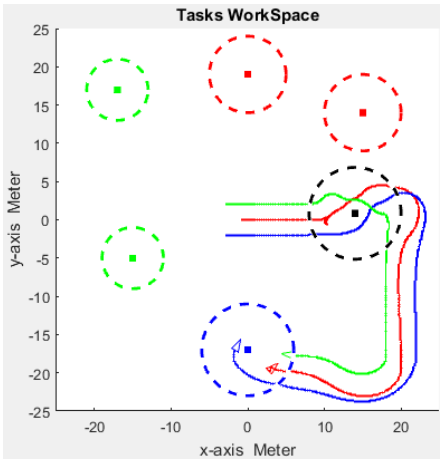
(d): Blue task completed by blue leader agent.



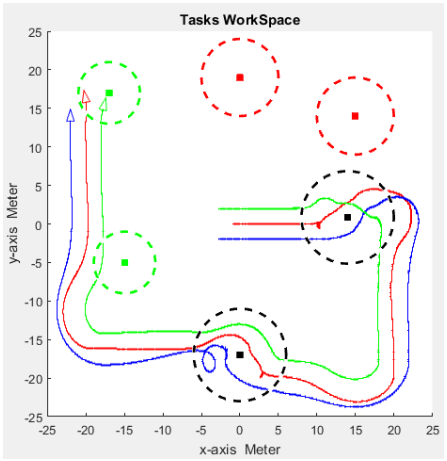
(e): Formation switched back to Search state with red default leader.



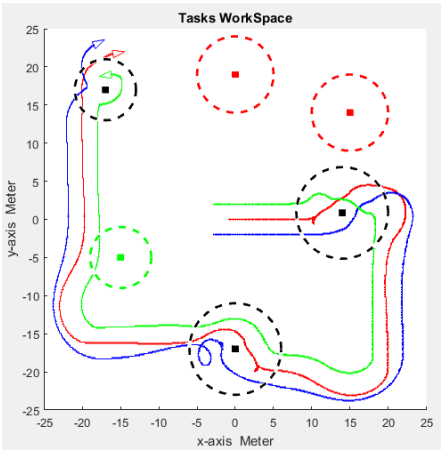
(f): Formation in Search state.



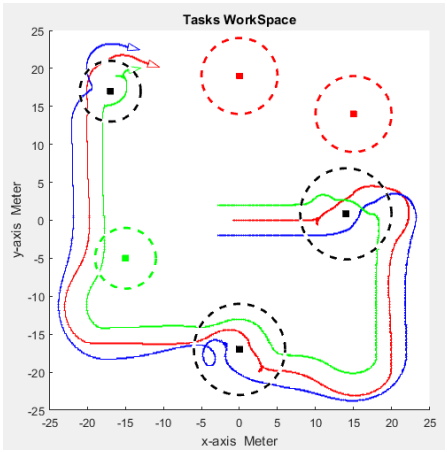
(g): Formation switched to execute the second blue task.



(h): Green task out of the formation sensing range (skipped).

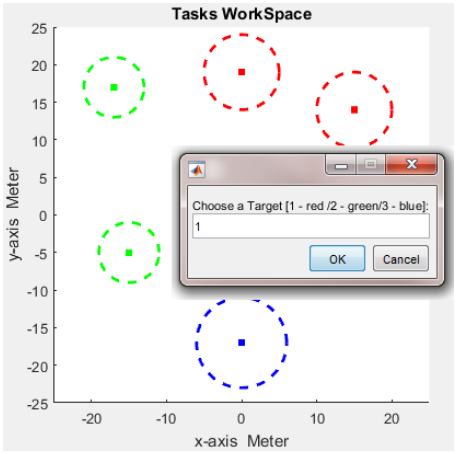


(i): Third detected task is executed with green leader.

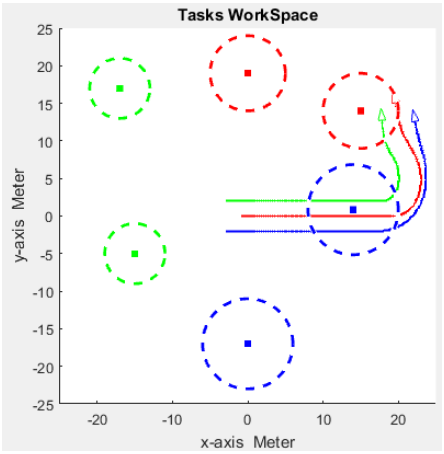


(j): Formation switched back to Search state again.

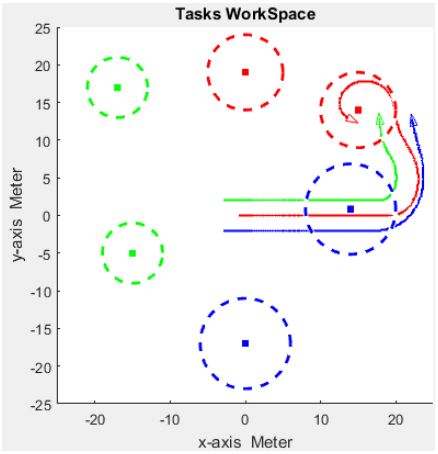
Figure 3.10 Successive tasks completion in automatic mode of operation.



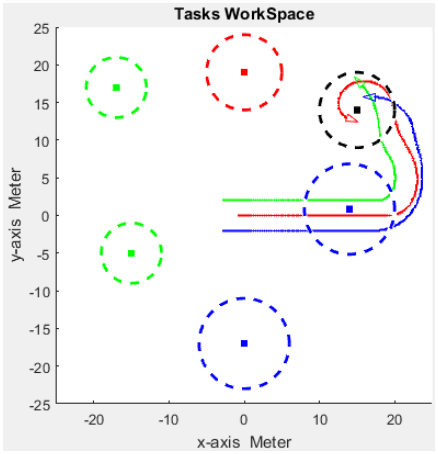
(a): Task type selection by supervisory layer (type 1 = “red”).



(b): Formation heading towards top priority “red” target.

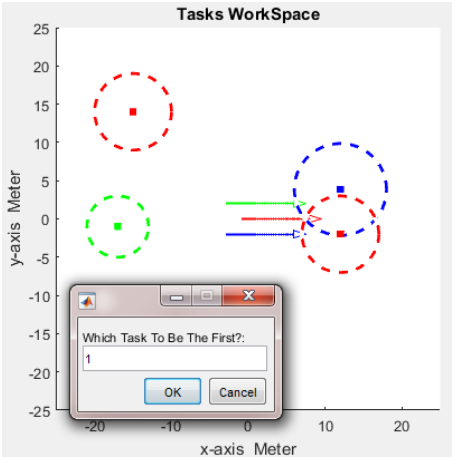


(c): Priority task under execution.

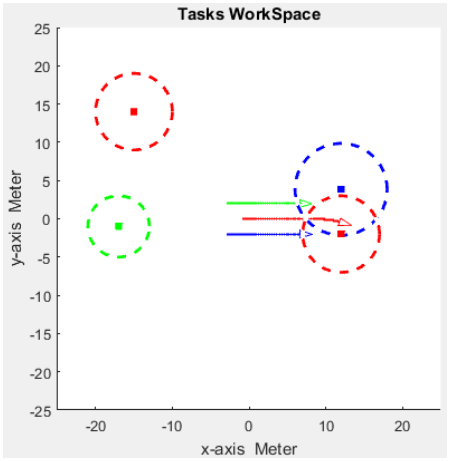


(d): Priority task completed.

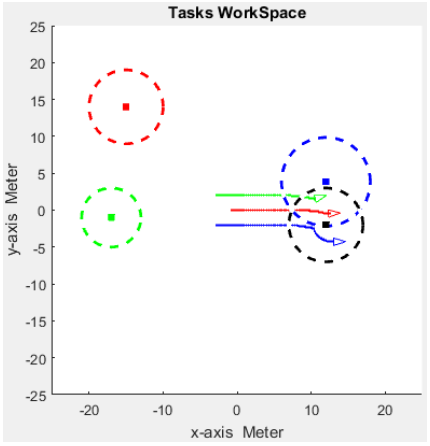
Figure 3.11 Specific task completion in manual mode of operation.



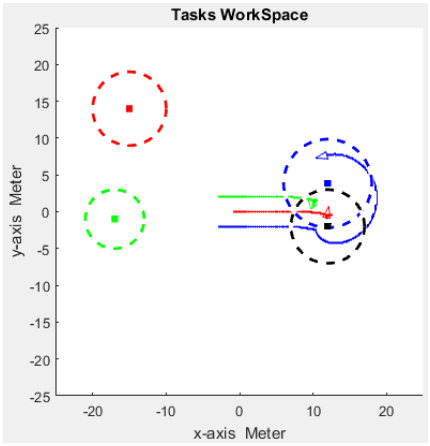
(a): Task 1 (red) selected as having priority.



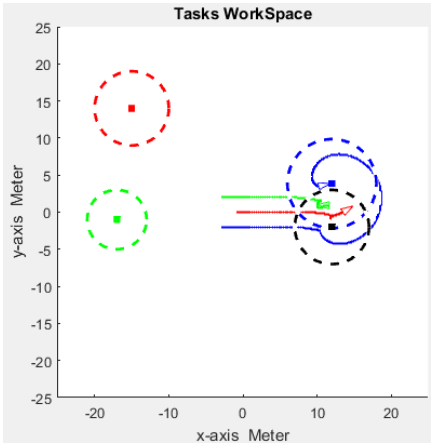
(b): Red robot remains leader until task completion.



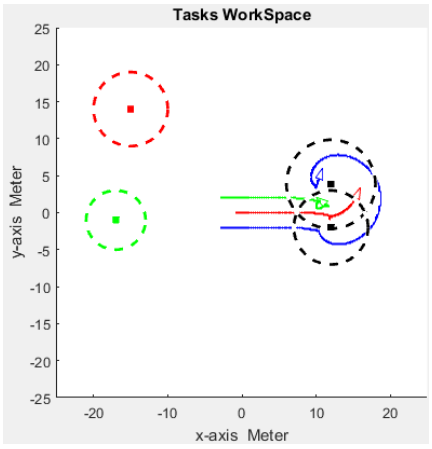
(c): Red task completed.



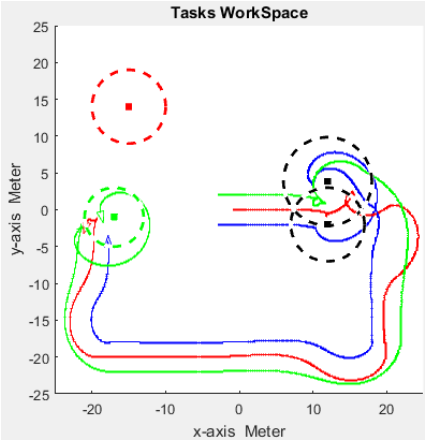
(d): Blue robot becomes leader until 2nd task completion.



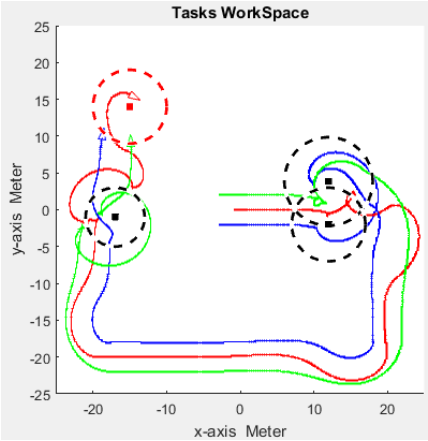
(e): Blue leader executing the blue task.



(f): Blue task completed.

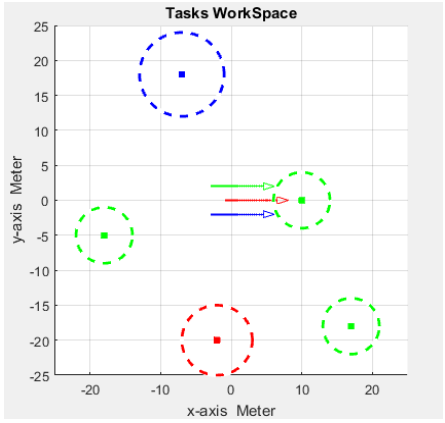


(g): Green task performed (regular process).

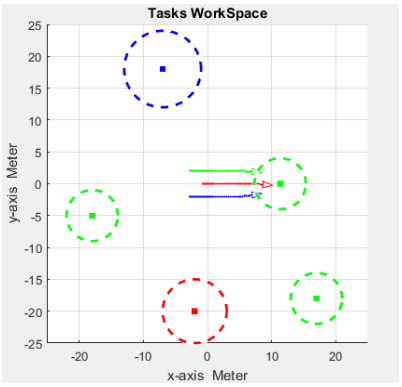


(h): Red task performed (regular process).

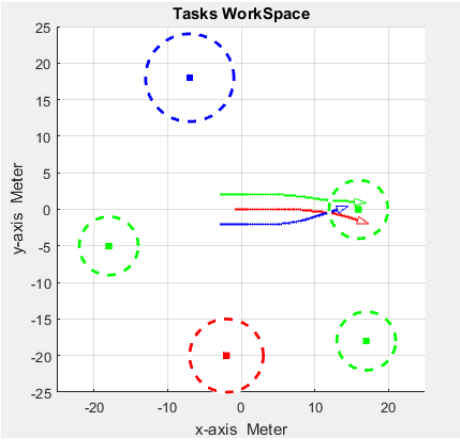
Figure 3.12 Successive tasks execution with (a-c) overlapping zones of influence with task 1 (red) selected by operator as having priority, which drives the red robot to remain the leader until the red task is completed; (d-f) second priority task (blue) being performed with blue robot transitioned to leader position; and (g-h) swarm resumes to the Search state outside the overlapping zones of influence, then searches for other tasks (green, then red) to be executed with corresponding leader robots.



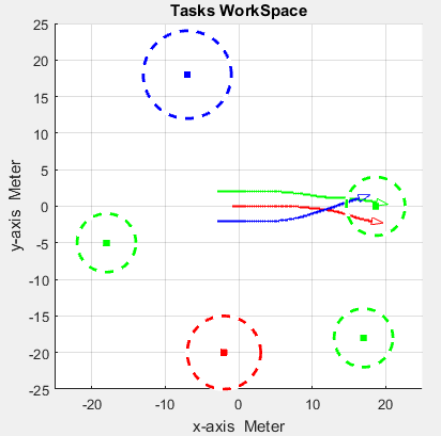
(a): Green moving target detected (Target Following state).



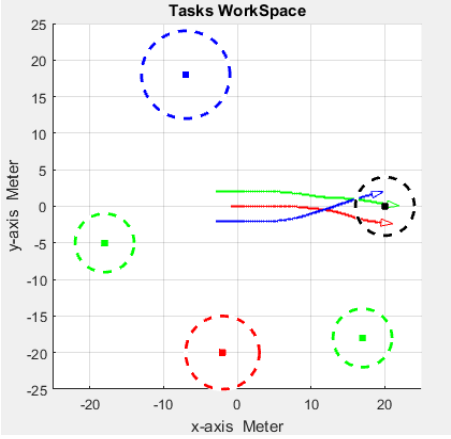
(b): Target reached while being pursued by the swarm.



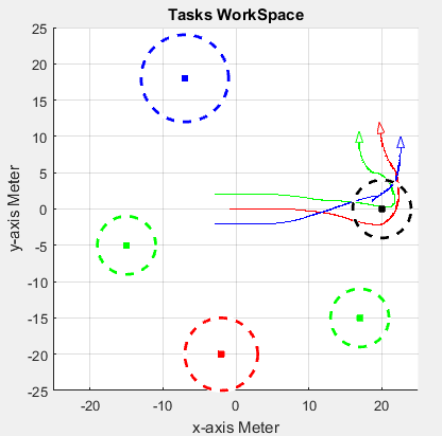
(c): Green agent transitioning toward leader position.



(d): Execution state with green leader.



(e): Green task completed.



(f): Swarm swithes back to Search state with red leader.

Figure 3.13 Scenario with a dynamic target:(a) green moving target detected, (b) green target being pursued by group of robots with green agent transitioning toward leader position, (c-d) green target reached and task executed by matching green leader robot, and (e-f) swarm pursuing search for other targets with default red leader agent returning to leader position.

3.7 Summary

The proposed task-agent matching mechanism is initially studied in two parts. The first one presents the design and implementation of a task switching and cooperative swarm formation transition for specialized mobile agents. The proposed system assigns the leading task to a specific agent depending on the position of the formation. A dynamic model of nonholonomic mobile robots is considered whereas a kinematic and torque controller is used to control path following and group formation.

In the second part, a generic conceptual framework to ensure coordination and formation control of a group of mobile agents is presented. A specialized task assignment and execution approach is formulated in an algorithmic manner that evolves the definition of agents' specialization to match with targets recognition. The proposed Automatic Task Selection Unit (ATSU) is implemented in simulation to operate in two possible modes, either under close supervision or in a more automated way. Both modes perform each specialized task in three successive states that ensure the search for a target, its detection and recognition via virtual embedded sensors, and finally the execution of a given task compatible with the resources available on a selected agent, which also serves as the group leader for the duration of a specific assignment.

To ensure safe navigation and smooth task switching without any inter-agent collisions, a potential field layer is integrated which generates internal repulsive forces between the robots. The collision avoidance and the formation control algorithms work together to keep the agents far enough from each other while managing transition in the group leader as zones requiring the

intervention of a given specialized agent are accessed by the swarm. The proposed framework is validated in simulation for a diversity of scenarios and demonstrates that the group of robots successfully coordinate themselves toward and around targets and dynamically allocate the appropriate specialized agent to automatically respond and perform specific tasks.

In Chapter 4, the specialty-based framework is substantially refined with the development of a task-agent specialty based matching scheme that further leverages the uncertainty associated with the recognition of a target object's characteristics from embedded sensors. The confidence level associated with the recognition of predefined characteristics on target objects is taken advantage of to best match tasks to be performed with the corresponding specialized robots' capabilities.

Chapter 4 Specialty-based Task-Agent Allocation - Probabilistic Approach

4.1 Introduction

In the research work reported in Chapter 3, the proposed approach is introduced as a swarm's cooperative control problem and the task-agent specialty assignment process is build based on a deterministic detection of the constrained task. In this chapter, the concept of task-agent specialty assignment is further developed and evolved. The framework is refined with an expanded formulation of the proposed speciality-based task-agent assignment as a generalized task allocation framework from a probabilistic perspective. For this purpose, this chapter introduces a probabilistic approach for assigning specialized individual agents among a heterogeneous robotic swarm to corresponding tasks, each imposing specific constraints. Based on the assumption that each individual agent possesses specialized capabilities, the proposed approach evaluates the probabilistic fitting of the available robot individuals based on the requirements imposed by the recognized task, which takes the form of a recognized target object in a specific environment. A formal probability-based task allocation matching scheme is developed to evaluate task-agent fitting probabilities of all available agents. This scheme assigns the most qualified and available specialized robotic agent or agents to respond to and perform the detected task.

The formalism for task allocation is still being developed in the context of a cooperative swarm of mobile robots. However, unlike previous work that considers heterogeneity among robotic agents based mainly upon their physical construction, here a specific definition of specialization is introduced which leverages the embedded hardware and software characterizing each agent's functionality. As a result, an advanced form of specialized labour division emerges in the swarm.

The labour is divided among the individual members based on best matching the specific requirements of the task to each robot's specific capabilities. This form of task allocation can increase the net efficiency of the swarm, whereas the most qualified agents can respond to the corresponding tasks based on the match between the agents' specialized capabilities and the requirements of the detected tasks. In addition, the proposed task allocation scheme computes the individual agents' specialty fitting probabilities to support optimal task allocation. As qualified agents are assigned to corresponding tasks with different probabilities, a form of a specialty-based prioritization among robotic agents also emerges.

The proposed approach contributes to advance the field of specialty-based task allocation in unmanned vehicles. Simulations of realistic scenarios are presented and discussed in Chapter 5 to validate and investigate the efficiency of the proposed approach. Finally, real-world experiments are conducted and the results are analyzed in Chapter 6 to validate the efficiency and feasibility of the proposed approach to its real-world desirable applications.

4.2 Proposed Approach

The proposed approach forms a probability-based allocation mechanism for robots that are equipped with specialized capabilities to best match with constrained tasks detected in their environment. Figure 4.1 illustrates the overall framework of the proposed specialty-based task-agent allocation approach.

When a group of specialized robotic agents navigates through a given workspace, observer agents search for target objects from a list of predefined objects of interest. Once a target is detected through visual pattern recognition, automated task allocation of a robotic agent is initiated

for the task to be completed (e.g. reach to the object for closer observation, track or pick the object, etc.). The goal is to assign the most capable specialized agent or agents to respond to a detected task. The proposed solution considers the swarm members to be different at their functionality level (e.g. on-board sensors or actuators, communication, or reasoning capabilities). This results in specialized capabilities for each individual agent forming an overall multi-agent system with non-homogeneous functionalities.

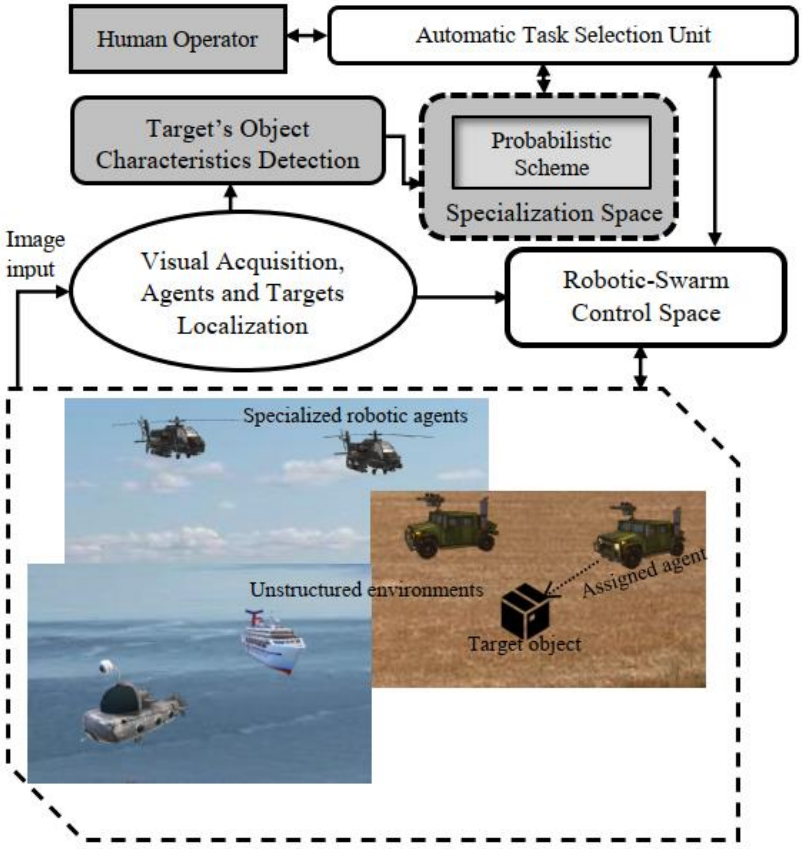


Figure 4.1 General framework for the proposed specialized task-agent allocation.

To achieve this goal, two coupled spaces are defined for coordinating the specialized robotic individuals in the swarm. A schematic diagram of the coupled coordination spaces is shown in Figure 4.2. The control space tackles control considerations related to robots' dynamics, swarm's navigation, group formation, and transition (described in Chapter 3 and with details published in

[128-130]). As the core contribution of this thesis, the specialization space optimizes the match between the task requirements and the individual robots' specializations. The objective of the specialization space is to divide the labour amongst the swarm from the perspective of the individuals' specialty in the presence of cooperation between the agents.

In the specialization space, a task-agent probabilistic approach is defined for matching characteristic features perceived on detected tasks with robot-embedded specialties. This approach relies on an uncertain representation [131] of the observed features, which provides a task specific signature that can be matched with predefined corresponding specialized functionalities of the swarm's individuals. As such, this work focuses on characterizing the specialization of the agents as heterogeneous individuals with respect to their functionalities and on the evaluation of a task-agent probabilistic fitting score that is required for assigning different robots to specific tasks. The development of the proposed approach in this chapter is focused on a generic formulation of the features that characterize target tasks and on the design aspects of the specialized agents' task allocator. The grey boxes in Figure 4.1 highlight the contribution of this chapter with a focus on the specialization space. The design considerations of the proposed framework address three main problems in this study:

1) Theoretical formulation of target task characteristic recognition;

2) Task-agent specialty matching; and

3) Coordination of the optimal specialized responders.

An automatic task selection unit (ATSU) (introduced in Chapter 3) is responsible for the decision-making process to assign the specialized robot to respond to a corresponding detected target. In addition, a human supervisor is preserved on the control loop for strategic guidance, as

depicted in Figures 4.1 and 4.2. In practice, target object detection and task recognition in the robot's unstructured environment can be performed via machine learning techniques. Formal development and implementation of this machine learning method is a deep learning target object recognition stage that is addressed by a colleague in SMART research group [131] and used in section 6.2.1.A as an input to the proposed approach. It does not form a contribution from this thesis. However, target object recognition is developed in this thesis as a generic theoretical formulation in section 4.3 and integrated with the proposed speciality-based task allocation framework, as shown in Figure 4.2, to support real-world experimental validation, as discussed in Chapter 6.

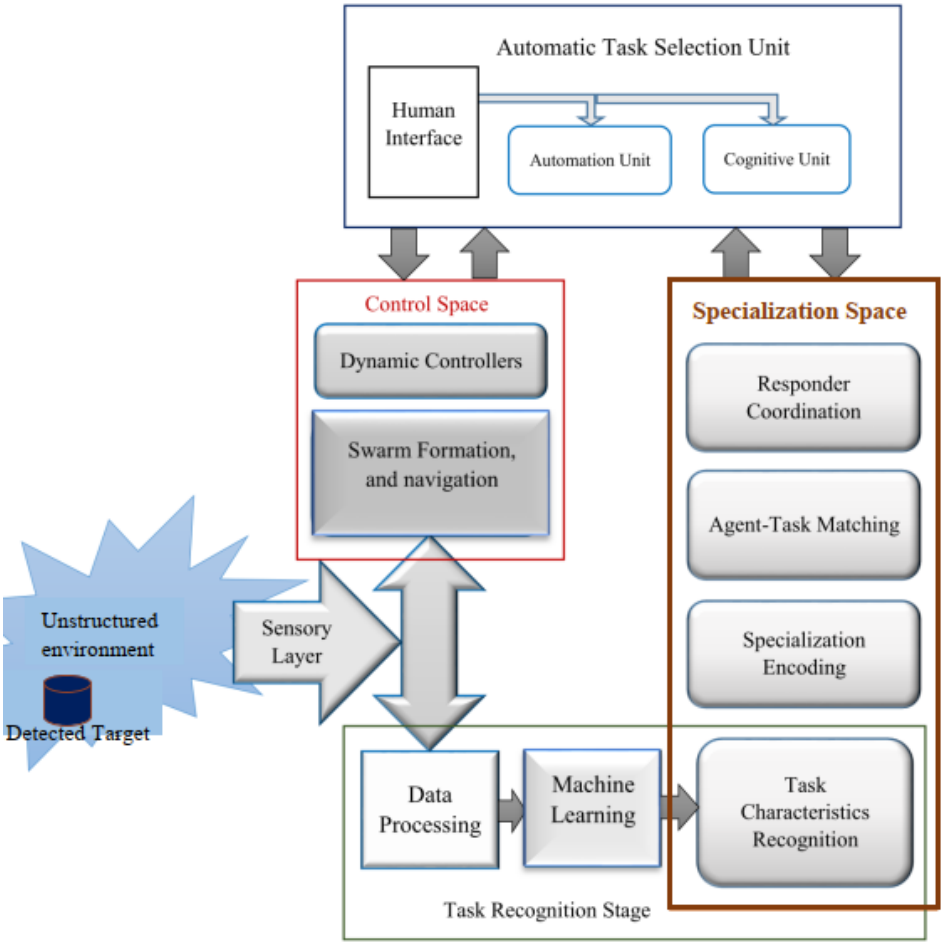


Figure 4.2 Architecture of the proposed specialty-based task allocation framework.

4.3 Recognition of Task Characteristics

From a more general perspective, the proposed theoretical solution considers predefined Gaussian distributed spatial features of different classes, $\boldsymbol{X}_k$, to characterize target objects in a two-dimensional environment. In a typical scenario, while the agents navigate the workspace, an embedded perception system collects information from the surrounding unstructured environment and extracts data from the dominant classes, $\boldsymbol{X}_k$, on the observed targets. Then, the detected features are categorized through a classification mechanism. This provides information to the task-agent matching stage that is discussed in section 4.4. The latter determines a probabilistic confidence level in the fit between the detected target and any specialized agent.

4.3.1 Target Object Recognition

To map the appropriate agents to the corresponding tasks independently, the dominant features on the considered target objects are separated into classes. The group of features on $\boldsymbol{X}_k$ that can be observed on each target type are encoded as a vector of F sample features, that is $\boldsymbol{X}_k = \{x_j : j = 1, \cdots, F\}$, where $x_j \in \mathcal{R}^{\boldsymbol{2}}$ is a Gaussian distributed random sample of a two-dimensional spatial feature that is observed with mean μ and variance σ^2. The observed sample features $x_j \in \boldsymbol{X}_k$ are assumed to be drawn independently and can be used to estimate the parameters, μ and σ^2, by maximum likelihood function, whereas the probability of the sample features x_j is maximum at the mean value μ, which is the central point of the feature. This property is a unique property of the standard Gaussian distribution. Therefore, the Gaussian distribution is considered the most descriptive form to mimic the real distribution of practical features of the target tasks to be considered in this study. Alternative distributions of the features would still support the operation of the proposed task allocator. However, a non-Gaussian distribution may influence the outcomes

of the task allocation process by assigning different agents to a detected task. Such a behavior would be coherent with a different perception of the target task nature, associated with a different distribution of the features that characterize that task.

Under the assumption of a Gaussian distribution, we consider that $k = 1,2, \dots, T$, where T is the maximum number of the different target objects classes that are expected to be detected on a task space. The total number of distinctive features to be estimated creates a set of vectors, here called the task space, $\boldsymbol{X} = [\boldsymbol{X_k}: k = 1,2, \dots, T]$. It is also assumed that each group of features, $\boldsymbol{X}_k$, is associated with one specific class, C_i, of one target object type. These classes are considered available for training during the target recognition stage on a number of predefined target objects. Moreover, each class is associated with an action of a specific nature to be performed by a specialized capability of a given robotic agent. The feature space is therefore comprised of $F \times T$ predefined features that characterize T possible target objects, and the latter are associated with the available specialized capabilities of the robots in the swarm. The confidence level in the recognition of these target objects, estimated as a probability, is leveraged in a matching scheme that computes the specialty fitting probabilities of the individual agents with respect to each type of target.

A pre-trained network with a set of T classes, C, where $C \in \{C_i: i = 1,2, \dots, T\}$, is used to retrieve an observed group of features, $\boldsymbol{X}_k$, from the visual perception of a given target. For example, in practice these classes can be pre-trained in a deep learning network from the sample image datasets representing them in order to perform target object recognition, which is reported in the experimental results. This leads to the posterior probability, $P(C_i|\boldsymbol{X}_k)$, that represents the probability that the observed features, $\boldsymbol{X}_k = \{x_j: j = 1, \cdots, F\}$, characterizing each class C_i, estimated by the Bayesian rule [132]:

$$P(C_i|\boldsymbol{X}_k) = \frac{p(\boldsymbol{X_k}|C_i)P(C_i)}{p(\boldsymbol{X_k})} \tag{4.1}$$

with $i = 1,2, \ldots, T$, and where $p(\boldsymbol{X}_k|C_i)$ is the class-conditioned probability density function that describes the Gaussian distribution of the group of features, $\boldsymbol{X}_k$, in each predefined class C_i; $P(C_i)$ is a prior probability of the class C_i, which is evaluated from the given training dataset. If M_{tr} is the total number of patterns that are available for training and M_{tr_i} of them belong to C_i, then a prior probability of this class can be computed as:

$$P(C_i) = \frac{M_{tr_i}}{M_{tr}} \tag{4.2}$$

$p(\boldsymbol{X}_k)$ is the probability density function of the group of features $\boldsymbol{X}_k$ over all classes, which is given by:

$$p(\boldsymbol{X}_k) = \sum_{i=1}^{T} p(\boldsymbol{X}_k|C_i)P(C_i) \tag{4.3}$$

This probability density function of the features $p(\boldsymbol{X}_k)$ is evaluated for all classes and does not affect the evaluation of the posterior probability, $P(C_i|\boldsymbol{X}_k)$, in Eq. (4.1) since all quantities are a function of C_i [132]. As such, the denominator in Eq. (4.1) can be considered a normalization constant, substituted by $\frac{1}{\xi}$, to ensure that the posterior distribution on the left-hand side integrates to one. Thus, the posterior probability is evaluated as:

$$P(C_i|\boldsymbol{X_k}) = \xi\, p(\boldsymbol{X}_k|C_i)P(C_i) \tag{4.4}$$

As a result, the estimated features, $\boldsymbol{X}_k$, recognized on the target objects are associated to the classes that have the maximum estimated posterior probability, Eq. (4.4). The estimated posterior probabilities, considered to be detection confidence levels of the observed targets, are used to

evaluate the specialty fitting probability of the proper agent/agents that to be allocated to the detected task, as detailed in section 4.4.

4.3.2 Probability Densities Estimation

The available data can be used to estimate the probability density functions (PDFs). To evaluate the parameters of the probability density function, such as its mean, μ, and variance, σ^2, the maximum likelihood function is used.

It is assumed that a group of features, $\boldsymbol{X}_k$, is distributed according to the class-conditioned probability density function $p(\boldsymbol{X}_k|C_i)$ for T classes. In addition, the class-conditioned probability density function, $p(\boldsymbol{X}_k|C_i)$, should be a likelihood function given in a parametric form, and the parameter vector, $\boldsymbol{\alpha} = [\mu,\ \sigma^2]^{\boldsymbol{T}}$, represents the corresponding unknown parameters of the extracted class. To highlight the dependence on the unknown parameter vector, $\boldsymbol{\alpha}$, the class-conditioned probability density function can be written in the form $p(\boldsymbol{X}_k|C_i;\boldsymbol{\alpha})$. This likelihood function can be used to estimate these unknown parameters by exploiting the available set of known data in each class. In theory, it is assumed that the classes are mutually independent and the parameter estimation problem can be solved for each class independently. The practical interpretation of this assumption is that the specialization of each robot allows for the performance of substantially different functions (e.g. inspect, explore, grab, operate on specific terrains or different environmental conditions, etc.), and is therefore related to substantially different task characteristics which ensure the required independence.

If $\boldsymbol{X}_k = \left[x_j : j = 1, 2, \ldots, F\right]^T$ is consisting of F statistically independent random sample features that are drawn from the probability density function $p(\boldsymbol{X}_k; \boldsymbol{\alpha})$, then we should have the following likelihood function of the parameter $\boldsymbol{\alpha}$ with respect to $\boldsymbol{X}_k$:

$$p(\boldsymbol{X}_k; \boldsymbol{\alpha}) = \prod_{j=1}^{F} p(x_j; \boldsymbol{\alpha}) \tag{4.5}$$

The estimation of the parameter's vector $\boldsymbol{\alpha}$ can be evaluated by using this function so that the maximum likelihood function takes its maximum value at $\boldsymbol{\alpha}$:

$$\widehat{\boldsymbol{\alpha}}_{\boldsymbol{ML}} = arg \max_{\boldsymbol{\alpha}} \prod_{j=1}^{F} p(x_j; \boldsymbol{\alpha}) \tag{4.6}$$

where $\boldsymbol{\alpha}$ is chosen as the value that maximizes the likelihood function $p(X_k; \boldsymbol{\alpha})$ which provides the necessary uncertainty level of the parameter $\boldsymbol{\alpha}$ [132].

4.4 Task-Agent Matching Scheme

The task-agent specialty matching problem, which is emphasized in this thesis, consists of matching the best suited specialized agents with their corresponding target objects, or tasks, with a maximum fitting probability. In practice, for a given matching assignment, an agent responds to a specific task when the agent's specialty offers a sufficient fitting probabilistic score with the task requirements. However, a given agent can also qualify for different tasks but with different fitting probabilities. The proposed task-agent matching approach leverages the target task recognition formulation introduced in section 4.3. It comprehends two sub-systems: 1) to evaluate the specialty fitting probability between the detected task and specialized agents; and 2) to allocate the most specialized and available robots to the detected task.

4.4.1 Specialization Encoding

In this section the individual agents' specialized capabilities are defined in two forms that are: a general binary definition of agents' specialization to serve as the basis for task-agent association. Then, the general binary form is further refined with a modulated definition of the agents' specialization based on their mechanical construction or embedded hardware or software.

A. Binary Specialization Encoding

The swarm of robots $\{R_i, i = 1,2, \dots, a\}$ consists of a specialized individual agents, R_i, and provides T different specialized roles or capabilities of each agent defined in a generic form as a binary specialty vector, $\boldsymbol{S}_i$: $\{s_k, k = 1,2, \dots, T\}$. Each entry of $\boldsymbol{S}_i \in \mathcal{R}^{1\times T}$ that is defined as $s_k = 1$ means that the robot possesses the corresponding capability; and $s_k = 0$ indicates that the robot is not equipped with the necessary capability to tackle a target type of a class C_k extracted from an observed group of features X_k that is meant to correspond to that class C_k, among T of them. This task allocator is called binary encoded task allocator (BETA). As an example, let us assume a firefighting scenario in which the individual agents of a robotic swarm are specialized to fight fires over large residential complexes, skyscrapers, small houses, and warehouses. The dominant classes of the task will be residential complex, skyscraper, house, and warehouse. In this case, the specialty vector of each agent, R_i, is defined as $\boldsymbol{S}_i \in \mathcal{R}^{1\times T}$, with $T = 4$ considered target types.

This binary definition of specialization is used as the basis of the expanded probabilistic form, which will be introduced in section 4.4.2. In the following section the definition of agents' specialization is presented in a non-binary form when different levels of sophistication of the agents' specialized capabilities are necessary based on the agents' mechanical construction or embedded hardware or software to capture different levels of suitability of the agents' specialization to different tasks.

B. Modulated Specialization Encoding

In order to further evolve the framework and provide additional flexibility in the modulation of robotic agents' competencies, the binary definition of the agents' specialty vector, $\boldsymbol{S}_i$, introduced in the previous section, can also be modulated in a non-binary form. This modulated representation of the agents' specialization can be adopted when increased sophistication of the task allocation scheme is either necessary or permitted by the application. Human beings may be considered as an example, given that individuals have many talents but some are more developed than others, and some are physically stronger or in better shape than others. For example, if a group of people enter a marathon, all of them may be able to complete the marathon successfully. However, they would not finish at the same time because they do not have equal training for it or they might be physically different. From this example we can realize that the objective of modulating the specialized capabilities is to mimic the nature and robustness of the individual agents' capabilities to different levels of their functional robustness with respect to corresponding targets. Therefore, this reformulation as a modulated specialization encoding is a fair representation of the potential of the individual agents, whereas the binary only specialty vector is not sufficient to represent such variable levels of the agents' specialized capabilities. The specialty vector, $\boldsymbol{S}_i$: $\{\mathrm{s}_k, k = 1,2, \dots, T\}$, can be modulated to capture relative specialty levels based on the agents' corresponding embedded functional qualities. Let us assume that the entries of $\boldsymbol{S}_i \in \mathcal{R}^{1\times T}$ are modelled to take values in the continuous range of 0 to 1, such that $\mathrm{s}_k \in [0\ \ 1]$, to represent the relative levels of the agents' specialized capabilities from the least efficient (0) to the most efficient capability level (1). This task allocator is called modulated encoded task allocator (META).

4.4.2 Probabilistic Matching

The goal of the matching scheme is to maximize the task-agent fitting probability that represents the match between confidence levels on the detected task constraints (target's features) and the agents' specialized capabilities. The confidence levels on the current target's detection that would be leveraged by the swarm's individuals are defined as:

$$\hat{\varphi}_{R_i} = \boldsymbol{S}_i \widehat{\boldsymbol{P}}_T \tag{4.7}$$

where $\hat{\varphi}_{R_i} \in \mathcal{R}^{1\times 1}$ represents the specialty fitting confidence level achieved by an individual agent of identity, i, based on the constraints raised by the detected classes, which correspond to given target objects. $\widehat{\boldsymbol{P}}_T \in \mathcal{R}^{T\times 1}$ represents the probability transition vector of the detected classes. This is a function of the estimated posterior probabilities, Eq. (4.4), of the detected features on the target object, which is given by:

$$\widehat{\boldsymbol{P}}_T = \left[\sum_{k=1}^{T} P(C_1|\boldsymbol{X}_k) \quad \sum_{k=1}^{T} P(C_2|\boldsymbol{X}_k) \quad \cdots \quad \sum_{k=1}^{T} P(C_T|\boldsymbol{X}_k)\right]^T \tag{4.8}$$

As a result, the specialty fitting confidence level, $\hat{\varphi}_{R_i}$, Eq. (4.7), of a robot R_i, emphasizes that each agent offers a specific level of competency with respect to the confidence levels of the recognized classes on a detected target object, corresponding to a given task.

To formulate the task-agent specialty matching in a probabilistic form, the concepts of probability theory are adopted [133]. The three main elements of the probabilistic model are:

1. An agent's sample space, which represents the *set of all possible outcomes*. In this formulation, the sample space is represented by the encoded specialties of each robotic agent. The agent's specialities are defined in its specialization vector, $\boldsymbol{S_i}$, defined in sections 4.4.1.A and 4.4.1.B. The

possible outcome of each individual agent is modelled as φ_{R_i}, which is discussed later on, Eq. (4.10).

2. An events spaces of the individual agents represents a set of outcomes in the agent's sample space. The outcome is defined, in the probability theory, to be the result of the trial or experiment at hand. In this model, the outcome is introduced as an agent's specialty fitting confidence level, $\hat{\varphi}_{R_i}$, on each detected target, which is a function of the agent's specialization vector $\boldsymbol{S}_i$ and uncertain measurements of target object detections, introduced by Eq. (4.7).

3. A probability function gives each event in the agent's event space a probability; the latter is a real number between 0 and 1. The computation of the event's probabilities within the agents' sample spaces can be given as:

$$\text{Fitting probability of each individual agent} = \frac{\text{The outcome of the agent's event space, } \hat{\varphi}_{R_i}}{\text{All possible outcomes of the same agent, } \varphi_{R_i}}$$

As a result, the *probabilistic* swarm's specialty matching matrix, $\boldsymbol{Q} \in \mathcal{R}^{a \times a}$, can be introduced as:

$$\boldsymbol{Q} = \begin{bmatrix} \frac{\hat{\varphi}_{R_1}}{\varphi_{R_1}} & 0 & \cdots & \cdots & \cdots & 0 \\ 0 & \frac{\hat{\varphi}_{R_2}}{\varphi_{R_2}} & & & & \vdots \\ \vdots & & \ddots & & & \vdots \\ \vdots & & & \ddots & & \vdots \\ \vdots & & & & \frac{\hat{\varphi}_{R_{a-1}}}{\varphi_{R_{a-1}}} & 0 \\ 0 & \cdots & \cdots & \cdots & 0 & \frac{\hat{\varphi}_{R_a}}{\varphi_{R_a}} \end{bmatrix} \tag{4.9}$$

where $\varphi_{R_i} \in \mathcal{R}^{1\times 1}$ is the agent's maximum expected collective score that results when all of the agent's capabilities are matched with their corresponding detected target's classes. $\boldsymbol{Q}$ is defined in matrix form because the objective is to estimate the individual specialty matching probability for each agent, as appears in Eq. (4.16). To define φ_{R_i}, the maximum number of specialized

capabilities that are built in each individual agent have to be considered. Then the maximum expected collective score, φ_{R_i}, of the agent R_i can be defined as:

$$\varphi_{R_i} = \boldsymbol{S}_i \boldsymbol{p}_{max} \tag{4.10}$$

and,

$$\boldsymbol{p}_{max} = [p(\boldsymbol{X}_1|C_1) \quad p(\boldsymbol{X}_2|C_2) \quad \cdots \quad p(\boldsymbol{X}_T|C_T)]^T \tag{4.11}$$

where $p(\boldsymbol{X}_k|C_i)$ is the class-conditional probability of $\boldsymbol{X}_k$ in its predefined class. In practice, it is realistic to consider $\varphi_{R_i} = \sum_{k=1}^{T} s_k : \{s_k \in \boldsymbol{S}_i\}$, which is the sum of the encoded basis of all agents' specialized capabilities.

4.4.3 Qualified Responders Coordination

It is important to support realistic scenarios in which a robot may not always be available because of a dead battery or damage that inhibits the robot from conducting the service. Alternatively, a robot might already be allocated to another task, or located at a far distance from the target position when called to service. To tackle such problems, the proposed framework addresses the agents' availability and attendance beyond their specialty. Therefore, a scheme for the best responder assignment coordination must consider the swarm members' availability states and attendance levels, denoted as $\boldsymbol{\vartheta}_{Av}$ and $\boldsymbol{\vartheta}_{Att}$ respectively, along with the probabilistic swarm's specialty matching matrix, $\boldsymbol{Q}$, defined in Eq. (4.9). The agents' availability, $\boldsymbol{\vartheta}_{Av}$, is a necessary attribute of this framework to be reliable to a great extent in terms of accomplishing the mission goals by replacing the assignments of unavailable agents with alternative available agents. Moreover, the consideration of agents' attendance, related to task-agent distance separation, also provides an important advantage to the framework for being adaptable to applications with wider range workspaces. Adaptability of the framework is defined here as an attribute referring to the

applicability of the proposed framework to suit different applications. This attribute is embedded in the proposed framework by the system's designer based on the application's demand. The importance of the agents' attendance is discussed in more detail later in this section.

The objective of the coordination scheme is to return the specialty fitting probabilities of only the available and attending agents. Thus, although it may be less competent, the most qualified agent among this group can be assigned to the detected task based on its availability and attendance at the time of target object discovery.

The availability vector, $\boldsymbol{\vartheta}_{Av} \in \mathcal{R}^{a\times 1}$, is defined based on the current internal status of each robot. At the time of swarm deployment, the internal flag of the deployed agents is raised to "available", while the internal flag of agents that are not available is set to "withdrawn". Then, whenever the system finds an "available" agent that is qualified for the detected task, the availability flag keeps its task allocation's fitting probability active and the detected task is assigned to this agent if the probability is optimal. In contrast, agents with an internal flag set to "withdrawn" or changed to "withdrawn" due to a battery depletion or mechanical failure are deactivated, and then the task allocator needs to find an alternative "available" agent with proper qualification to accomplish the mission goal. In addition, when an available agent is assigned to a given task, then its availability state is changed to "busy", which means that this agent is temporarily withdrawn because it is already assigned to a task. The proposed availability status vector of the robots, $\boldsymbol{\vartheta}_{Av} \in \mathcal{R}^{a\times 1}$, is defined as:

$$\vartheta_{Av_i} = \begin{cases} 1, & |R_i \quad is\ "available" \\ 0, & |R_i \quad is\ "withdrawn"\ or\ "busy" \end{cases} \tag{4.12}$$

A diagonal attendance matrix, $\boldsymbol{\vartheta}_{Att} \in \mathcal{R}^{a\times a}$, is defined based on the distance, d_i, between the current position of each robot and the position of the detected target. In addition, it also considers

the velocities of the individual robotic agents, v_i, and the velocity of the detected target, v_t. The distance, d_i, is taken into consideration with the objective to optimize the response of the specialized individuals of the swarm over time. Accordingly, the agents that are closer to the target position when it is detected are promoted by assigning them higher task allocation probabilities compared to agents that are located farther away from the target position as well as to increase the system's capacity. The increase in the system's capacity means that the number of agents that are qualified to perform the task is increased. Additionally, the robots' velocities are considered in the case of moving targets to ensure that the velocities of the available agents are sufficient for them to reach the detected moving targets. This issue is discussed in section 3.6 with the example presented in Figure 3.13. The respective robots' attendance levels are evaluated at the moment a target object is detected. The role of the corresponding weights is to boost the robots' specialty fitting probabilities and increase the chance of assigning the closer agents with higher fitting probabilities. Similarly to the definition of $\boldsymbol{Q}$ in Eq. (4.9), the attendance matrix, $\boldsymbol{\vartheta}_{Att}$, is defined in matrix form to support the estimation of individuals' specialty matching probabilities for each agent, as detailed in Eq. (4.16). The attendance matrix of the swarm is formed as:

$$\boldsymbol{\vartheta}_{Att} = \begin{bmatrix} \vartheta_{Att_1} & 0 & \cdots & \cdots & \cdots & 0 \\ 0 & \vartheta_{Att_2} & & & & \vdots \\ \vdots & & \ddots & & & \vdots \\ \vdots & & & \ddots & & \vdots \\ \vdots & & & & \vartheta_{Att_{a-1}} & 0 \\ 0 & \cdots & \cdots & \cdots & 0 & \vartheta_{Att_a} \end{bmatrix} \tag{4.13}$$

where,

$$\vartheta_{Att_i} = \begin{cases} \dfrac{\overline{\mathrm{T}}\left(\epsilon\left(\dfrac{v_i}{v_{i_{max}}}\right) + \left(\dfrac{r_{task}}{d_i}\right)^{\Upsilon}\right)}{2^{\epsilon}} & \begin{cases} \Upsilon = 1\ \{\ d_i > r_{task} \\ \Upsilon = 0\ \{\ d_i \leq r_{task} \\ \epsilon = 1\ \{moving\ target\ \text{AND}\ v_t\ is\ positive \\ \epsilon = 0\ \{static\ target\ \text{OR}\ v_t\ is\ negative \\ \overline{\mathrm{T}} = 1\ \{\ v_t < v_{i_{max}} \\ \overline{\mathrm{T}} = 0\ \{\ v_t \geq v_{i_{max}} \end{cases} \\ 0\ , & \{R_i \quad is\ \text{"}withdrawn\text{"}\ \text{or}\ \text{"}busy\text{"} \end{cases} \tag{4.14}$$

and $\vartheta_{Att_i} \in [0\ 1]$, d_i is the Euclidean distance between the current location (x_i, y_i) of the robot R_i and the location of the detected target (x_T, y_T) in their shared 2-D plane, which is given by:

$$d_i = \left|\sqrt{(x_i - x_T)^2 + (y_i - y_T)^2}\right| \tag{4.15}$$

where v_i is the current linear velocity of the robot R_i, $v_{i_{max}}$ is the maximum pre-initialized limit of the linear velocity of a robot R_i, v_t is the linear velocity of the detected target. v_t is positive when the target moves in the direction of agent's movement, and it is considered negative when it moves in an opposite direction with respect to the agent's movement. $\overline{\mathrm{T}}, \epsilon$, and Υ are control variables that take binary values, 1 or 0, to activate or omit the impact of the robots' velocity and the distance to the target's location based on the sensors' output. The binary value of $\overline{\mathrm{T}}$ is defined based on the comparison between the estimated velocity of the target and the maximum predefined velocity of each agent. This variable plays a prominent role to even consider the impact of the agents' attendance, to boost the task allocation probability, or to omit it. In the case that $v_t \geq v_{i_{max}}$ then $\overline{\mathrm{T}}$ equals 0 so the weight of the agent's attendance term in Eq. (4.14) will be 0. Thus, the target cannot be reached as long as its velocity is higher than or equal to the agent's maximum velocity. The developments of the proposed framework provide a formulation in Eq. (4.14) to address all of these considerations. However, the discussion in this thesis is limited to the cases of static targets. Therefore, the estimation of the moving target's velocity is conditioned to an advanced robust sensory layer which is beyond the scope of this thesis.

The impact of the distance, $\frac{r_{task}}{d_i}$, in the agent's attendance, Eq. (4.14), is rounded to 1 by the control variable Υ when the agent is available within the zone of the detected target with $d_i \leq r_{task}$. In this case, the agents that are very close to the target's location must be promoted with a higher chance to be allocated to the detected tasks. r_{task} is the radius of the designated task zone of influence which is selected by the system's designer, as defined in section 3.5.2-C.ii. Also, the binary value of the control variable ϵ is defined in Eq. (4.14) based on the target's velocity and its direction with respect to the direction of the individuals' movements. Whereas this thesis presents a generalized speciality-based task allocation framework, the contribution of the agents' attendance can be chosen, by the system's designer, based on the application demand. For applications that have a wide area workspace such as military, outdoor search-and-rescue, and firefighting operations, the agents' attendance is critical. In these applications, considering that the targets can be detected over a wide area, the most qualified agent (or the first qualified responder) needs time to reach the target. However, if another agent with a lower level of competence is closer to the target, this agent should be given the opportunity to attend the task earlier. This closer agent may be adequate to provide a specific level of service to tackle the situation until the arrival of agents that have more extensive competencies. Therefore, the agents' attendance plays a critical role in weighting the individuals' specialty-based task allocation probabilities, especially when targets must be reached within a reasonable time period, for example to avoid major damage in firefighting or to save lives in search-and-rescue, as well as to present a level of resistance in military applications. In contrast, for applications that are limited to small workspaces such as indoor search-and-rescue applications and service robots, the agents are usually available within the target's zone of influence and the robotic agents' attendance factor may neglected.

Consequently, the overall task-agent coordination scheme can be synthesized as a summation of two components: fixed specialty fitting probabilities, $\boldsymbol{Q}$, weighted by ℘, and a varied agents' attendance, $\boldsymbol{\vartheta}_{Att}$, weighted by $(1 - ℘)$. The overall function is further filtered by the availability status of the agents. Therefore, the overall task allocation scheme is defined as:

$$\boldsymbol{\Psi} = (℘\mathbf{Q} + (1 - ℘)\boldsymbol{\vartheta}_{Att})\boldsymbol{\vartheta}_{\boldsymbol{Av}} \tag{4.16}$$

where $\boldsymbol{\Psi} \in \mathcal{R}^{a\times 1}$ returns the task allocation fitting probabilities for each available agent, or 0 for withdrawn or busy units, with respect to the current detected task. The parameter $℘ \in [0\ 1]$ is defined by the system's designer to distribute the weight of the overall task allocation fitting probabilities between $\mathbf{Q}$ and $\boldsymbol{\vartheta}_{\boldsymbol{Att}}$ based on the nature of the application. The parameter ℘ is adjusted by the system's designer to enhance this framework with more flexibility to adapt to different applications. For example, in applications that require a higher level of specialty matching between the agents' specialized capabilities and the corresponding targets over a limited size workspace, such as service robotics, then the agents' specialized functionalities matter the most. Therefore, ℘ is selected to be equal to 1. However, in emergency cases for such applications where an agents' prompt attendance is critical, then ℘ can be set to lower values. In applications that cover wider workspaces and the period of time required to reach the task is important, such as law enforcement, fire extinguishing operations, or outdoor search-and-rescue applications, ℘ is preferably selected in a range $0.7 \leq ℘ \leq 0.9$ or lower, which will give an opportunity for the agents' attendance weighted by $(1 - ℘)$ to boost the overall task allocation fitting probability.

4.4.4 Human on the Loop

For a more responsive dynamic operation of the proposed approach, a human supervisor is given a role to initialize or to modify the system's operational conditions. However, the human

does not need to intervene frequently as the automated process responds autonomously. From the literature, having "a human in the loop" [121] means that a person has complete control over starting or stopping any action that is to be performed by the system after receiving the human cue. Instead, this framework moves the design toward a "human on the loop" standard, pushing human control farther from the centre of the automated decision-making. In this framework which gives humans oversight of the automated system [48], the level of human intervention [122] is minor whereas the automated process would jump right into action and not need frequent human interventions as it otherwise would with a "human in the loop" design.

For example, in the proposed approach the human supervisor can initialize or control a threshold involved in the task allocation, as addressed in this section, but does not need to change it frequently.

A minimum fitting threshold (MFT), η, is implemented as a safety measure that guarantees a minimum fitting probability below which no agent will be selected. To control this parameter, a human operator who supervises the swarm sets the MFT value for the task allocation system, either before the deployment of the swarm or during the operation. The MFT can vary to different levels based on the requirements of the application or in association with the operational conditions. This way, human skills and understanding of the situation can be shared with the robots by changing this parameter, and the minimum required level of trust in the recognized target objects can be influenced.

As one of the overall objectives of the proposed solution is to reduce the cognitive load on the operator, pre-setting the distribution of MFT over two ranges can reduce the number of times the human operator needs to intervene on the MFT set point. Therefore, the desired MFT, η, is defined as $\eta \in (0\ 1]$, but is distributed over two predefined ranges corresponding to a low specialty fitting

level (LSFL) and a high specialty fitting level (HSFL). The lower limit of LSFL, $\eta \in (0\ B]$, drives the task-agent allocation scheme to match the very minimum specialized capabilities of the available agents to respond to the detected targets. However, in many situations a higher level of specialty-based task allocation must be ensured to fit the most capable agents with the requirements of the detected task. In such a case, the task allocator is enforced by the human supervisor to work in the HSFL range, $\eta \in (B\ 1]$, that is by setting η above a specific level B to ensure that only robots with a higher level of competence can intervene, where:

$$\begin{cases} LSFL: & 0 < \eta \leq \mathrm{B}, \\ HSFL: & \mathrm{B} < \eta \leq 1, \end{cases} \tag{4.17a}$$

For applications that require an influence from the agents' attendance, $\boldsymbol{\vartheta}_{Att}$, with $\mathfrak{p} < 1$ in Eq. (4.16), to boost the specialty fitting probability, $\boldsymbol{Q}$, the low specialty fitting level (LSFL) is conditioned to $(1 - \mathfrak{p}) < \eta$ which results in:

$$\begin{cases} LSFL: & (1 - \mathfrak{p}) < \eta \leq \mathrm{B}, \\ HSFL: & \mathrm{B} < \eta \leq 1, \end{cases} \tag{4.17b}$$

This formulation ensures that the maximum level of the agents' attendance, $\boldsymbol{\vartheta}_{Att}$, would only boost the fixed specialty-based task allocation probability, $\boldsymbol{Q}$, but it cannot initiate the task allocation process without an existing level of the agents' specialty.

Therefore, $\boldsymbol{\Psi}$, defined in Eq. (4.16), is further refined in Eq. (4.18) to only consider the task allocation fitting probabilities of the available agents that achieve the desired MFT. The fitting probabilities of the available responders, $\boldsymbol{\Psi}_{MFT} \in \mathcal{R}^{a\times 1}$, are given by:

$$\boldsymbol{\Psi}_{MFT} = [\Psi_{MFT_1}, \Psi_{MFT_2}, \cdots, \Psi_{MFT_a}]^T \tag{4.18}$$

where,

$$\Psi_{MFT_i} = \begin{cases} \Psi_i, & \mid \Psi_i \geq \eta : \Psi_i \in \boldsymbol{\Psi} \\ 0, & \mid \Psi_i < \eta : \Psi_i \in \boldsymbol{\Psi} \end{cases} \tag{4.19}$$

and with $i = 1,2, \dots, a$. Accordingly, the qualified available agents are automatically selected and allocated to the detected tasks considering the human supervisor's strategic guidance. The identification index of the best-suited and available agent above the MFT is given by:

$$\emptyset_{BEST\ RESPONDER\ INDEX} = i \mid i \in max\{\boldsymbol{\Psi}_{MFT}\} \tag{4.20}$$

The function of the max operator is to extract either the identification index of the agent with the highest fitting probability, or else a group of identification indexes if more than one agents share the same fitting probability and the latter is maximum. In the case where more than one agent shares the highest score, the max operator extracts their identification indexes sequentially based on their order in $\boldsymbol{\Psi}_{MFT}$. In this thesis, when multiple targets are detected simultaneously, this process of Eq. (4.20) runs iteratively and individually on each detected target in order to allocate the most competent agent to each detected target. Alternatively, when the goal is to allocate all of the qualified competent agents to the detected target, then the process of Eq. (4.18) is considered, whereas an iterative loop runs inside the vector, $\boldsymbol{\Psi}_{MFT}$, in Eq. (4.18) and extracts the identification indexes of all agents with fitting probabilities that are higher than η.

4.4.5 Specialty-Based Task Allocation on Complex Tasks

Complex tasks are characterized by simultaneous detection of multiple classes, and the combination of these classes creates the target task. To deal with such tasks, the proposed task allocation approach must be more selective and deterministic in response to the individual classes involved in these complex tasks.

The development explored in this section could open doors for future in-depth developments of the proposed framework when the robots' specialized capabilities must be involved in the task allocation process based upon different priorities. To advance the development of the proposed approach, a prioritization scheme is introduced in this section. While the concept is proposed here,

the functionality of this prioritization scheme is conditioned to the availability of an advanced sensing stage. However, this advanced sensing stage has not been fully achieved in this work beyond simulation. As such it remains beyond the scope of this thesis from a practical implementation perspective.

The concept of the prioritization scheme is designed to work in parallel with the original specialty-based task allocation scheme for the purpose of further filtering the agents' task allocation priorities based on their primary capabilities. The original task allocation scheme, presented in previous sections of 4.4, continues to evaluate the task-agent specialty-based task allocation probability, but it now considers primary and secondary competencies. In other words, this enhanced prioritization scheme associates the task allocation priority to agents possessing functional competencies that are considered dominant, or primary, in a given context of operation, while preventing the agents that do not possess the primary capabilities to be totally excluded from the operation upon the detection of a given target, as these less qualified agents may still intervene and contribute important support or complementary functionalities. The main benefit of this evolved scheme is that it preserves the evaluation of the overall task allocation probability for the qualified agents based on all of the agents' primary and secondary competencies. Then, the final selection for the task allocation emphasizes and is conditional upon the qualified agents' primary capabilities. This adds more restrictions to the potential assignments of less competent agents and permits finer control of the swarm's coordination.

This prioritization approach is proposed to use the simulated sensor's information to automatically adjust the agents' assignment priorities based on the understanding of the agents' capabilities with respect to the detected tasks. This approach plays a prominent role in responding to safety problems that lead to risky task allocation operations, such as incorrect assignments that

might result in a collision of the robotic agents. The process of switching the task allocation scheme to the prioritization mode can be activated automatically based on the sensor's input.

For example, let us consider a complex outdoor scenario in which the individual agents of a robotic swarm are specialized to perform specific tasks such as picking up a box, or rescuing a person, and these tasks are to be performed on land or on water-covered areas. The classes of the tasks considered will be land, water, box, and person. In this scenario, the individual agents of a robotic swarm have different capabilities of either picking up a box, or rescuing a person, with these tasks to be performed on land or on water-covered areas. The multiple classes of the potential task are {land, box}, {land, person}, {water, box}, and {water, person}. In this task allocation scenario, the swarm of robots consists of four robotic agents $\{R_i: i = 1, \cdots, 4\}$. The definition of the specialty vector of each agent R_i is the one introduced in section 4.4.1 as $\boldsymbol{S}_i \in \mathcal{R}^{1\times T}$, with $T = 4$ considered classes. Now, let us assume that, for safety purposes, the agents' capability, such as performing tasks on water-covered areas, is categorized as the agents' primary capability, whereas the agents' capabilities to perform tasks on land to pick up a box or rescue a person are categorized as the secondary capabilities. Targets such as boxes or persons can be detected on land or over water surfaces, and the agents can respond to these tasks. However, only some of these agents can succeed in navigating and working on water-covered areas while others will sink into the water and thereby be destroyed. To protect these agents from such incorrect assignments, the task allocator must prioritize and allocate the agents based on the detection of all the classes that characterize the potential task while also considering the agents' primary capabilities. As a result, the agents that possess the primary capabilities with respect to the detected tasks will be given priority in performing the detected tasks. In contrast, the agents that do not possess the primary capabilities will be suspended.

To implement this solution, a primary selector is proposed to work in parallel with the original specialty-based task allocator (shown in Figure 4.2). The main function of the primary selector is to automatically redefine the essential definition of the individuals' specialty vector that is to be used in this scheme, $\boldsymbol{S}_i$, based on the detected classes of the task at hand and the pre-initialized mission's primary requirements. The mission's primary and secondary requirements are pre-initialized and embedded in the primary selector by the system's designer and set to "ON" or "OFF" based on the application demand. Let us first consider a case where the agent's specialization vector, $\boldsymbol{S}_i$, defined in section 4.4.1, is automatically redefined in this prioritization scheme based on the input received from the simulated sensors. $Prim$ is the index of the primary class in the vector, $\widehat{\boldsymbol{P}}_T$, which is the class that requires the robot to be equipped with the corresponding primary capability that is critical for the robotic operation at hand, whereas $\bar{\bar{q}}_{\rho_{Prim}} = \widehat{\boldsymbol{P}}_{T_{Prim}}$ means that for "ON" states of the primary selector the diagonal value of the matrix $\boldsymbol{B}_\rho$takes the corresponding scores of the detected class from the vector, $\widehat{\boldsymbol{P}}_T$, in Eq. (4.8). Then, upon the detection of the corresponding target's classes, a new $\boldsymbol{B}_\rho$ will be generated which will automatically encode the agents' specialization vector in this scheme for the purpose of only activating the agents that possess the primary capabilities and suspending the others in response to the allocation of the detected task.

To prioritize the robots that possess primary capabilities, the categorized specialization vector, $\boldsymbol{S}_{\rho_i} \in \mathcal{R}^{1\times T}$, is defined in this scheme as:

$$\boldsymbol{S}_{\rho_i} = \boldsymbol{S}_i \boldsymbol{B}_\rho \tag{4.21}$$

Then, $\hat{\varphi}_{R_i}$, in Eq. (4.7) is redefined in this scheme as:

$$\hat{\varphi}_{\rho_i} = \sum \boldsymbol{S}_{\rho_i} \tag{4.22}$$

Thus, the swarm's primary probability is defined in a vector form, $\boldsymbol{Q}_{\rho} \in \mathcal{R}^{1\times a}$, as:

$$\boldsymbol{Q}_{\rho} = \left[\frac{\hat{\varphi}_{\rho_i}}{\varphi_{\rho}} : i = 1, \dots, a\right] \tag{4.23}$$

where $\varphi_{\rho} = Trsc\{\boldsymbol{B}_{\rho}\} + \kappa$, κ is a very small constant which is added to avoid the division by zero in the cases when no primaries are detected and $Trsc\{\boldsymbol{B}_{\rho}\} = 0$. In addition, the value of κ should be selected to calibrate the minimum level of confidence on the primary class detection, $\widehat{\boldsymbol{P}}_{T_{Prim}}$, and the minimum level of trust on the agents' primary probabilities that are introduced by Eq. (4.23). The latter are conditioned to $\text{PRIM}_{min} = \frac{1}{1+2\kappa}$ in Algorithm 4.1. The calibration of κ and PRIM_{min} is discussed at the end of Appendix A. Eq. (4.23) returns the normalized probabilities of the agents that are qualified with the primary capabilities with respect to the detected target. Then, the robotic agents that possess the required primary capabilities with respect to the detected task can be activated for allocating the detected task using a binary activation matrix, $\boldsymbol{R}_{\rho} \in \mathcal{R}^{a\times a}$, based on a conditioned minimum level of the agents' resulting primary probability, $\text{PRIM}_{min} = \frac{1}{1+2\kappa}$. The activation matrix, $\boldsymbol{R}_{\rho}$, is introduced by the following algorithm:

Algorithm 4.1: Primary Capabilities Prioritization

Input: $[Q_{\rho_i}, i] = (\boldsymbol{Q}_{\rho}(i))$;
Q_{ρ_i} is the normalized probability of a robot, R_i, that possesses primary capabilities in the presence of the detected target, and:
i in $\{1, \dots, a\}$ and

Output: $\boldsymbol{R}_{\rho} = diag[\rho_l : l = 1 : a]$;
If $Q_{\rho_i} \geq \text{PRIM}_{min}$;
then $\rho_l = 1$;
else $\rho_l = 0$
end;
Return $\boldsymbol{R}_{\rho}$;

Finally, the task-agent assignment scheme, Eq. (4.18), is integrated with this prioritization layer to only activate the assignments of agents that possess primary capabilities with respect to the detected task, which is given by:

$$\boldsymbol{h}_{\rho} = \boldsymbol{R}_{\rho}\boldsymbol{\Psi}_{MFT} \tag{4.24}$$

where $\boldsymbol{h}_{\rho} \in \mathcal{R}^{1\times a}$ represents the fitting probabilities of the active agents based on their primary capabilities in the presence of the detected classes of a given task. The flowchart in Figure 4.3 details the process of prioritization for the robots' primary capabilities. The identification index of the best-suited and available agent is given by:

$$\emptyset_{BEST\ RESPONDER\ INDEX} = i \mid i\ \in max\{\boldsymbol{h}_{\rho}\} \tag{4.25}$$

Eq. (4.25) runs iteratively on every detected complex task in the case of multiple simultaneous task assignments on different targets. Alternatively, when the goal is to allocate all of the qualified competent agents to a detected complex task, the process of Eq. (4.24) is considered.

Upon the detection of a primary class of a given target object, then the switch that is shown at the bottom of Figure 4.3 switches the task allocation operation to give a role for the prioritization scheme to filter the task allocation probabilities of the agents that possess the corresponding primary capabilities and suspend the others. The switch is controlled by $Trac(\boldsymbol{B}_{\rho}) \geq 0.5$ which is the minimum confidence level the system can trust that can be detected on a target object of a single primary class.

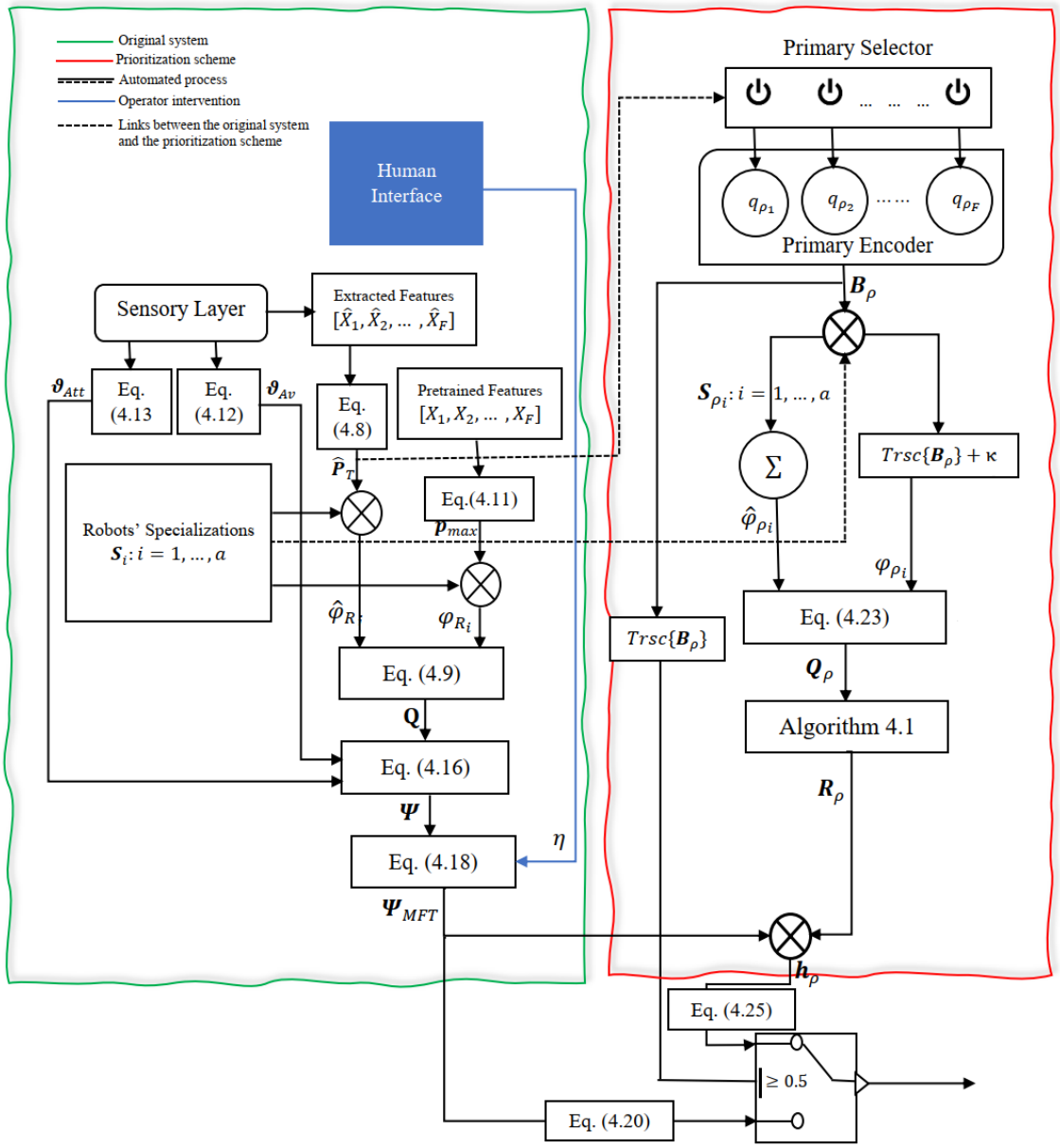


Figure 4.3 Prioritization of the robots' specialized primary capabilities.

4.5 Summary

This chapter addresses the design aspects of the proposed task allocation framework for a swarm of heterogeneous robotic agents in the specialization space. A representation for specializing individual members of a robotic swarm is introduced. The specialized capabilities of individual robotic agents are modelled in two forms. First, a general binary definition of agents'

specialization serves as the basis for task-agent association. Then, a non-binary definition of the agents' specializations encodes the agents' specialization vector in a non-binary form to modulate the agents' specialized functionalities based on the robustness of their hardware and software implementation and to capture different levels of suitability of the agents' specializations to different tasks. The specialized capabilities of individual robotic agents are matched to corresponding classes of visual features recognized on target objects with a quantified confidence level. That confidence level is associated with specific task requirements and is used to tune a novel task-agent probabilistic matching scheme. The latter computes the overall specialty-based task allocation probabilities of the individual agents for the purpose of assigning specialized agents to corresponding tasks that require specialized functionalities such as specific mechanical or instrumentation characteristics from autonomous robots.

For the purpose of a reliable speciality-based task allocation operation of the proposed framework, the specialized individual agents are coordinated with corresponding tasks while considering the agents' availability state along with their probabilistic specialty fitting level. The agents' availability represents a high-level reasoning capability of the proposed framework when it is required to adjust the task allocation process and select amongst the available qualified agents to pursue the mission goals. In addition, the implemented framework is expanded to coordinate heterogeneous robotic teams for wider workspace applications, and the agents' attendance levels are considered along with the agents' specialty fitting probabilities to compute the individual agents' overall task allocation probabilities. As such, the agents' attendance expands the adaptability of the proposed framework to capture wider workspace applications. In addition, the framework is extended to tackle the tasks that are characterized based on simultaneous detection

of multiple classes. A strategy which prioritizes the agents' competencies is proposed to further prioritize agents that possess primary capabilities with respect to the detected task. The framework also supports strategic guidance from a human operator to refine the task assignment process with situational awareness. The process is designed to keep the human cognitive load low while adjusting the system's operational conditions at a high level of coordination to result in a safer and more selective task allocation operation.

The proposed framework, introduced in this chapter, is implemented in MATLAB. The simulation results are extensively investigated and analyzed in Chapter 5.

Chapter 5 Experimental Validation in Simulation

5.1 Introduction

To validate the proposed framework, MATLAB simulation experiments are conducted on the advanced specialty-based task allocation framework that is introduced in Chapter 4 with an attempt to cover all types of robots in the level of their specialization space. This simulator is implemented to validate the generalized proposed approach, which addresses the concept of the specialty-based task allocation for teams of heterogeneous individuals of unmanned systems in the level of the individual agents' specializations. This simulator is independent from the simulator that covers the early developments of the proposed approach (detailed in Chapter 3). The simulator used in Chapter 3 covers developments in the level of agents' control space that involve the kinematic and dynamic models, formation control, and collision avoidance of a sample team of ground vehicles. The second simulator introduced in this chapter regroups all of the developments introduced in Chapter 4, namely detecting target objects, encoding individual agents' specializations (i.e. binary and modulated encoding of agents' specialized competencies), probabilistic matching, and responders' coordination in consideration of agents' availability and attendance. In addition, the human operator is given a role in these simulations to intervene on the operational loop but without necessary frequent intervention. In addition, the same simulator is used to test the developments that are introduced in section 4.4.5 but for extended operational conditions.

5.2 Experimental Setup

The proposed task allocation framework is validated in the context of a large-scale system. The scenario considered assumes a team of 20 robotic agent that are specialized to fight different sources of fire and provide different types of services such as ambulance services and protection

services. The proposed system is exemplified by $T = 14$ classes of target objects to be potentially detected in an environment. The agents' specializations are encoded based on the two definitions that are introduced in section 4.4.1. First, the agents' specializations are encoded as a vector of binary coefficients, detailed in section 4.4.1.A, and as a vector of relative coefficients, associated with each robot's mechanical construction or embedded hardware or software, which modulates the relative robustness of the agents to respond to given targets, as detailed in section 4.4.1.B. A swarm of robots $\{R_i, i = 1,2, \dots, a\}$ consists of $a = 20$ specialized individual agents, R_i, that can provide $T = 14$ different task allocation responses. The formulation of the probabilistic specialization matching between the agents' specialized competencies and the corresponding constraints of given tasks is detailed in section 4.4.2. The specialized responders' coordination based on the agents' availability and attendance is detailed in section 4.4.3, whereas the human intervention on the operational loop is introduced in section 4.4.4.

Table 5.1 provides a binary representation of competencies for each of the 20 agents considered based on the agents possessing or not the specialized competencies. Table 5.2 provides a non-binary representation of competencies for each of the 20 agents considered (for the same group of specialized robots) based on level of their respective suitability. The first group of robots consists of eight robotic agents, S_1 to S_8, that are specialized to fight fire at different types of properties or places, the latter of which are of different types of construction. The second group consists of two

Table 5.1 Formulation of $a = 20$ robotic agents with binary encoded specialization functionalities to serve on 14 tasks with different requirements illustrated in the context of a large scale robotic team

Agents' specialization binary vectors	Firefighting Tasks					Ambulance Tasks		External Threats (Defence Tasks)						
	T_{F_1}	T_{F_2}	T_{F_3}	T_{F_4}	T_{F_5}	T_{Am_1}	T_{Am_2}	T_{E_1}	T_{E_2}	T_{E_3}	T_{E_4}	T_{E_5}	T_{E_6}	T_{E_7}
$\boldsymbol{S_1}$	1	1	0	0	0	0	0	0	0	0	0	0	0	0
$\boldsymbol{S_2}$	1	1	0	0	0	0	0	0	0	0	0	0	0	0
$\boldsymbol{S_3}$	1	0	1	0	0	0	0	0	0	0	0	0	0	0
$\boldsymbol{S_4}$	1	1	1	0	0	0	0	0	0	0	0	0	0	0
$\boldsymbol{S_5}$	0	1	1	0	0	0	0	0	0	0	0	0	0	0
$\boldsymbol{S_6}$	0	0	1	1	0	0	0	0	0	0	0	0	0	0
$\boldsymbol{S_7}$	0	0	0	1	1	0	0	0	0	0	0	0	0	0
$\boldsymbol{S_8}$	0	0	0	1	1	0	0	0	0	0	0	0	0	0
$\boldsymbol{S_9}$	0	0	0	0	0	1	1	0	0	0	0	0	0	0
$\boldsymbol{S_{10}}$	0	0	0	0	0	1	1	0	0	0	0	0	0	0
$\boldsymbol{S_{11}}$	0	0	0	0	0	0	0	1	1	0	0	0	0	0
$\boldsymbol{S_{12}}$	0	0	0	0	0	0	0	1	1	0	0	0	0	0
$\boldsymbol{S_{13}}$	0	0	0	0	0	0	0	0	1	1	0	0	0	0
$\boldsymbol{S_{14}}$	0	0	0	0	0	0	0	0	0	1	1	0	0	0
$\boldsymbol{S_{15}}$	0	0	0	0	0	0	0	0	0	0	1	1	0	0
$\boldsymbol{S_{16}}$	0	0	0	0	0	0	0	0	0	0	0	1	1	0
$\boldsymbol{S_{17}}$	0	0	0	0	0	0	0	0	0	0	0	1	1	0
$\boldsymbol{S_{18}}$	0	0	0	0	0	0	0	0	0	0	0	1	1	1
$\boldsymbol{S_{19}}$	0	0	0	0	0	0	0	0	0	0	0	0	1	1
$\boldsymbol{S_{20}}$	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Table 5.2 Formulation of $a = 20$ robotic agents with modulated encoding of different levels of specialization functionalities to serve on 14 tasks with different requirements illustrated in the context of a large scale robotic team

Agents' specialization modulated vectors	Firefighting Tasks					Ambulance Tasks		External Threats (Defence Tasks)						
	T_{F_1}	T_{F_2}	T_{F_3}	T_{F_4}	T_{F_5}	T_{A_1}	T_A	T_{E_1}	T_{E_2}	T_{E_3}	T_{E_4}	T_{E_5}	T_{E_6}	T_{E_7}
$\boldsymbol{S_1}$	0.8	0.2	0	0	0	0	0	0	0	0	0	0	0	0
$\boldsymbol{S_2}$	0.3	0.7	0	0	0	0	0	0	0	0	0	0	0	0
$\boldsymbol{S_3}$	0.6	0	0.4	0	0	0	0	0	0	0	0	0	0	0
$\boldsymbol{S_4}$	0.4	0.2	0.4	0	0	0	0	0	0	0	0	0	0	0
$\boldsymbol{S_5}$	0	0.5	0.5	0	0	0	0	0	0	0	0	0	0	0
$\boldsymbol{S_6}$	0	0	0.2	0.8	0	0	0	0	0	0	0	0	0	0
$\boldsymbol{S_7}$	0	0	0	0.65	0.35	0	0	0	0	0	0	0	0	0
$\boldsymbol{S_8}$	0	0	0	0.5	0.5	0	0	0	0	0	0	0	0	0
$\boldsymbol{S_9}$	0	0	0	0	0	0.6	0.4	0	0	0	0	0	0	0
$\boldsymbol{S_{10}}$	0	0	0	0	0	0.4	0.6	0	0	0	0	0	0	0
$\boldsymbol{S_{11}}$	0	0	0	0	0	0	0	0.7	0.3	0	0	0	0	0
$\boldsymbol{S_{12}}$	0	0	0	0	0	0	0	0.38	0.62	0	0	0	0	0
$\boldsymbol{S_{13}}$	0	0	0	0	0	0	0	0	0.8	0.2	0	0	0	0
$\boldsymbol{S_{14}}$	0	0	0	0	0	0	0	0	0	0.7	0.3	0	0	0
$\boldsymbol{S_{15}}$	0	0	0	0	0	0	0	0	0	0	0.44	0.56	0	0
$\boldsymbol{S_{16}}$	0	0	0	0	0	0	0	0	0	0	0	0.36	0.64	0
$\boldsymbol{S_{17}}$	0	0	0	0	0	0	0	0	0	0	0	0.4	0.6	0
$\boldsymbol{S_{18}}$	0	0	0	0	0	0	0	0	0	0	0	0.3	0.4	0.3
$\boldsymbol{S_{19}}$	0	0	0	0	0	0	0	0	0	0	0	0	0.3	0.7
$\boldsymbol{S_{20}}$	0	0	0	0	0	0	0	0	0	0	0	0	0	1

agents, S_9 and S_{10}, that are specialized to provide different classes of ambulance services. Finally, the third group consists of ten robotic agents, S_{11} to S_{20}, that are specialized to respond to external threats. In addition, a heterogeneous sensing layer is assumed to feed the targets' detection stage of the proposed approach. A flying robot is used in these simulations as a part of the system's sensing stage.

For example, let us consider that agent R_1 is an aerial vehicle that possesses the capability to fight fire in two different types of facilities such as skyscrapers, T_{F_1}, and large residential complexes, T_{F_2}. Therefore, its specialized functionality to be allocated to these two different tasks is defined as a binary vector with 1's values, as encoded in section 4.4.1 A, which means that R_1 possesses a specialized functionality to respond to the tasks, T_{F_1} and T_{F_2}, as defined in specialty vector $\boldsymbol{S}_1$ (Table 5.1). However, consider that agent R_1 has sophisticated mechanical construction that provides it with a robust functionality to fight fire on skyscrapers, T_{F_1}, that are accessible for this flying robot. Its functionality level for this type of task can be modulated to 0.8 based on the agents' specialization modulated encoding that is introduced in section 4.4.1.B, as defined in its specialty vector S_1 in Table 5.2. Moreover, the same agent has difficulty fighting fire in large residential complexes, T_{F_2}. Its limited ability to access this target will not allow this flying robot to perform this task perfectly. As well it would be very expensive to deal with this target using this type of robot whereas another type of robot (i.e. ground vehicles) could perform it at lower cost. This fact can be encoded with a lower functionality level at 0.2 as indicated in Table 5.2. On the other hand, robot R_2 runs a powerful capability which allows it to function robustly on task T_{F_2} and with lower functionality on task T_{F_1}. This robot could be a large ground vehicle equipped with robust resources to fight fire in large residential complexes. As well, this robot is capable to some

extent of responding to fire on skyscrapers by having an end effector that can reach the skyscrapers' middle floors or slightly higher, but it cannot reach the top floors. Accordingly, its competencies are modulated with a functionality level equal to 0.3 and 0.7 on the corresponding tasks, T_{F_1} and T_{F_2}, respectively. The overall specialties of the individual members of this robotic team are represented as binary encoded or non-binary modulated specialty vectors (defined in Tables 5.1 and 5.2 respectively).

5.3 Results of Simulated Experiments

In this simulated test set, the proposed mechanism is simulated in many scenarios. In the first two scenarios, (5.3.C1 and 5.3.C2), the specialty-based task allocation approach is validated for the binary definition (section 4.4.1.A.) and modulated definition (section 4.4.1.B) of robots' specialization considering a high level and a low level confidence on the target object detection in the first case and the second case respectively. These two test cases are performed to validate the performance of the proposed specialty-based task allocation framework for the two definitions of the agents' specializations encoding and how each case assigns heterogeneous agents with different specialized functionalities to corresponding tasks that impose specific constraints. These test cases can also demonstrate the benefits of and the differences between the binary and modulated specialization definitions of the individual agents.

From the perspective of human operator supervision, discussed in section 4.4.4, these test scenarios consider a $LSFL$ for the task-agent matching process with an imposed MFT value, η. The $LSFL$ range can drive the task allocation scheme to match the available agents' minimum specialized capabilities with corresponding tasks; the latter are detected with different confidence levels on the target object detection. The main role of the human supervisor is to initialize η in

advance. The latter is set by Eq. (4.17), with $\eta = 0.3$. Then, the human supervisor does not need to change η frequently.

The third test case, (5.3.C3), tests the proposed approach in terms of the specialized agents' availability, introduced in section 4.4.3, Eq. (4.12), to demonstrate the proposed system's reliability when it replaces the assignment of the unavailable or "withdrawn" specialized agent with another available specialized agent which might be less competent. The fourth test case, (5.3.C4), demonstrates the adaptability of the proposed framework for wide-ranging workspace applications based on the considered agents' attendance level introduced in section 4.4.3, Eq. (4.13). The fifth test case, (5.3.C5), shows the proposed framework's efficiency in achieving a sequential task allocation process when multiple targets are detected simultaneously. Finally, the sixth test case, (5.3.C6), validates the extended approach of the proposed framework to tackle complex tasks introduced in section 4.4.5. This level of extended development is considered in this test case only. A sample example that provides detailed step-by-step calculations for each one of the above mentioned test scenarios is provided in Appendix A.

5.3.C1 Validating the specialty-based task allocation approach based on a high confidence level of targets' object detection

In this test scenario, simulation experiments are conducted to test the proposed specialty-based task allocation framework, introduced in section 4.4, in two test cases. The first case considers the binary specialization definition (introduced in section 4.4.1.A); and the second test case considers the agents' modulated specialization definition (introduced in section 4.4.1.B). In this scenario, a high confidence level of the target object detection is considered (Table 5.3, second column from the left) when all of the team members are available (Table 5.3, sixth column from the left) for the task assignment process. Attendance, from section 4.4.3, is deactivated and the task allocation

probability is computed by placing full weight on the agents' specialty-based qualification, $\boldsymbol{Q}$, considering $\wp = 1$ in Eq. (4.16). The proposed scheme deals with the detection of one target object at a time. Here the object detection and task allocation frameworks are both simulated.

The task allocator computes the a individual agents' probabilistic specialty fitting levels to assign the most competent agents, as defined in Eqs. (4.18) and (4.20), to respond to the detected tasks. The task allocation dynamics of Eq. (4.16) are considered based on the task-agent specialty matching, Q, as given by Eq. (4.9), along with the agents' availability status, defined in Eq. (4.12). This test case considers $LSFL$ for the task-agent matching fitting process with an imposed MFT value of $\eta = 0.3$. The dynamics of agents' attendance level, defined in Eq. (4.14), are not considered here as the attendance is deactivated in this test scenario. Figure 5.1a shows that a target object of type T_{F_1} is detected by a flying robot (shown in red). The flying robot is considered as a component of a simulated sensing layer which collects the data from the workspace and feeds the target's object detection stage. The latter provides the proposed task allocator with a confidence level on the detection of the target object.

The resulting task allocation fitting probabilities of the available team members based on the binary encoding, introduced in section 4.4.1.A, are shown in Figure 5.1b. Agents R_1, R_2, and R_3 present a specialty fitting score of 0.43 that satisfies the MFT, which is set to 0.3. In addition, agent R_4 presents a lower level of probabilistic fitting of 0.29 with respect to the detected target, which is less than the minimum level of the system's trust of the task allocation response (MFT). As a result, this agent, R_4, cannot be allocated to the detected task T_{F_1}. Therefore, in the case that the system should assign only one agent as a first specialized responder to be allocated

to the detected target, the system, Eq. (4.20), automatically assigns the agent that has a maximum probabilistic fitting as indicated in Figure 5.1a.

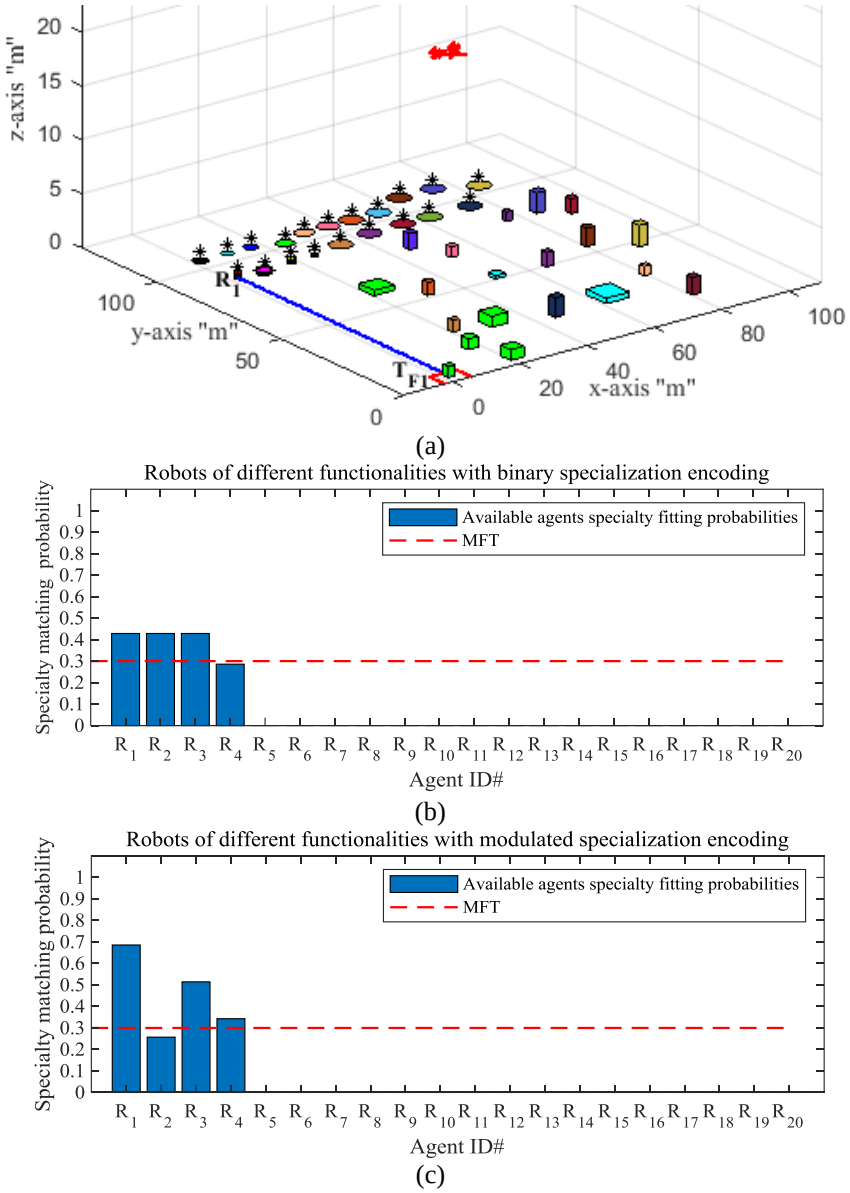


Figure 5.1 a) Detected target object T_{F1} inside a red square; **b)** Fitting probabilities of the available specialized agents with binary specialization encoding; and **c)** Fitting probabilities of the available specialized agents with modulated specialization encoding

For the purpose of clarifying the specialty-based task allocation responses of our approach, these simulation scenarios usually consider the most qualified agent to respond to the detected task. Therefore, in the case that multiple agents share the same maximum score, the max operator,

Eq. (4.20), assigns the first agent based on the order of the stored scores in the vector resulting from Eq. (4.18). The system's response to the cases when all qualified agents that achieve MFT are required to respond to the detected task are discussed below. Here, R_1 has the first maximum fitting level based on the order of the maximum scores in the vector of Eq. (4.18).

Detailed results are introduced in Table 5.3. According to Table 5.3, in the column of the binary encoding results (third column from the right), the task allocation probabilistic fitting level of the agents R_1, R_2, and R_3 with the detected target equals 0.43, indicated in bold font. This means that these three specialized agents are all qualified for the first allocation of the detected task based on the agents' specialization given in Table 5.1. On the other hand, when the agents' specializations are modulated in a non-binary format (introduced in section 4.4.1.B) based on the level of the agent's functionality robustness or the embedded equipment suitability (introduced in Table 5.2), the resulting task allocation fitting probabilities of the available team members present different levels of probabilistic fitting with respect to the detected task (shown in Figure 5.1c). Agent R_1 presents the maximum specialty fitting probability of 0.69 (or the most specialized agent), and agent R_3 presents the second-highest specialty probabilistic fitting that satisfies the MFT. In addition, agent R_4 presents a lower level of probabilistic fitting (0.34), but it satisfies the minimum MFT with respect to the detected target to represent the third-highest specialized agent. However, the agent R_2 presents a specialty probabilistic fitting which is less than the minimum level of the system's trust of the task allocation response (MFT).

Table 5.3 Specialty-based task allocation dynamics and the specialized agents capabilities associated with team members with respect to a task recognized with a high recognition confidence, (75-95%), and a deactivated attendance, $\wp = 1$, in Eq. (4.16). * indicates the score of the selected agent (first responder) that is to be allocated to the detected task. **Bold** values means that all agents are qualified for the same level of the task allocation response.

Target object recognition confidence level		*Agent ID#*	*Target-agent specialty matching probability* ($\wp \boldsymbol{Q}$)		*Agents' availability* 1:*Available* 0:*Withdrawn* ($\boldsymbol{\vartheta}_{Av}$)	*Available agents' attendance level* ($(1-\wp)\boldsymbol{\vartheta}_{Att}$)	*Available qualified agents' fitting scores* ($\boldsymbol{\Psi}_{MFT}$)		*MFT* (η)
			Binary encoding	***Modulated encoding***			***Binary Encoding***	***Modulated Encoding***	
T_{F_1}	0.86	$\boldsymbol{R_1}$	**0.43**	**0.69**	**1**	**---**	**0.43***	**0.69***	
T_{F_2}	0.00	R_2	0.43	0.26	1	---	**0.43**	---	
T_{F_3}	0.00	R_3	0.43	0.52	1	---	**0.43**	0.52	0.3
T_{F_4}	0.00	R_4	0.29	0.34	1	---	---	0.34	
T_{F_5}	0.00	R_5	0.00	0.00	1	---	---	---	
T_{A_1}	0.00	R_6	0.00	0.00	1	---	---	---	
T_{A_2}	0.00	R_7	0.00	0.00	1	---	---	---	
T_{E_1}	0.00	R_8	0.00	0.00	1	---	---	---	
T_{E_2}	0.00	R_9	0.00	0.00	1	---	---	---	
T_{E_3}	0.00	R_{10}	0.00	0.00	1	---	---	---	
T_{E_4}	0.00	R_{11}	0.00	0.00	1	---	---	---	
T_{E_5}	0.00	R_{12}	0.00	0.00	1	---	---	---	
T_{E_6}	0.00	R_{13}	0.00	0.00	1	---	---	---	
T_{E_7}	0.00	R_{14}	0.00	0.00	1	---	---	---	
		R_{15}	0.00	0.00	1	---	---	---	
		R_{16}	0.00	0.00	1	---	---	---	
		R_{17}	0.00	0.00	1	---	---	---	
		R_{18}	0.00	0.00	1	---	---	---	
		R_{19}	0.00	0.00	1	---	---	---	
		R_{20}	0.00	0.00	1	---	---	---	

As a result, this agent, R_2, cannot be allocated to the detected task (T_{F_1}). Therefore, in the case that the system should assign only one agent as a first specialized responder to deal with the detected target, then the system's Eq. (4.20) automatically assigns the agent that has a maximum probabilistic fitting (as indicated in Figure 5.1a). Here, R_1 has the only maximum fitting probability (most specialized agent) out of the team's probabilistic scores resulting from the vector of Eq. (4.18), as indicated in Figure 5.1c and assigned by Eq. (4.20). Detailed results are introduced in Table 5.3.

In contrast to the binary case in which three agents were equally qualified to the first allocation of the detected task, in this test case the modulated encoding results in different levels of the agents' probabilistic fitting. Only one agent is qualified to respond to the first allocation of the detected task. A detailed discussion is presented at the end of the third test scenario (5.3.C3) to show the differences between the binary and the modulated specialization encodings.

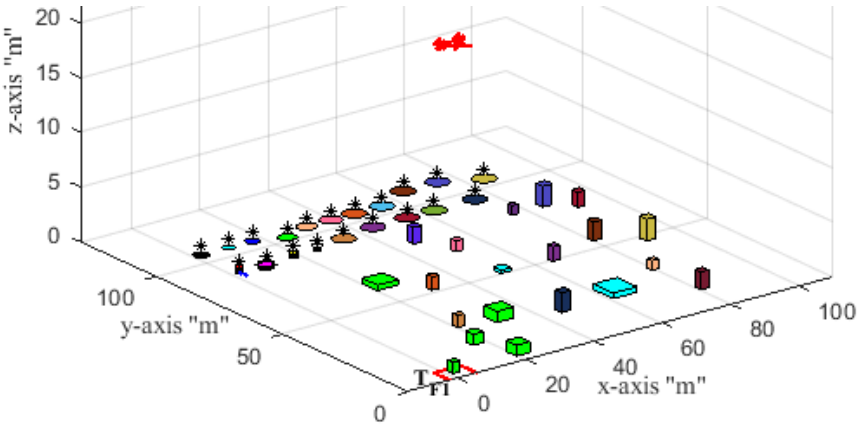
Detailed computations of the results presented in Table 5.3 are provided in Appendix A. Numerical example 5.3.C1.1 is introduced for the binary specialization encoding case in Figure 5.1b, and numerical example 5.3.C1.2 is introduced for the modulated specialization encoding case in Figure 5.1c.

5.3.C2 Validating the specialty-based task allocation approach based on a low confidence level of target object detection

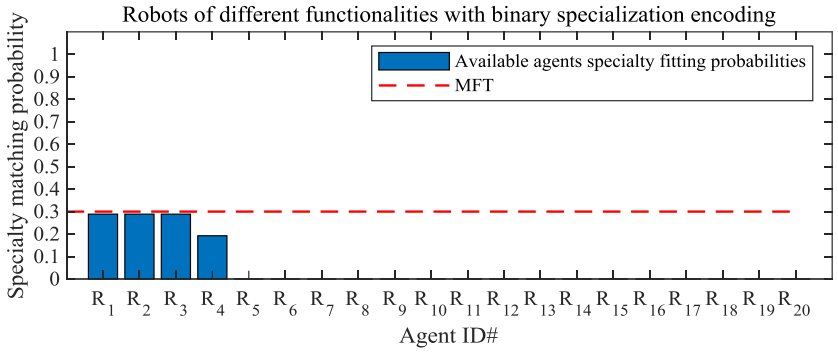
This scenario is similar to the previous one; however, a low confidence level of the target object detection is considered (Table 5.4, second column from the left) when all of the team members are available (Table 5.4, sixth column from the left) for the task assignment process. Attendance, from section 4.4.3, is deactivated with $\mathfrak{p} = 1$ in Eq. (4.16).

Figure 5.2a shows that the target, $T_{F_{1,}}$ is detected (inside red square) and the resulting task allocation fitting probabilities of the available team members, based on the binary encoding, are shown in Figure 5.2b. The agents R_1, R_2 , and R_3 present low levels of specialty probabilistic fitting ($\boldsymbol{Q}$, fourth column from the left, Table 5.4 and Figure 5.2b) which equal 0.29, and the agent R_4 presents a lower level of specialty probabilistic fitting which equals 0.19 with respect to the detected target. All of these probabilities are less than the minimum level of the system's trust required (MFT) to respond to the detected target. As a result, these four agents are not qualified. Therefore, these agents cannot respond to the detected target T_{F_1} as indicated in Figure 5.2a. Detailed results are presented in Table 5.4. According to Table 5.4, in the column of the binary encoding results ($\boldsymbol{\Psi}_{MFT}$, third column from the right), no agent is qualified. This means that when the target object is detected with a low level of confidence, the possibilities of these four specialized agents to deal with the detected target are dropped based on the binary specialization encoding given in Table 5.1. However, in the case of the modulated specialization encoding, the team members are still presenting higher levels of probabilistic fitting. The latter are higher than the minimum level of the system's trust (MFT) that is required to respond to the detected target (shown in Figure 5.3b). The agent R_1 presents the maximum specialty fitting probability of 0.46 (as the most specialized agent) and is selected to perform the detected task, as indicated in Figure 5.3a. As well, the agent R_3 presents the second-highest specialty probabilistic fitting of 0.35 that satisfies the MFT. Therefore, R_3 is qualified to respond to the detected task in the eventuality that the most specialized agent, R_1, would not be available. (The agents' availability is discussed in the third test scenario 5.3.C3 below.). However, agents R_2 and R_4 present a specialty probabilistic fitting, $\boldsymbol{Q}$, as indicated in Table 5.4 (fifth column from

the left) that is less than the minimum level of the system's trust of the task allocation response (MFT).



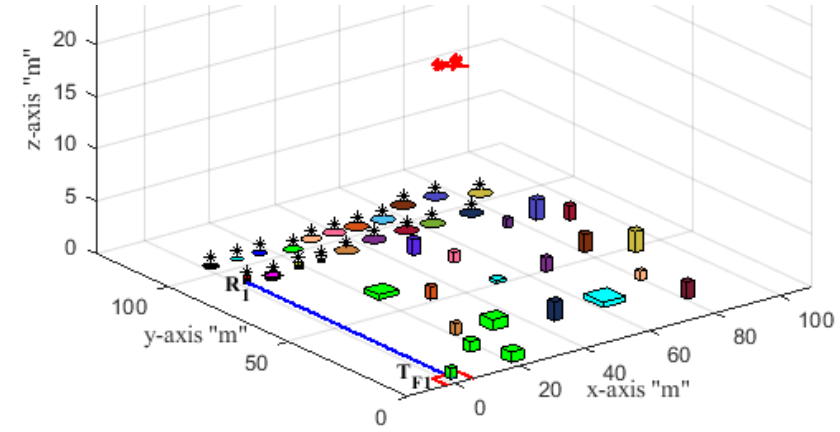
(a)



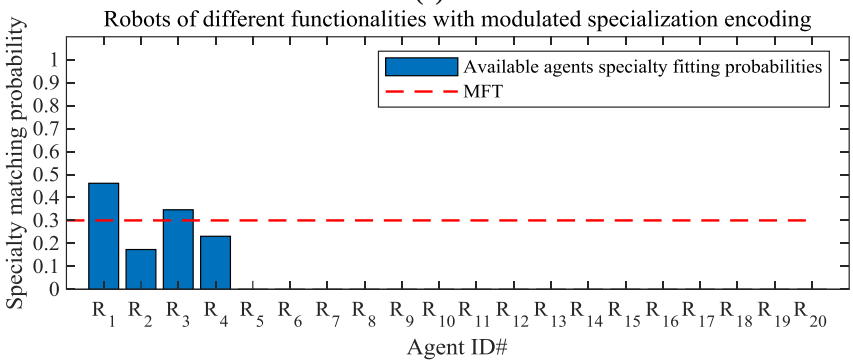
(b)

Figure 5.2 a) Detected target object T_{F1} inside a red square; **b)** Fitting probabilities of the available specialized agents with binary specialization encoding

As a result, agents R_2 and R_4 are not qualified to respond to the detected task, T_{F_1}, as indicated in Table 5.4 ($\boldsymbol{\Psi}_{MFT}$, second column from the right). This means that when the target object is detected with a low level of confidence, the chances of the same specialized agents to be allocated to the same detected target are still possible based on the modulated specialization encoding given in Table 5.2. A detailed discussion is introduced at the end of the third test scenario (5.3.C3.) to emphasize the differences between the binary and modulated specialization encodings.



(a)



(b)

Figure 5.3 a) Detected target object T_{F1} inside a red square; **b)** Fitting probabilities of the available specialized agents with modulated specialization encoding

Table 5.4 Specialty-based task allocation dynamics and the specialized agents' capabilities associated with team members with respect to a task recognized with **a low recognition confidence**, (50-60%), with a deactivated attendance, $\mathbb{p} = 1$, in Eq. (4.16)

* indicates the score of the selected agent (first responder) that is to be allocated to the detected task.

Target object recognition confidence level	*Agent ID#*	*Target-agent specialty matching probability* ($\mathbb{p}\boldsymbol{Q}$)		*Agents' availability 1:Available 0:Withdrawn* ($\boldsymbol{\vartheta}_{Av}$)	*Available agents' attendance level* ($(1-\mathbb{p})\boldsymbol{\vartheta}_{Att}$)	*Available qualified agents' fitting scores* ($\boldsymbol{\Psi}_{MFT}$)		*MFT* (η)
		Binary encoding	***Modulated encoding***			***Binary Encoding***	***Modulated Encoding***	
T_{F_1} 0.58	R_1	0.29	0.46	1	---	---	**0.46***	
T_{F_2} 0.00	R_2	0.29	0.17	1	---	---	---	
T_{F_3} 0.00	R_3	0.29	0.35	1	---	---	0.35	0.3
T_{F_4} 0.00	R_4	0.19	0.23	1	---	---	---	
T_{F_5} 0.00	R_5	0.00	0.00	1	---	---	---	
T_{A_1} 0.00	R_6	0.00	0.00	1	---	---	---	
T_{A_2} 0.00	R_7	0.00	0.00	1	---	---	---	
T_{E_1} 0.00	R_8	0.00	0.00	1	---	---	---	
T_{E_2} 0.00	R_9	0.00	0.00	1	---	---	---	
T_{E_3} 0.00	R_{10}	0.00	0.00	1	---	---	---	
T_{E_4} 0.00	R_{11}	0.00	0.00	1	---	---	---	
T_{E_5} 0.00	R_{12}	0.00	0.00	1	---	---	---	
T_{E_6} 0.00	R_{13}	0.00	0.00	1	---	---	---	
T_{E_7} 0.00	R_{14}	0.00	0.00	1	---	---	---	
	R_{15}	0.00	0.00	1	---	---	---	
	R_{16}	0.00	0.00	1	---	---	---	
	R_{17}	0.00	0.00	1	---	---	---	
	R_{18}	0.00	0.00	1	---	---	---	
	R_{19}	0.00	0.00	1	---	---	---	
	R_{20}	0.00	0.00	1	---	---	---	

5.3.C3 Validating the specialty-based task allocation approach in terms of agents' availability

This test scenario validates the proposed specialty-based task allocation framework, introduced in section 4.4, in terms of the specialized agents' availability, introduced in section 4.4.3, Eq. (4.12). The purpose of this simulation scenario is to test the reliability of the proposed approach. When the qualified agent or agents are "withdrawn," the system must reliably accomplish the mission goals by replacing the function of "withdrawn" individuals and functioning in a reliable manner. This test is performed for the two definitions of the agents' specialization encoding: the binary and modulated specialization encoding introduced in sections 4.4.1.A and 4.4.1.B respectively. However, the attendance from section 4.4.3 is deactivated and the task allocation probability is computed by placing full weight on the agents' specialty-based qualification, $\boldsymbol{Q}$, that considers $\mathfrak{p} = 1$ in Eq. (4.16) which deactivates the attendance dynamics in Eq. (4.16) as well. In this test scenario, the proposed scheme deals with the detection of one target object at a time. Here the object detection and task allocation framework are both simulated. Based on the agents' availability state, $\boldsymbol{\vartheta}_{Av}$, Eq. (4.12), the task allocator, defined in Eq. (4.18), computes the team members' specialty fitting probabilities of the "available" agents to respond to a target detected with a confidence level equal to 0.77. This test scenario considers $LSFL$ for the task-agent matching process with an imposed MFT value of $\eta = 0.3$.

Figure 5.4a shows another situation from the same simulation where a second target object of type T_{E_2} is detected, given that all team members are available. The system processes the agents' probabilistic fitting levels based on Eqs. (4.18) and (4.20) to respond to the detected task. In this test case, the binary encoding is considered (Table 5.1). The fitting probabilities of the available team members are shown in Figure 5.4b. This time, agents R_{11}, R_{12}, and R_{13} present equal fitting probabilities above the MFT. According to Table 5.5, the fitting probabilities of these agents with

the detected target equal 0.38. Therefore, the system automatically assigns agent R_{11} to respond to the detected task T_{E_2}. Detailed results are indicated in Table 5.5 (third column from the right).

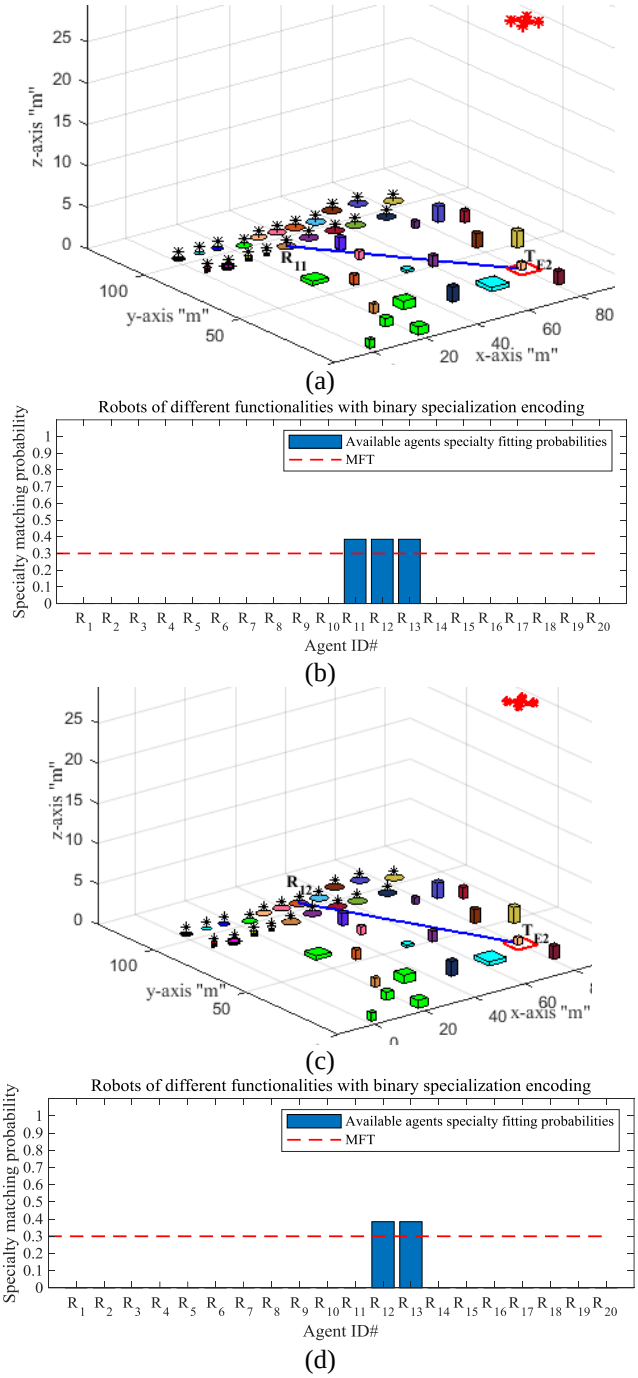
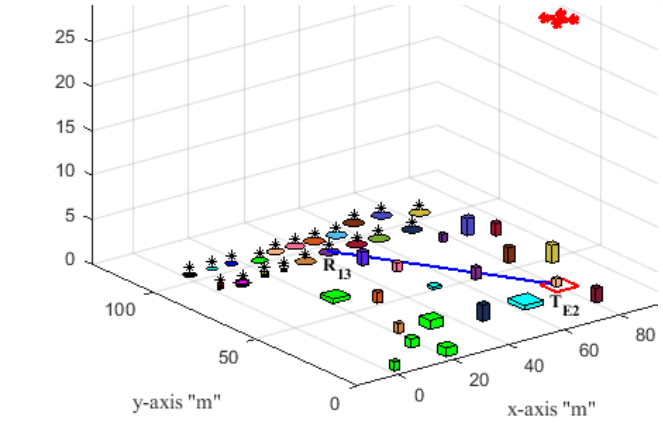


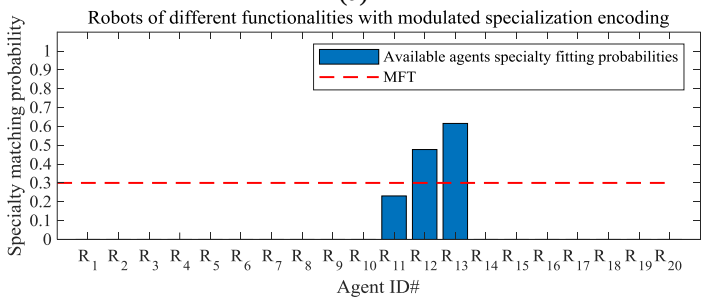
Figure 5.4 a) Detected target object T_{E_2} inside a red square and allocated to the most specialized agent R_{11}; **b)** Fitting probabilities of all specialized agents with binary encoding when the specialized agents are all available; **c)** Detected target object T_{E_2} allocated to the second available specialized agent R_{12}; and **d)** Fitting probabilities of the remaining specialized agents with binary encoding when the most specialized agent is "withdrawn"

However, in a situation where this agent R_{11} is not available or "withdrawn," as indicated in Table 5.6 (sixth column from the left), the system assigns the next specialized available agent which is R_{12} (Figure 5.4c). The fitting probabilities of the remaining available agents are shown in Figure 5.4d. Detailed results are indicated in Table 5.6 (third column from the right). Considering the proposed agents' availability in the case of the binary encoding of the agents' specialized functionalities, the system responds in a reliable manner and successfully assigns a replacement qualified agent in terms of the existing required specialized functionality.

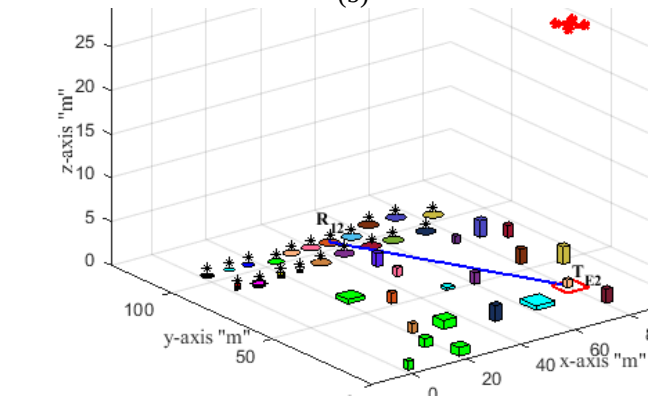
On the other hand, Figure 5.5a shows that when modulated encoding (Table 5.2) is considered, the system presents another response when the same target, T_{E_2}, is detected, given that all of the team members are available. The system processes the agents' probabilistic fitting levels based on Eqs. (4.18) and (4.20) to respond to the detected task. The fitting probabilities of the available team members are shown in Figure 5.5b. This time, the available agents present different levels of task allocation suitability with respect to the detected task. Agent R_{13} presents the highest fitting probability (the most specialized) above the MFT, which equals 0.62, as indicated in Table 5.5 (fifth column from the left). Then, agent R_{12} presents the second-highest probability (0.48) that achieves the system's minimum level of trust (MFT). However, agent R_{11} shows a lower level of the specialty-based fitting probability equal to 0.23, which does not achieve the MFT. As a result, the system automatically assigns agent R_{13} to respond to the detected task, T_{E_2}. Agent R_{12} may be allocated to the task if needed, whereas the task allocation possibility of agent R_{11} is dropped. Detailed results are presented in Table 5.5 (second column from the right).



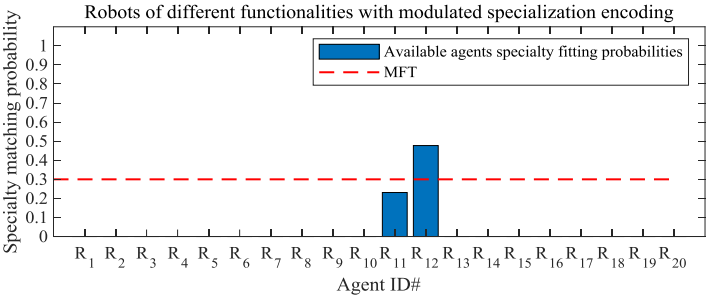
(a)



(b)



(c)



(d)

Figure 5.5 a) Detected target object T_{E_2} inside a red square and allocated to the most specialized agent R_{13}; **b)** Fitting probabilities of all specialized agents with modulated encoding when the specialized agents are all available; **c)** Detected target object T_{E_2} allocated to the second available specialized agent R_{12}; and **d)** Fitting probabilities of the remaining specialized agents with modulated encoding when the most specialized agent is "withdrawn"

However, when this agent R_{13} is not available ("withdrawn") as indicated in Table 5.6 (seventh column from the left), the system assigns the next specialized available agent which is R_{12} (Figure 5.5c). The fitting probabilities of the remaining available agents are shown in Figure 5.5d. Detailed results are indicated in Table 5.6 (second column from the right). Considering the proposed agents' availability, in the case of the modulated encoding of the agents' specialized functionalities (Table 5.2) when the most specialized agent is not available ("withdrawn"), this test case illustrates that the system responds with a great extent of reliability and successfully assigns the second-highest qualified agent in terms of the suitability level or robustness of the required specialized functionality to perform the detected task.

In general, these test scenarios show that when the proposed scheme considers the agents' availability along with their specialized capabilities, the system responds with robustness and reliability to replace the role of the "withdrawn" agent with another sufficiently qualified specialized agent to achieve the mission goals.

Table 5.5 Specialty-based task allocation dynamics when the most specialized agents are "**available**" with a deactivated attendance, $\wp = 1$, in Eq. (4.16).

* indicates the score of the selected agent (first responder) that is to be allocated to the detected task. **Bold** values in the same column mean that all agents are qualified for the same level of the task allocation response. (This table details the results that are shown in Figures 5.4b and 5.5b).

Target object recognition confidence level	*Agent ID#*	*Target-agent specialty matching probability* $(\wp \boldsymbol{Q})$		*Agents' availability* ***1:Available*** *0:Withdrawn* $(\boldsymbol{\vartheta}_{Av})$		*Available agents' attendance level* $((1-\wp)\boldsymbol{\vartheta}_{Att})$	*Available qualified agents' fitting scores* $(\boldsymbol{\Psi}_{MFT})$		*MFT* (η)
		Binary encoding	***Modulated encoding***	***Binary encoding***	***Modulated encoding***		***Binary Encoding***	***Modulated Encoding***	
T_{F_1} 0.00	R_1	0.00	0.00	1	1	---	---	---	
T_{F_2} 0.00	R_2	0.00	0.00	1	1	---	---	---	
T_{F_3} 0.00	R_3	0.00	0.00	1	1	---	---	---	0.3
T_{F_4} 0.00	R_4	0.00	0.00	1	1	---	---	---	
T_{F_5} 0.00	R_5	0.00	0.00	1	1	---	---	---	
T_{A_1} 0.00	R_6	0.00	0.00	1	1	---	---	---	
T_{A_2} 0.00	R_7	0.00	0.00	1	1	---	---	---	
T_{E_1} 0.00	R_8	0.00	0.00	1	1	---	---	---	
T_{E_2} **0.77**	R_9	0.00	0.00	1	1	---	---	---	
T_{E_3} 0.00	R_{10}	0.00	0.00	1	1	---	---	---	
T_{E_4} 0.00	R_{11}	0.38	0.23	**1**	**1**	---	**0.38***	---	
T_{E_5} 0.00	R_{12}	0.38	0.48	**1**	**1**	---	**0.38**	0.48	
T_{E_6} 0.00	R_{13}	0.38	0.62	**1**	**1**	---	**0.38**	**0.62***	
T_{E_7} 0.00	R_{14}	0.00	0.00	1	1	---	---	---	
	R_{15}	0.00	0.00	1	1	---	---	---	
	R_{16}	0.00	0.00	1	1	---	---	---	
	R_{17}	0.00	0.00	1	1	---	---	---	
	R_{18}	0.00	0.00	1	1	---	---	---	
	R_{19}	0.00	0.00	1	1	---	---	---	
	R_{20}	0.00	0.00	1	1	---	---	---	

Table 5.6 Specialty-based task allocation dynamics when the most specialized agents are "**withdrawn"** with a deactivated attendance, $\mathfrak{p} = 1$, in Eq. 4.16.

* indicates the score of the selected agent (first responder) that is to be allocated to the detected task. **Bold** values in the same column mean that all agents are qualified for the same level of the task allocation response. (This table details the results that are shown in Figures 5.4d and 5.5d).

Target object recognition confidence level		*Agent ID#*	*Target-agent specialty matching probability* ($\mathfrak{p}\boldsymbol{Q}$)		*Agents' availability 1:Available **0:Withdrawn*** ($\boldsymbol{\vartheta}_{Av}$)		*Available agents' attendance level* $((1-\mathfrak{p})\boldsymbol{\vartheta}_{Att})$	*Available qualified agents' fitting scores* ($\boldsymbol{\Psi}_{MFT}$)		*MFT* (η)
			Binary encoding	***Modulated encoding***	***Binary encoding***	***Modulated encoding***		***Binary Encoding***	***Modulated Encoding***	
T_{F_1}	0.00	R_1	0.00	0.00	1	1	---	---	---	
T_{F_2}	0.00	R_2	0.00	0.00	1	1	---	---	---	
T_{F_3}	0.00	R_3	0.00	0.00	1	1	---	---	---	0.3
T_{F_4}	0.00	R_4	0.00	0.00	1	1	---	---	---	
T_{F_5}	0.00	R_5	0.00	0.00	1	1	---	---	---	
T_{A_1}	0.00	R_6	0.00	0.00	1	1	---	---	---	
T_{A_2}	0.00	R_7	0.00	0.00	1	1	---	---	---	
T_{E_1}	0.00	R_8	0.00	0.00	1	1	---	---	---	
T_{E_2}	**0.77**	R_9	0.00	0.00	1	1	---	---	---	
T_{E_3}	0.00	R_{10}	0.00	0.00	1	1	---	---	---	
T_{E_4}	0.00	R_{11}	0.38	0.23	**0**	1	---	---	---	
T_{E_5}	0.00	R_{12}	0.38	0.48	1	**1**	---	**0.38***	**0.48***	
T_{E_6}	0.00	R_{13}	0.38	0.62	1	**0**	---	**0.38**	---	
T_{E_7}	0.00	R_{14}	0.00	0.00	1	1	---	---	---	
		R_{15}	0.00	0.00	1	1	---	---	---	
		R_{16}	0.00	0.00	1	1	---	---	---	
		R_{17}	0.00	0.00	1	1	---	---	---	
		R_{18}	0.00	0.00	1	1	---	---	---	
		R_{19}	0.00	0.00	1	1	---	---	---	
		R_{20}	0.00	0.00	1	1	---	---	---	

5.3.D1 Discussion: Performance Analysis of the Original Framework

Table 5.7 introduces comprehensive results for all of the 14 given targets when each target is detected with a high recognition confidence level in the range of 76-95%. The results shown in Table 5.7 are presented for the binary encoding task allocator (BETA), whereas the first sample case of these results is tested in the scenario 5.3.C1 (Figure 5.1b). The task allocation results of each test case show the specialty-based task allocation possibilities to each detected task. Agent/task relative robustness is used in this analysis, which is introduced in Table 5.7 (forth column from the left) based on the modulated specialization levels of the agent's robustness suitability with respect to each task, introduced in Table 5.2 based on the definition of META in section 4.4.1 B. The latter is assumed to be the real robustness suitability of the individual agent's functionality (or capability). In scenario 5.3.C1, introduced above, we demonstrated that the binary definition of the agents' specialization is appropriate to allocate a specialized robot to a detected task based on robot's corresponding specialty. However, the modulated definition of the agents' specialization is even more suitable to assign the most specialized agent based on the level of the agent's suitability to the corresponding task. In this discussion, the agent's suitability level is considered to analyze the proposed framework's performance and the differences between the two specialization encoding definitions. Based on the agent's relative robustness, the measurements of individual agents' specialization suitability with respect to each task are quantified into most specialized (green color), second specialized (blue color), third specialized (grey color), and least specialized (orange color). In addition, when the task allocation is dropped, this case is depicted by the red color (Table 5.7). Table 5.7 covers the case of binary specialty encoding (BETA), while Table 5.8 shows the results with the modulated encoding task allocator (META). Both tables are considering the same confidence level on target detection in order to distinguish from the influence

of that parameter. The first sample case of these results is tested in scenario 5.3.C1 (Figure 5.1c), that is for a high confidence level on target detection. Table 5.9 and Table 5.10 present the same results for the two cases when a low level of target recognition confidence is considered in the range of 50-60%.

Table 5.7 shows four task allocation attempts to respond to each target, each of which can be performed by a different agent. For example, in the case when T_{F_1} is detected, agent R_1 presents the potential to be the first responder and the agents R_2 and R_3 show different potentials to be the second and third responders respectively. However, the task allocation potential of agent R_4 is dropped, and its specialty-based fitting score is therefore less than the MFT (η=0.3). Considering the above-motioned measurements, we can now compare the system efficiency in the two cases of binary and modulated specialization encoding. Based on these results, the performance of the proposed approach is analyzed below:

Table 5.7 Specialty-based task allocation statistics based on **binary encoded specializations** with respect to all of the given tasks that are recognized with a **high recognition confidence level, (76-95%),** with a deactivated attendance, MFT (η=0.3)

Task		First Assignment Possibility			Second Assignment Possibility			Third Assignment Possibility			Fourth Assignment Possibility		
Type	*Dectection Score*	*Agent ID#*	*Agent/Task Relative Robustness*	*Qualified agents' fitting scores* ($\boldsymbol{\Psi}_{MFT}$)	*Agent ID#*	*Agent/Task Relative Robustness*	*qualified agents' fitting scores* ($\boldsymbol{\Psi}_{MFT}$)	*Agent ID#*	*Agent/Task Relative Robustness*	*Qualified agents' fitting scores* ($\boldsymbol{\Psi}_{MFT}$)	*Agent ID#*	*Agent/Task Relative Robustness*	*Qualified agents' fitting scores* ($\boldsymbol{\Psi}_{MFT}$)
T_{F_1}	0.85	R_1	80%	0.43	R_2	30%	0.43	R_3	60%	0.43	R_4	40%	0.29
T_{F_2}	0.76	R_1	20%	0.38	R_2	70%	0.38	R_5	50%	0.38	R_4	20%	0.29
T_{F_3}	0.87	R_3	40%	0.43	R_5	50%	0.43	R_6	20%	0.43	R_4	40%	0.29
T_{F_4}	0.77	R_6	80%	0.38	R_7	65%	0.38	R_8	50%	0.38			
T_{F_5}	0.95	R_7	35%	0.48	R_8	50%	0.48						
T_{W_1}	0.88	R_9	60%	0.44	R_{10}	40%	0.44						
T_{W_2}	0.90	R_9	40%	0.45	R_{10}	60%	0.45						
T_{E_1}	0.95	R_{11}	70%	0.48	R_{12}	38%	0.48						
T_{E_2}	0.76	R_{11}	30%	0.38	R_{12}	62%	0.38	R_{13}	80%	0.38			
T_{E_3}	0.89	R_{13}	20%	0.45	R_{14}	70%	0.45						
T_{E_4}	0.91	R_{14}	30%	0.46	R_{15}	44%	0.46						
T_{E_5}	0.92	R_{15}	56%	046	R_{16}	36%	0.46	R_{17}	40%	0.46	R_{18}	30%	0.31
T_{E_6}	0.88	R_{16}	64%	0.44	R_{17}	60%	0.44	R_{19}	30%	0.44	R_{18}	30%	0.29
T_{E_7}	0.93	R_{20}	100%	0.93	R_{19}	70%	0.47	R_{18}	30%	0.31			

Most specialized | 2nd specialized | 3rd specialized | Least specialized | Dropped task allocation possibility

Table 5.8 Specialty-based task allocation statistics based on **modulated encoded specializations** with respect to all of the given tasks that are recognized with a **high recognition confidence level, (76-95%),** with a deactivated attendance, MFT (η=0.3)

Task		First Assignment Possibility			Second Assignment Possibility			Third Assignment Possibility			Fourth Assignment Possibility		
Type	*Dectection Score*	*Agent ID#*	*Agent/Task Relative Robustness*	*Qualified agents' fitting scores* ($\boldsymbol{\Psi}_{MFT}$)	*Agent ID#*	*Agent/Task Relative Robustness*	*Qualified agents' fitting scores* ($\boldsymbol{\Psi}_{MFT}$)	*Agent ID#*	*Agent/Task Relative Robustness*	*Qualified agents' fitting scores* ($\boldsymbol{\Psi}_{MFT}$)	*Agent ID#*	*Agent/Task Relative Robustness*	*Qualified agents' fitting scores* ($\boldsymbol{\Psi}_{MFT}$)
T_{F_1}	0.85	R_1	80%	0.69	R_3	60%	0.52	R_4	40%	0.34	R_2	30%	0.26
T_{F_2}	0.76	R_2	70%	0.54	R_5	50%	0.38	R_1	20%	0.15	R_4	20%	0.15
T_{F_3}	0.87	R_5	50%	0.44	R_3	40%	0.35	R_4	40%	0.35	R_6	20%	0.18
T_{F_4}	0.77	R_6	80%	0.62	R_7	65%	0.50	R_8	50%	0.38			
T_{F_5}	0.95	R_8	50%	0.48	R_7	35%	0.33						
T_{W_1}	0.88	R_9	60%	0.53	R_{10}	40%	0.35						
T_{W_2}	0.90	R_{10}	60%	0.54	R_9	40%	0.36						
T_{E_1}	0.95	R_{11}	70%	0.67	R_{12}	38%	0.36						
T_{E_2}	0.77	R_{13}	80%	0.62	R_{12}	62%	0.48	R_{11}	30%	0.23			
T_{E_3}	0.89	R_{14}	70%	0.63	R_{13}	20%	0.18						
T_{E_4}	0.91	R_{15}	44%	0.40	R_{14}	30%	0.27						
T_{E_5}	0.92	R_{15}	65%	0.51	R_{17}	40%	0.37	R_{16}	36%	0.33	R_{18}	30%	0.27
T_{E_6}	0.88	R_{16}	64%	0.56	R_{17}	60%	0.52	R_{18}	40%	0.35	R_{19}	30%	0.26
T_{E_7}	0.93	R_{20}	100%	0.31	R_{19}	70%	0.65	R_{18}	30%	0.28			

Most specialized | 2nd specialized | 3rd specialized | Least specialized | Dropped task allocation possibility

Table 5.9 Specialty-based task allocation statistics based on **binary encoded specializations** with respect to all of the given tasks that are recognized with **a low recognition confidence level, (50-60%),** with a deactivated attendance, MFT (η=0.3).

Task		First Assignment Possibility			Second Assignment Possibility			Third Assignment Possibility			Fourth Assignment Possibility		
Type	*Dectection Score*	*Agent ID#*	*Agent/Task Relative Robustness*	*Qualified agents' fitting scores* ($\boldsymbol{\Psi}_{MFT}$)	*Agent ID#*	*Agent/Task Relative Robustness*	*Qualified agents' fitting scores* ($\boldsymbol{\Psi}_{MFT}$)	*Agent ID#*	*Agent/Task Relative Robustness*	*Qualified agents' fitting scores* ($\boldsymbol{\Psi}_{MFT}$)	*Agent ID#*	*Agent/Task Relative Robustness*	*Qualified agents' fitting scores* ($\boldsymbol{\Psi}_{MFT}$)
T_{F_1}	0.58	R_1	80%	0.29	R_2	30%	0.29	R_3	60%	0.29	R_4	40%	0.19
T_{F_2}	0.56	R_1	20%	0.28	R_2	70%	0.28	R_5	50%	0.28	R_4	20%	0.19
T_{F_3}	0.56	R_3	40%	0.28	R_5	50%	0.28	R_6	20%	0.28	R_4	40%	0.19
T_{F_4}	0.56	R_6	80%	0.28	R_7	65%	0.28	R_8	50%	0.28			
T_{F_5}	0.56	R_7	35%	0.28	R_8	50%	0.28						
T_{W_1}	0.56	R_9	60%	0.28	R_{10}	40%	0.28						
T_{W_2}	0.56	R_9	40%	0.28	R_{10}	60%	0.28						
T_{E_1}	0.56	R_{11}	70%	0.28	R_{12}	38%	0.28						
T_{E_2}	0.56	R_{11}	30%	0.28	R_{12}	62%	0.28	R_{13}	80%	0.28			
T_{E_3}	0.56	R_{13}	20%	0.28	R_{14}	70%	0.28						
T_{E_4}	0.56	R_{14}	30%	0.28	R_{15}	44%	0.28						
T_{E_5}	0.56	R_{15}	56%	0.28	R_{16}	36%	0.28	R_{17}	40%	0.28	R_{18}	30%	0.18
T_{E_6}	0.56	R_{16}	64%	0.28	R_{17}	60%	0.28	R_{19}	30%	0.28	R_{18}	30%	0.18
T_{E_7}	0.56	R_{20}	100%	0.56	R_{19}	70%	0.28	R_{18}	30%	0.18			

Most specialized | 2nd specialized | 3rd specialized | Least specialized | Dropped task allocation possibility

Table 5.10 Specialty-based task allocation statistics based on **modulated encoded specializations** with respect to all of the given tasks that are recognized with a **low recognition confidence level, (50-60%),** with a deactivated attendance, MFT (η=0.3).

Task		First Assignment Possibility			Second Assignment Possibility			Third Assignment Possibility			Fourth Assignment Possibility		
Type	*Dectection Score*	*Agent ID#*	*Agent/Task Relative Robustness*	*Qualified agents' fitting scores* ($\boldsymbol{\Psi}_{MFT}$)	*Agent ID#*	*Agent/Task Relative Robustness*	*Qualified agents' fitting scores* ($\boldsymbol{\Psi}_{MFT}$)	*Agent ID#*	*Agent/Task Relative Robustness*	*Qualified agents' fitting scores* ($\boldsymbol{\Psi}_{MFT}$)	*Agent ID#*	*Agent/Task Relative Robustness*	*Qualified agents' fitting scores* ($\boldsymbol{\Psi}_{MFT}$)
T_{F_1}	0.58	R_1	80%	0.46	R_3	60%	0.35	R_4	40%	0.23	R_2	30%	0.17
T_{F_2}	0.56	R_2	70%	0.39	R_5	50%	0.28	R_1	20%	0.11	R_4	20%	0.11
T_{F_3}	0.56	R_5	50%	0.28	R_3	40%	0.22	R_4	40%	0.22	R_6	20%	0.11
T_{F_4}	0.56	R_6	80%	0.44	R_7	65%	0.36	R_8	50%	0.28			
T_{F_5}	0.56	R_8	50%	0.28	R_7	35%	0.19						
T_{W_1}	0.56	R_9	60%	0.34	R_{10}	40%	0.22						
T_{W_2}	0.56	R_{10}	60%	0.34	R_9	40%	0.22						
T_{E_1}	0.56	R_{11}	70%	0.39	R_{12}	38%	0.21						
T_{E_2}	0.56	R_{13}	80%	0.45	R_{12}	62%	0.34	R_{11}	30%	0.17			
T_{E_3}	0.56	R_{14}	70%	0.39	R_{13}	20%	0.11						
T_{E_4}	0.56	R_{15}	44%	0.25	R_{14}	30%	0.17						
T_{E_5}	0.56	R_{15}	65%	0.31	R_{17}	40%	0.22	R_{16}	36%	0.20	R_{18}	30%	0.17
T_{E_6}	0.56	R_{16}	64%	0.36	R_{17}	60%	0.34	R_{19}	40%	0.22	R_{18}	30%	0.17
T_{E_7}	0.56	R_{20}	100%	0.56	R_{19}	70%	0.39	R_{18}	30%	0.17			

Most specialized | 2nd specialized | 3rd specialized | Least specialized | Dropped task allocation possibility

A- First Responder Assignment

In terms of specialty-based task allocation, the first responder agent must be the most specialized available agent that can respond to the detected target. In this example, the first responders are located in the column of the first assignment in Table 5.7 and Table 5.8.

Table 5.7 shows that the first responders in the case of the binary encoding of the agents' competencies can be in different classes of specialization suitability represented by different colors (green, blue, or grey). This means that in the case of binary specialization encoding, the proposed task allocator successfully selects a specialized agent that to be allocated to the detected task; however, the level of the specialized agent's suitability cannot be estimated. The latter might result in selecting a first responder of a lower suitability level, as indicated in Table 5.7 (blue and grey colors) in the column of the first assignment. On the other hand, in the case of modulated encoding of the agents' competencies, Table 5.8 shows that the first responder is usually selected to be the most suitable specialized agent (green color). This means that in the case of modulated specialization encoding the proposed task allocator is most successful at assigning the most qualified agent to respond to a detected task.

Comparing the binary encoded task allocator (BETA) and the modulated encoded task allocator (META) leads to the following observations. Based on the binary encoding (Table 5.7, first assignment column), the first responder is selected seven times to be the most qualified agent but it fails in another seven attempts. In spite of this, the success of the first seven cases is based in the order of the agents' stored probabilistic score in the vector that resulted from Eq. (4.18). This interpretation indicates that in the case of binary encoding of the agents' specializations, the system usually succeeds at assigning a specialized agent but it cannot guarantee it to be the most suitable

agent unless its probability is stored first in the vector of Eq. (4.18) amongst the other agents. However, Table 5.8 (first assignment column) indicates that, based on modulated encoding, the system successfully selects the first responder to be the most qualified agent amongst all 14 trials. This indicates that in the case of modulated encoding of the agents' specializations, the system usually succeeds with high efficiency at assigning the most qualified agent to be the first responder to the detected task.

Table 5.9 shows the case when a low target recognition confidence level is considered in the range of 50-60%. The binary encoded task allocator fails to respond to all of the detected targets expect in one trial out of the 14 simulated trials. However, Table 5.10 shows that the modulated encoded task allocator still presents a high level of performance to allocate the most specialized agents to the detected tasks, succeeding in 11 cases out of the 14 trials. The results are summarized in Table 5.11.

Table 5.11 Successful task allocations with **the most specialized agent** out of the simulated trials

Task	High confidence level of target object detection:		Low confidence level of target object detection:	
	BETA (Table 5.7)	META (Table 5.8)	BETA (Table 5.9)	META (Table 5.10)
	Allocation possibility/Task	Allocation possibility/Task	Allocation possibility/Task	Allocation possibility/Task
T_{F_1}	√	√	Failed	√
T_{F_2}	×	√	Failed	√
T_{F_3}	×	√	Failed	Failed
T_{F_4}	√	√	Failed	√
T_{F_5}	×	√	Failed	Failed
T_{A_1}	√	√	Failed	√
T_{A_2}	×	√	Failed	√
T_{E_1}	√	√	Failed	√
T_{E_2}	×	√	Failed	√
T_{E_3}	×	√	Failed	√
T_{E_4}	×	√	Failed	Failed
T_{E_5}	√	√	Failed	√
T_{E_6}	√	√	Failed	√
T_{E_7}	√	√	√	√
$Total$	7	14	1	11
√ : With the most qualified agent; × : With another specialized agent; Failed: Task allocation possibility is dropped				

A) with high confidence level on target object detection:

The efficiency of the binary encoded task allocator (BETA) with respect to the 14 task scenarios used in this simulation is defined as follows:

$$\text{BETA Efficiency} = \frac{Number\ of\ successful\ cases\ of\ selecting\ the\ most\ qualified\ agent}{Total\ number\ of\ trials} \times 100\% \quad (5.1)$$

$$= \frac{7}{14} \times 100\% = 50\%$$

The efficiency of the modulated encoded task allocator (META) with respect to these 14 task scenarios is defined as follows:

$$\text{META Efficiency} = \frac{Number\ of\ sucessful\ cases\ of\ selecting\ the\ most\ qualified\ agent}{Total\ number\ of\ trials} \times 100\% \quad (5.2)$$

$$= \frac{14}{14} \times 100\% = 100\%$$

The relative efficiency of BETA with respect to the efficiency of META can be estimated for these 14 task scenarios as follows:

$$\text{Relative efficiency} = \frac{Number\ of\ successful\ cases\ of\ BETA}{Number\ of\ successful\ cases\ of\ META} \times 100\% \quad (5.3)$$

$$= \frac{7}{14} \times 100\% = 50\%$$

B) with low confidence level on target object detection:

Similarly, when target detection is less reliable:

$$\text{BETA Efficiency} = \frac{Number\ of\ successful\ cases\ of\ selecting\ the\ most\ qualified\ agent}{Total\ number\ of\ trials} \times 100\% \quad (5.4)$$

$$= \frac{1}{14} \times 100\ \% = 7.14\%$$

$$\text{META Efficiency} = \frac{Number\ of\ successful\ cases\ of\ selecting\ the\ most\ qualified\ agent}{Total\ number\ of\ trials} \times 100\% \quad (5.5)$$

$$= \frac{11}{14} \times 100\ \% = 78.6\%$$

$$\text{Relative efficiency} = \frac{Number\ of\ successful\ cases\ of\ BETA}{Number\ of\ successful\ cases\ of\ META} \times 100\% \quad (5.6)$$

$$= \frac{1}{11} \times 100\% = 9.1\%$$

2- All Qualified Responders' Assignments

In many applications such as firefighting, a large number of firefighting vehicles are necessary to control the situation. In the case of such applications, all of the available specialized agents within the service sector must respond to the task. The proposed specialty-based task allocator can work well in such a case by assigning all of the qualified agents that achieve the minimum MFT. Similarly, ambulance services and safety/defence resources must respond in higher capacities in severe situations when a call is received or a threat is detected. The efficiency of the proposed system in this case can be inferred from the statistics introduced in Tables 5.7, 5.8, 5.9, and 5.10 based on the quantified measurements of BETA and META successful possibilities in response to each detected task. Table 5.12 presents the successful task allocation possibilities per task inferred from the simulated trials.

Table 5.12 The successful task allocation possibilities/task inferred from the simulated trials when **all qualified agents** must be allocated to the detected task

Task	High confidence level of target object detection:		Low confidence level of target object detection:	
	BETA (Table 5.7)	META (Table 5.8)	BETA (Table 5.9)	META (Table 5.10)
	Successful responders /Task	Successful responders /Task	Successful responders /Task	Successful responders /Task
T_{F_1}	3	3	Failed	2
T_{F_2}	3	2	Failed	1
T_{F_3}	3	3	Failed	Failed
T_{F_4}	3	3	Failed	2
T_{F_5}	2	2	Failed	Failed
T_{A_1}	2	2	Failed	1
T_{A_2}	2	2	Failed	1
T_{E_1}	2	2	Failed	1
T_{E_2}	3	2	Failed	2
T_{E_3}	2	1	Failed	1
T_{E_4}	2	1	Failed	Failed
T_{E_5}	4	3	Failed	1
T_{E_6}	3	3	Failed	2
T_{E_7}	3	2	1	2
$Total$	37	31	1	16

The total efficiency of the system can be measured as follows:

$$\text{Total Efficiency } = \frac{Number\ of\ successful\ task\ allocation\ possibilities}{Total\ number\ of\ tasks} \times 100\% \qquad (5.7)$$

The relative efficiency of the performance of BETA with respect to META can be introduced for these 14 simulated trials. Given that the task allocators BETA and META are tested using the same test trials, the relative efficiency of a task allocator BETA with respect to the task allocator META can be determined as:

$$\text{Relative efficiency} = \frac{Number\ of\ successful\ cases\ of\ task\ allocator\ BETA}{Number\ of\ successful\ cases\ of\ task\ allocator\ META} \times 100\% \qquad (5.8)$$

A) with high confidence level on target object detection (all qualified agents):

$$\text{BETA Total Efficiency} = \left(\frac{37}{14}\right) \times 100\% = 260\%$$

$$\text{META Total Efficiency} = \left(\frac{31}{14}\right) \times 100\% = 220\%$$

$$\text{BETA Relative Efficiency} = \left(\frac{37}{31}\right) \times 100\% = 116.1\%$$

B) with low confidence level on target object detection (all qualified agents):

$$\text{BETA Total Efficiency} = \left(\frac{1}{14}\right) \times 100\% = 7.1\%$$

$$\text{META Total Efficiency} = \left(\frac{16}{14}\right) \times 100\% = 114.3\%$$

$$\text{BETA Relative Efficiency} = \left(\frac{1}{16}\right) \times 100\% = 6.2\%$$

These observations promote META for many applications rather than BETA, particularly the applications that require assigning the most specialized robot to perform a specific task, or the applications that consider low levels of recognition confidence on target object detection.

5.3.C4 Validating the specialty-based task allocation approach in terms of agents' attendance

The coordination of multi-robot systems is a major challenge in robotics, particularly with wide workspace applications. Particularly in emergency response, multi-robot coordination plays a prominent role to efficiently and timely drive the robots to the locations that are affected by a disaster. For example, let's consider a heterogeneous firefighting team that is distributed over a large sector and operates from different fire stations. In total, this team consists of dozens of heterogeneous fire ground vehicles equipped with different levels of firefighting equipment, aerial vehicles, and ambulance vehicles distributed over a wide area. The proposed framework is designed to be responsive to operational conditions over wide workspace applications. To achieve

this purpose, the agents' attendance, $\boldsymbol{\vartheta}_{Att}$, Eq. (4.13), is activated in Eq. (4.16) with $\mathfrak{p} < 1$ to optimize the system's time responses of the selected agents to visit the location, where a specific level of services is requested, and to perform tasks in a specialty-based manner. The agents' attendance, $\boldsymbol{\vartheta}_{Att}$, is weighted by the term $(1 - \mathfrak{p})$. At the same time, the task-agent specialty-based matching, $\boldsymbol{Q}$, must be preserved as the essential component of the task allocation process by selecting this term based on the condition that $(1 - \mathfrak{p}) < \eta$. This test scenario validates the proposed framework (introduced in section 4.4) in terms of the specialized agents' attendance (introduced in section 4.4.3), Eq. (4.13). The purpose of this simulation scenario is to test the adaptivity of the proposed approach when the most qualified agent or agents in terms of task-agent specialty-based matching, $\boldsymbol{Q}$, are located far away from the target position. While other qualified agents may be closer to the target, the latter might be less competent. In these cases, the system must adapt to control the situation until the arrival of the more adequate agents to avoid major damage in firefighting applications, to provide lifesaving services in search-and-rescue applications, or to resist to threats in safety/defence applications.

The modulated specializations' encoding task allocator (META) defined in Table 5.2 provides a high level of efficiency in comparison with the binary encoded task allocator (BETA), as demonstrated above; therefore, the remaining simulation test scenarios (5.3.C4 and 5.3.C5) are performed for META.

This test scenario considers $LSFL$ for the task-agent matching process with an imposed MFT value of $\eta = 0.3$. An assumption is made that the system is equipped with robust capabilities to provide accurate measurements of the agents' current locations, velocities, and the velocity of the target, given that pose estimation and localization considerations are beyond the scope of this

thesis. Appropriate assumptions are made to mimic these considerations in the simulated trials. First, the velocities of all agents are assumed to be 50 km/hour as an average velocity of a ground vehicle. For robots that have high velocities as a specialized attribute, such as aerial vehicles, this attribute is weighted in the corresponding robots' defined specialty encoding vector (Table 5.2). The Euclidian distance is simulated as a travelling distance that separates the robot from the target's location. In addition, a static target is considered for the robots' attendance state, Eq. (4.14), that results in $\epsilon = 0$ and $v_t < v_{max}$. The latter leads to $\overline{\mathrm{T}} = 1$.

Example C4-1:

Figure 5.6a shows the same situation of the target object of type T_{F_1} considered in the test case (5.3.C1) with modulated specializations encoding and detected with a confidence level equal to 0.86, given that all of the team members are available but distributed over a wide area. The system processes the agents' probabilistic fitting levels based on Eqs. (4.18) and (4.20) to respond to the detected task. The fitting probabilities of the available team members shown in Figure 5.6b and Table 5.13 are produced with deactivated attendance, $\boldsymbol{\vartheta}_{Att}$, with $\mathfrak{p} = 1$ for comparison purposes. Agents R_1, R_3, and R_4 present fitting probabilities above the MFT. According to Table 5.13, the distances separating the agents R_1, R_3, and R_4 from the target location are $78\ km$, $23.9\ km$, and $25.6\ km$ respectively. Although agent R_2 is located closer to the target location ($14.1\ km$ away from the target location), this agent is not qualified to respond to the current task because it does not achieve the MFT in terms of task-agent specialization matching, $\boldsymbol{Q}$, indicated in Figure 5.6b. Considering that the agent R_1 is a very robust robot that is equipped with a very expensive equipment. This type of robot should be operated from global stations to cover a large-scale area such as a province or multiple cities. The overall system has a limited number of units of this type

of robot. On the other hand, the other robots R_2, R_3, and R_4 are operated from different local stations. When the agent's attendance, $\boldsymbol{\vartheta}_{Att}$, is deactivated, then the task allocation results in Table 5.13 (fourth column from the right) show two cases, which are discussed below. These can be compared to another two cases when the agent's attendance, $\boldsymbol{\vartheta}_{Att}$, is activated (as shown in Table 5.14, fourth column from the right). These cases are:

1- First Responder Assignment: In this case, the most qualified agent to be the first responder is R_1. However, R_1 is located 78 km away from the target location and would require approximately 94 minutes to arrive and conduct the task (Table 5.13, first column from the right). This can be a weakness of the framework for wide workspace applications, given that the 94-minute delay for the responder's arrival could incur loss of lives and possessions in emergency applications. On the other hand, when the agents' attendance, $\boldsymbol{\vartheta}_{Att}$, is activated in Eq. (4.16) with $\text{p} < 1$, then the system shows a different response, as indicated in Figure 5.7a, and assigns a closer agent, R_3, as a first responder to the current task, T_{F_1} (indicated in Figure 5.7b, Table 5.14, fourth column from the right). This agent can arrive within 29 minutes as shown in Table 5.14 (first column from the right). As a result, the selected responder can arrive within 29 minutes, which is 65 minutes earlier than the case that does not consider attendance.

2- All Qualified Responders' Assignment: In the case of deactivated attendance, the qualified responders are R_1, R_3, and R_4 (Table 5.13, fourth column from the right). The closest responders, R_3, and R_4, can arrive within 29 and 31 minutes respectively, and the possibility of the agent R_2 is dropped (Table 5.13, first and second columns from the right). However, when the agents' attendance is activated, the system shows different response and higher capacity. The increase in the system's capacity means that the number of the agents that are qualified to allocate

the task is increased. Table 5.2 shows that four agents are specialized in the task, T_{F_1}, when the attendance is deactivated results in that three of them are only qualified to respond to this task, as indicated in Figure 5.6a, which represents 75% of the agents that are specialized in this task. However, when the attendance is activated, agent R_2 achieves the minimum MFT, and is added to the list of qualified agents that can respond to the current task (indicated in Figure 5.7b, which increases the system's task allocation capacity to 100% of the specialized agents (Table 5.14, fourth column from the right). This agent, R_2, can arrive within 17 minutes as shown in Table 5.14 (first column from the right). As a result, the system capacity is increased to four responders and the time of the first arrival is reduced to 17 minutes.

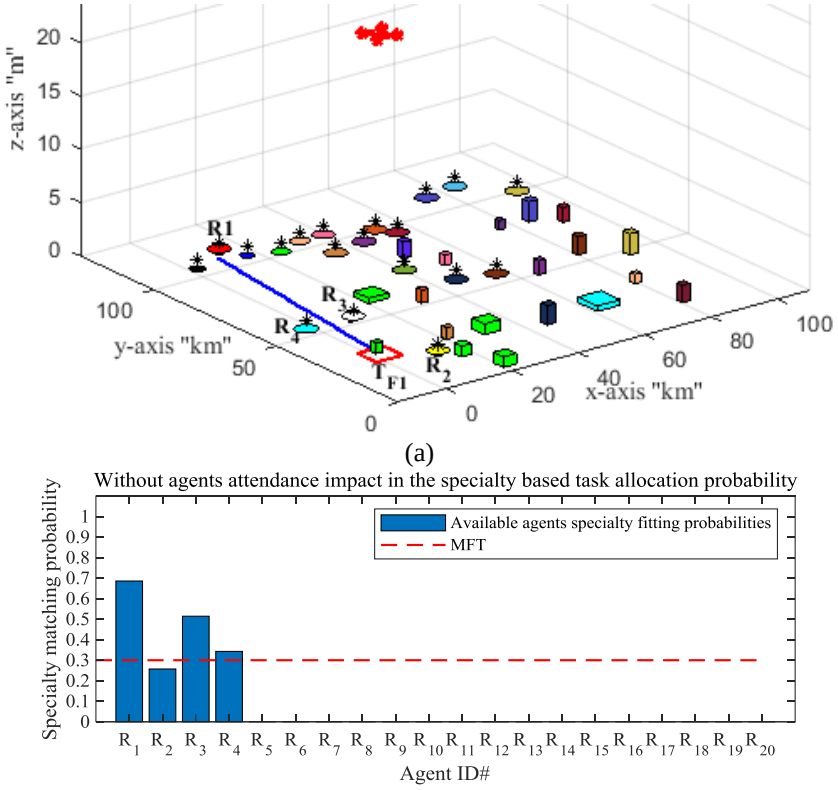
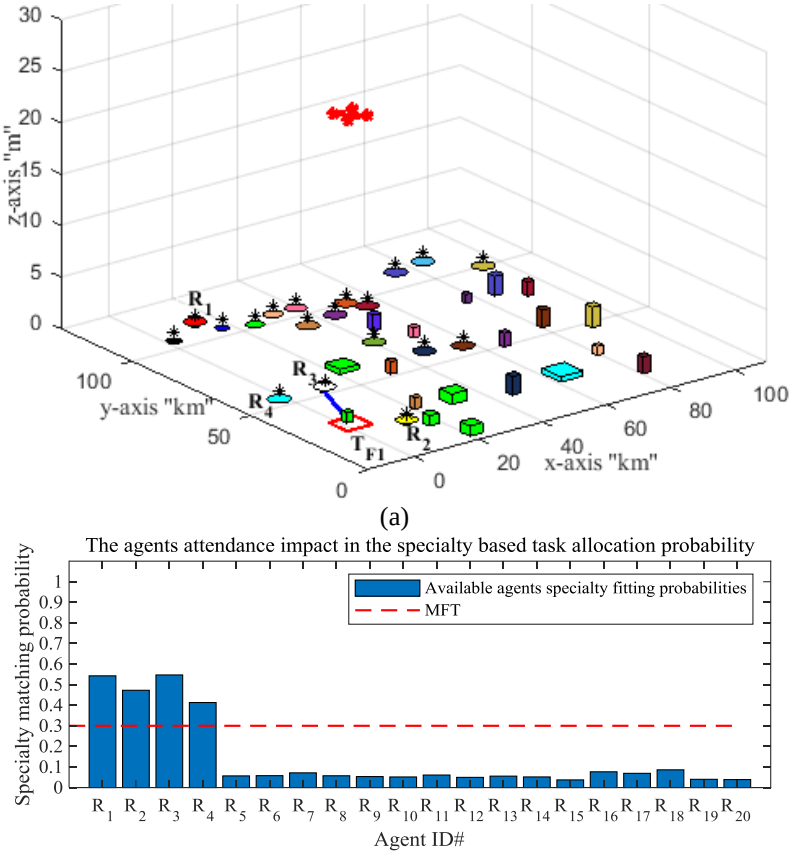


Figure 5.6 a) Detected target object T_{F1}, wide-ranging workspace inside a red square; **b)** Fitting probabilities of the available specialized agents,$\boldsymbol{Q}$ only, with modulated specialization encoding

Table 5.13 Specialty-based task allocation dynamics with a deactivate attendance, $\mathfrak{p} = 1$, in Eq. 4.16.
* indicates the score of all qualified agents that are to be allocated to the detected task. **Bold** font indicates the task allocation dynamics of the first responder. MFT (η=0.3) $v = 50\ km/hr$.

Target object recognition confidence level		*Agent*		*Agent-target specialty matching probability* $(\mathfrak{p}\boldsymbol{Q})$	*Agents' availability 1:Available 0:Withdrawn* $(\boldsymbol{\vartheta}_{Av})$	*Available agents' attendance level* $(1-\mathfrak{p})\boldsymbol{\vartheta}_{Att}$	*Available qualified agents' fitting scores* $(\boldsymbol{\Psi}_{MFT})$	*Agent-target travelling distance* $(d_i)Km$	*Qualified Responders*	
		ID#	*Agent-target relative robustness*	*Modulated encoding*					*Order*	*Time of arrival (minutes)*
T_{F_1}	0.86	$\boldsymbol{R_1}$	**80%**	**0.69**	**1**	**0**	**0.69***	**78**	**1st**	**94**
T_{F_2}	0.00	R_2	30%	0.26	1	0	---	14.1	Dropped	17
T_{F_3}	0.00	R_3	60%	0.52	1	0	0.52*	23.9	2nd	29
T_{F_4}	0.00	R_4	40%	0.34	1	0	0.34*	25.6	3rd	31
T_{F_5}	0.00	R_5	0%	0.00	1	0	---	---	---	---
T_{A_1}	0.00	R_6	0%	0.00	1	0	---	---	---	---
T_{A_2}	0.00	R_7	0%	0.00	1	0	---	---	---	---
T_{E_1}	0.00	R_8	0%	0.00	1	0	---	---	---	---
T_{E_2}	0.00	R_9	0%	0.00	1	0	---	---	---	---
T_{E_3}	0.00	R_{10}	0%	0.00	1	0	---	---	---	---
T_{E_4}	0.00	R_{11}	0%	0.00	1	0	---	---	---	---
T_{E_5}	0.00	R_{12}	0%	0.00	1	0	---	---	---	---
T_{E_6}	0.00	R_{13}	0%	0.00	1	0	---	---	---	---
T_{E_7}	0.00	R_{14}	0%	0.00	1	0	---	---	---	---
		R_{15}	0%	0.00	1	0	---	---	---	---
		R_{16}	0%	0.00	1	0	---	---	---	---
		R_{17}	0%	0.00	1	0	---	---	---	---
		R_{18}	0%	0.00	1	0	---	---	---	---
		R_{19}	0%	0.00	1	0	---	---	---	---
		R_{20}	0%	0.00	1	0	---	---	---	---



(a)

(b)

Figure 5.7 a) Task T_{F1} allocated to a closer robot R_3 in a wide-ranging workspace inside a red square; **b)** Fitting probabilities of the available specialized agents with activated attendance, $\boldsymbol{\vartheta}_{Att}$, and modulated specialization encoding.

Table 5.14 Specialty-based task allocation dynamics with an **activate attendance**, $\mathfrak{p} = 1 - (\eta - 0.1)$, in Eq. (4.16.),
* indicates the score of all qualified agents that are to be allocated to the detected task. **Bold** font indicates the task allocation dynamics of the first responder. MFT (η=0.3) $v = 50\ km/hr$.

Target object recognition confidence level	Agent		Agent-taregt specialty matching probability ($\mathfrak{p}\boldsymbol{Q}$) Modulated encoding	Agents' availability 1:Available 0:Withdrawn ($\boldsymbol{\vartheta}_{Av}$)	Available Agents' attendance level ($(1-\mathfrak{p})\boldsymbol{\vartheta}_{Att}$)	Available qualified agents' fitting probability ($\boldsymbol{\Psi}_{MFT}$)	Agent-target travelling distance $(d)Km$	Qualified Responders	
	ID#	Agent-target relative robustness						Order	Time of arrival (minutes)
T_{F_1} 0.86	R_1	80%	0.48	1	0.05	0.53*	78.0	2nd	94
T_{F_2} 0.00	R_2	30%	0.18	1	0.29	0.47*	14.1	3rd	17
T_{F_3} 0.00	$\boldsymbol{R_3}$	**60%**	**0.36**	**1**	**0.18**	**0.54***	**23.9**	**1st**	**29**
T_{F_4} 0.00	R_4	40%	0.24	1	0.17	0.41*	25.4	4th	31
T_{F_5} 0.00	R_5	0%	0.00	1	0.05	---	---	---	---
T_{A_1} 0.00	R_6	0%	0.00	1	0.05	---	---	---	---
T_{A_2} 0.00	R_7	0%	0.00	1	0.07	---	---	---	---
T_{E_1} 0.00	R_8	0%	0.00	1	0.05	---	---	---	---
T_{E_2} 0.00	R_9	0%	0.00	1	0.05	---	---	---	---
T_{E_3} 0.00	R_{10}	0%	0.00	1	0.05	---	---	---	---
T_{E_4} 0.00	R_{11}	0%	0.00	1	0.06	---	---	---	---
T_{E_5} 0.00	R_{12}	0%	0.00	1	0.04	---	---	---	---
T_{E_6} 0.00	R_{13}	0%	0.00	1	0.06	---	---	---	---
T_{E_7} 0.00	R_{14}	0%	0.00	1	0.05	---	---	---	---
	R_{15}	0%	0.00	1	0.03	---	---	---	---
	R_{16}	0%	0.00	1	0.07	---	---	---	---
	R_{17}	0%	0.00	1	0.07	---	---	---	---
	R_{18}	0%	0.00	1	0.08	---	---	---	---
	R_{19}	0%	0.00	1	0.04	---	---	---	---
	R_{20}	0%	0.00	1	0.03	---	---	---	---

Detailed computation of the results presented in Table 5.14 are provided in Appendix A. Numerical example 5.3.C4 is introduced for the modulated specialization encoding case, Figure 5.7b.

5.3.D2 Performance Analysis of Refined Framework

As illustrated above, the proposed specialty-based task allocation framework can be refined to tackle wide workspace applications by weighting the overall task allocation probability with the agents' attendance. The previous example, C4-1, is introduced to evaluate the overall performance of the system given that the most robust agent is located too far from the target's locations. This example tests the system's ability to allocate a qualified agent to the task within a reasonable time, or to offer an opportunity for more agents to join the task allocation operation and increase the total system capacity. In this simulated trial the system shows good performance, as summarized in Tables 5.15.

Table 5.15 Comparison of the system's performance in presence of agents' attendance (Example C4-1)

Attendance	Most specialized agent		All qualified agents	
	Time of arrival of the selected agent (minutes)	Agent-target relative robustness	Time of arrival of the closest agent (minutes)	System's capacity
Deactivated	94	80%	29	75%
Activated	29	60%	17	100%
Observations	65 minutes earlier	20% lower	13 minutes earlier	Increased with 25%

Table 5.15 shows that for the application that desires to allocate the most qualified agent to the detected task, the agent's attendance improves the system's response in selecting another specialized agent. Even though the latter is 20% less competent than the most available qualified agent, its selection cuts the time of arrival by 65 minutes. For example, consider that an emergency call is received which requires an ambulance vehicle with advanced medical equipment. However, this vehicle is too far from the target position, whereas another vehicle with basic medical equipment is located very close to the target position. In this case, the situation calls for a fast response rather than waiting for the most qualified vehicle to arrive from a long distance, as the delay might result in loss of life. On the other hand, for the application that desires the intervention of all of the qualified agents such as firefighting or safety/defence operations to control the

situation, Table 5.15 also shows that the agents' attendance increases the capacity of the system from three responders to four responders. In addition, the time of arrival of the first responder is reduced from 29 minutes to 17 minutes. As a result, the proposed agents' attendance extends the application of the specialty-based framework to lead teams of specialized individuals distributed over wider workspaces.

5.3.C5. Simultaneous Task Detection and Sequential Allocation

This test scenario validates the reliability of the proposed specialty-based task allocation framework when more than one target is detected simultaneously and qualified agents are withdrawn or involved in parallel task allocations. When the latter is occurring, the availability state of the agent becomes dynamic and changes to "busy" based on the logic given by Eq. (4.12). In spite of that, the system must be sufficiently reliable to proceed and select another qualified agent to accomplish the mission goals without waiting for the "busy" agents to return to the service and without failing to respond to the simultaneously detected tasks. The purpose of this test scenario is to validate the reliability of the proposed framework in response to simultaneous detection of multiple tasks when the qualified agent or agents are "withdrawn" or involved in the task allocation process. This test is performed for modulated specialization encoding (introduced in 4.4.1.B). Four examples are introduced in this test scenario. However, the attendance from section 4.4.3 is deactivated in the first three examples and activated in the fourth one. The proposed scheme deals with two targets detected at one time. Here the object detection and task allocation frameworks are both simulated. This test scenario considers $LSFL$ for the task-agent matching process with an imposed MFT value of $\eta = 0.3$.

Example C5-1: Basic Operation

In this example, the system must respond to multiple target objects detected simultaneously and sequentially allocate different robots to these targets. Figure 5.8 shows that two target objects of types T_{E_5} and T_{E_6} are detected simultaneously. Then, the task allocator processes the confidence levels on the detected objects and computes each individual agent's specialty probabilistic fitting level to sequentially allocate these tasks to the most competent agents using Eqs. (4.18) and (4.20). The team members' availability upon the detection of the first target is shown in Figure 5.9a. The specialty fitting probabilities of the available agents with respect to the first task, T_{E_5}, with a deactivated attendance, in Eq. (4.16), are shown Figure 5.9b, where it is evident that agents R_{15}, R_{16}, and R_{17} are all qualified with specialty fitting probabilities that achieve the MFT which is still set to 0.3. However, R_{15} is the most qualified specialized agent. According to Table 5.16, the specialty fitting probability of R_{15} equals 0.48 with respect to a target of type T_{E_5} that is detected with a recognition confidence level equal to 0.86. As a result, the system, using Eq. (4.20), selects this agent as the most qualified specialized agent amongst the three qualified agents to allocate to the task T_{E_5}. Due to this allocation, the availability state of R_{15} is changed to "busy" (Figure 5.9c).

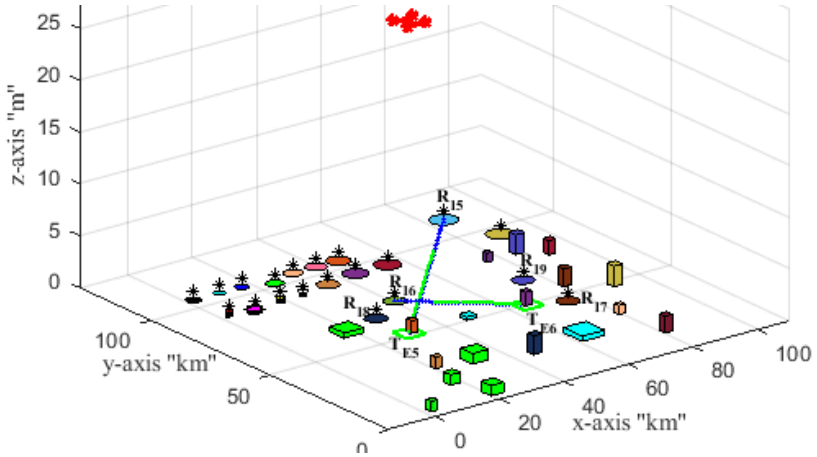


Figure 5.8 Detection of two target objects T_{E_5} and T_{E_6} (inside green squares) and simultaneous allocation of agents R_{15} and R_{16} (blue trajectories); agents R_{15} and R_{16} return to service (green trajectories)

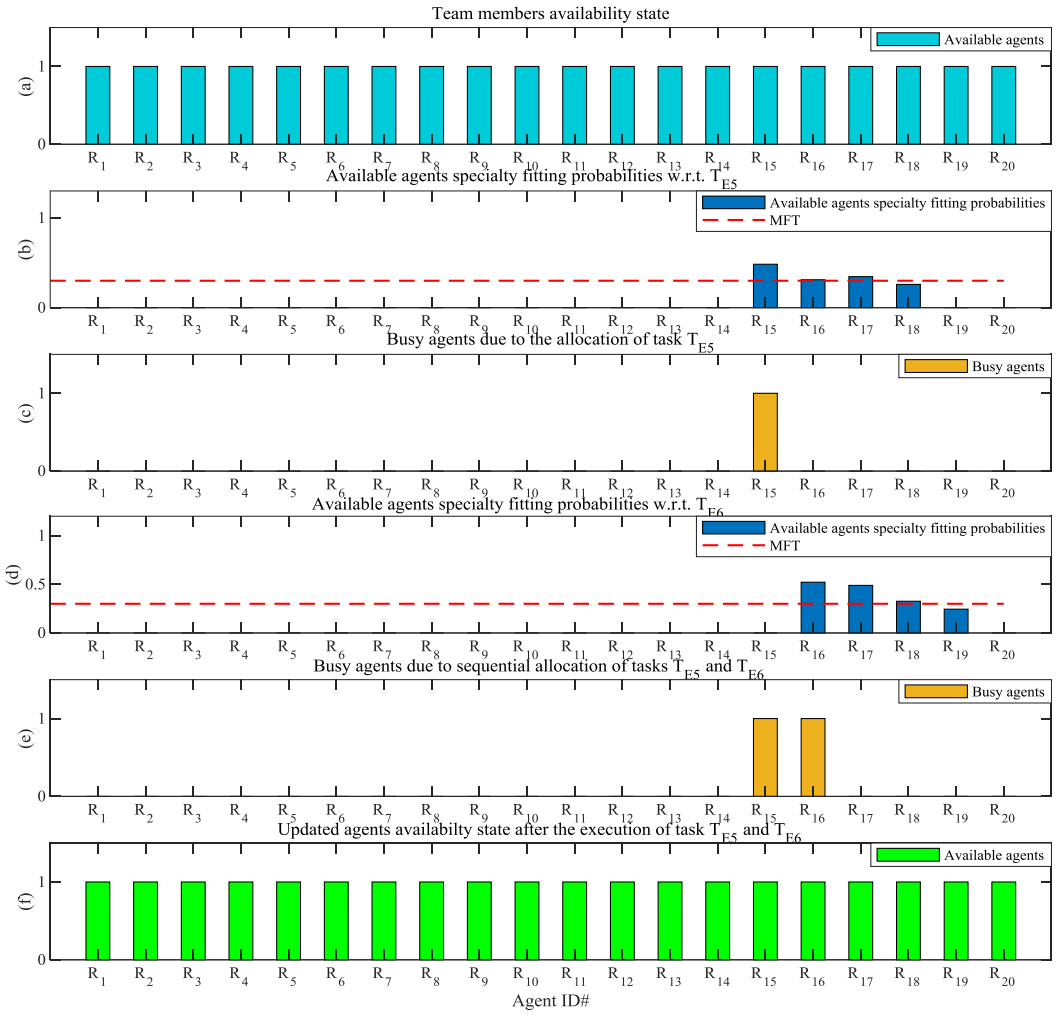


Figure 5.9 Sequential allocation of multiple agents to two simultaneously detected targets T_{E_5} and T_{E_6} and the agents' dynamic availability state

After that, the system processes the second detected task. The latter is detected simultaneously with a recognition confidence level equal to 0.81 (indicated in Table 5.16, second column from the left) considering the updated availability vector by Eq. (4.12). The resulting specialty fitting probabilities of agents R_{16}, R_{17}, and R_{18} achieve the minimum MFT, as indicated in Figure 5.9d. However, agent R_{16} is the most qualified agent with respect to a target of type T_{E_6}. Therefore, the system, through Eq. (4.20), responds and sequentially assigns R_{16} to allocate to T_{E_6} along with the first assignment of R_{15} to T_{E_5}. This allocation results in an updated availability state of agent R_{16}

which is changed to “busy” (Figure 5.9e). Finally, when the allocated tasks are performed, then the agents R_{15} and R_{16} return to the service.

In Figure 5.8, the blue paths are the forward paths (busy state) and the green paths are the returning paths (available state). Their availability states become “available,” as indicated in Figure 5.9f. This example is introduced to demonstrate the basic operation of the sequential assignments, whereas the agent that is selected to respond to the first task allocation is not the most specialized one that is to be allocated to the second task. In the following examples, additional complex cases are introduced.

Table 5.16 Team fitting probabilities and availability state with respect to two target objects of types T_{E5} and T_{E6} that are detected simultaneously and allocated sequentially with deactivated attendance $\wp = 1$ in Eq. (4.16)
* indicates the score of the most qualified agents allocated to the corresponding detected tasks. MFT (η=0.3) (all agents are available)

Target objects' recognition confidence level		*Agent ID*	*Agent's Availability 1:Available; 0:Withdrawn; ƥ:Busy (ƥ=0)* ($\boldsymbol{\vartheta}_{Av}$)			*Available agents' fitting probabilities* ($\boldsymbol{\Psi}_{MFT}$)		*MFT*
			w.r.t. 1st target	*w.r.t. 2nd target*	*Due to the allocation of the two tasks*	*w.r.t. 1st target*	*w.r.t. 2nd target*	
T_{F_1}	0	R_1	1	1	1	0	0	
T_{F_2}	0	R_2	1	1	1	0	0	
T_{F_3}	0	R_3	1	1	1	0	0	
T_{F_4}	0	R_4	1	1	1	0	0	
T_{F_5}	0	R_5	1	1	1	0	0	0.3
T_{A_1}	0	R_6	1	1	1	0	0	
T_{A_2}	0	R_7	1	1	1	0	0	
T_{E_1}	0	R_8	1	1	1	0	0	
T_{E_2}	0	R_9	1	1	1	0	0	
T_{E_3}	0	R_{10}	1	1	1	0	0	
T_{E_4}	0	R_{11}	1	1	1	0	0	
T_{E_5}	0.86	R_{12}	1	1	1	0	0	
T_{E_6}	0.81	R_{13}	1	1	1	0	0	
T_{E_7}	0	R_{14}	1	1	1	0	0	
		$\boldsymbol{R_{15}}$	**1**	***ƥ***	***ƥ***	**0.48***	0	
		$\boldsymbol{R_{16}}$	**1**	**1**	***ƥ***	0.31	**0.52***	
		R_{17}	1	1	1	0.34	0.49	
		R_{18}	1	1	1	0.26	0.33	
		R_{19}	1	1	1	0	0.24	
		R_{20}	1	1	1	0	0	

Example C5-2: Most Qualified Agent Is Withdrawn

This example is similar to the previous one except that the agent R_{15} is withdrawn when the tasks T_{E_5} and T_{E_6} are detected. Figure 5.10 indicates that the system through Eq. (4.20) selects alternative agent R_{17} to respond to and allocate the task T_{E_5} because R_{17} is the most qualified agent amongst the available qualified agents (Figure 5.11b). This allocation updates the availability state of agent R_{17} to "busy" as indicated in Figure 5.11c. In parallel, the system continues to process the simultaneously detected task T_{E_6}. The agent R_{16} achieves the highest fitting probability that meets MFT (Figure 5.11d). Therefore, the system responds sequentially and assigns the agent R_{16} to the task T_{E_6}. Table 5.17 shows that the agent R_{15} is withdrawn, whereas the agents R_{16} and R_{17} are available to be allocated to the first task. Then, when the system allocates T_{E_5}, the availability state of R_{17} is changed to "busy" whereas the agent R_{16} is still available to be allocated to task T_{E_6}. As a result, the availability state of this agent is changed to "busy" (Figure 5.11e). This example illustrates the system's reliability to continue searching amongst the remaining available agents to allocate to the tasks that are detected simultaneously. A numerical example is introduced in Appendix A as 5.3.C5. This example provides a logical procedure of computing the fitting probabilities of the sequential task allocation process of this sample case.

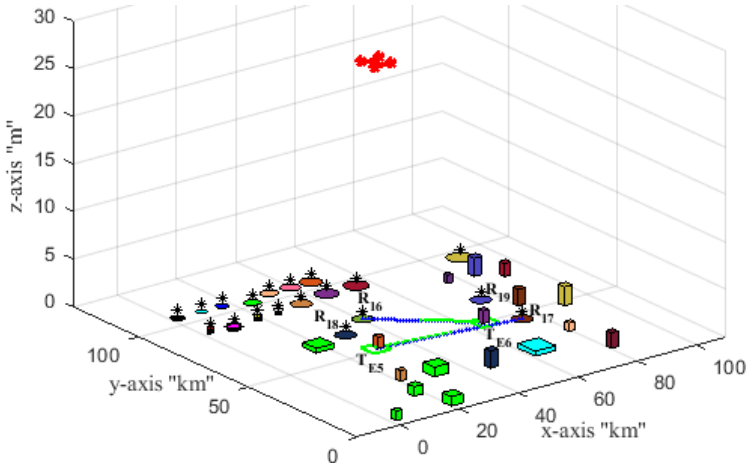


Figure 5.10 Detection of two target objects T_{E_5} and T_{E_6} (inside green squares) and simultaneous allocation of agents R_{17} and R_{16} (blue trajectories). Then agents R_{17} and R_{16} return to service (green trajectories)

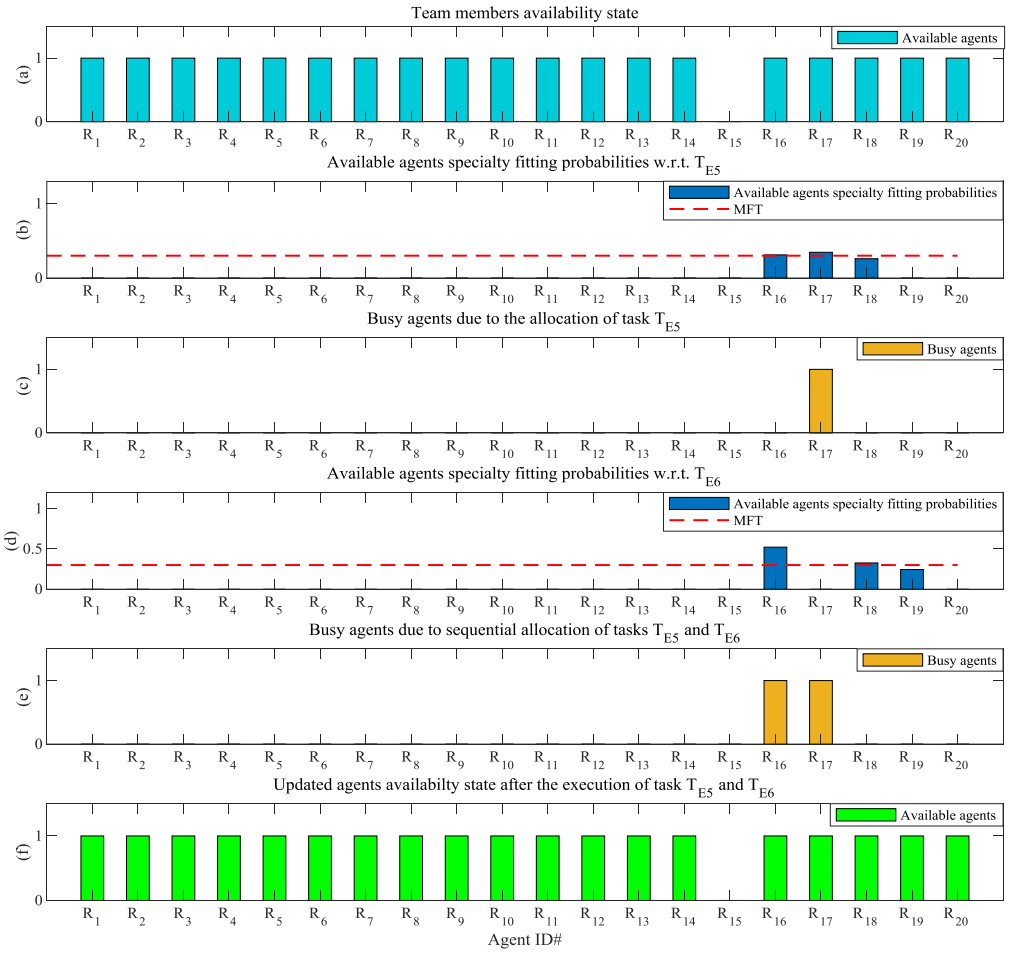


Figure 5.11 Sequential allocation of multiple agents to two simultaneously detected targets T_{E_5} and T_{E_6} and the agents' dynamic availability state.

Table 5.17 Team fitting probabilities and availability state with respect to two target objects of types T_{E5} and T_{E6} detected simultaneously and allocated sequentially with deactivated attendance ($\eth = 1$) in Eq. (4.16)
* indicates the score of the most qualified agents that are to be allocated to the corresponding detected tasks, MFT (η=0.3). (The most specialized agent R_{15} is withdrawn.)

Target objects' recognition confidence level		*Agent ID*	*Agent's availability 1:Available; 0:Withdrawn; þ:Busy* ($\boldsymbol{\vartheta}_{Av}$)			*Available agents' fitting probabilities* ($\boldsymbol{\Psi}_{MFT}$)		*MFT*
			w.r.t. 1st target	*w.r.t. 2nd target*	*Due to the allocation of the two tasks*	*w.r.t. 1st target*	*w.r.t. 2nd target*	
T_{F_1}	0	R_1	1	1	1	0	0	
T_{F_2}	0	R_2	1	1	1	0	0	
T_{F_3}	0	R_3	1	1	1	0	0	
T_{F_4}	0	R_4	1	1	1	0	0	
T_{F_5}	0	R_5	1	1	1	0	0	0.3
T_{A_1}	0	R_6	1	1	1	0	0	
T_{A_2}	0	R_7	1	1	1	0	0	
T_{E_1}	0	R_8	1	1	1	0	0	
T_{E_2}	0	R_9	1	1	1	0	0	
T_{E_3}	0	R_{10}	1	1	1	0	0	
T_{E_4}	0	R_{11}	1	1	1	0	0	
T_{E_5}	0.86	R_{12}	1	1	1	0	0	
T_{E_6}	0.81	R_{13}	1	1	1	0	0	
T_{E_7}	0	R_{14}	1	1	1	0	0	
		R_{15}	**0**	0	0	0	0	
		$\boldsymbol{R_{16}}$	**1**	**1**	***þ***	0.31	**0.52***	
		$\boldsymbol{R_{17}}$	**1**	***þ***	***þ***	**0.34***	**0**	
		R_{18}	1	1	1	--	0.32	
		R_{19}	1	1	1	0	---	
		R_{20}	1	1	1	0	0	

Example C5-3: Most Qualified Agents Are Withdrawn

This example is an extension of the previous one with the difference that the agents R_{15} and R_{16} are "withdrawn" at the time when the tasks T_{E_5} and T_{E_6} are detected. Figure 5.12 indicates that the system through Eq. (4.20) selects an alternative agent R_{17} to respond to and allocate the task T_{E_5} because R_{17} is the only qualified agent amongst the available agents (Figure 5.13b). This allocation updates the availability state of agent R_{17} to "busy" as indicated in Figure 5.13c. In parallel, the system keeps processing the simultaneously detected task T_{E_6}. Although the agent R_{16} is the most specialized agent and the agent R_{17} is the second specialized agent that can respond to this task (as indicated in Table 5.18, fourth column from the right), these two agents are not available, given that R_{16} is "withdrawn" and agent R_{17} is "busy" with the allocation of the task T_{E_5} (as indicated in Table 5.18, sixth column from the right). Therefore, the system keeps searching for another available agent to respond to and allocate this task. Figure 5.13d indicates that agent R_{18} is available and meets the minimum MFT. Then, the system intervenes sequentially and assigns this agent to the task T_{E_6}.

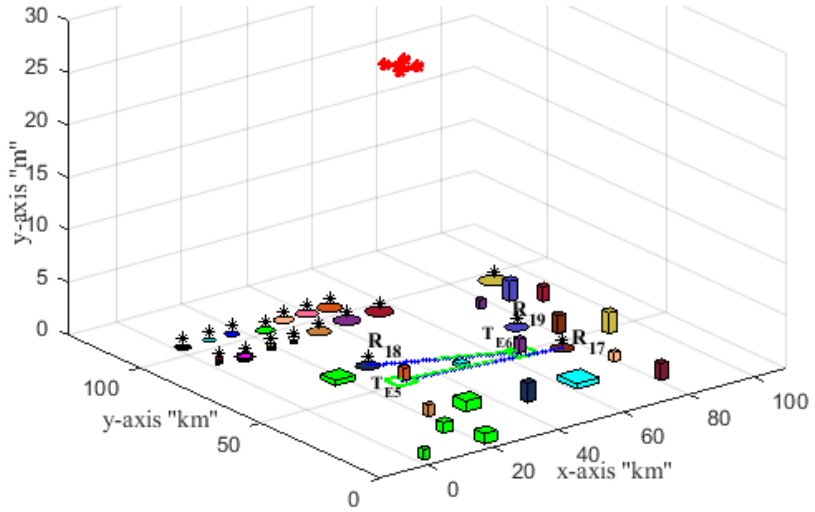


Figure 5.12 Detection of two target objects T_{E_5} and T_{E_6} (inside green squares) and simultaneous allocation of agents R_{17} and R_{18} (blue trajectories). Then agents R_{17} and R_{18} return to service (green trajectories).

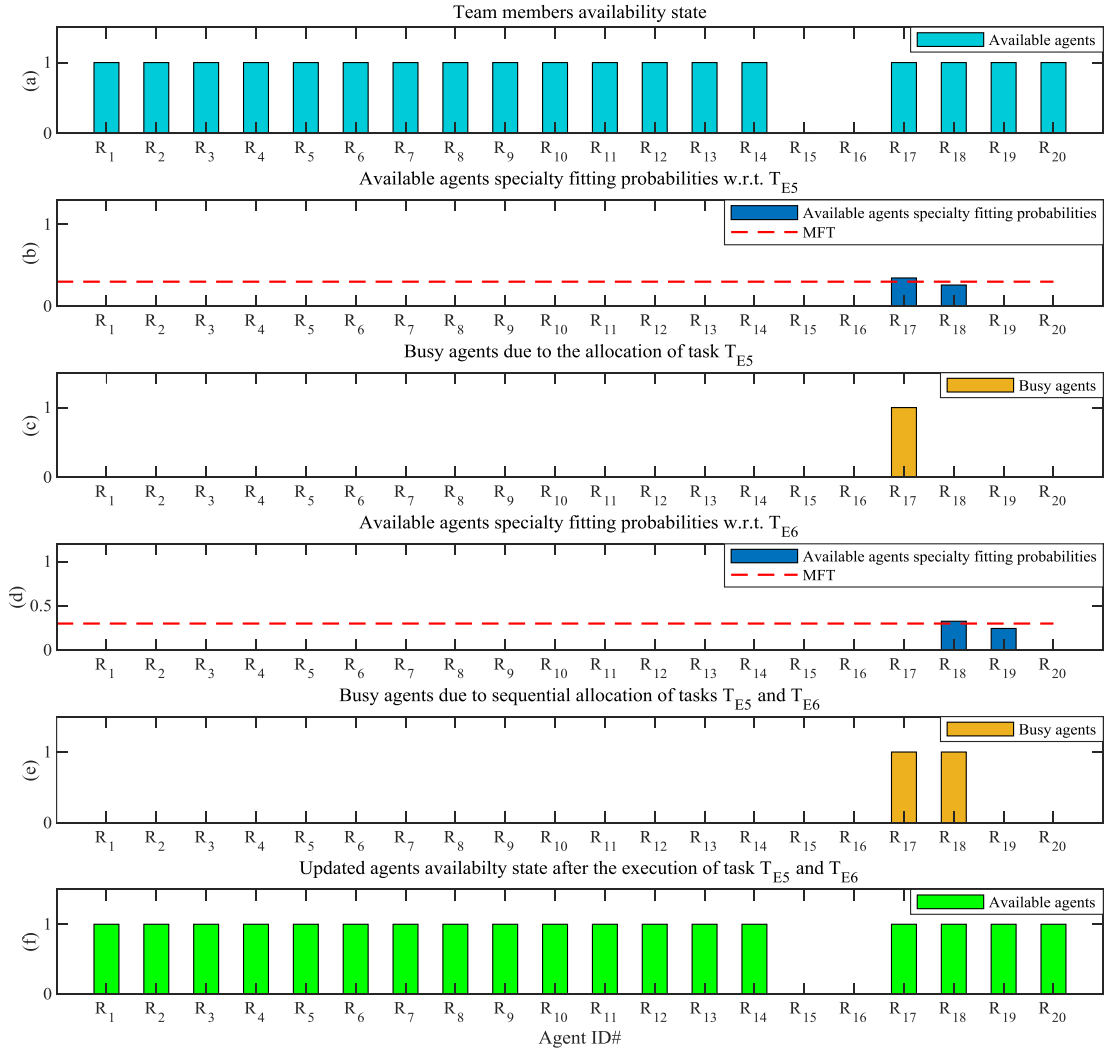


Figure 5.13 Sequential allocation of multiple agents to two simultaneously detected targets T_{E_5} and T_{E_6} and the agents' dynamic availability state

This example also illustrates the system's reliability when the most qualified agents are not available and the less competent agents are already involved in the task allocation process of alternative tasks. In this example, the system demonstrates reliability as it continues to search for another specialized agent amongst the remaining available agents to accomplish the mission goals supported by the dynamic availability state of the individual agents.

Table 5.18 Team fitting probabilities and availability state with respect to two target objects of types T_{E5} and T_{E6} detected simultaneously and allocated sequentially with deactivated attendance $\mathbb{p} = 1$ in Eq. (4.16)
*indicates the score of the most qualified agents to be allocated to the detected corresponding tasks. MFT (η=0.3). (The two most specialized agents R_{15} and R_{16} are withdrawn.)

Target objects' recognition confidence level		*Agent ID*	*Agent's availability 1:Available; 0:Withdrawn; þ:Busy* ($\boldsymbol{\vartheta}_{Av}$)			*Target-agent specialty matching probability* ($\boldsymbol{Q}$)		*Available agents' fitting probabilities* ($\boldsymbol{\Psi}_{MFT}$)		*MFT*
			w.r.t. 1st target	*w.r.t. 2nd target*	*Due to the allocation of the two tasks*	*w.r.t. 1st target*	*w.r.t. 2nd target*	*w.r.t. 1st target*	*w.r.t. 2nd target*	
T_{F_1}	0	R_1	1	1	1	0	0	0	0	0.3
T_{F_2}	0	R_2	1	1	1	0	0	0	0	
T_{F_3}	0	R_3	1	1	1	0	0	0	0	
T_{F_4}	0	R_4	1	1	1	0	0	0	0	
T_{F_5}	0	R_5	1	1	1	0	0	0	0	
T_{A_1}	0	R_6	1	1	1	0	0	0	0	
T_{A_2}	0	R_7	1	1	1	0	0	0	0	
T_{E_1}	0	R_8	1	1	1	0	0	0	0	
T_{E_2}	0	R_9	1	1	1	0	0	0	0	
T_{E_3}	0	R_{10}	1	1	1	0	0	0	0	
T_{E_4}	0	R_{11}	1	1	1	0	0	0	0	
T_{E_5}	0.86	R_{12}	1	1	1	0	0	0	0	
T_{E_6}	0.81	R_{13}	1	1	1	0	0	0	0	
T_{E_7}	0	R_{14}	1	1	1	0	0	0	0	
		R_{15}	**0**	**0**	**0**	0.49	0	---	---	
		R_{16}	**0**	**0**	**0**	0.31	0.52	---	---	
		$\boldsymbol{R_{17}}$	**1**	***þ***	***þ***	0.34	0.49	**0.34***	---	
		$\boldsymbol{R_{18}}$	**1**	**1**	***þ***	0.26	0.32	---	**0.32***	
		R_{19}	1	1	1	0	0.24	0	---	
		R_{20}	1	1	1	0	0	0	0	

Example C5-4: Activated Attendance

This example is similar to Example C5-2 above but with active attendance: $\mathfrak{p} = 1 - (\eta - 0.1)$ in Eq. (4.16). In this example, agent R_{15} is withdrawn at the time when the targets T_{E_5} and T_{E_6} are detected. Figure 5.14 indicates that the system selects an alternative agent R_{18} to respond and be allocated to the task T_{E_5} because R_{18} is equipped with the required specialized functionality and is positioned very close the target location. As a result, R_{18} becomes the most qualified agent amongst the available qualified agents based on Eq. (4.20) with active attendance, $\boldsymbol{\vartheta}_{Att}$ (Figure 5.15b). In contrast to the case when the attendance is deactivated in Example C5-2, the system response differs and agent R_{17} should be selected to be allocated to this task. However, agent R_{17} must travel $45\,km$ to the target location whereas R_{18} is only $13.6\,km$ away from the target location. In addition, the difference in the agent/task relative robustness of these two agents with respect to the task T_{E_5} is negligible given that agent R_{18} is less suitable than agent R_{17} by only 10%, as defined in Table 5.2 (third column from the right); however, the latter can arrive at the target location in 54 minutes whereas agent R_{18} requires only 16 minutes to arrive at the target location. This allocation updates the availability state of agent R_{18} to "busy" as indicated in Figure 5.15c.

In parallel, the system continues processing the simultaneously detected task T_{E_6}. Agent R_{17} achieves the highest fitting probability that meets MFT (Figure 5.15d) because it is specialized and located in close proximity to the target position (Figure 5.15d). Therefore, the system responds sequentially and assigns agent R_{17} to allocate to the task T_{E_6}. In contrast to the case when the attendance is deactivated in Example C5-2, the system response differs and agent R_{16} is selected to deal with this task. However, agent R_{16} must travel $33.3\,km$ to the target location whereas R_{17}

is positioned only 10.5 km away from the target location. In addition, the difference in agent/task relative robustness of the agents' functionality with respect to the task T_{E_6} is negligible. The agent R_{17} is less suitable than agent R_{16} by only 4% as defined in Table 5.2 (second column from the right); however, the latter can arrive at the target location in 40 minutes whereas the agent R_{17} requires only 12 minutes to arrive at the target location. This allocation updates the availability state of agent R_{17} to "busy" as indicated in Figure 5.15e. Table 5.19 provides detailed results for this test case.

This example illustrates that the proposed framework is adaptable to wide ranging workspace applications as it can optimize the time of the specialized responders to be allocated to corresponding detected tasks. In addition, this example also provides evidence that the system is sufficiently reliable to continue searching amongst the remaining available and closer qualified agents to accomplish the mission goals.

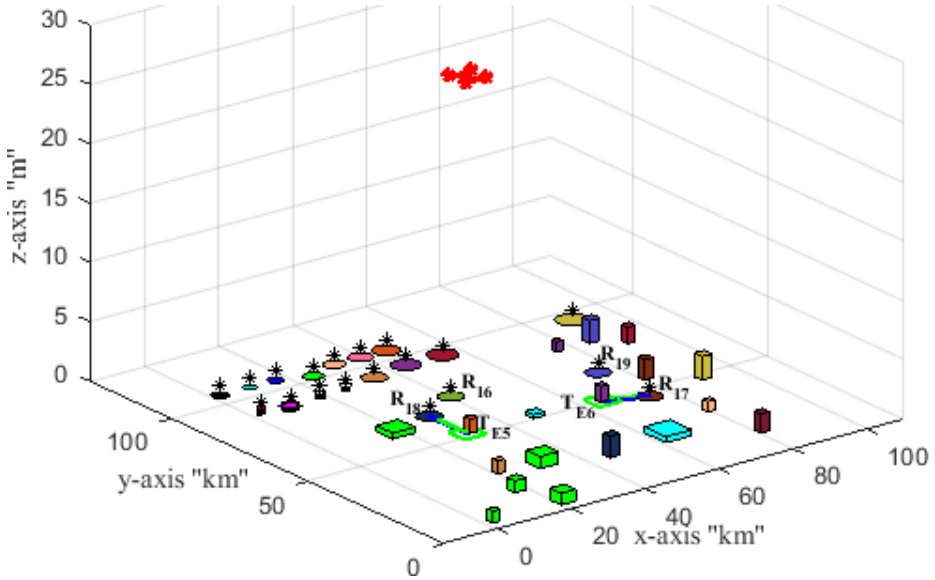


Figure 5.14 Detection of two target objects T_{E_5} and T_{E_6} (inside green squares) and simultaneous allocation of agents R_{18} and R_{17} (blue trajectories), with active attendance. Then agents R_{18} and R_{17} return to service (green trajectories).

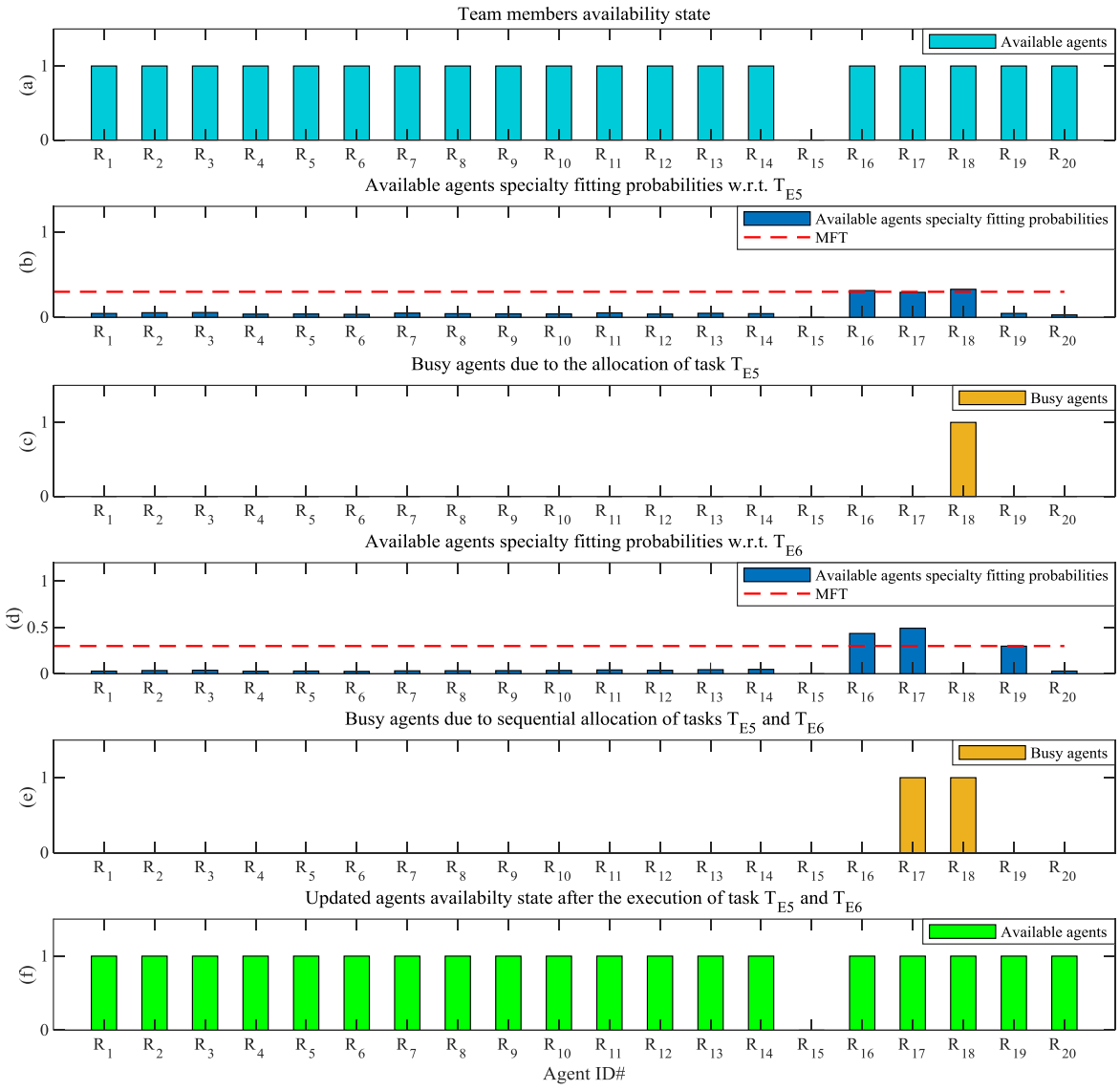


Figure 5.15 Sequential allocation of multiple agents to two simultaneously detected targets T_{E_5} and T_{E_6} and the agents' dynamic availability state, with active attendance.

Table 5.19 Team members' fitting probabilities and availability state with respect to two target objects of types T_{E5} and T_{E6} that are detected simultaneously and allocated sequentially with activated attendance $\mathbb{p} = 1 - (\eta - 0.1)$ in Eq. (4.16)
*indicates the score of the most qualified agents allocated to the corresponding detected tasks, MFT (η=0.3). The most specialized agents are withdrawn.

Target objects' recognition confidence level		*Agent ID*	*Agent's Availability 1:Available; 0:Withdrawn; þ:Busy* ($\boldsymbol{\vartheta}_{Av}$)			*Available agents' fitting probabilities* ($\boldsymbol{\Psi}_{MFT}$)		*MFT*
			w.r.t. 1st target	*w.r.t. 2nd target*	*Due to the allocation of the two tasks*	*w.r.t. 1st target*	*w.r.t. 2nd target*	
T_{F_1}	0	R_1	1	1	1	0	0	
T_{F_2}	0	R_2	1	1	1	0	0	
T_{F_3}	0	R_3	1	1	1	0	0	
T_{F_4}	0	R_4	1	1	1	0	0	
T_{F_5}	0	R_5	1	1	1	0	0	0.3
T_{A_1}	0	R_6	1	1	1	0	0	
T_{A_2}	0	R_7	1	1	1	0	0	
T_{E_1}	0	R_8	1	1	1	0	0	
T_{E_2}	0	R_9	1	1	1	0	0	
T_{E_3}	0	R_{10}	1	1	1	0	0	
T_{E_4}	0	R_{11}	1	1	1	0	0	
T_{E_5}	0.86	R_{12}	1	1	1	0	0	
T_{E_6}	0.81	R_{13}	1	1	1	0	0	
T_{E_7}	0	R_{14}	1	1	1	0	0	
		R_{15}	**0**	**0**	**0**	**0**	**0**	
		R_{16}	1	1	1	0.31	0.44	
		$\boldsymbol{R_{17}}$	**1**	**1**	***þ***	0.29	**0.49***	
		$\boldsymbol{R_{18}}$	**1**	***þ***	***þ***	**0.33***	0	
		R_{19}	1	1	1	0	0.29	
		R_{20}	1	1	1	0	0	

5.3.C6 Complex Tasks

This test scenario is used to validate the proposed approach for complex tasks. The latter are characterized based on simultaneous detection of target objects from multiple classes. Replicating the example introduced in section 4.4.5, three different target objects (represented by red, blue, and green markers) are assumed to be distributed over two different environments. Each environment is characterized by terrains that are water-covered (cyan) or land (brown) areas, as shown in Figure 5.16.

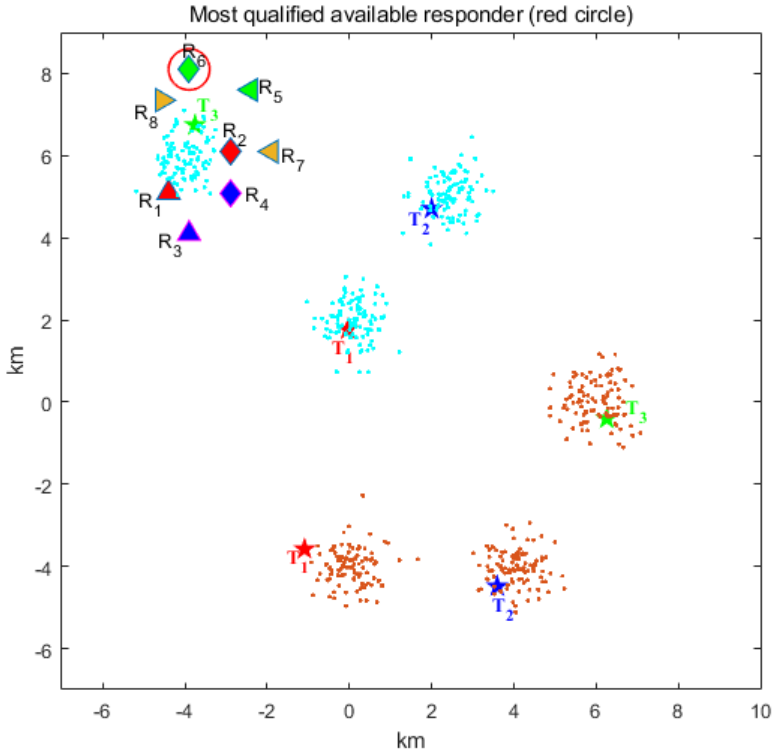


Figure 5.16 Assigning robot R_6 to a task of type T_3 (green star) on water-covered area

A swarm of eight individual robots is deployed in a two-dimensional workspace. Six of these robots, $\{R_1,\ R_2, R_3, R_4, R_5, R_6\}$, are divided in groups of two robots. Each pair of robots is specialized to perform on each type of target (T_1, T_2, T_3), and each robot has an individual preference for a specific environment of either water workspace (W_{WS}) or land workspace (L_{WS})) associated with its physical construction. However, the last two robots $\{R_7, R_8\}$ are amphibious

robotic agents which can function on water-covered space or on land. These two robots are specialized to perform on targets of type T_3 and T_2, respectively. The class of X_k, can be related to the nature of the target object or to specific environment constraints. Therefore, $X_k \in \{T_1, T_2, T_3, W_{WS}, L_{WS}\}$. The targets and the environmental constraints are color-coded in the figures below under consideration of the binary and modulated specialization encoding, introduced in section 4.4.1, and defined in Table 5.20 and Table 5.21 respectively.

Table 5.20 Formulation of $a = 8$ robotic agents with binary encoded specialization functionalities to serve on three complex tasks with different requirements. Each task is characterized by two classes.

Agents' specialization binary vectors	Secondary Classes				Primary Class
	T_1	T_2	T_3	L_{WS}	W_{WS}
$\boldsymbol{S_1}$	1	0	0	1	0
$\boldsymbol{S_2}$	1	0	0	0	1
$\boldsymbol{S_3}$	0	1	0	1	0
$\boldsymbol{S_4}$	0	1	0	0	1
$\boldsymbol{S_5}$	0	0	1	1	0
$\boldsymbol{S_6}$	0	0	1	0	1
$\boldsymbol{S_7}$	0	0	1	1	1
$\boldsymbol{S_8}$	0	1	0	1	1

Table 5.21 Formulation of $a = 8$ robotic agents with modulated encoded specialization functionalities to serve on three complex tasks with different requirements. Each task is characterized by two classes.

Agents' specialization modulated vectors	Secondary Classes				Primary Class
	T_1	T_2	T_3	L_{WS}	W_{WS}
$\boldsymbol{S_1}$	0.7	0	0	0.6	0
$\boldsymbol{S_2}$	0.7	0	0	0	0.6
$\boldsymbol{S_3}$	0	0.7	0	0.6	0
$\boldsymbol{S_4}$	0	0.7	0	0	0.6
$\boldsymbol{S_5}$	0	0	0.7	0.6	0
$\boldsymbol{S_6}$	0	0	0.7	0	0.6
$\boldsymbol{S_7}$	0	0	0.7	0.3	0.3
$\boldsymbol{S_8}$	0	0.7	0	0.3	0.3

Table 5.22 defines the categorization of the given targets and their specific surrounding environment constraints in five different classes. The categorizations of the primary and secondary classes considered for the primary encoder, Eq. (4.21), that is implemented in this simulation, are indicated in Table 5.23, whereas water covered workspaces, W_{WS}, is considered a primary class

because in order to intervene on water a robot must be floating. The features belonging to each class are considered available in the simulation.

Table 5.22 Categorization of targets and environment constraints

Target or environment constraint	Class
T_1 *(red)*	C_1
T_2 *(blue)*	C_2
T_3 *(green)*	C_3
L_{WS} *(brown)*	C_4
W_{WS} *(cyan)*	C_5

Table 5.23 Categorization of targets and environment constraints associated with corresponding primary encoder entities

Target or environment constraint	Class	q_{ρ_k} (competency priority category)
T_1 (red)	C_1	$\bar{q}_{\rho_1}$ (secondary)
T_2 (blue)	C_2	$\bar{q}_{\rho_2}$ (secondary)
T_3 (green)	C_3	$\bar{q}_{\rho_3}$ (secondary)
L_{WS} (brown)	C_4	$\bar{q}_{\rho_4}$ (secondary)
W_{WS} (cyan)	C_5	$\bar{\bar{q}}_{\rho_5}$(primary)

In these tests, the proposed mechanism is simulated in two scenarios. Each scenario presents a test case and considers the binary and modulated definitions of robots' specialization (defined in section 4.4.1.A and 4.4.1.B respectively). From the perspective of human operator supervision, discussed in section 4.4.4, these tests consider a $LSFL$ for the task-agent matching process with an imposed MFT value of η. The latter is set by Eq. (4.17): $\eta = 0.3$. The proposed approach, with the extension introduced in section 4.4.5, is tested for a complex task to be allocated to an agent. This task is characterized by multiple classes. The latter are detected simultaneously amongst four secondary classes and one primary class, related to intervention over water which calls for specific characteristics of the robotic agent. The first test scenario is performed when all of the team members are available. On the other hand, the second test scenario is performed when the most specialized agents are not available or "withdrawn." The agents' availability is set by Eq. (4.12). However, the agents' attendance (4.13) is deactivated in Eq. (4.16) with $\mathfrak{p} = 1$, and the task allocation probability is computed by placing full weight on the agents' specialty-based

qualifications, $\boldsymbol{Q}$. Thus, the purpose of these test cases is to validate selectively of the extended approach through Eq. (4.24), introduced in section 4.4.5, amongst the agents' primary and secondary specialized capabilities, within $\boldsymbol{Q}$, when a complex task is detected.

Example C6-1: All Specialized Agents Are Available

In this test case, the agents' task allocation fitting probabilities are computed based on the match between the agents' specialized capabilities and the detected target, $\boldsymbol{Q}$, Eq. (4.18). Then the qualified agent/agents are selected to be allocated to the detected task based on the agents' primary capabilities by using Eq. (4.24). Figure 5.16 shows that a target of type T_3 (green star) is detected over a water-covered area W_{ws} (cyan dots) on the upper left hand side of the figure, Table 5.24 (second column from the left), Eq. (4.8). Figure 5.17a shows all team members' fitting probabilities considering the binary encoded task allocator (Table 5.24, fourth column from the left), and Figure 5.17b indicates the fitting probabilities of the "available" team members only (Table 5.24, third column from the right) given that a primary class, W_{ws}, is detected. This detection is leveraged by the prioritization scheme to activate the qualified robots that achieve MFT and possess the corresponding primary capability as assignment priorities. These robots are $R_2, R_4, R_6,$ and R_7. The allocation of R_5, which is no longer considered due to lacking the primary capability to operate on water (as shown in Figure 5.17c, Table 5.24, second column from the right) is suspended. As a result, in the case that the most qualified agent must be assigned, then the system assigns the robot R_6 to this task because it is the most qualified in terms of task-agent specialty-based matching. This robot is active for assignment priority because it possesses a primary capability, $\bar{\bar{q}}$, as indicated in Table 5.23, with respect to the detected task that imposes operating on water. For the same reason, when all qualified agents must be assigned, the agents $R_2, R_4, R_6,$ and R_7 can be assigned to the current task. This selectivity process increases the

safety and robustness of the task allocation operation. When the qualified agents must be selected based on the detection of multiple classes, the latter characterize a given complex task. In this example, the most specialized agent is qualified with the primary capability. Detailed results are introduced in Table 5.24. Numerical calculations of this test case are detailed in Appendix A, 5.3.C6 Numerical Example.

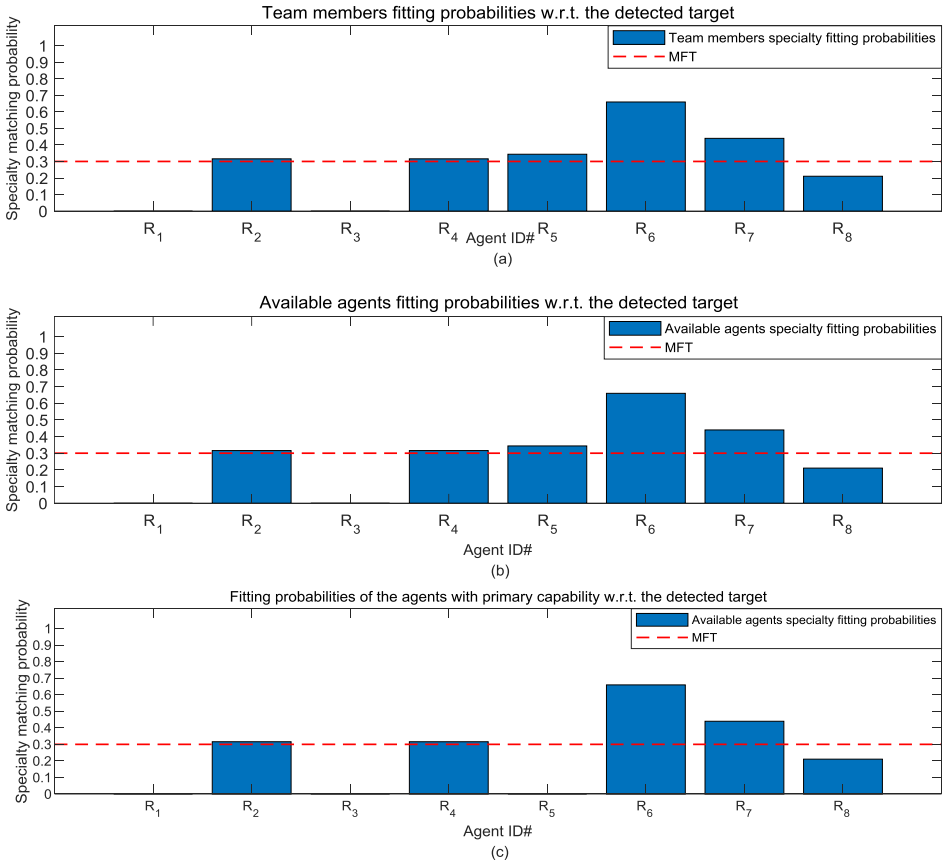


Figure 5.17 Binary task allocator: (a) team members' fitting probabilities w.r.t. the detected task, T_3 over W_{ws}; (b) specialty fitting probabilities of the available agents; and (c) specialty fitting probabilities of the agents that have active assignment priority w.r.t. the detection of the current task and the agent's primary capability

Alternatively, a modulated task allocator, defined in Table 5.21, is also tested for the same detected task when the team members are all available. The results are introduced in Figure 5.18 and Table 5.25. The modulated task allocator returns similar results in terms of the selectively of the extended approach. The main difference is that the fitting levels of overall task allocation

probabilities of the individual robots are different, whereas the modulated task allocator is highly deterministic in the evaluation of the task allocation probabilities based on the modulated level of the agents' suitability with respect to the detected task. As indicated in Table 5.21, the agents' suitability to the given workspaces is modulated with lower levels than the agents' suitability to perform the target task types. This modulation, indicated in Table 5.21, gives more importance to the agent's specialized functionality with respect to the target type and less importance to the agent's ability to work on a specific workspace. This results in different overall task allocation fitting probabilities. As a result, when a target of type T_3 is detected over a water-covered area, W_{ws}, the modulated task allocator places higher weight on T_3 than on W_{ws}. Figure 5.18 and Table 5.25 show that the robots R_2 and R_4 no longer achieve MFT and are no longer qualified. In spite of that, the detection of W_{ws} enables the primary selector to suspend the robots that are not capable of operating on water. In this test example, the binary and modulated task allocators introduce similar behaviour in terms of selecting the most qualified agent; however, in the second example which will be analyzed below (Example C6-2), a different response will result.

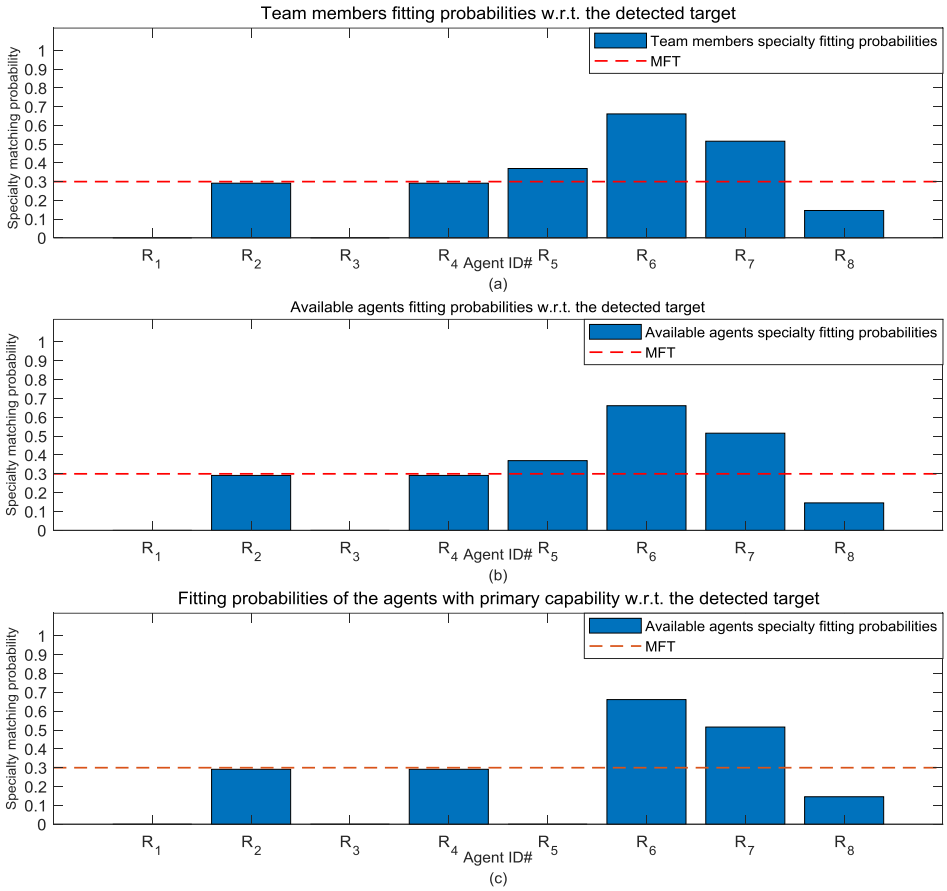


Figure 5.18 Modulated task allocator: (a) team members' fitting probabilities w.r.t. the detected task, T_3 over W_{ws}; (b) specialty fitting probabilities of the available agents; and (c) specialty fitting probabilities of the agents that have active assignment priority w.r.t. the detection of the current task and the agent's primary capability

Table 5.24 Specialty-based task allocation dynamics for team members with respect to a complex task recognized based on the detection of multiple classes; binary encoded task allocator with a deactivated attendance with $\mathfrak{p} = 1$ in Eq. (4.16). (All agents are available.)

Target objects' recognition confidence level	*Agent ID#*	*Target-agent specialty matching probability* ($\mathfrak{p}\boldsymbol{Q}$) ***Binary encoding***	*Agents' availability* 1:*Available* 0:*Withdrawn* ($\boldsymbol{\vartheta}_{Av}$)	*Available agents' attendance level* ($(1-\mathfrak{p})\boldsymbol{\vartheta}_{Att}$)	*Available qualified agents' fitting probabilities* ($\boldsymbol{\Psi}_{MFT}$) ***Binary encoding***	*Fitting probabilities of the agents with the detected primary capability* ($\boldsymbol{h}_{\rho}$)	*MFT* (η)
T_1 0.00	R_1	0.00	1	0.00	0.00	0.00	
T_2 0.00	R_2	0.31	1	0.00	0.31	0.31	
T_3 0.68	R_3	0.00	1	0.00	0.00	0.00	0.3
W_L 0.00	R_4	0.31	1	0.00	0.31	0.31	
W_W 0.63	R_5	0.34	1	0.00	0.34	Suspended	
	$\boldsymbol{R_6}$	**0.65**	**1**	**0.00**	**0.65**	**0.65***	
	R_7	0.43	1	0.00	0.43	0.43	
	R_8	0.21	1	0.00	0.00	0.00	

Table 5.25 Specialty-based task allocation dynamics for team members with respect to a complex task recognized based on the detection of multiple classes; modulated encoded task allocator with a deactivated attendance with $\mathfrak{p} = 1$ in Eq. (4.16). (All agents are available.)

Target objects' recognition confidence level	*Agent ID#*	*Target-agent specialty matching probability* ($\mathfrak{p}\boldsymbol{Q}$) ***Modulated encoding***	*Agents' availability* 1:*Available* 0:*Withdrawn* ($\boldsymbol{\vartheta}_{Av}$)	*Available agents' attendance level* ($(1-\mathfrak{p})\boldsymbol{\vartheta}_{Att}$)	*Available qualified agents' fitting probabilities* ($\boldsymbol{\Psi}_{MFT}$) ***Modulated encoding***	*Fitting probabilities of the agents with the detected primary capability* ($\boldsymbol{h}_{\rho}$)	*MFT* (η)
T_1 0.00	R_1	0.00	1	0.00	0.00	0.00	
T_2 0.00	R_2	0.29	1	0.00	0.00	0.00	
T_3 0.68	R_3	0.00	1	0.00	0.00	0.00	0.3
W_L 0.00	R_4	0.29	1	0.00	0.00	0.00	
W_W 0.63	R_5	0.37	1	0.00	0.37	suspended	
	$\boldsymbol{R_6}$	**0.66**	**1**	**0.00**	**0.66**	**0.66***	
	R_7	0.51	1	0.00	0.51	0.51	
	R_8	0.14	1	0.00	0.00	0.00	

Example C6-2: Most Specialized Agents Are Withdrawn

This test case is similar to the previous example except that robots R_6 and R_7 are withdrawn (Table 5.26, fifth column from the left). The purpose of this example is to test the selectivity of the extended approach, Eq. (4.25), amongst available agents which are less competent. Figure 5.19 shows that a target of type T_3 (green star) is detected over a water-covered area, W_{ws} (cyan dots) (Table 5.26, second column from the left), Eq. (4.8). Figure 5.20a shows all team members' fitting probabilities for the binary encoded task allocator (Table 5.26, fourth column from the left). Figure 5.20b indicates the fitting probabilities of the "available" team members only (Table 5.26, third column from the right), given that a primary class, W_{ws}, is detected. This detection is leveraged by the prioritization scheme, Eq. (4.24), to activate the qualified robots that achieve MFT and possess the corresponding primary capability as assignment priorities. As a result, the robots R_2 and R_4 are activated for the task allocation priority and the system suspends robot R_5 from the task allocation operation. Although R_5 presents a higher fitting probability with respect to the detected task, indicated in Figure 5.20b, it is no longer considered due to lacking the primary capability to operate on water (as shown in Figure 5.20c, Table 5.26, second column from the right). As a result, in the application that requires all qualified agents to respond to the detected task, the agents R_2 and R_4 can be assigned. In the case where only the most specialized agent must respond, then the system assigns robot R_2 (Figure 5.20, inside red circle) to the detected task, because it is the most qualified agent, in terms of task-agent specialty-based matching, and it is still active for assignment priority for the reason that it possesses a primary capability $\bar{\bar{q}}$ as indicated in Table 5.23, with respect to the detected task that imposes operating on water.

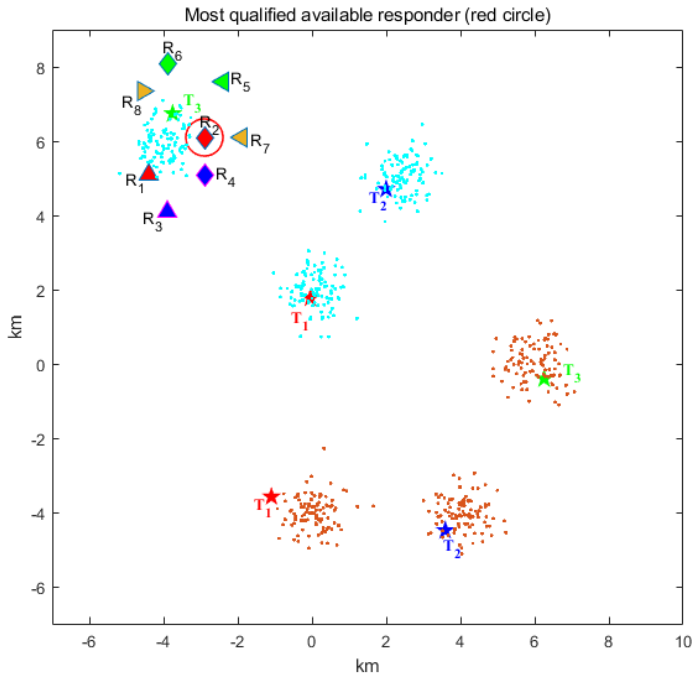


Figure 5.19 Assigning robot R_2 to a task of type 3 (green star) on water-covered area

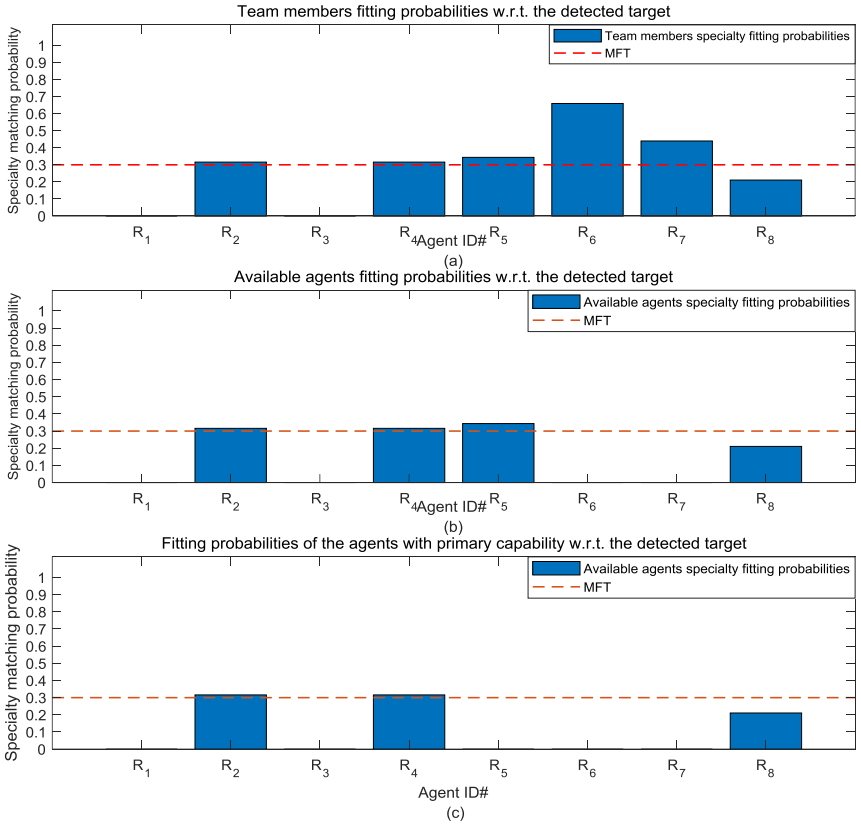


Figure 5.20 Binary task allocator: (a) team members' fitting probabilities w.r.t. the detected task, T_3 over W_{ws}; (b) specialty fitting probabilities of the available agents; and (c) specialty fitting probabilities of the agents that have active assignment priority w.r.t. the detection of the current task and the agent's primary capability

A modulated task allocator, defined in Table 5.21, is also tested for the same detected task when the most competent agents are not available or "withdrawn". The results are indicated in Figure 5.21 and Table 5.27. Agent R_5 is the only qualified agent that achieves MFT as indicated in Figure 5.24b. The modulated task allocator returns similar results to the binary task allocator in terms of the selectivity of the extended approach, Eq. (4.24), where it suspends agent R_5. This time, agents R_2 and R_4 are no longer qualified. The modulated task allocator, defined in Table 5.21, gives lower importance to the agent's ability to work on a specific workspace, given that agents R_2 and R_4 have the capability to operate on water but are not specialized in the detected target type T_3. This results in a different overall task allocation fitting probability. As a result, when a target of type T_3 is detected over a water-covered area, W_{ws}, the modulated task allocator puts higher weight on T_3 than on W_{ws}. Figure 5.21 and Table 5.27 show that robots R_2 and R_4 no longer achieve MFT and are no longer qualified. In spite of that, the detection of W_{ws} enables the primary selector, Eq. (4.21), to suspend the robots that are not capable of operating on water, in this case R_5. As the latter was the only one qualified agent for the task, the result is that no agent is assigned to respond to this task (Table 5.27, second column from the right).

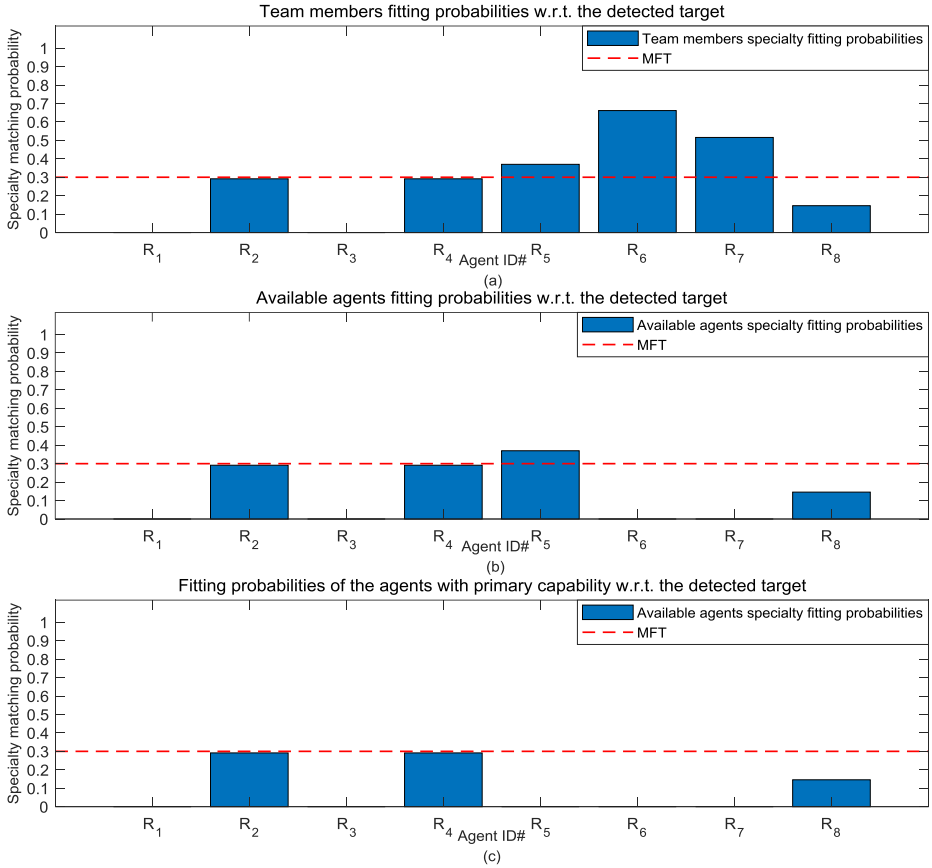


Figure 5.21 Modulated task allocator: (a) team members' fitting probabilities w.r.t. the detected task, T_3 over W_{ws}; (b) specialty fitting probabilities of the available agents; (c) specialty fitting probabilities of the agents that have active assignment priority w.r.t. the detection of the current task and the agent's primary capability

In these two examples, the binary and modulated task allocators generate different behaviours in terms of selecting the qualified agent based on their properties (as discussed in 4.5D1), but the prioritization scheme introduced in section 4.4.5 deals with the task complexity and prevents the agents that may not possess critical capabilities to perform a task, in this case navigating the water. This increased level of safety and robustness is what motivated the development of a more advanced task allocation formalism, as detailed in section 4.4.5, to tackle such complex tasks with an additional level of strategic guidance.

Table 5.26 Specialty-based task allocation dynamics for team members with respect to a complex task recognized based on the detection of multiple classes. Binary encoded task allocator with a deactivated attendance with $\mathfrak{p} = 1$ in Eq. (4.16). (Most qualified agents are "withdrawn.")

Target objects' recognition confidence level	*Agent ID#*	*Target-agent specialty matching probability* $(\mathfrak{p}\boldsymbol{Q})$ ***Binary encoding***	*Agents' availability* 1:*Available* 0:*Withdrawn* $(\boldsymbol{\vartheta}_{Av})$	*Available agents' attendance level* $((1-\mathfrak{p})\boldsymbol{\vartheta}_{Att})$	*Available qualified agents' fitting probabilities* $(\boldsymbol{\Psi}_{MFT})$ ***Binary Encoding***	*Fitting probabilities of the agents with the detected primary capability* $(\boldsymbol{h}_{\rho})$	*MFT* (η)
T_1 0.00	R_1	0.00	1	0.00	0.00	0.00	
T_2 0.00	$\boldsymbol{R_2}$	**0.31**	**1**	**0.00**	**0.31**	**0.31***	
T_3 0.68	R_3	0.00	1	0.00	0.00	0.00	0.3
W_L 0.00	R_4	0.31	1	0.00	0.31	0.31	
W_W 0.63	R_5	0.34	1	0.00	0.34	Suspended	
	R_6	0.65	**0**	0.00	0.00	0.00	
	R_7	0.43	**0**	0.00	0.00	0.00	
	R_8	0.21	1	0.00	0.00	0.00	

Table 5.27 Specialty-based task allocation dynamics for team members with respect to a complex task recognized based on the detection of multiple classes. Modulated encoded task allocator with a deactivated attendance with $\mathfrak{p} = 1$ in Eq. (4.16). (Most qualified agents are "withdrawn.")

Target objects' recognition confidence level	*Agent ID#*	*Target-agent specialty matching probability* $(\mathfrak{p}\boldsymbol{Q})$ ***Modulated encoding***	*Agents' availability* 1:*Available* 0:*Withdrawn* $(\boldsymbol{\vartheta}_{Av})$	*Available agents' attendance level* $((1-\mathfrak{p})\boldsymbol{\vartheta}_{Att})$	*Available qualified agents' fitting probabilities* $(\boldsymbol{\Psi}_{MFT})$ ***Modulated Encoding***	*Fitting probabilities of the agents with the detected primary capability* $(\boldsymbol{h}_{\rho})$	*MFT* (η)
T_1 0.00	R_1	0.00	1	0.00	0.00	0.00	
T_2 0.00	R_2	0.29	1	0.00	0.00	0.00	
T_3 0.68	R_3	0.00	1	0.00	0.00	0.00	0.3
W_L 0.00	R_4	0.29	1	0.00	0.00	0.00	
W_W 0.63	R_5	0.37	1	0.00	0.34	Suspended	
	R_6	0.66	**0**	0.00	0.00	0.00	
	R_7	0.51	**0**	0.00	0.00	0.00	
	R_8	0.14	1	0.00	0.00	0.00	

5.4 Summary

The developments of the proposed specialty-based probabilistic framework that are addressed in Chapter 4 are validated and the performance of the system is analyzed through an extensive series of MATLAB simulation experiments in this chapter. Performance is evaluated based on two definitions of the agents' specializations encoding: the binary and modulated. The simulator also implements the developments of the proposed task recognition stage, introduced in section 4.3, to provide an input to the proposed task allocation framework, as well as the agents' coordination scheme. The latter considers the addressed agents' availability states and attendance levels. In addition, the behaviour of the extended framework, introduced in section 4.4.5, is validated to tackle tasks that involve simultaneous detection of multiple classes of target objects. The human operator is also given an interface to refine the task assignment process, providing situational awareness, and the simulator is implemented with the objective to keep the human's cognitive load low.

The results of simulation experiments show that the proposed framework provides an efficient task allocation operation to coordinate the specialized individual agents with corresponding tasks based on the match between the task's specific constraints and the agents' specialized capabilities. Comparing the binary and modulated specialization encoding strategies, experiments demonstrate that the proposed task allocator generates a higher task allocation operational efficiency with modulated encoding of the agents' specializations than the binary encoding.

The experiments also demonstrate that consideration of the agents' availability results in a reliable framework with a high-level reasoning capability to adjust the task allocation process and

select amongst the available qualified agents to pursue the mission goals. The agents' attendance further expands the adaptability of the proposed framework to lead specialized swarms over wider workspaces. The simulation results show that agents' attendance expands the capacity of the individuals that are involved in the task allocation operation, which gives opportunities for additional agents to join the task allocation operation while providing more time efficient and secure responses.

To conclude, the experiments conducted in simulation in this chapter demonstrate that the proposed probabilistic task allocation approach is successful and efficient at properly assigning specialized agents to corresponding tasks that impose constraints on completion requirements while guaranteeing a minimum safety level. The proposed framework presents a high level of reliability to pursue the goals of the specialty-based task allocation missions in a broad diversity of scenarios. As such, an adaptability is introduced to tackle different task allocation applications with a variable number of involved tasks and specialized robotic agents, and dealing with small, medium or wide workspaces.

Chapter 6 will extend the evaluation of the proposed framework on real-life applications and perform experimental validation on real robots.

Chapter 6 Applications and Experiments

6.1 Introduction

The specialty-based task allocation approach is described in the previous chapters. In this chapter, the proposed approach is examined in terms of its potential applications. First, the performance of the framework is investigated in the context of an indoor search-and-rescue (SAR) application. In addition, to apply the proposed framework to real world's physical systems, experiments are also conducted on a team of real mobile robotic agents.

The confidence level on detected objects in Chapter 5 was simulated, while here it is estimated. Realistic test cases are introduced in this chapter for SAR, and in these cases target object recognition confidence is estimated with CNN on real images. Moreover, in physical experiments the confidence level on detected objects is also estimated using a color camera on color-coded features. Therefore, the presented examples in this chapter are more representative of reality.

6.2 Test Cases Related to Search-and-Rescue Application

In this section, an indoor search-and-rescue (SAR) application is considered. Initially, the proposed probabilistic task allocation approach leverages the output of a target object detection and recognition stage operating on real images, as detailed in [131]. The object recognition stage, which involves deep learning convolutional neural networks (CNN) methodologies to detect target objects from vision sensors mounted on mobile robotic agents, is integrated with the proposed task allocator to support a search-and-rescue task allocation for a group of cooperating robots, each with different functional capabilities. The target object recognition stage was developed by a

colleague and is not considered part of this thesis beyond its integration as the input to the task allocator. Its structure is summarized here to properly explain how it is integrated with the proposed task allocation framework. The deep learning-based object recognition stage supports the detection and recognition over five predefined classes of target objects in a typical indoor SAR scenario. The predefined classes include doors to be opened, stairs to be climbed, a person to be rescued, fire to be extinguished, and posted signs or floor maps to be read and interpreted to support robots' navigation. In our experiments, the class "fire" is depicted in image samples by a computer monitor for obvious safety reasons. The inference results obtained from the target object detection stage provide information about the class category and a corresponding confidence score for each detected object. This serves as the input for the multi-robot task allocation scheme. The probabilistic task allocation mechanism proposed in Chapter 4 works as a matching scheme for assigning the qualified individual agents among a SAR robotic team to corresponding tasks. The most qualified responders are assigned to the detected tasks based on the fitting probabilistic score estimated in the task-agent specialty matching process, given the available agents' specializations and the requirements imposed by the detected SAR targets.

6.2.1 Experimental Setup

The target objects' recognition scheme is used in this application to serve as a task recognition stage for the proposed framework, as shown in Figure 4.2. It collects real measurements on the classes observed on target objects, which help recognize them as part of one of the five classes considered. The experimental validation attempts to demonstrate the reliability of the integrated solutions for realistic SAR applications.

A. Detection of Target Objects

A pre-trained target object detection stage is integrated with the proposed framework. It was fed with a series of real images collected on the University of Ottawa campus which depict instances of objects from each of the five classes defined in section 6.2. Table 6.1 summarizes the object recognition confidence level experimentally estimated from visual features on each class and the corresponding specialized characteristics or functionalities of virtual robotic agents considered for experimentation. These characteristics are related to specific embedded sensors, actuators, or physical properties that target five classes of action to be performed for SAR tasks (open doors, climb stairs, assist people, extinguish fire, read signs).

In order to support integration with the task allocation stage, the output of the object detection stage is formulated as follows:

$$\widehat{\boldsymbol{P}}_T = \left[P_{C_1}, P_{C_2}, P_{C_3}, P_{C_4}, P_{C_5}\right]^t \tag{6.1}$$

where C_k: k from 1 to 5 denote the classes of door, stairs, person, fire (e.g. computer monitor), and signs respectively. P_{C1}~P_{C5} is the recognition confidence level on a target object associated with each class. This recognition confidence level on a target object, which is produced using modern convolutional neural network (CNN) architectures, serves as a task recognition stage that is to be used as an entry to the proposed task allocation process, as indicated in Figure 4.2. The practical implementation details of this recognition stage are introduced in [131] and are beyond the scope of this study. In this SAR test case, the object recognition stage performs target object recognition from vision sensors mounted on-board unmanned vehicles. Table 6.1 shows selected samples of the five classes (out of 140 testing cases considered in this experiment) for object detection with respective confidence level estimates for each class of target objects, as well as the related robot's

specialized functionalities which are further detailed in upcoming sections. These examples illustrate the visual features on the detected target objects and the expected corresponding robot's specialized capabilities. The task-agent fitting probabilities are computed with the framework proposed in Chapter 4 based on the confidence level estimated by the object recognition stage for each class of detected objects. Specialized robotic agents are expected to respond and be allocated to specific SAR-related tasks when a given agent's specialty characteristics offer sufficient compatibility with the task's requirements, as measured visually on a detected target object. As such, a given agent can be qualified to fulfill different tasks with different fitting levels.

Table 6.1 Object recognition from visual features with confidence level, and corresponding specialized functionalities of robotic agents

Input image with segmentation mask	*Detected target object class*	*Object recognition stage output with confidence level per class*	*Corresponding robot specialized functionality*
	Door (C_1)	$\hat{P}_T = [0.995\ \ 0\ \ 0\ \ 0\ \ 0]$	Open doors
	Stairs (C_2)	$\hat{P}_T$=[0 0.963 0 0 0]	Climb stairs
	Person (C_3)	$\hat{P}_T = [0\ \ 0\ \ 0.958\ \ 0\ \ 0]$	Assist people
	Fire (computer monitor (C_4)	$\hat{P}_T = [0\ \ 0\ \ 0\ \ 0.954\ \ 0]$	Extinguish fire

	Sign (C_5)	$\hat{P}_T = [0\ \ 0\ \ 0\ \ 0\ \ 0.983]$	Read signs

B. Specialization Definition and Encoding

In this study, SAR scenarios are considered and exemplified by the five classes of target objects to be potentially detected in the environment of the robotic agents. The definition of agent specialization is encoded as a binary vector as defined in section 4.4.1A, associated with each robot, that describes the presence or absence of specific hardware or physical construction essential to completing a given task (e.g. opening a door). A swarm of robots $\{R_i, i = 1,2, \dots, a\}$ consists of $a = 7$ specialized individual agents, R_i, and provides $T = 5$ specialized roles or capabilities, defined in Table 6.1, that are encoded in each agent's binary specialty vector, $\boldsymbol{S}_i$: $\{s_k, k = 1,2, \dots,5\}$, where $\boldsymbol{S}_i \in \mathcal{R}^{1\times5}$. The encoding for the individual robots' specialization characteristics is summarized in Table 6.2 for a group of seven agents considered for this SAR scenario test case.

Table 6.2 Formulation of seven robotic agents' specialization for SAR tasks with five classes of target objects

Agent ID#	Agent's specialized functionalities 1 = Possesses functionality; 0 = Does not possess functionality					
	Specialty vector	*Open doors*	*Climb stairs*	*Assist people*	*Extinguish fire*	*Read signs*
R_1	$\boldsymbol{S_1}$	1	1	0	0	0
R_2	$\boldsymbol{S_2}$	0	1	1	0	0
R_3	$\boldsymbol{S_3}$	0	1	0	1	0
R_4	$\boldsymbol{S_4}$	0	1	0	0	1
R_5	$\boldsymbol{S_5}$	0	0	1	0	0
R_6	$\boldsymbol{S_6}$	0	0	0	1	0
R_7	$\boldsymbol{S_7}$	0	0	0	0	1

6.2.2 Experimental Results

6.2.C1 Search-and-Rescue with general probabilistic framework

In the first series of experimental test cases, the general probabilistic task allocation framework, detailed in section 4.4, is applied on a number of test images. First, 140 images samples were manually acquired with a camera while patrolling different sectors of the UOttawa SITE building. Next, these images were processed to retrieve every instance of the five classes of objects considered. A typical test scenario is demonstrated as the robotic team navigates the ground floor of a building.

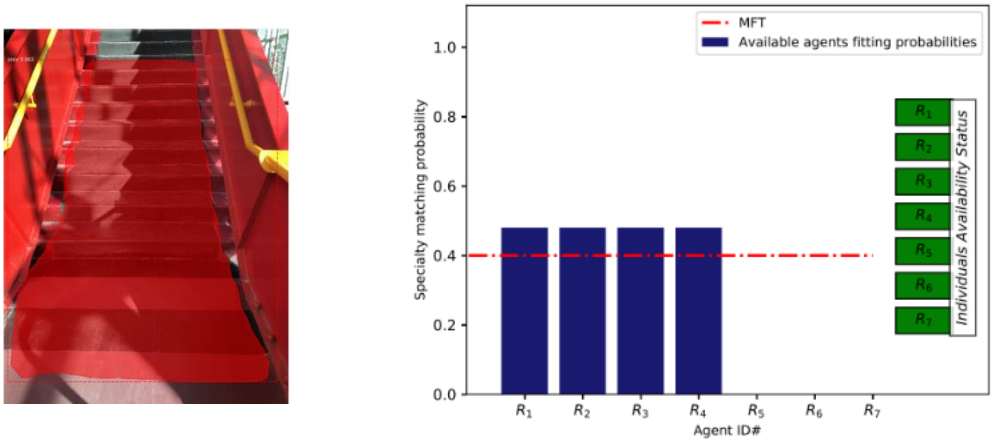
As an example, the task recognition stage, shown in Figure 4.2, initially detects stairs, as shown in Figure 6.1a. The class of object and associated confidence level from the object recognition stage are leveraged by the proposed probabilistic task allocation scheme, Eq. (4.18), to compute the seven individual robots' probabilistic fitting scores with the purpose of assigning the qualified agents to the detected task. The agents' attendance, $\boldsymbol{\vartheta}_{Att}$, is deactivated with $\mathfrak{p} = 1$ in Eq. (4.16) for this indoor application given that the workspace is bounded and all agents are available within a limited area, whereas the overall task allocation probability is computed by placing full weight on the agents' specialty based qualification, $\boldsymbol{Q}$. The extended developments in section 4.4.5 are not considered whereas this example tests the original framework, Eqs. (4.18) and (4.20).

Figure 6.1b shows that robots $R_1, R_2, R_3, \& R_4$ are sufficiently qualified to proceed and climb the recognized stairs, as their specialty matching probability satisfies the MFT set at 0.4 in this case. However, the other available robots $R_5, R_6, \& R_7$ are not qualified based on their functional characteristics defined in Table 6.2 that do not make them suitable to climb stairs. The availability status, $\boldsymbol{\vartheta}_{\boldsymbol{Av}}$, of the robots at the moment when the target (i.e. stairs) is detected is also shown in

Figure 6.1b, with green color-coding indicating an "available state". The respective target object detection confidence levels and the robots' probabilistic fitting scores are presented in Table 6.3. Mathematical details of computing the fitting probabilities for this example, C.1, are reported in Appendix B.

Table 6.3 Team members' fitting probabilities to climb stairs during SAR task in an indoor workspace, with $\wp = 1$

Target recognition confidence level	*Robot ID#*	*Target-agent specialty matching probability* ($\boldsymbol{Q}$) *Binary encoding*	*Availability 1:Available 0:Withdrawn* ($\boldsymbol{\vartheta_{Av}}$)	*Available agents' attendance level* ($\boldsymbol{\vartheta_{Att}}$)	*Available qualified agents' fitting scores* ($\boldsymbol{\Psi_{MFT}}$)	*MFT* (η)
Door: 0.00 Stairs: 0.96 Person: 0.00 Fire: 0.00 Sign: 0.00	R_1	0.48	1	0	0.48	0.4
	R_2	0.48	1	0	0.48	
	R_3	0.48	1	0	0.48	
	R_4	0.48	1	0	0.48	
	R_5	0.00	1	0	0.00	
	R_6	0.00	1	0	0.00	
	R_7	0.00	1	0	0.00	



(a) (b)

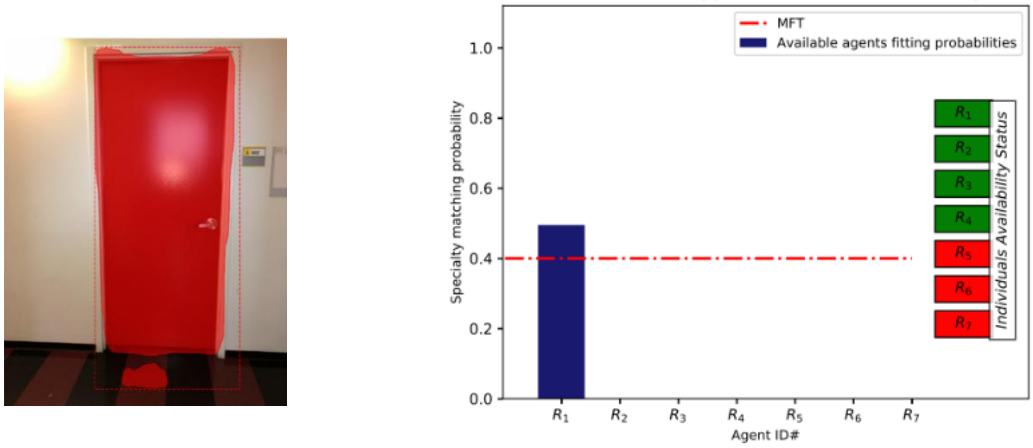
Figure 6.1 a) Detected target object (stairs); and **b)** Specialized agents' fitting probabilities and availability status (available = green)

Pursuing their displacement in the building, the team members that were able to climb the stairs ($R_1, R_2, R_3, \& R_4$) leave the area and begin navigating the hallway. A door is eventually detected, as shown in Figure 6.2a. The task allocation system then obtains the confidence level on the

detected target object and computes the corresponding individual agents' specialty fitting probabilities. The most competent agent according to Eqs. (4.18) and (4.20) is assigned to the current task of opening the detected door. The fitting probabilities of the team members and their availability status are shown in Figure 6.2. Agents $R_1, R_2, R_3,$ & R_4 are still available, while agents $R_5, R_6,$ & R_7 are not (i.e. withdrawn since they previously failed to climb the stairs with other team members). Given that only agent R_1 is qualified to open doors (Table 6.2), three agents achieve a null fitting score with the current task, while the specialty fitting probability of R_1 with the detected target equals 0.49 (Table 6.4). This level also satisfies the MFT (here set to 0.4) while robot R_1 is available. Therefore, the system automatically assigns agent R_1 to open the door, while $R_2, R_3,$ & R_4 remain behind in support. Once R_1 opens the door, the team then accesses the inside workspace and the vision sensor is used to conduct a new survey of the environment to detect additional target objects. A fire (computer-monitor) and a human victim are simultaneously detected, as shown in Figure 6.3a. The new images are processed through the target object recognition stage, and the classes and related confidence levels are leveraged by the task allocation scheme, Eqs. (4.18) and (4.20), to determine the specialty fitting probabilities for the agents that are still available, as shown in Figure 6.3b and detailed in Table 6.5. As a result, the most competent available agents, R_2 and R_3, are assigned to the latest detected tasks, that is R_2 to rescue the person and R_3 to extinguish the fire, respectively. This simulated test scenario demonstrates that, while guaranteeing a minimum fitting threshold (MFT) to ensure that decisions are taken only when a sufficient confidence level is achieved on target object recognition, task allocation is performed on those detected targets, whenever possible, with the most competent and available agent or agents that are selected as the qualified responders to perform the detected tasks.

Table 6.4 Team members' fitting probabilities to open a detected door during SAR task in an indoor workspace, with $\mathfrak{p} = 1$. * is the most qualified agent

Target object recognition confidence level	*Agent ID#*	*Target-agent specialty matching probability* ($\boldsymbol{Q}$) *Binary encoding*	*Availability 1:Available 0:Withdrawn* ($\boldsymbol{\vartheta_{Av}}$)	*Available agents' attendance level* ($\boldsymbol{\vartheta_{Att}}$)	*Available agents' fitting probabilities* ($\boldsymbol{\Psi_{MFT}}$)	*MFT* (η)
Door: 0.98 Stairs: 0.00 Person: 0.00 Fire: 0.00 Sign: 0.00	R_1	0.49	1	0.0	0.49*	0.4
	R_2	0.0	1	0.0	0.0	
	R_3	0.0	1	0.0	0.0	
	R_4	0.0	1	0.0	0.0	
	R_5	0.0	0	0.0	0.0	
	R_6	0.0	0	0.0	0.0	
	R_7	0.0	0	0.0	0.0	

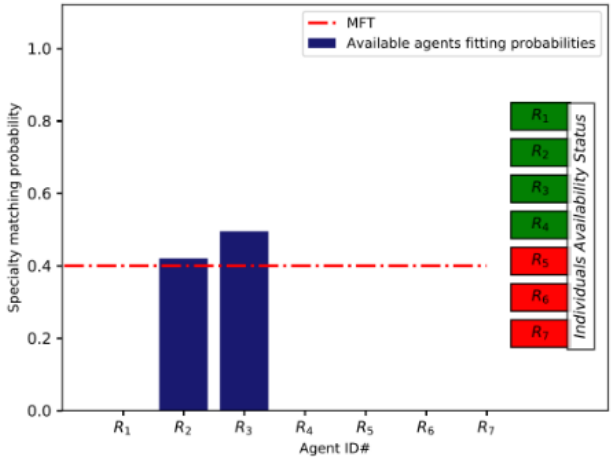


(a) (b)

Figure 6.2 a) Detected target (door); and **b)** Specialized agents' fitting probabilities and availability status (available = green; withdrawn = red).

Table 6.5 Team members' fitting probabilities to respond to two detected tasks (person and fire) during SAR in an indoor workspace. (Two agents are assigned sequentially to each detected target.)
* is the most qualified agent/target

Target object recognition confidence level	*Agent ID#*	*Target-agent specialty matching probability* ($\boldsymbol{Q}$) *Binary encoding*	*Availability 1:Available 0:Withdrawn* ($\boldsymbol{\vartheta_{Av}}$)	*Available agents' attendance level* ($\boldsymbol{\vartheta_{Att}}$)	*Available agents' fitting probabilities* ($\boldsymbol{\Psi_{MFT}}$)	*MFT* (η)
Door: 0.00 Stairs: 0.00 Person: 0.84 Fire: 0.99 Sign: 0.00	R_1	0.00	1	0.0	0.00	0.4
	R_2	0.42	1	0.0	0.42*	
	R_3	0.49	1	0.0	0.49*	
	R_4	0.0	1	0.0	0.00	
	R_5	0.0	0	0.0	0.00	
	R_6	0.0	0	0.0	0.00	
	R_7	0.0	0	0.0	0.00	



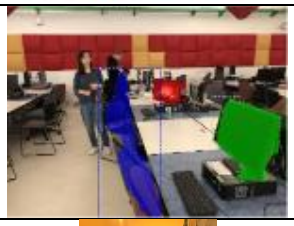
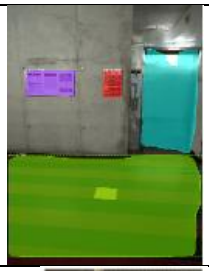
(a) (b)

Figure 6.3 a) Detected target objects: (left) person to be rescued, and (right) fire to be extinguished; and **b)** Specialized agents' fitting probabilities and availability status (available = green; withdrawn = red)

More results of the task allocation process are added in Table 6.6 which covers cases where target object recognition offers variable confidence levels. These examples support the demonstration of the performance of the proposed framework in more realistic operational conditions, where vision-based target recognition does not perform perfectly.

Table 6.6 Sample images exhibiting target objects recognition with lower confidence levels among the five classes considered and automatically assigned robotic agents for detected target(s) by the proposed approach. MFT = 0.4.

No.	Input Image with Segmented Detected Target(s)	Recognized Target Object Confidence Level		Assigned Agents	
				Agent	Fitting Probability
1		Door	0.991	$\boldsymbol{R_1}$	0.49
		Stairs	0.000		
		Person	0.000		
		Fire	0.000		
		Sign	0.000		
2		Door	0.000		
		Stairs	0.965	$\boldsymbol{R_1, R_2, R_3, R_4}$	0.48
		Person	0.000		
		Fire	0.000		
		Sign	0.000		
3		Door	0.000		
		Stairs	0.000		
		Person	0.858	$\boldsymbol{R_5}$	0.86
		Fire	0.901	$\boldsymbol{R_6}$	0.90
		Sign	0.000		

4		Door	0.000		
		Stairs	0.000		
		Person	0.987	R_5	0.99
		Fire	0.000		
		Sign	0.788	R_7	0.79
5		Door	0.000		
		Stairs	0.000		
		Person	0.985	R_5	0.98
		Fire	0.968; 0.628	R_6	0.97
				R_3 (not assigned)	0.31<MFT
		Sign	0.000		
6		Door	0.847	R_1	0.42
		Stairs	0.978	$R_2, R_3, R_4,$	0.49
		Person	0.000		
		Fire	0.000		
		Sign	0.000		
7		Door	0.000		
		Stairs	0.000		
		Person	0.000		
		Fire	0.000		
		Sign	0.984; 0.940; 0.657	R_7	0.98
				R_4	0.47
				No specialized agent is available for allocating to the third target	
8		Door	0.654	R_1 (not assigned)	0.33<MFT
		Stairs	0.970	R_1, R_2, R_3, R_4	0.49
		Person	0.000		
		Fire	0.000		
		Sign	0.995; 0.991	R_7	1
				No specialized agent is available for allocating to the second target	
9		Door	0.993	R_1	0.49
		Stairs	0.000		
		Person	0.000		
		Fire	0.000		
		Sign	0.733	R_7	0.73
10		Door	0.000		
		Stairs	0.995	R_1, R_2, R_3, R_4	0.5
		Person	0.000		
		Fire	0.000		
		Sign	0.000		
11		Door	0.000		
		Stairs	0.000		
		Person	0.000		
		Fire	0.987	R_6	0.99
		Sign	0.843; 0.624	R_7	0.84
				R_4 (not assigned)	0.31<MFT

12		Door	0.000		
		Stairs	0.000		
		Person	0.000		
		Fire	0.000		
		Sign	0.928, 0.824	$\boldsymbol{R_7}$	0.93
				$\boldsymbol{R_4}$	0.41
13		Door	0.000		
		Stairs	0.977	$\boldsymbol{R_1, R_2, R_3, R_4}$	0.49
		Person	0.000		
		Fire	0.000		
		Sign	0.000		
14		Door	0.000		
		Stairs	0.000		
		Person	0.630	$\boldsymbol{R_5}$	0.63
		Fire	0.913; 0.879	$\boldsymbol{R_6}$	0.91
				$\boldsymbol{R_3}$	0.44
		Sign	0.963	$\boldsymbol{R_7}$	0.96
15		Door	0.000	------	
		Stairs	0.000		
		Person	0.000		
		Fire	0.000		
		Sign	0.000		

6.2.C2 Search-and-Rescue task allocation with multiple classes tasks

This test case demonstrates the behaviour of the extended approach that considers complex tasks; the latter are characterized based on the detection of multiple classes and the qualified agents are selected based on a prioritization process between the primary and secondary detected classes, as detailed in section 4.4.5. A practical example of the importance of such prioritization in the specialty of robotic agents in the context of indoor search-and-rescue application (SAR) is a task of multiple classes {i.e., stairs, sign}. The agent R_4, listed in Table 6.2, is qualified to climb stairs and read signs. In this SAR context, the capability to climb stairs could be defined as a primary capability, while reading signs could be considered a secondary capability. Such an initialization would be justified in the context of the task allocation when all of the classes that characterize the given task are detected simultaneously. Then, the agent would only make for a suitable allocation to the given task provided that it can first overcome the stairs; otherwise, it would be of no use. As

a result, R_7 is prevented from taking part in the intervention, and R_4 is assigned amongst four agents that are all qualified to climb the stairs: $R_1, R_2, R_3,$ and R_4.

In this case, a sample output of the target object detection stage, Figure 6.4a, shows that stairs and a posted sign are detected within the same workspace. The extended approach, introduced in section 4.4.5, processes this detection as one task of two classes. This case is tested when all team members are available. Based on the individuals' specialty encoding, as defined in Table 6.2, the task allocation scheme, as defined by Eq. (4.18), evaluates that agents $R_1, R_2, R_3,\ R_4,$ and R_7 all achieve fitting probabilities that exceed the MFT (here set at 0.4), as detailed in Figure 6.4b. On the other hand, if the task allocation operation is to allocate the most qualified agent, then R_7 will be selected based on the process of Eq. (4.20). However, while agent R_7 is specialized to read the sign, it does not possess the capability to climb the stairs. Therefore, agent R_7 would fail to reach the detected sign since it is located beyond the stairs. For complex tasks, the system must deal with simultaneous detected objects (e.g. stairs and sign) and prevent the allocation of less qualified agents, such as agent R_7 in this situation, which would fail at their mission. In this case, the system must respond with a higher level of reasoning without increasing the human supervisor's cognitive load. A solution to such a scenario requires the task allocation scheme to consider another criterion to ensure that higher task allocation priority is given to specific agents among the available specialized individuals, while suspending others. To address this problem and increase the system's reasoning capability, the task allocation priority is addressed based on a recoding of the agents' primary capabilities, as indicated in Table 6.7, by leveraging a received input from the system's visual sensors and using the primary encoder defined in Eq. (4.21). This automated process is detailed in section 4.4.5 and Figure 4.3. Figure 6.4c shows how the system assigns

agents R_1, R_2, R_3, and R_4 to respond to the detected targets using Eq. 4.25. In this case, agent R_7 is suspended because it does not possess the primary capability, which is climbing stairs, as defined in Table 6.2. Detailed results of this case are presented in Table 6.8.

Table 6.7 Categorization of targets' constraints associated with corresponding primary encoder entities

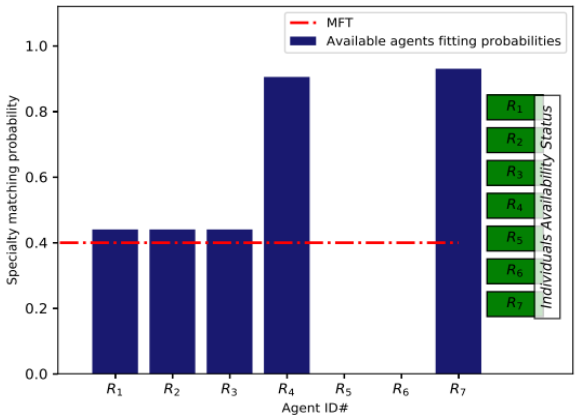
Target object	Class	$\boldsymbol{B}_\rho$ (competency priority category)
Door	C_1	$\bar{q}_{\rho_1}$ (secondary)
Stairs	C_2	$\bar{\bar{q}}_{\rho_2}$ (primary)
Person	C_3	$\bar{q}_{\rho_3}$ (secondary)
Fire	C_4	$\bar{q}_{\rho_4}$ (secondary)
Sign	C_5	$\bar{q}_{\rho_5}$ (secondary)

These results demonstrate that extending the proposed task allocator by considering the allocation priorities (discussed in section 4.4.5) for the complex tasks that are characterized based on the detection of multiple classes resulted in a higher reasoning capability and a more selective task allocation scheme to tackle complicated intervention scenarios which may have resulted in mission failure if these priorities were not considered. Task allocation prioritization ensures a safe task allocation operation in the scenarios of the complex tasks which allow only the robots that are fully equipped with the required functionalities to tackle more sensitive situations. In addition, it prevents other agents from engaging in operations that may otherwise result in a failed mission.

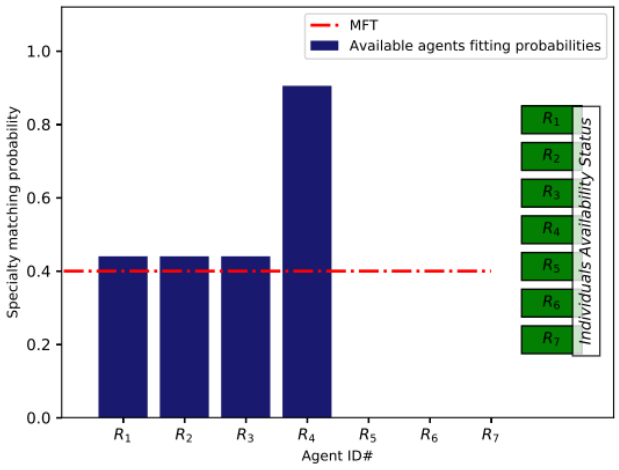
Table 6.8 Team members' fitting probabilities to respond to two detected tasks (stairs and sign) during SAR in an indoor workspace with/without considering the task allocation prioritization.
* is the most qualified agent

Target object recognition confidence level	*Agent ID#*	*Target-agent specialty matching probability* ($\boldsymbol{Q}$) *Binary encoding*	*Availability 1:Available 0:Withdrawn* ($\boldsymbol{\vartheta}_{Av}$)	*Available agents' attendance level* ($\boldsymbol{\vartheta}_{Att}$)	*Available agents' fitting probabilities* ($\boldsymbol{\Psi}_{\boldsymbol{MFT}}$)	*Available agents' fitting probabilities* ($\boldsymbol{h}_\rho$)	*MFT* (η)
Door: 0.00 Stairs: 0.88 Person: 0.00 Fire: 0.00 Sign: 0.93	R_1	0.44	1	0.0	0.44	0.44	0.4
	R_2	0.44	1	0.0	0.44	0.44	
	R_3	0.44	1	0.0	0.44	0.44	
	R_4	0.90	1	0.0	0.90	0.90*	
	R_5	0.0	1	0.0	0.0	0.00	
	R_6	0.0	1	0.0	0.0	0.00	
	R_7	0.93	1	0.0	0.93*	0.00	

(a)



(b)



(c)

Figure 6.4 a) Detected target's objects: (front) stairs, and (back right) sign; **b)** Specialized agents' fitting probabilities and availability status; and **c)** Specialty fitting probabilities of the agents that have active assignment priority w.r.t. the detected task, while considering the agents' ability to climb stairs is a primary capability

6.3 Implementation for Experimental Validation on Real Robots

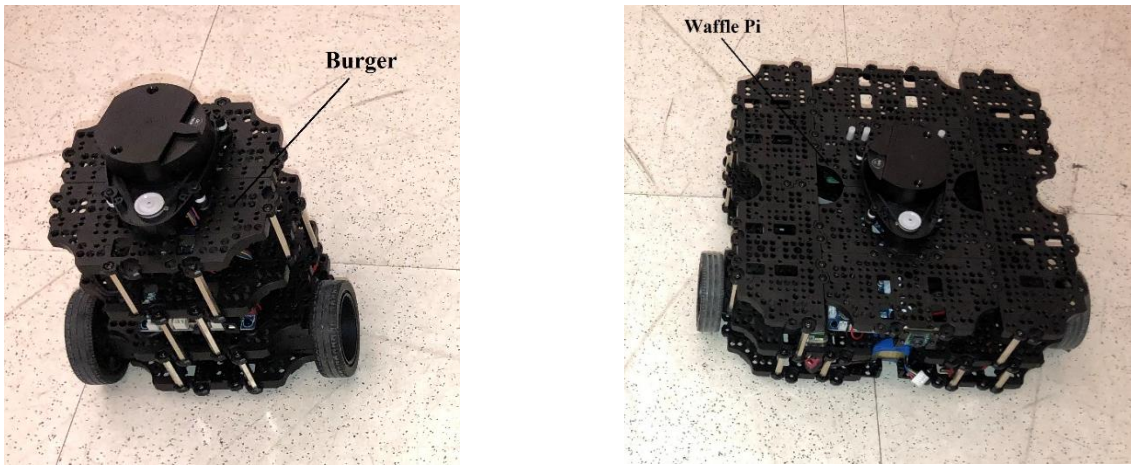
To know whether the test set derived from the simulation trials can be transferred to real-world systems, a series of real-world experiments are conducted on physical robotic platforms. These robots represent the real behaviour of the robotic system under testing. In addition, realistic data is fed from a real sensor to the target object recognition stage, which represents the input of the proposed framework. The results of what is actually measured from the real-world experiments

validate the viability of the proposed framework for its desired applications. In this section, the physical implementation used to conduct the validation experiments is described.

6.3.1 System Description

For these experiments, a multi-agent robotic system is implemented using two different physical platforms: one TurtleBot3 Burger and one TurtleBot3 Waffle Pi [135], as shown in Figure 6.5. With the purpose of performing this test case in consideration of the robotic agents' availability, the proposed framework is tested in this section by involving a third virtual robotic agent, called Milkshake, that is assumed to be unavailable or "withdrawn." Each one of the two real robotic agents has a differential drive base formed with a pair of Dynamixel servo motors. The Burger model has two wheels of $66\,mm$ in diameter and $160\,mm$ of (rear axle) distance between them, while the wheels of the Waffle Pi are the same diameter but with a rear axle length of $287\,mm$. The two robotic platforms are equipped with an OpenCR controller and a Raspberry Pi microcomputer, and they carry an odometer sensor. The developmental environment for OpenCR is Arduino IDE, and the Raspberry Pi is operated by the Raspbian operating system.

The area of the operational workspace is approximately $4.80\,m \times 2.11\,m$, as illustrated in Figure 6.6. The experimental system uses a central processing unit (CPU) on a laptop configured [136] with the Linux (Ubuntu) operation system which runs a ROS navigation stack to provide communication between the system components via a shared Wi-Fi network podcast by a smart device. The required communications, robots' localization, and path-planning capabilities are mastered by ROS [137].



(a) (b)

Figure 6.5 TurtleBot3 mobile robots including (a) a TurtleBot3 Burger, and (b) a TurtleBot3 Waffle Pi

Figure 6.6 Operational workspace to conduct experiments with real robots

The target considered here corresponds to a colored cylindrical shape positioned at known coordinates in the 2D workspace. The experiments are conducted for different scenarios considering different targets that consist of a combination of two vertically neighbouring color features with the following distinct color combinations: red-green, red-blue, and green-blue. Each color combination represents a target type associated with one robotic agent that is to be the most qualified agent to deal with it, whereas the other robotic agents can present different levels of qualifications to respond to the same target. Many test cases are conducted considering different recognition confidence levels of the target's features. Based on the nature of the target used in these experiments, which is a combination of two color features, tasks of multiple classes are considered. The corresponding specialization vectors, $\boldsymbol{S_i}$, of the robots are formulated in the original binary form and defined in Table 6.9.

Table 6.9 Formulation of robotic agents' specialization for experiments with real robots

Agent name	Agent's specialized functionalities 1 = Possesses functionality; 0 = Does not possess functionality			
	Specialty vector	*Red feature*	*Blue feature*	*Green feature*
Burger	$\boldsymbol{S_1}$	1	0	1
Waffle Pi	$\boldsymbol{S_2}$	1	1	0
Milkshake	$\boldsymbol{S_3}$	0	1	1

6.3.2 Color Camera

Adding cameras to robotic systems could help the robots to better recognize their surroundings. In these experiments, a color sensor is used to detect the color features of the target object. This sensor is equipped with a See3CAM_130 USB 3.1 camera, from E-con systems [138], which has an accurate color response and resolution. Table 6.10 shows the SeeCAM_130 USB 3.1 camera's specifications:

Table 6.10 See3CAM_130 USB 3.1 camera specifications

Size	(L×W): $80mm \times 15mm$
Image resolution	4224×3156
Horizontal field of view	60.2^{o}

The camera offers an auto white balance and autofocus which provide consistent detection results. To ensure usable response in these experiments, the camera operating resolution is set at 1920×1080 with a frame rate of 30 Hz. The recognition stage of the proposed framework develops a solution to leverage the detection results of color features in hue range, based on Hue-Saturation-Value (HSV) color space representation, which can be detected by the color camera. If a saturated color-coded target is properly lighted, the color features can be detected individually by their hues. The color hues that are acquired from the camera are in the RGB space, then OpenCV is used to convert it to HSV space, whereas OpenCV uses the range [0, 179]. Figure 6.7 depicts a color bar of fully lighted and saturated colors plotted by hue in the range of [0, 179]. Based on the noise consideration in the camera, the target is coded with two colors that are distinctive from each other with apparent hue range. In these experiments, the colors red, green, and blue are adopted to encode and detect the target [139].

Figure 6.7 A Hue-Saturation-Value color bar

As a result of the presence of noise in the camera model it cannot autofocus at the exact value of the color hue and usually it returns uncertain detection results in a hue range. Based on the observation from the experiments each color can be detected within a range of ± 5 around its hue value. To leverage these uncertain detection results provided by the color camera, the following formula is derived to compute the confidence level on the color features detection:

$$\text{Confidence level of the detected color feature} = \frac{10-[actual\ hue-|detected\ hue|]}{10} \quad (6.2)$$

The actual hue values are indicated in Figure 6.7 and the color camera provides the detected values. In order to support integration with the task allocation stage, the output of the object detection stage is formulated as follows:

$$\widehat{\boldsymbol{P}}_T = \left[P_{C_1}, P_{C_2}, P_{C_3}\right]^T \quad (6.3)$$

where $C_k: k = 1:3$ denote the classes of the used color features that are red, blue, and green respectively. $P_{C1} \sim P_{C3}$ is the recognition confidence level on a target object feature associated with each class.

6.3.3 Experimental Setup

A central decision-making unit, corresponding to the proposed automatic task selection unit (ATSU), is implemented on the laptop CPU as a high-level master node which is responsible for facilitating interaction between other nodes and the decision-making process for agents' assignments to the detected tasks. The ATSU runs the proposed probabilistic task allocation approach (discussed in Chapter 4) and evaluates the probabilistic fitting levels of the individual agents in accordance with detected target objects in the environment. The proposed approach is implemented in this central unit considering the agents' binary specialization encoding that is introduced in section 4.4.1.A. The agents' availability is also considered along with their specialty matching probability. The agents' attendance is deactivated in these experiments, given that the experiments are performed in an indoor lab environment over a limited operational workspace area of approximately $4.80 \times 2.11\ m$ as indicated in Figure 6.6. In addition, the deactivation of the agents attendance places a full weight of computing the overall task allocation probability on the agents' specialty-based qualification, $\boldsymbol{Q}$, which dedicates the experiments to test the system

response to the task allocation operation based on the core component of the proposed task allocation framework that is the match between the agents' specialized capabilities and the task's specific constraints. Next, the results are turned into individual agents' task allocation probabilities and assignment commands.

The ATSU communicates with the two robotic agents' low-level local sensors and actuators. The Raspberry Pi is a microcomputer board installed on each robotic platform. The Raspberry Pi has a wired connection with the OpenCR and a wireless connection with the central controller. The Raspberry Pi is responsible for exchanging information and commands between the robotic agent's local controller, the OpenCR, and the central controller unit, the ATSU. The OpenCR is an agent's local microcontroller board responsible for executing the commands sent by the central controller via the Raspberry Pi on Dynamixel servo motors. The OpenCR also collects the pose information from the odometer sensor. The battery of the robot is directly connected to this board as it serves as a bridge between the power source and the motors. The OpenCR microcontroller is programmed in Arduino IDE.

The color camera is used to detect color-coded target objects. Then, the confidence levels of target object detection, Eq. (6.2), are collected and fed offline to the central control unit. The latter processes the collected data and sends control commands back to enable the dynamic controllers of the individual robotic agents. Each agent controller receives the task allocation commands from the central control unit, whereas the central control unit (ATSU) determines the qualified agents and allocates each task accordingly. The dynamic controllers of the individual robots are designed to receive the task allocation control commands from the central unit (ATSU), located in the central CPU, and respond locally to control the angular and linear velocities of the robotic agents

for the purpose of leading the robotic agents towards the target task. These local controllers drive the operation of the automated mission of the robotic team over the three task allocation states that are discussed in section 3.5.2.C. Considering the dynamics of the agents at hand, the control design in the level of the agent's local controller is slightly different than the detailed control design that is introduced in section 3.3. In the real robots we used here, the robots' movements are controlled by adjusting the angular and linear velocity of the differential driver of the robotic agent, which consists of two DC motors, based on online position information collected by odometer sensors embedded on the physical robots. That is slightly different than the dynamics control of the ideal rigid body dynamics that we used for the simulations trials which control the movements of virtual robotics by controlling the movements of their position points along an ideally-generated path. In general, to achieve the concepts of the formation control design that have been introduced in section 3.3, the dynamic controllers of the real robotic agents are designed to mimic the same behaviour. While searching for a target, the individual robotic agents follow a predefined path and keep a specific distance between them to avoid collisions. Then, upon the task detection, the team switches to task state and selects the most qualified robotic agent to respond to the detected task. After that, the selected agent is assigned to move towards the position of the detected task. Finally, the assigned agent stops at the zone of influence in front of the target position with the purpose of excusing the current task. The practical description of this process is detailed in section 6.4.

6.3.4 Software Design

The central control unit and each robotic agent run pieces of software implemented using the ROS framework. The schematic diagram of the software configuration, depicted in Figure 6.8, details the interaction and communication links between the implemented nodes.

Each robot has a Kinetic Kame version of ROS installed on its Raspberry Pi. The robotic agents are connected to a CPU (Dell laptop) running Ubuntu 16.04 with ROS Kinetic Kame version installed on it as well. The entire system's components are connected through a local Wi-Fi network. The ROS network uses the laptop as a master node whereas both of the robotic agents' nodes are connected as clients. Communication between the central control unit and the robots is created over a local Wi-Fi hotspot that is established on a smartphone [137].

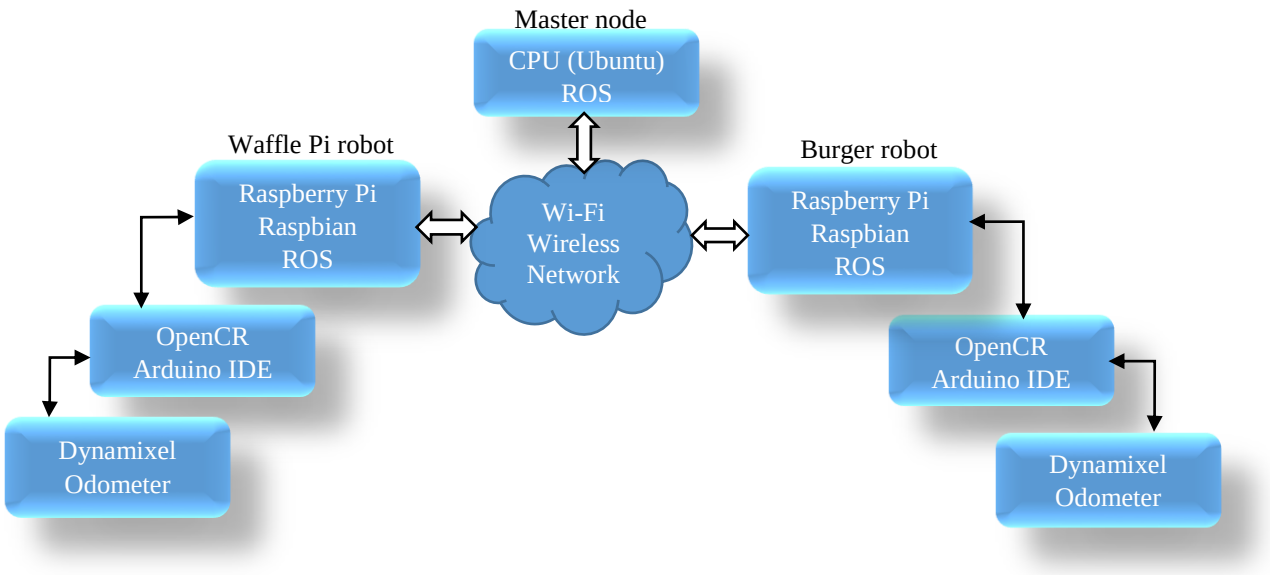


Figure 6.8 Software configuration for the system control network

6.3.5 ROS Network Description

ROS provides powerful conventions to create complex robotic software for robotic platforms. ROS network is composed of coupled components that are called nodes, topics, and messages. Figure 6.9 shows the architecture of the ROS network which has been designed and implemented to validate the proposed framework. The implementation of the ROS network was achieved in cooperation with an internship undergraduate student in SMART group. Each ROS node, outlined in red, is a ROS component that can perform a computation process. The ROS nodes in this case are written with Python using rospy, which is a ROS client library.

ROS topics, outlined in black, serve as message buses. These buses allow the nodes to send and receive messages routed through the ROS network via publishers and subscribers. In this network, the tf special topic is used to receive different information from other publisher nodes saved under two different names. In this case, tf topic is used to collect information from the odometers of the two robots.

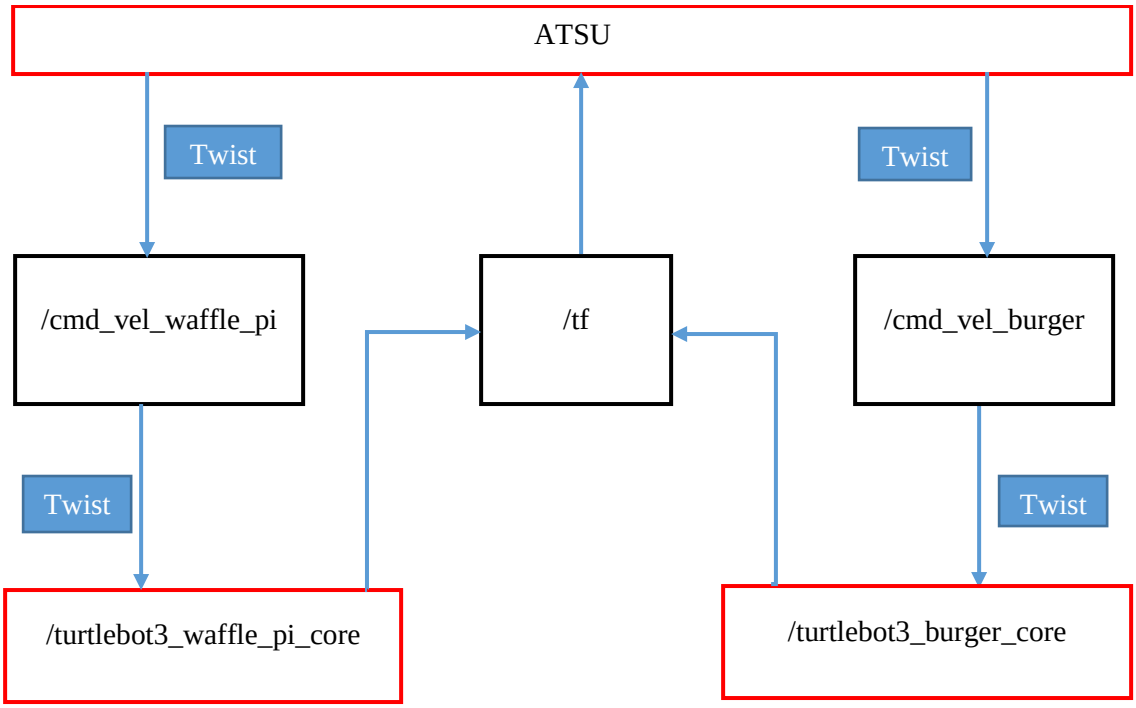


Figure 6.9 Architecture of the implemented ROS network

In this experimental implementation, the algorithms on the central node, that is the ATSU, are implemented by the author and programmed using Python. The robotic agents' nodes turtlebot3_burger_core and turtlebot3_waffle_pi_core are developed by a colleague in SMART group and programmed in Arduino IDE. The topics cmd_vel_waffle_pi and cmd_vel_burger are responsible for transferring the information on the angular and linear velocities of the robots from the ATSU to the robots using a twist type of message. Finally, ROS messages, filled in blue, are

simply a data structure. Standard primitive types of ROS messages are integer, floating point, Boolean, etc. ROS messages are supported as arrays of primitive types.

The software package which has been developed to handle the functions of this central unit is documented in a technical report [137].

6.4 Experimental Results with Real Robots

The physical experiments are conducted to illustrate the real robots' response to the matched tasks using the proposed probabilistic specialty-based task allocation approach. Considering a fully automated process, the robotic platforms are centrally commanded and locally controlled to follow a pre-defined path in the searching state, or to respond to and reach the detected targets in the execution state. Figure 6.10 shows that a predefined path (green-dotted arrow) is initialized between two predefined positions: the initial position of Waffle Pi robotic agent, and the border of the task zone (red-dotted line). The odometers of this robot feed the current position information. The latter is used to control the angular and linear velocities of this robot along this path until the robot arrives at the second point, that is the border point of the task zone. However, the second robotic agent, Burger robot, follows the Waffle Pi robot in parallel and keeps a preserved distance between them (purple-dotted double arrow) with the purpose of avoiding collision with the Waffle Pi. To achieve this condition the Burger robot copies the online linear and angular velocities of the Waffle Pi robot. The task allocation scheme is configured in this system such that the target has to be detected when the robots reach the task zone (red-dotted line). Therefore, due to the arrival of the two robots at the border of the task zone, the robots are stopped and the central control unit processes the stored confidence levels, Eq. (6.3), on the recognition of color features imaged by a camera which is placed beside the robots in the workspace. Task

allocation commands are sent to the qualified agent to respond to the detected task. After that, a new path is initialized between the current position of the selected agent and the position of the target (blue-dotted arrows). The selected agent is assigned and moves towards the target position. The movement of the selected robot in the task zone is independent from the other robotic agent. This robot navigates the new path towards the target position whereas its dynamical controller uses the current position data that are fed by its odometers and leads it towards the target position. However, the second robot stays behind the border point of the task zone. Due to the arrival of the assigned robot to the execution zone (red circle around the target), the robot stops in front of the target in preparation for executing the task.

In the first test case, shown in Figure 6.11, the multi-agent system consists of two available agents: a Burger and a Waffle Pi, and a virtual "withdrawn" agent, called Milkshake. For clarity, the available robotic agents are marked with a colored indicator on the top of each robot. The Burger agent, on the right, carries a red/green tag, which means that this agent is qualified to be assigned to target objects that have green and red features. Similarly, the Waffle Pi agent, on the left, is specialized to deal with target objects that have red and blue features. The corresponding specialization vectors, S_i, of the robots are defined in Table 6.9 above. Upon initialization, the team navigates the workspace and searches for targets. Figure 6.11 also shows the team mission at different points from start to task completion. As well, this figure shows a target with red and green features which is supposed to be assigned to the Burger robot based on the agents' specialization encoding indicated in Table 6.9.

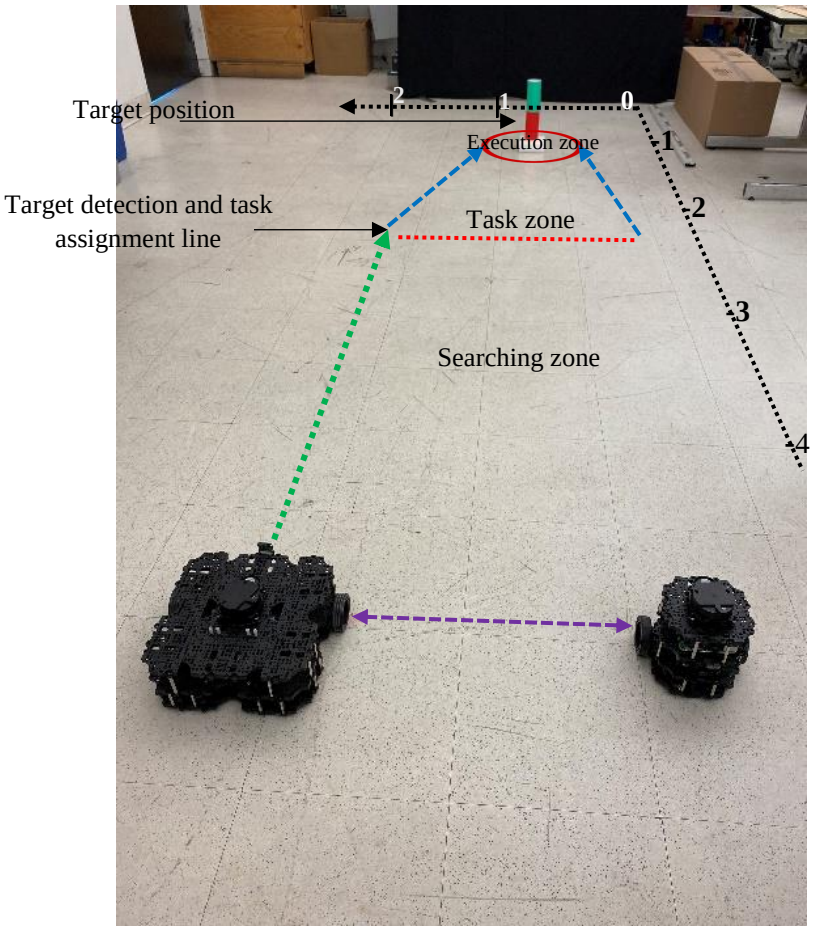
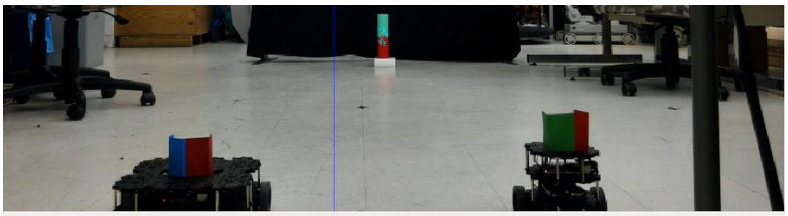
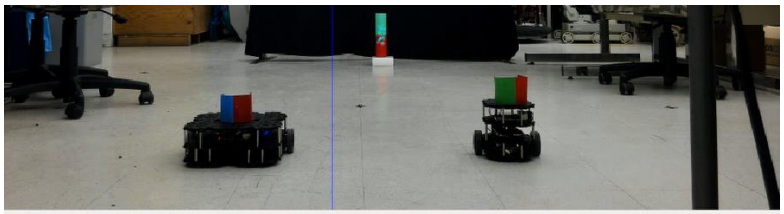


Figure 6.10 Experimental workspace configuration with distances marked in meters

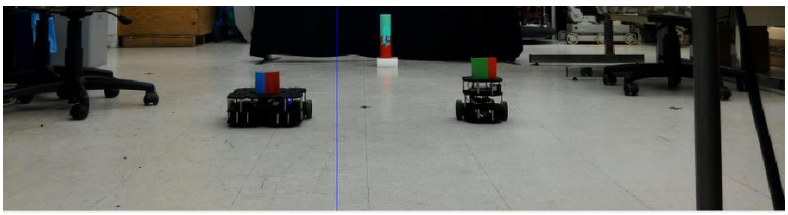
In the experiments with real robots, the proposed specialty-based task allocation approach is tested using Eqs. (4.18) and (4.20) by considering $\mathfrak{p} = 1$ in Eq. (4.16). $\mathfrak{p} = 1$ means that the proposed task allocation scheme computes the agents' task allocation probabilities based on the match between the available specialized agents and the detected features of the given target while neglecting the impact of the agents' attendance. The attendance is neglected since the workspace is of limited size and distance between agents and the target object does not have a significant impact. Table 6.11 presents the complete results for this task allocation process, including the individual agents' availability status, the task-agent matching probabilities of the available agents with to the detected target, and the selected MFT.



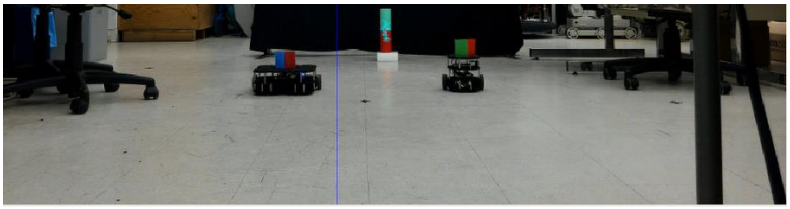
(a) t=0 sec.: Searching for target.



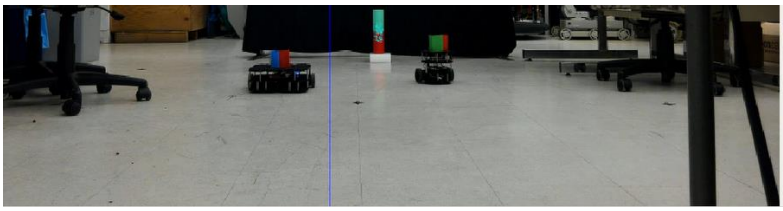
(b) t = 7 sec.: Searching and approaching target.



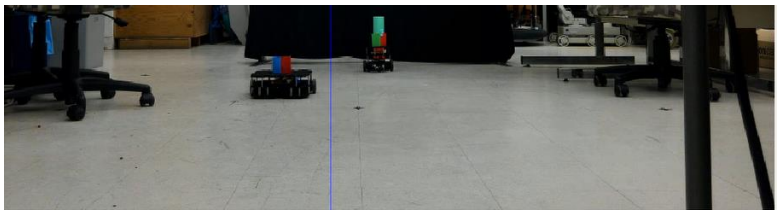
(c) t = 15 sec.: Searching and approaching target.



(d) t = 21 sec.: Task zone border reached, assigning the most qualified agent.



(e) t = 26 sec.: Assigned agent approaching the target.



(f) t = 32 sec.: Assigned agent executing the task.

Figure 6.11 Multiple specialized agents navigate the workspace, search for the target, reach it, and perform the task

Table 6.11 Available agents' fitting probabilities with respect to a target with red and green colors

Target features' recognition confidence level	*Agent name*	*Target-agent specialty matching probability* ($\boldsymbol{Q}$) *Binary encoding*	*Availability 1:Available 0:Withdrawn* ($\boldsymbol{\vartheta_{Av}}$)	*Available agents' attendance level* ($\boldsymbol{\vartheta_{Att}}$)	*Available qualified agents' fitting probabilities* ($\boldsymbol{\Psi_{MFT}}$)	*MFT* (η)
Red: 0.70	**Burger**	**0.75**	**1**	**0.00**	**0.75***	
Blue: 0.00	Waffle Pi	0.35	1	0.00	0.00	0.3
Green: 0.80	Milkshake	0.40	0	0.00	0.00	

Alternatively, in the second test case shown in Figure 6.12, the task allocation system assigns the Waffle Pi robot to a target with red and blue color features detected in the workspace, where each color feature also carries different confidence levels on their recognition. The full results are shown in Table 6.12.

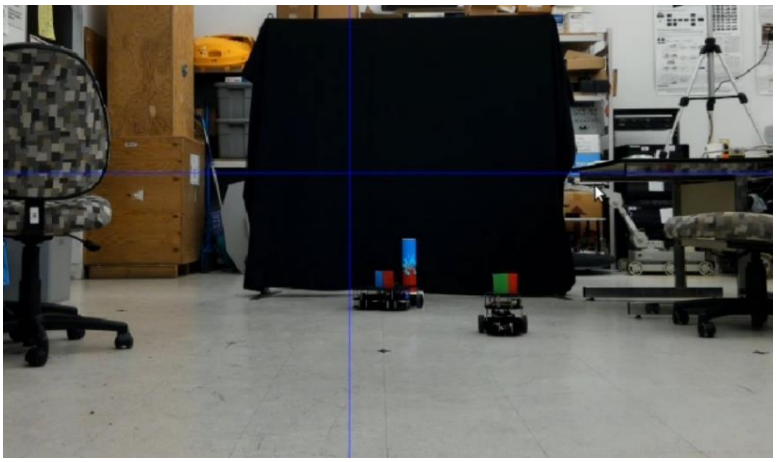


Figure 6.12 The most specialized agent, Waffle Pi robot, responds by reaching the detected target carrying red and blue features

Table 6.12 Available agents' fitting probabilities with respect to a target with red and blue colors

Target features' recognition confidence level	*Agent name*	*Target-agent specialty matching probability* ($\boldsymbol{Q}$) *Binary encoding*	*Availability 1:Available 0:Withdrawn* ($\boldsymbol{\vartheta_{Av}}$)	*Available agents' attendance level* ($\boldsymbol{\vartheta_{Att}}$)	*Available qualified agents' fitting probabilities* ($\boldsymbol{\Psi_{MFT}}$)	*MFT* (η)
Red: 0.70	Burger	0.35	1	0.00	0.00	
Blue: 0.90	**Waffle Pi**	**0.80**	**1**	**0.00**	**0.80***	0.3
Green: 0.00	Milkshake	0.45	0	0.00	0.00	

Third, a test scenario is considered where the system detects a target object but the most specialized agent is not available. Figure 6.13 shows that a target with blue/green features is detected. Based on the team members' specialization encoding, defined in Table 6.9, the most specialized agent in this test case would be the Milkshake robot but it is not available. The recognition confidence level of the target object results in fitting probabilities for the other agents that are below the set MFT level, which is 0.7 in this case. The Burger and Waffle Pi robotic agents' fitting probabilities are 0.3 and 0.4 respectively, which are less than the new MFT. In such a case, the task allocation solution does not assign any robotic agent to the detected target, as shown in Figure 6.13. Then, the available agents stop at the transition line between the searching and task zones (red-dotted line as indicated in Figure 6.10). Detailed results of this test case are presented in Table 6.13.

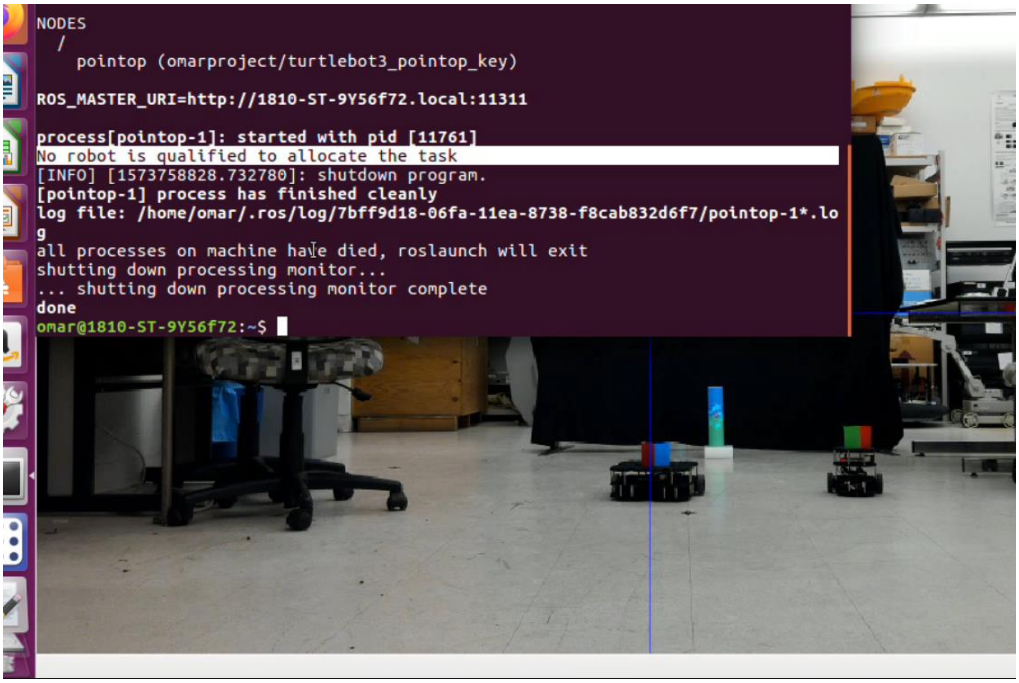


Figure 6.13 The available agents cannot respond to the detected task due to lack of qualification

Table 6.13 Available agents' fitting probabilities with respect to a target with green and blue colors

Target features' recognition confidence level	*Agent name*	*Target-agent specialty matching probability* ($\boldsymbol{Q}$) *Binary encoding*	*Availability 1:Available 0:Withdrawn* ($\boldsymbol{\vartheta_{Av}}$)	*Available agents' attendance level* ($\boldsymbol{\vartheta_{Att}}$)	*Available qualified agents' fitting probabilities* ($\boldsymbol{\Psi_{MFT}}$)	*MFT* (η)
Red: 0.00 Blue: 0.80 Green: 0.60	Burger	0.30	1	0.00	0.00	0.7
	Waffle Pi	0.40	1	0.00	0.00	
	Milkshake	**0.70**	**0**	0.00	0.00	

Finally, Figure 6.14 and Table 6.14 show a scenario where no robot is allocated to a detected target when the recognition confidence level from the target object detection stage decreases in proportion to the set MFT, which indicates a low confidence in the exact nature of the task detected while imposing a given level of trust and security in the robotic intervention process. In this case, the detection of the green and red features on the target remains doubtful, with only 50% of confidence on the recognition of each. As a consequence, the available Burger agent, which is the most specialized agent to respond to this target, is not assigned to it. The fitting probability being influenced by the target recognition confidence level, as detailed in section 4.4, results in the fitting probabilities not satisfying the MFT which is set to 0.7.

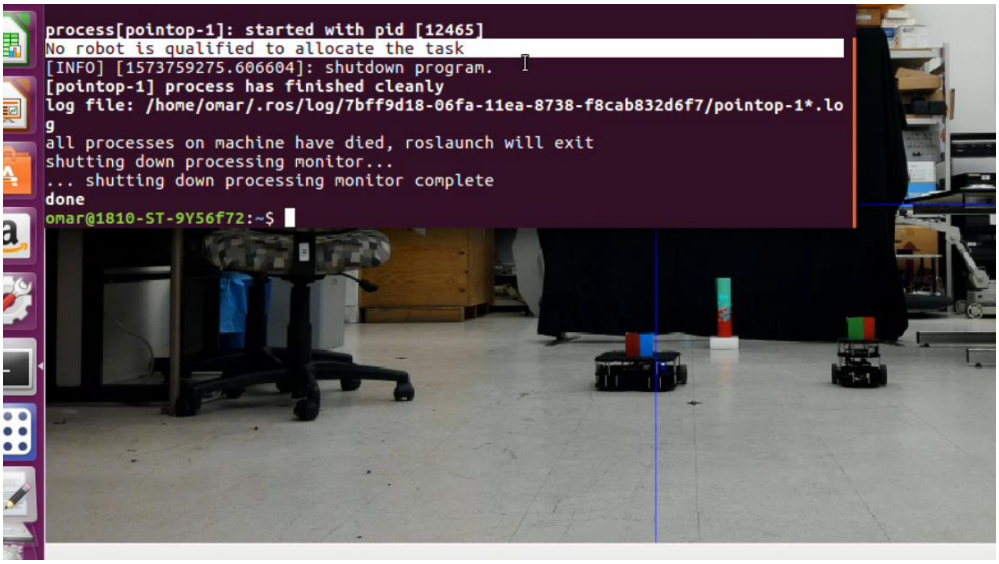


Figure 6.14 No robotic agent responds to the detected task due to a low confidence in target recognition

Table 6.14 Available agents' fitting probabilities with respect to a target with red and green colors.

Target features' recognition confidence level	*Agent name*	*Target-agent specialty matching probability* ($\boldsymbol{Q}$) *Binary encoding*	*Availability 1:Available 0:Withdrawn* ($\boldsymbol{\vartheta_{Av}}$)	*Available agents' attendance level* ($\boldsymbol{\vartheta_{Att}}$)	*Available qualified agents' fitting probabilities* ($\boldsymbol{\Psi_{MFT}}$)	*MFT* (η)
Red: 0.50	Burger	0.50	1	0.00	0.00	
Blue: 0.00	Waffle Pi	0.25	1	0.00	0.00	0.7
Green: 0.50	Milkshake	0.25	0	0.00	0.00	

In addition, it is obvious from the last two cases that when the human operator changes the operational conditions of the task allocation process and increases the minimum level of MFT to 0.7, the result is a more selective task allocation operation.

The results from these real experiments show that the proposed specialty-based task allocation approach applies and performs as expected in matching and assigning the specialized individual members of a realistic robotic team to corresponding specific constrained tasks. The experimental results verify the feasibility and usability of the proposed framework to the degree needed for intended purpose or application.

6.5 Comparison Study

Many factors are considered in this study to design a specialty-based task allocation framework to divide the labour amongst different individual agents with different levels of specialty with respect to a given range of tasks. The purpose is to maximize the task execution efficiency based on the match between individual agents' specialized capabilities and the requirements that are imposed by detected tasks. In addition, this framework automates the missions of robotic teams that consist of specialized individuals and expands the range of the potential applications. The cost function considered here is to maximize a task-agent specialty fitting probability based on matching the detected classes on targets with the respective robotic agents' specialized

capabilities. In addition, a centralized control strategy is adopted to process the individual agents' responses to the task allocation operation. The central unit is implemented to compute the task allocation probabilities whereas the agents' local controllers receive the task allocation commands and respond locally to drive the individual agents to reach the detected tasks. Processing the task allocation dynamics through a central unit alleviates the high cost of the individual local processing units. The proposed specialty-based task-agent matching evaluation is computed in the central control unit because it needs high performance processors which cannot be installed on the individual agents. The agents only run their local decentralized dynamic controllers to control the interactions of the agents with the detected targets and to ensure that the overall system can run a robust task assignment process.

In this section, the essence of the proposed approach is compared with alternative task allocation mechanisms proposed in the literature. In section 6.5.1, the performance of the proposed probabilistic task allocation scheme is compared one-on-one with the same image input datasets to a task allocator using machine learning with self-organizing contextual maps (SOCM) [140]. In addition, a comparison study of the proposed specialty-based task allocation approach with some state-of-the-art task alternative allocation techniques for multi-agent robotic systems coordination is presented in section 6.5.2 to highlight the main differences with the proposed approach as a new framework for tasks allocation in multi robot systems. These comparisons contribute to better position the solution elaborated in this thesis and emphasize its relevance to address several limitations observed with techniques from the literature.

6.5.1 Quantitative Comparison

In this section, the overall performance of the proposed approach to a SAR application is discussed over a complete set of 140 test cases that used real images and simulated robotic agents. A formal comparison is derived where the performance of the proposed probabilistic task allocation scheme was compared one-on-one with the same image input datasets to a task allocator using machine learning with self-organizing contextual maps (SOCM) [140] while considering the same SAR scenarios and the same agents' specialty definition from Table 6.2.

The 140 test cases described in section 6.2.1 were considered with a total of 7 available agents and a binary definition of qualifications as defined in Table 6.2. Table 6.15 shows 20 sample images among the 140 test cases with experimental results from the object recognition stage and mask segmentation corresponding to each of the five classes of target objects defined in Table 6.1, each with the related recognition confidence levels. Task allocation to the robotic agents (R_1 to R_7) respectively with the proposed probabilistic task allocation approach and with the alternative SOCM-based method are also reported to validate the relevance and accuracy of the task allocation process.

The two approaches present different responses to the task allocation operation when a task (or target object) is properly recognized. Of the 140 cases, the object recognition stage, which serves as an input to the task allocation scheme, failed to recognize any object and resulted in no agent allocation in 12 cases (8.6%), similar to case 20 in Table 6.15. Additionally, out of the 140 test cases, 9 (6.4%) presented a misclassification error similar to case 11 in Table 6.15 where non-existing stairs are detected in the scene due to floor tiling with varying colors that appear visually similar to a staircase.

Without considering the refinements proposed for task allocation prioritization introduced in section 4.4.5, the original probabilistic approach was successful 100% of the time including the situations where no objects were detected and the correct response was no allocation as demonstrated in Table 6.15, case 20. This approach was also highly efficient computationally. It took approximately 0.04 second to allocate agents to all 140 cases. This method brings no computational bottleneck to the pipeline, considering that target object recognition running on GPUs necessitated 0.22 second per image. Conversely, the comparative SOCM-based task allocation approach was successful 87.2% of the time on the overall 140 test cases, including situations where no objects were detected and the correct response was no allocation. This approach takes approximately 0.08 second to allocate agents to all 140 cases, which is 50% slower than the time response of the proposed probabilistic approach.

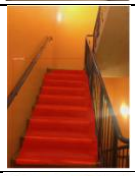
A key difference between the proposed probabilistic approach and the SOCM approach becomes clear when multiple objects are recognized. For example, in Table 6.15, case 10, the posted board is recognized as three separate signs by the task recognition stage, which indicates three separate tasks. The probabilistic method assigns the available specialized agents, R_4 and R_7, whereas the SOCM method assigns three separate agents accordingly. However, only agents R_4 and R_7 are specialized and capable of reading signs, according to Table 6.2. In the current implementation, the SOCM assigns these two agents and identifies a third task that needs to be performed. However, because the first two agents are already assigned to a task, the SOCM finds R_5 as the next available agent with the least number of additional specializations. The latter is then appointed to the third task, even though it is not specialized for reading signs. Depending on the

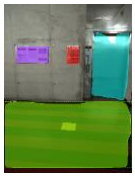
user's expectation, this extra robot assignment can be considered a mismatch error in the SOCM framework for the scenario considered. This accounted for a total of 8 cases (5.7%) out of the 140.

Moreover, unlike the probabilistic approach, the SOCM may consider the detected instances as a single task. In cases 2 and 18 of Table 6.15, the SOCM only assigns one agent to the stairs task even though multiple agents are capable of climbing stairs; this accounted for a total of 10 cases (7.1%) out of the 140. In contrast, the proposed approach sees this task as a way to access the workspace. That is, the proposed approach is flexible enough to allocate either the most specialized agent or all the qualified agents that achieve the minimum desired MFT when they are available to the detected tasks. As a result, the proposed probabilistic approach offers the advantage of assigning all of the qualified agents that possess the capability to perform the detected task. This allows a larger number of agents to pursue the mission and be prepared to face other constrained tasks along their path.

Finally, when the number of detected tasks is less than or equal to the available qualified specialized agents, with both approaches the order of selecting the best-suited agent to allocate to a task is based on selecting the agent with the least number of additional features. An example of this is in case 3 of Table 6.15 where a person is detected. Both R_5 and R_2 are specialized to assist people, but R_5 will be assigned first because it has the least number of other specializations (i.e. none from Table 6.2) compared to R_2 (also qualified to climb stairs). This is meant to conserve the agent with more specializations for future tasks.

Table 6.15 Sample images containing recognized targets among five classes considered and automatically assigned to robotic agents with the proposed approach and alternative SOCM-based method

No.	Input image with detected target(s)	Recognized target object(s) confidence level		Assigned agents with proposed method	Assigned agents with SOCM method
1		Door	0.997	$\mathbf{R_1}$	$\mathbf{R_1}$
		Stairs	0.000		
		Person	0.000		
		Fire	0.000		
		Sign	0.000		
2		Door	0.000	$\mathbf{R_1}$ $\mathbf{R_2}$ $\mathbf{R_3}$ $\mathbf{R_4}$	$\mathbf{R_1}$
		Stairs	0.965		
		Person	0.000		
		Fire	0.000		
		Sign	0.000		
3		Door	0.000	$\mathbf{R_5}$	$\mathbf{R_5}$
		Stairs	0.000		
		Person	0.995		
		Fire	0.000		
		Sign	0.000		
4		Door	0.000	$\mathbf{R_7}$	$\mathbf{R_7}$
		Stairs	0.000		
		Person	0.000		
		Fire	0.000		
		Sign	0.987		
5		Door	0.000	$\mathbf{R_6}$	$\mathbf{R_6}$
		Stairs	0.000		
		Person	0.000		
		Fire	0.971		
		Sign	0.000		
6		Door	0.000	$\mathbf{R_5}$ $\mathbf{R_6}$	$\mathbf{R_5}$ $\mathbf{R_6}$
		Stairs	0.000		
		Person	0.858		
		Fire	0.901		
		Sign	0.000		
7		Door	0.000	$\mathbf{R_5}$ $\mathbf{R_7}$	$\mathbf{R_5}$ $\mathbf{R_7}$
		Stairs	0.000		
		Person	0.987		
		Fire	0.000		
		Sign	0.788		
8		Door	0.000	$\mathbf{R_5}$ $\mathbf{R_6}$	$\mathbf{R_5}$ $\mathbf{R_6}$ $\mathbf{R_3}$
		Stairs	0.000		
		Person	0.985		
		Fire	0.968; 0.628		
		Sign	0.000		
9		Door	0.847	$\mathbf{R_1}$ $\mathbf{R_2}$ $\mathbf{R_3}$ $\mathbf{R_4}$	$\mathbf{R_1}$ $\mathbf{R_5}$
		Stairs	0.978		
		Person	0.000		
		Fire	0.000		
		Sign	0.000		

10		Door Stairs Person Fire Sign	0.000 0.000 0.000 0.000 0.984; 0.940; 0.657	R_7 R_4	R_7 R_4 R_5
11		Door Stairs Person Fire Sign	0.654 0.970 0.000 0.000 1.000; 0.998	R_1 R_2 R_3 R_4 R_7	R_7 R_4 R_1 R_2
12		Door Stairs Person Fire Sign	0.981 0.000 0.989 0.000 0.000	R_1 R_5	R_5 R_1
13		Door Stairs Person Fire Sign	0.000 0.000 0.998 0.665 0.000	R_5 R_6	R_5 R_6
14		Door Stairs Person Fire Sign	0.997 0.000 0.000 0.000 0.733	R_1 R_7	R_1 R_7
15		Door Stairs Person Fire Sign	0.000 0.889 0.000 0.000 0.936	R_1 R_2 R_3 R_4 R_7	R_1 R_4
16		Door Stairs Person Fire Sign	0.000 0.000 0.000 0.987 0.843; 0.624	R_6 R_7	R_6 R_7 R_4
17		Door Stairs Person Fire Sign	0.000 0.000 0.000 0.000 0.928; 0.824	R_7 R_4	R_7 R_4
18		Door Stairs Person Fire	0.000 0.977 0.000 0.000	R_1 R_2 R_3 R_4	R_1

		Sign	0.000		
19		Door	0.000	$\mathbf{R_5}$ $\mathbf{R_6}$ $\mathbf{R_3}$ $\mathbf{R_7}$	$\mathbf{R_7}$ $\mathbf{R_6}$ $\mathbf{R_3}$ $\mathbf{R_5}$
		Stairs	0.000		
		Person	0.630		
		Fire	0.913 0.879		
		Sign	0.963		
20		Door	0.000	------	------
		Stairs	0.000		
		Person	0.000		
		Fire	0.000		
		Sign	0.000		

6.5.2 Framework Comparisons

In this section, the attributes of the proposed framework as a new general task allocation approach for multi-robot systems are compared with some state-of-the-art task alternative allocation techniques for multi-agent robotic systems.

A. Interface Delay Task Allocation (IDTA)

The task allocation approach presented in [141] partitions the foraging task into simpler subtasks called harvesting and storing subtasks. These two subtasks are sequentially inter-dependent, which means that the execution of one sub-task is conditioned by the execution of the other one. As a result, an item is transported from a source position to a task interface area by a harvesting agent. Next, the harvesting agent waits for an available agent that is involved in the storing subtask to deliver the item to that agent, which will then pass it to the nest area. Similarly, a storing agent waits at the task interface border for an available agent that is engaged in the harvesting subtask to pick up the item. This task allocation technique is introduced based on a waiting time that is measured by the agents at the task's interface. It enables a swarm of service

robots to dynamically partition the agents into two specialized groups. The individual agents work autonomously and divide the labour amongst them similar to the proposed approach in this thesis. However, this task allocation scheme does not require the agents to communicate, whereas each individual agent switches between the harvesting and storing subtasks using the locally measured information about the time that the robot must wait to transfer the item at the task interface. The interface delay task allocation method might be an efficient method to enable the robotic agents to move between two subtasks; however, it does not offer an efficient approach for a swarm that has a wide variety of functionalities involved in allocating tasks with different requirements that demand specific agents' functionalities. It also imposes the existence of a formal interface between the agents where their role is transformed, a constraint that the proposed specialty-based task allocation scheme does not bring into the formulation, therefore providing superior flexibility into the definition of tasks and the freedom of movement for every agent.

B. Multiple Travelling Salesman Assignment (MTSA)

This task allocation approach selects the next navigational goal using the famous Travelling Salesman Problem (TSP) distance cost [142]. The distance cost is defined as the travelled distance on the shortest path that connects the robot position with the candidate goals. This task allocation mechanism is developed for a single robot exploration that navigates many goal points and from which the exploration mission can cover all frontier cells. This task allocation approach is optimal for a single robot mission to perform exploration tasks; however, the problem of computing the optimal distance between the robot's position and a set of goals only considers the shortest travelling distance. In comparison, the proposed specialty-based task allocation method deals with an indefinite number of agents; it optimizes the selection of agents beyond just the travelling distance; and it easily adapts to a wide range of robot's specialization considerations according to

the nature of the tasks to be performed. The travelling distance is only considered in the proposed approach to boost the robots' specialty fitting probabilities and to increase the chance of assigning the closer agents with higher fitting probabilities to optimize the system's time responses for wide workspace applications.

Table 6.16 summarizes the main differences between the task allocation approach proposed in this thesis and three alternative methods in the context of multi-agent robotic teams.

Table 6.16 Comparison of the proposed approach to three state-of-the-art task allocation approaches

Task Allocation Approach	**Utility function**	**Constraints**		**Location**		**Control Strategy**		**Applications**
		Task	Agent	Agent's location	Target's location	Centralized	De-Centralized	
Specialty-based (proposed)	- Target-agent specialty-based probability - Target-agent distance - Agents' availability - Agents attendance - Agents primary capability	Target characteristics / specific features (task requirements)	Agent's specialized capabilities	√	√	√		- Firefighting - Search-and-rescue - Service robotics - Military/security - Parcel delivery - Public transportation - Space
IDTA [141]	Agents' waiting times at the task interface	Target location	Agents' arrival times at the task position		√		√	- Service robotics
MTSA [142]	Distance	Minimal length of the travelled trajectory		√	Target cell		√	- Exploration tasks
SOCM [140]	- Euclidean distance between the specialty of the robot (input of the network) and the unit in the output layer of the network - Agent's availability	Target characteristics / specific features	Agent's specialized capabilities	√	√	√		- Search-and-rescue

6.6 Summary

The work presented in this chapter illustrates different applications of the proposed specialty-based task allocation scheme and details a variety of experimental results conducted in simulation as well as on real robotic platforms. The experiments considered for the first application in indoor

search-and-rescue scenarios (SAR) validate the effectiveness of the proposed approach to assign the individual members of a team of specialized robotic agents to corresponding SAR tasks imposing different requirements. The efficiency of the proposed approach is also formally compared to a machine learning-based task allocation approach addressing the exact same tasks.

A real-world implementation of the solution is also demonstrated on a number of physical mobile robots as a centralized task allocator. Various test cases considered also validate the feasibility of the proposed approach to desirable real-life applications.

Finally, a comparison study of the proposed specialty-based task allocation approach with some state-of-the-art task alternative allocation techniques for multi-agent robotic systems coordination highlighted the main differences with the proposed approach. This comparison contributes to better position the solution elaborated in this thesis and emphasizes its relevance to address several limitations observed with techniques from the literature, especially in terms of the flexibility to adapt to operational contexts and constraints.

Chapter 7 Conclusion

7.1 Summary

Following an extensive study of previous and related works focusing on topics such as swarm formation, coordination strategies, homogeneous and heterogeneous multi-agent systems, research gaps are identified in Chapter 2. In order to take advantage of cooperation among non-homogeneous agents that possess specialized functionalities attributed to swarm individuals, a rigorous mechanism is designed for task-agent specialty-based assignments to achieve an efficient specialization framework among a cooperative robotic swarm.

The first phase of the research work, reported in Chapter 3, addresses the design problems of task-agent specialty-based assignment at the agent's control level, such as a swarm's formation control and navigation. A task-agent specialization-based matching approach is introduced in two progressive stages. The first stage of the design addresses the idea of each individual's specialization as a cooperative control problem on the basis of switching the swarm's formation to a new group formation in response to a detected task. The proposed system assigns the leading role to a specific agent based solely on the position of the formation over its workspace. For the purpose of system validation, a dynamical model of non-holonomic mobile robot is considered whereas a kinematic and torque controller is used to control the path following and group formation in simulated scenarios. Collision avoidance is also addressed using repulsive potential forces to prevent the agents from colliding during formation transition. In the second stage of this phase of research, task recognition based on perceivable characteristics on a target object is considered. An agent's specialty definition and a task allocation approach are rigorously

formulated in an algorithmic manner to evolve the definition of agent specialization in alignment with target recognition. Detailed results of this phase of development, evaluated through MATLAB simulations, are presented in Chapter 3.

The following step, detailed in Chapter 4, expands the initial design and implementation of the specialty-based task allocation framework to two coupled spaces: a control space and a specialization space. Next, the task-agent matching and allocation approach is derived in the specialization space. This approach represents the core contribution of this thesis. A representation for specializing individual members of a robotic swarm is introduced. The agents' specializations are defined in two forms: a general binary form which is used as the basis of the probabilistic specialty-based task allocation approach, and a modulated form of the agents' specialization which supports different levels of sophistication in the agents' mechanical construction or embedded resources. The specialized capabilities of individual agents are modelled and matched to the corresponding features recognized on target objects with a quantified fitting level. The confidence levels on the recognized features on a target object (corresponding to a specific task) are combined to drive the agent-task specialty matching scheme. This scheme computes an agent-task specialty fitting score encoded in the form of probabilities. The framework is also extended to coordinate the specialized individuals with the corresponding tasks by considering the agents' availability and attendance level along with their specialty fitting probabilities. In addition, the framework supports strategic guidance from a human operator to orient the task assignment process with situational awareness. The process is designed to keep human's cognitive load low while adjusting the system's operational conditions at a high level of coordination only to result in safer and more selective task allocation operation. The proposed approach is also extended to tackle tasks with

multiple classes. Agents' assignment priorities are addressed at the task allocation stage, with priority being attributed to selected agents among the swarm based on the agents' primary capabilities and the nature of the detected tasks. This process results in safe task allocation operations and ensures higher coherence in the task-agent specialty fitting.

In Chapter 5, a simulator is implemented to investigate and validate the efficiency of the proposed framework introduced in Chapter 4. Simulation results demonstrate that the proposed approach is successful at properly assigning specialized agents to corresponding tasks that require a specific level of robots' specialty-based suitability, such as specific mechanical or instrumentation characteristics from autonomous robots. Simulation results also demonstrate that the proposed framework is sufficiently reliable to deal with various confidence levels on the target object detection as well as to assign alternative robotic agents, when necessary, to accomplish the team's mission goals. In addition, simulations show that the proposed framework operates at a high level of flexibility to lead multi-robot specialized teams to tackle tasks over different scales of workspaces.

In Chapter 6, the proposed approach is further studied in relation to various potential application areas such as search-and-rescue (SAR). The experimental results support a realistic evaluation of the overall performance of the proposed framework based on real target object detection results as well as demonstrating the viability of the proposed specialty-based task allocation approach for such real applications under various levels of constraints and complexity levels. The proposed framework is revealed to be an efficient task allocator for individual specialized members of a team of robots. The feasibility and effectiveness of the proposed approach is further demonstrated by transferring the simulated system to a limited scale implementation on physical robotic agents.

The latter experiments confirm the potential to transpose the main concepts developed in this thesis to real robotic systems. This forms a first demonstration of the impact that the proposed technology may have on future generations of autonomous unmanned robotic systems. Finally, a formal comparative study is presented that positions the proposed specialty-based task allocation framework with respect to alternative task allocation approaches from the literature, as well as methods studied in margin of this research. The comparison emphasizes the main differences between the approaches and highlights the potential applications. Most importantly, it highlights the generality of the original framework that places a unique emphasis on the specific physical, mechanical, or computational functionalities embedded in robotic agents that form a cooperative swarm.

7.2 Original Contributions

The main original contributions of this thesis are twofold:

Primary Contributions:

1. The design of an elegant and practical formulation of the concept of specialization at the level of functionalities among individual members of a robotic swarm. The formal consideration of robots' specializations is formulated using two explicit specialization encoding schemes:

A. A binary form, which represents generic definitions of the agents' specialized capabilities as the basis of the specialty-based task allocator;

B. A modulated form of the agents' specialization to represent different skill levels in the agents' specialized capabilities based on their mechanical, physical or computational structure. Inspired by how human beings are more or less talented in different domains or activities, this refinement provides additional flexibility and sophistication to the

speciality-based task allocator to select amongst the specialized individuals based on different levels of agents' suitability with respect to the detected task.

2. The design and implementation of an original probabilistic task-agent matching mechanism working from a confidence level on task's characteristics recognition with the objective to optimally fit the most competent, available and attending robotic agent, or agents, to a recognized corresponding task.

Secondary Contributions:

3. The extension of the original framework to tackle tasks belonging to multiple classes and the implementation of an assignment prioritization scheme to automatically prioritize the agents that possess primary capabilities that reveal essential with respect to the detected task.

4. The theoretical development of a task features' recognition stage that leverages output of the systems sensing layer and provides an input to the specialty-based task allocation scheme.

5. The design and implementation of the specialty-based task allocation framework as a formation transition strategy for a group of robots that involves specialized agents with different specialized capabilities. This strategy extends the concept of cooperative formation in multi-robot systems to a rigorous process in the context of specialty-based task allocation responses of individual robots.

6. The integration of a supporting task recognition sensing and data processing stage with uncertainty handling to formally adapt the specialty-based task allocation framework to various real-life applications as well as to permit extended experimental validation of the proposed concepts both in simulation and on physical robotic systems.

The outcomes of this research present an innovative formalism for task allocation in the context of a cooperative swarm of robots. Unlike previous work that considered heterogeneity among robotic agents mainly from their physical construction, here a specific definition of specialization is introduced which leverages the embedded hardware and software characteristics of each agent. As a result, an advanced form of specialized labour division emerges which divides the workload among the individual members of the swarm based on best matching of the tasks' specific requirements to the robots' capabilities. This form of task allocation can reduce equipment cost and increase the net efficiency of the swarm. In addition, the proposed probabilistic fitting scheme leverages modern deep learning object recognition approaches for autonomous target detection in the working environment, and consequently computes a task-agent specialty probabilistic fitting score which supports optimal task allocation and increases the specificity of task execution in a broad range of application scenarios.

The research outcomes of this thesis have led to a number of publications. The contents of Chapter 3 are documented in [128], [129], [130]. The original methodology for probabilistic specialty-based task allocation detailed in Chapter 4, along with some of the experimental results presented in Chapter 5 and Chapter 6, are reported in [134], [143], [144], [145], and [146]. Additionally, the integration with complementary work related to target object recognition from vision sensors is detailed in [131].

7.3 Future Work

A new methodology approach for multi-agent task allocation is proposed in this thesis for a multi-agent robotic system. Throughout the design and implementation of the proposed framework, many constraints have been considered such as simulated sensing layer, offline object

detection input, as well as well-structured laboratory testing environment. In addition, the solutions are designed and implemented to perform well in the context of the considered constraints. Alternatively, it would be a very interesting future direction to integrate the proposed framework with an advanced sensing layer to measure online the task characteristics and test the system responses. In addition, complex tasks have to be studied in a complex environment, which will align the proposed framework to its greatest extent with real-life applications.

The overall situational understanding is usually shared with the system by the human operator intervention on the MFT set point. However, the system operator may struggle to read the situation especially with larger numbers of robots and increased requests for task allocations. Therefore, further reducing the human cognitive load while expanding the number of agents could be an interesting direction for further developments to increase the system's reasoning capabilities. Alternatively, a model to provide desirable measures of situational awareness can be implemented to automatically drive MFT through desirable values in response to changing the system's operational conditions. The modelling of the situational awareness should consider many parameters such as the total number of robots, the number of robots that are available in service, the desirable level of task allocation accuracy, and the progressive requests of task allocations.

REFERENCES

[1] Beni, G. "From swarm intelligence to swarm robotics," *International Workshop on Swarm Robotics. Springer*, Berlin, Heidelberg, pp. 1-9, July 2004.

[2] Beni, G., and Wang, J. "Swarm intelligence in cellular robotic systems," In *Robots and Biological Systems: Towards a New Bionics*. *Springer*, Berlin, Heidelberg, pp. 703-712, 1993.

[3] Şahin, E. "Swarm robotics: From sources of inspiration to domains of application," *International Workshop on Swarm Robotics*, Springer, Berlin, Heidelberg. pp. 10-20, July 2004.

[4] Dudek, G., Jenkin, M.R., Milios, E., and Wilkes, D. "A taxonomy for multi-agent robotics". *Autonomous Robots*, vol. 3(4), pp. 375-397, 1996.

[5] Cao, Y. U., Fukunaga, A. S. and Kahng, A. "Cooperative mobile robotics: Antecedents and directions", *Autonomous Robots*, vol. 4(1), pp. 7-27, 1997.

[6] Gazi, V., and Passino, K. M., "Stability analysis of social foraging swarms," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 34(1), pp. 539-557, 2004.

[7] Farinelli, A., Iocchi, L., and Nardi, D. "Multirobot systems: a classification focused on coordination," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 34(5), pp. 2015-2028, 2004.

[8] Susca, S., Bullo, F., and Martinez, S. "Monitoring environmental boundaries with a robotic sensor network," *IEEE Transactions on Control Systems Technology*, vol. 16(2), pp. 288-296, 2008.

[9] Fiorelli, E., Leonard, N.E., Bhatta, P., Paley, D. A., Bachmayer, R., and Fratantoni, D. M., "Multi-AUV control and adaptive sampling in Monterey Bay," *IEEE Journal of Oceanic Engineering*, vol. 31(4), pp. 935-948, 2006.

[10] Antonelli, G., Arrichiello, F., and Chiaverini, S. "The entrapment/escorting mission," *IEEE Robotics and Automation Magazine*, vol. 15(1), 2008.

[11] Wu, H., Tian, G. and Huang, B. "Multi-robot collaborative localization methods based on Wireless Sensor Network," *IEEE International Conference in Automation and Logistics*, pp. 2053-2058, Qingdao, China, 2008.

[12] Viguria Jimenez, L. A., and Howard, A.M. "An integrated approach for achieving multi-robot task formations". *IEEE/ASME Transactions on Mechatronics*, vol. 14(2), pp. 176-186, 2009.

[13] Dahl, T. S., Mataric, M. J., and Sukhatme, G.S. "Multi-robot task-allocation through vacancy chains," *IEEE International Conference in Robotics and Automation (ICRA'03)*, vol. 2, pp. 2293-2298, 2003.

[14] Brutschy, A. "Task allocation in swarm robotics. Towards a method for self-organized allocation to complex tasks," *University Libre de Bruxelles*, 1050 Bruxelles, Belgium, Technical Report TRlIRIDIA12009–007, 52009, May 2009.

[15] Horling, B., and Victor L. "A survey of multi-agent organizational paradigms," *The Knowledge Engineering Review*, vol. 19(4), pp. 281-316, 2004.

[16] Al-Yafi, K., Habin L., and Afshin M. "Mtap-masim: a multi-agent simulator for the mobile task allocation problem," *IEEE International Workshops in Enabling Technologies: Infrastructures for Collaborative Enterprises WETICE'09*, pp. 25-27. 2009.

[17] Coltin, B., and Manuela V. "Mobile robot task allocation in hybrid wireless sensor networks," *IEEE International Conference in Intelligent Robots and Systems (IROS)*, pp. 2932-2937, 2010.

[18] Liu, C., and Kroll, A. "A centralized multi-robot task allocation for industrial plant inspection by using a* and genetic algorithms". *International Conference on Artificial Intelligence and Soft Computing*, pp. 466-474. Springer, Berlin, Heidelberg. 2012.

[19] Atsushi Y., Tamio A., Jun O., and Hajime A. "Motion planning of multiple mobile robots for cooperative manipulation and transportation". *IEEE Transactions on Robotics and Automation*, vol. 19(2), pp. 223-237, 2003.

[20] Clark, C. M., Rock, S.M. and Latombe, J.C., "Motion planning for multiple mobile robots using dynamic networks," *IEEE International Conference in Robotics and Automation ICRA'03*, vol. 3, pp. 4222-4227, 2003.

[21] Brumitt, B. L., and Stentz, A.. "GRAMMPS: A generalized mission planner for multiple mobile robots in unstructured environments," *IEEE International Conference in Robotics and Automation ICRA'98*, vol. 2, pp. 1564-1571, 1998.

[22] Giordani, S., Lujak, M., and Martinelli, F. "A distributed algorithm for the multi-robot task allocation problem," *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, pp. 721-730. Springer, Berlin, Heidelberg, 2010.

[23] Choi, H.L., Brunet, L., and How, J. P. "Consensus-based decentralized auctions for robust task allocation," *IEEE Transactions on Robotics*, vol. 25(4), pp. 912-926, 2009.

[24] Rooker, M. N., and Birk, A. "Multi-robot exploration under the constraints of wireless networking", *Control Engineering Practice*, vol. 15(4), pp. 435-445, 2007.

[25] Bicchi, A., Danesi, A., Dini, G., La Porta, S., Pallottino, L., Savino, I.M., and Schiavi, R. "Heterogeneous wireless multirobot system. " *IEEE Robotics and Automation Magazine*, vol. 15(1), 2008.

[26] Bayram, H., and Bozma, H. I. "Multi-robot navigation with limited communication-deterministic vs game-theoretic networks". *IEEE International Conference on Intelligent Robots and Systems (IROS)*, pp. 1825-1830, 2010.

[27] Reif, J. H., and Wang, H. "Social potential fields: A distributed behavioral control for autonomous robots", *Robotics and Autonomous Systems*, vol. 27(3), pp. 171-194, 1999.

[28] Song, P., and Kumar, V. "A potential field based approach to multi-robot manipulation". *IEEE International Conference on Robotics and Automation*, vol. 2, pp. 1217-1222, 2002.

[29] Sabattini, L., Cristian S., and Cesare F. “Arbitrarily shaped formations of mobile robots: artificial potential fields and coordinate transformation”, *Autonomous Robots*, vol. 30(4), pp. 385-397, 2011.

[30] Chaimowicz, L., Sugar, T., Kumar, V., and Campos, M. F. M. “An architecture for tightly coupled multi-robot cooperation”, *IEEE International Conference on Robotics and Automation*, vol. 3, pp. 2992-2997, 2001.

[31] Desai, J. P., Ostrowski, J., and Kumar, V. “Controlling formations of multiple mobile robots”. *IEEE International Conference on Robotics and Automation*, vol. 4, pp. 2864-2869, 1998.

[32] Fierro, R., Das, A. K., Kumar, V., and Ostrowski, J. P. “Hybrid control of formations of robots”, *IEEE Intl Conference on Robotics and Automation*. Vol. 1, pp. 157-162, 2001.

[33] Shao, J., Xie, G., and Wang, L. “Leader-following formation control of multiple mobile vehicles”, *IET Control Theory and Applications*, vol. 1(2), pp. 545-552, 2007.

[34] Spry, S., and Hedrick, J. K. “Formation control using generalized coordinates”, *IEEE Conference on Decision and Control.* vol. 3, pp. 2441-2446, 2004.

[35] Zhang, F., Goldgeier, M., and Krishnaprasad, P. S. “Control of small formations using shape coordinates”, *IEEE Intl Conference on Robotics and Automation*, vol. 2, pp. 2510-2515, Sept. 2003.

[36] Yamakita, M., and Saito, M. “Formation control of SMC with multiple coordinate systems”, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1, pp. 1023-1028, Sept. 2004.

[37] Tan, K. H., and Lewis, M. A. “Virtual structures for high-precision cooperative mobile robotic control.” *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1, pp. 132-139, Nov. 1996.

[38] Gazi, V. “Formation control of a multi-agent system using non-linear servomechanism”, *International Journal of Control*, vol. 78(8), pp. 554-565. 2005.

[39] Kobayashi, F., Tomita, N., and Kojima, F. “Re-formation of mobile robots using genetic algorithm and reinforcement learning”, *SICE Annual Conference*, vol. 3, pp. 2902-2907, 2003.

[40] Reynolds, C. W. “Flocks, herds and schools: A distributed behavioral model,” *ACM SIGGRAPH Computer Graphics*, vol. 21(4), pp. 25-34. July 27-31, 1987.

[41] Balch, T., and Arkin, R. C. “Behavior-based formation control for multi-robot teams”, *IEEE Trans. on Robotics and Automation*, vol. 14(6), pp. 926-939. 1998.

[42] Cao, Z., Xie, L., Zhang, B., Wang, S., and Tan, M. “Formation constrained multi-robot system in unknown environments”, *IEEE Intl Conference on Robotics and Automation*, vol. 1, pp. 735-740. Sep. 2003.

[43] Cao, Z., Tan, M., Wang, S., Fan, Y., and Zhang, B. “The optimization research of formation control for multiple mobile robots”, *World Congress on Intelligent Control and Automation*. vol. 2, pp. 1270-1274, 2002.

[44] Monteiro, S., and Bicho, E. “A dynamical systems approach to behavior-based formation control”, Proc. of *IEEE Intl Conference on Robotics and Automation*, vol. 3, pp. 2606-2611, 2002.

[45] Desai, J. P., Ostrowski, J.P., and Kumar, V. “Modeling and control of formations of nonholonomic mobile robots." *IEEE Transactions on Robotics and Automation*, vol. 17(6) pp. 905-908, 2001.

[46] Mesbahi, M., and Hadaegh, F.Y. “Formation flying control of multiple spacecraft via graphs, matrix inequalities, and switching,” *IEEE International Conference on Control Applications*, vol. 2, pp. 1211-1216, 1999.

[47] Howard, C. E. “Scientists to develop miniature military robots” *Military and Aerospace Electronics*, 2008.

[48] L. G. Weiss, “Autonomous robots in the fog of war,” *IEEE Spectrum*, vol. 48 (8), Aug. 2011.

[49] Santana, P., Barata, J., Cruz, H., Mestre, A., Lisboa, J., and Flores, L. “A multi-robot system for landmine detection,” *IEEE Conference on in Emerging Technologies and Factory Automation, ETFA’05*, vol. 1, pp. 8-15, 2005.

[50] Kakalis, N. M., and Ventikos, Y. “Robotic swarm concept for efficient oil spill confrontation” *Journal of Hazardous Materials*, vol. 154(1-3), pp. 880-887, 2008.

[51] Maglogiannis, I. ”Robotic sensor networks: an application to monitoring electro-magnetic fields”, *Emerging Artificial Intelligence Applications in Computer Engineering: Real Word AI Systems with Applications in EHealth, HCI, Information Retrieval and Pervasive Technologies*, vol. 160, pp. 384, 2007.

[52] Doherty, M. B., “A Weight-distributing Placement Algorithm for Large Teams of Lifting Robots”, M.Sc. thesis, Carleton University, 2011.

[53] Eoh, G., Jeon, J. D., Choi, J. S., and Lee, B. H. “Multi-robot cooperative formation for overweight object transportation”, *IEEE/SICE Intl Symposium on System Integration (SII)*, pp. 726-731. Dec. 2011.

[54] Suzuki, T., Sekine, T., Fujii, T., Asama, H., and Endo, I. “Cooperative formation among multiple mobile robot teleoperation in inspection task”, *IEEE Intl Conference on Decision and Control*. vol. 1, pp. 358-363. Dec. 2000.

[55] Wang, Z., and Gu, D. “Cooperative target tracking control of multiple robots”, *IEEE Transactions on Industrial Electronics*, vol. 59(8), pp. 3232-3240, 2012.

[56] Parker, L. E. “Current state of the art in distributed autonomous mobile robotics”, *Distributed Autonomous Robotic Systems*, vol. 4, pp. 3-12, Springer, 2000.

[57] Matsuda, K., and Ishihara, H. “Improvement of organizational efficiency by multi-robot system for manufacturing operation,” *International Conference in Mechatronics and Automation* (ICMA’07), pp. 2014-2019, 2007.

[58] Doring, K., and Petersen, H. G. "Multi-robot task scheduling in micro-manufacturing," *IEEE International Symposium in Assembly and Task Planning: From Nano to Macro Assembly and Manufacturing* (*ISATP'05*), pp. 125-131, 2005.

[59] Gregory, J., Fink, J., Stump, E., Twigg, J., Rogers, J., Baran, D., Fung, N., and Young, S. "Application of multi-robot systems to disaster-relief scenarios with limited communication," *Field and Service Robotics*, pp. 639-653. Springer, Cham, 2016.

[60] Barea, R., Bergasa, L. M., Lopez, E., Ocana, M., Schleicher, D., and León, A. "Robotic assistants for health care," *IEEE International Conference in Robotics and Biomimetics* (*ROBIO'08*), pp. 1099-1104, 2009.

[61] Cohn, M. B., Crawford, L. S., Wendlandt, J. M., and Sastry, S. S., "Surgical applications of milli-robots", *Journal of Field Robotics*, vol. 12 (6), pp. 401-416, 1995.

[62] Brambilla, M., Ferrante, E., Birattari, M., and Dorigo, M. "Swarm robotics: a review from the swarm engineering perspective". *Swarm Intelligence*, vol. 7, (1), pp. 1-41, 2013.

[63] Soysal, O., and Şahin, E. "A macroscopic model for self-organized aggregation in swarm robotic systems." *In International Workshop on Swarm Robotics*, pp. 27-42. Springer, Berlin, Heidelberg, 2006.

[64] Shucker, B., and Bennett, J. K. "Scalable control of distributed robotic macrosensors", *Distributed Autonomous Robotic Systems*, vol. pp. 379-388, 2007.

[65] Hettiarachchi, S., Maxim, P. M., Spears, W. M., and Spears, D. F. "Connectivity of collaborative robots in partially observable domains". *International Conference on Control in Automation and Systems*, pp. 14-17. 2008.

[66] O'Grady, R., Christensen, A. L., and Dorigo, M. "SWARMORPH: multirobot morphogenesis using directional self-assembly". *IEEE Transactions on Robotics*, vol. 25 (3), pp. 738-743, 2009.

[67] Beckers, R., Holland, O. E., and Deneubourg, J. L. "From local actions to global tasks: Stigmergy and collective robotics". *Artificial Life IV*, vol. 181, p. 189, 1994.

[68] Wawerla, J., Sukhatme, G. S., and Mataric, M. J. "Collective construction with multiple robots". *IEEE/RSJ International Conference in Intelligent Robots and Systems*, vol. 3, pp. 2696-2701, 2002.

[69] Payton, D., Regina E., and Mike H. "Pheromone robotics and the logic of virtual pheromones." *International Workshop on Swarm Robotics*, pp. 45-57. Springer, Berlin, Heidelberg, 2004.

[70] Howard, A., Mataric, M. J., and Sukhatme, G. S. "Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem". *Distributed Autonomous Robotic Systems*, vol. 5 pp. 299-308, 2002.

[71] Sabattini, L, Secchi, C., and Fantuzzi, C. "Arbitrarily shaped formations of mobile robots: artificial potential fields and coordinate transformation." *Autonomous Robots*, vol. 30 (4), pp. 385-397, 2011.

[72] Balch, T., and Hybinett, M. “Social potentials for scalable multi-robot formations." *IEEE International Conference in Robotics and Automation, (ICRA'00)*, vol. 1, pp. 73-80, 2000.

[73] Baldassarre, G., Nolfi, S., and Parisi, D., “Evolving mobile robots able to display collective behaviors,” *Artificial Life*, vol. 9(3), pp. 255-267, 2003.

[74] Turgut, A. E., Çelikkanat, H., Gökçe, F., and ¸Sahin, E. “Self-organized flocking in mobile robot swarms”. *Swarm Intelligence*, vol. 2(2-4), pp. 97–120, 2008.

[75] Stranieri, A., Ferrante, E., Turgut, A. E., Trianni, V., Pinciroli, C., Birattari, M., and Dorigo, M. “Self-organized flocking with a heterogeneous mobile robot swarm”. *Advances in Artificial Life, ECAL'*11. pp. 789–796, Cambridge: MIT Press, 2011.

[76] Ferrante, E., Turgut, A. E., Huepe, C., Stranieri, A., Pinciroli, C., and Dorigo, M. “Self-organized flocking with a mobile robot swarm: a novel motion control method.” *Adaptive Behavior*, vol. 20(6), pp. 460-477, 2012.

[77] Ferrante, E., Brambilla, M., Birattari, M., and Dorigo, M. “Socially-mediated negotiation for obstacle avoidance in collective transport”. *Distributed Autonomous Robotic Systems*, pp. 571-583. Springer Berlin Heidelberg, 2013.

[78] Wessnitzer, J., and Melhuish, C. “Collective decision-making and behaviour transitions in distributed ad hoc wireless networks of mobile robots: Target-hunting*.” European Conference on Artificial Life (ECAL)*, vol. 2801, pp. 893-902, 2003.

[79] Campo, A., Garnier, S., Dédriche, O., Zekkri, M., and Dorigo, M., “Self-organized discrimination of resources.” *PLoS One*, vol. 6(5), p.e19888, 2011.

[80] Chamoun, P. “Rigorous movement of convex polygons on a path using multiple robots”. M.Sc. thesis, Carleton University; 2012.

[81] Ma, H., and S. Koenig, S. “Optimal target assignment and path finding for teams of agents”. *International Conference on Autonomous Agents and Multiagent Systems*, pp. 1144–1152, 2016.

[82] Abukhalil, T., Patil, M., Patel, S., and Sobh, T. “Coordinating a heterogeneous robot swarm using Robot Utility-based Task Assignment (RUTA)”, *IEEE 14th International Workshop on Advanced Motion Control (AMC)*, pp. 57-62, 2016.

[83] Scheidler, A., Arne B., Eliseo, F., and Marco D. “The k-Unanimity Rule for Self-Organized Decision-Making in Swarms of Robots,” *IEEE Transactions on Cybernetics*, vol. 46(5), pp. 1175-1188, 2016.

[84] Alitappeh, R. J., Jeddisaravi, K., and Guimarães, F. G. “Multi-objective multi-robot deployment in a dynamic environment”, *Soft Computing*, pp. 1-17. 2016.

[85] Roy, B. “Classement et choix en présence de points de vue multiples.” *Revue française d'informatique et de recherche opérationnelle*, vol. 2, no. 8, pp. 57-75, 1968.

[86] O'Hara, K. J., and Balch. T. R. “Pervasive sensor-less networks for cooperative multi-robot tasks.” *Distributed Autonomous Robotic Systems*, vol. 6, pp. 305-314, 2007.

[87] Mathews, N., Christensen, A. L., O'Grady, R., and Dorigo, M. "Spatially targeted communication and self-assembly." *IEEE/RSJ International Conference in Intelligent Robots and Systems (IROS'12)*, pp. 2678-2679. 2012.

[88] Sadeghi, A., and Smith, S. L. "Heterogeneous Task Allocation and Sequencing via Decentralized Large Neighborhood Search". *Unmanned Systems*, vol. 5, no. 02, pp. 79-95, 2017.

[89] Ferrante E, Duéñez-Guzmán E., Turgut, A.E., and Wenseleers, T. "GESwarm: Grammatical evolution for the automatic synthesis of collective behaviors in swarm robotics". *Annual Conference on Genetic and Evolutionary Computation, ACM*, pp. 17-24, July 2013.

[90] Ferrante, E., Turgut, A.E., Duéñez-Guzmán, E., Dorigo, M., and Wenseleers, T. "Evolution of self-organized task specialization in robot swarms". *PLoS Computational Biology* 6; 11(8):e1004273, Aug. 2015.

[91] Ijspeert, A., Martinoli, A., Billard, A., and Gambardella, L. M. "Collaboration through the Exploitation of Local Interactions in Autonomous Collective Robotics: The Stick Pulling Experiment," *Autonomous Robots*, vol. 11, no. 2, pp. 149–171, 2001.

[92] Hsieh, M. A., Halász, Á., Cubuk, E. D., Schoenholz, S., and Martinoli, A. "Specialization as an optimal strategy under varying external conditions". *IEEE International Conference in Robotics and Automation (ICRA'09),* vol. 12, pp. 1941-1946, May 2009.

[93] Halász, A. M., Liang, Y., Hsieh, M. A., and Lai, H. J. "Emergence of specialization in a swarm of robots". *Distributed Autonomous Robotic Systems*, pp. 403-416, Springer Berlin Heidelberg, 2013.

[94] Quinn, M. "A comparison of approaches to the evolution of homogeneous multi-robot teams". Congress in *Evolutionary Computation*, vol. 1, pp. 128-135, 2001.

[95] Tuci, E., and Trianni, V. "On the evolution of homogeneous multi-robot teams: Clonal versus aclonal approaches", *Neural Computing and Applications*, vol. 25(5), pp. 1063–1076. 2014.

[96] Pini, G, Brutschy, A, Scheidler, A, Dorigo, M, and Birattari, M. "Task partitioning in a robot swarm: Object retrieval as a sequence of subtasks with direct object transfer". *Artificial Life*, vol. 20(3), pp. 291-317, July 2014.

[97] Buchanan, E., Pomfret, A., and Timmis, J. "Dynamic Task Partitioning for Foraging Robot Swarms." *International Conference on Swarm Intelligence*, pp. 113-124, Springer International Publishing, 2016.

[98] Trueba, P., Prieto, A., Bellas, F., Caamaño P., and Duro, R. J. "Specialization analysis of embodied evolution for robotic collective tasks", *Robotics and Autonomous Systems*; vol. 61(7), pp. 682-693, July 2013.

[99] Montanier, J. M., Carrignon, S., and Bredeche, N. "Behavioral specialization in embodied evolutionary robotics: Why so Difficult," *Frontiers in Robotics and AI*, vol. 3, p. 38, July 2016.

[100] Gigliotta, O., Mirolli, M., and Nolfi, S. "Communication based dynamic role allocation in a group of homogeneous robots", *Natural Computing*, vol. 13(3), pp. 391-402, Sep. 2014.

[101] Gigliotta O. “Task Allocation in Evolved Communicating Homogeneous Robots: The Importance of Being Different”, *Trends in Practical Applications of Scalable Multi-Agent Systems, the PAAMS Collection*, pp. 181-190, Springer International Publishing, 2016.

[102] Gigliotta, O. “Equal but different: Task allocation in homogeneous communicating robots”, *Neurocomputing*. June 2017.

[103] Zavlanos, M. M., Spesivtsev, L., and Pappas, G. J., “A distributed auction algorithm for the assignment problem,” *IEEE Conf. on Decision Control*, pp. 1212–1217, Cancun, Mexico, Dec. 2008.

[104] Gerkey, B. P., and Mataric, M. J., “Sold!: auction methods for multirobot coordination,” *IEEE Trans. Robotics and Automation*, vol. 18, no. 5, pp. 758–768, Oct. 2002.

[105] Ratnieks, F. L., and Anderson C. “Task partitioning in insect societies.” *Insectes Sociaux*, vol. 46(2), pp. 95-108, May 1999.

[106] Vasconcelos, H. L., and Cherrett., J. M. “The effect of wilting on the selection of leaves by the leaf-cutting ant Atta laevigata.” *Entomologia Experimentalis et Applicata*, vol. 78, no. 2, pp. 215-220, 1996.

[107] Gorelick, R., and Bertram S. M. “Quantifying division of labor: borrowing tools from sociology, sociobiology, information theory, landscape ecology, and biogeography”. *Insectes Sociaux*, 2007.

[108] Goldsby H. J., Dornhaus A., Kerr B., and Ofria C. “Task-switching costs promote the evolution of division of labor and shifts in individuality”. *Proceedings of the National Academy of Sciences*. Aug. 2012.

[109] Simpson, C. “The evolutionary history of division of labour”. *Proceedings of the Royal Society of London B: Biological Sciences*. May 2011.

[110] Rueffler, C., Hermisson, J., and Wagner G. P. “Evolution of functional specialization and division of labor”. *Proceedings of the National Academy of Sciences*, vol. 109(6), pp. E326-335, Feb. 2012.

[111] Korte, B., and Vygen, J. “Combinatorial Optimization: Theory and Algorithms”, *Algorithmics and Combinatorics*, vol. 21(3), Springer, 2005.

[112] Hall, P. "On representatives of subsets." *Classic Papers in Combinatorics*. Birkhäuser Boston, pp. 58-62, 2009.

[113] Jones, C., and Mataric, M.J., “Adaptive division of labor in large-scale minimalist multi-robot systems”. *IEEE/RSJ International Conference in Intelligent Robots and Systems*, vol. 2, pp. 1969-1974, 2003.

[114] Smith, S. L., and Bullo, F. “Target assignment for robotic networks: Worst-case and stochastic performance in dense environments.” *IEEE Conference in Decision and Control*, pp. 3585-3590, 2007.

[115] Claes, D., Robbel, P., Oliehoek, F. A., Tuyls, K., Hennes, D., and Van der Hoek, W. "Effective approximations for multi-robot coordination in spatially distributed tasks", *International Conference on Autonomous Agents and Multiagent Systems*, pp. 881-890, 2015.

[116] Jain, D., Mosenlechner, L., and Beetz, M. "Equipping robot control programs with first-order probabilistic reasoning capabilities". *IEEE International Conference in Robotics and Automation, ICRA'09*. pp. 3626-3631, 2009.

[117] Lang, T., and Toussaint, M. "Relevance grounding for planning in relational domains," *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 736-751, Springer, Berlin, Heidelberg, Sep. 2009.

[118] Toussaint, M., Plath, N., Lang, T., and Jetchev, N. "Integrated motor control, planning, grasping and high-level reasoning in a blocks world using probabilistic inference", *IEEE International Conference in Robotics and Automation, (ICRA'10)*, pp. 385-391, May 2010.

[119] Yasuda, T., Kage, K., and Ohkura, K. "Response threshold-based task allocation in a reinforcement learning robotic swarm," *IEEE International Workshop in Computational Intelligence and Applications, (IWCIA'14)*, pp. 189-194, 2014.

[120] Wu, H., Li, H., Xiao, R., and Liu, J. "Modeling and simulation of dynamic ant colony's labor division for task allocation of UAV swarm", *Physica A: Statistical Mechanics and its Applications*, pp. 127-41, Feb. 2018.

[121] Murphy, R., and Burke, J. "The safe human–robot ratio," *Human–Robot Interactions in Future Military Operations*, pp. 51–70, CRC press, 2016.

[122] Chen, J. Y., and Barnes, M.J., "Human–agent teaming for multi-robot control: A review of human factors issues," *IEEE Transactions on Human-Machine Systems*, vol. 44(1), pp.13-29, 2014.

[123] Valero-Gomez, A., De La Puente, P., and Hernando, M. "Impact of two adjustable-autonomy models on the scalability of single-human/multiple-robot teams for exploration missions," *Human Factors*, vol. 53, pp. 703–716, 2011.

[124] Kidwell, B., Calhoun, G., Ruff, H., and Parasuraman, R. "Adaptable and adaptive automation for supervisory control of multiple autonomous vehicles," *Annual HFES Meeting*, pp. 428–432, 2012.

[125] Miller, C., and Parasuraman, R. "Designing for flexible interaction between humans and automation: Delegation interfaces for supervisory control," *Human Factors*, vol. 49, pp. 57–75, Feb. 2007.

[126] Fierro, R., and Lewis, F. L. "Control of a nonholonomic mobile robot: backstepping kinematics into dynamics", *IEEE Conference on Decision and Control*, vol. 4, pp. 3805-3810, 1995.

[127] Khalil, H. "Nonlinear systems", *Prentice-Hall*, New Jersey, 1996.

[128] Al-Buraiki, O., Payeur, P., and Castillo, Y. R. "Task switching for specialized mobile robots working in cooperative formation" In *Proc. of the IEEE International Symposium in Robotics and Intelligent Sensors* (IRIS), pp. 207-212, Tokyo, Japan, Dec. 2016.

[129] Al-Buraiki, O., Payeur, P., and Busiquia, H. M. "Automatic Task Selection from Targets Recognition for Swarm Mobile Robots with Specialized Agents", *MDPI Proceedings*. vol. 2 (3)-116, 2017.

[130] Al-Buraiki, O., and Payeur, P. "Task-Agent Assignation Based on Target Characteristics for a Swarm of Specialized Agents", *Annual IEEE International Systems Conference*, pp. 268-275, Orlando, USA, April 2019.

[131] Wu, W., Payeur, P., Al-Buraiki, O., and Ross, M. "Vision-Based Target Objects Recognition and Segmentation for Unmanned Systems Task Allocation". Proceedings of the *International Conference on Image Analysis and Recognition (ICIAR 2019)*, LNCS, vol. 11662, pp. 252-263, Springer Nature Switzerland, Aug. 2019.

[132] Bishop, C. M. "Pattern Recognition and Machine Learning" *Springer*, New York, 2016.

[133] Papoulis, A., and Pillai, S. U. "Probability, random variables, and stochastic processes". *Tata McGraw-Hill Education*, New York, 2002.

[134] Al-Buraiki, O., and Payeur, P. "Probabilistic Task Assignment for Specialized Multi-Agent Robotic Systems", in *Proc. of the IEEE International Symposium on Robotic and Sensors Environments*, pp. 43-49, Ottawa, ON, June 2019.

[135] "Overview" ROBOTIS e-Manual. 2020. [Online]. Available: http://emanual.robotis.com/docs/en/platform/turtlebot3/overview/ [Accessed: July, 25, 2020].

[136] "PC Setup." ROBOTIS e-Manual. 2020. [Online]. Available: http://emanual.robotis.com/docs/en/platform/turtlebot3/pc_setup/ [Accessed: July, 25, 2019].

[137] Cardoso, P., AL-Buraiki, O., and Payeur, P., "Experimental Implementation of Specialty-Bask probabilistic Task-Allocator on the Multi-Agent System using Turtlebot robots" *Technical Report - Sensing and Machine Vision for Automation and Robotic Intelligence Research Laboratory*. Ottawa, ON, February 2020.

[138] "See3CAM_130 - Technical Documents," e-con Systems, 2020. [Online]. Available: https://www.e-consystems.com/doc_13MP_autofocus_USB3_Camera.asp. [Accessed: July, 25, 2020].

[139] Yang, S. "Active Sensing for Collaborative Localization in Swarm Robotics." M.Sc. thesis, University of Ottawa, Ottawa, ON, 2020.

[140] Ross, M., Payeur, P., and Chartier. S. "Task Allocation for Heterogeneous Robots Using a Self-Organizing Contextual Map", in Proc. of the *IEEE International Symposium on Robotic and Sensors Environments* (ROSE), pp. 57-62, Ottawa, ON, June 2019.

[141] Brutschy, A., Pini, G., Pinciroli, C., Birattari, M., and Dorigo, M. "Self-organized task allocation to sequentially interdependent tasks in swarm robotics". *Autonomous Agents and Multi-Agent Systems*, pp. 101-125, 2014.

[142] Kulich, M., Faigl, J., and Přeučil, L. M. “On distance utility in the exploration task”. *IEEE International Conference on Robotics and Automation*, pp. 4455-4460, 2011.

[143] Al-Buraiki, O., Wu, W., and Payeur, P., “Probabilistic Allocation of Specialized Robots on Targets Detected Using Deep Learning Networks”, *Robotics,* 9(3), 54, 16 July 2020.

[144] Al-Buraiki, O., Payeur, P., “Specialized Agents Allocation in Autonomous Multi-Agent Unmanned Systems”. Presented in *Unmanned Canada Conference and Exhibition (UC20Remote),* November, 2020.

[145] Al-Buraiki, O., and Payeur, P. “Task Allocation in Multi-Robot Systems Based on the Suitability Level of the Individual Agents” Submitted to *IEEE International Conference in Robotics Automation, (ICRA21)*, Xi’an China.

[146] Al-Buraiki, O., Payeur, P., and Cardoso, P. “Task Allocation in Multi-Agent Unmanned Systems Based on Analog Suitability of Specialized Robots” In preparation for *IEEE Transactions on Robotics.*

Appendix A

5.3.C1.1 Numerical Example (Binary encoding case)

This example provides the mathematical details of computing the fitting probabilities of 5.3.C1 sample case, presented in Figure 5.1b and Table 5.3. This example is detailing the numerical processing of the proposed scheme, which shows how the probabilities of agents $R_1, R_2,$ and R_3 are obtained at the output starting from a target recognition confidence level of 0.86 at the input.

Step 0: System parameters and initialization:

Step 0.1: $\eta = 0.3$, initial value of MFT, (it also can be initialized by a human operator).

$\mathfrak{p} = 1$, $r_{task} = 2$ design parameters, embedded by the system designer.

Step1: Inputs:

Step 1.1: Target object recognition confidence level Eq. (4.8), and number of targets is $T = 14$;

$$\widehat{\boldsymbol{P}}_T = [0.86\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0]^t \in \mathcal{R}^{T\times 1}$$

Step 1.2: The agents' availability vector, $\boldsymbol{\vartheta}_{Av} \in \mathcal{R}^{a\times 1}, a = 20$ robot, is defined based on the current internal status of each robot, Eq. (4.12):

$$\boldsymbol{\vartheta}_{Av} = [1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ ]^t \in \mathcal{R}^{a\times 1}$$

Step 1.3: d_i the Euclidean distance Eq. (4.15) between the current location (x_i, y_i) of the robot R_i and the location of the estimated target (x_T, y_T), v_t target velocity and $v_{i_{max}}$ the maximum velocity of the robot, all are provided by the sensing system.

Step 1.4: The robotic agents attendance parameters are generated based on the logic that is given by Eq. (4.14), and used to evaluate the agents attendance, given by Eq. (4.13). These parameters are:

Υ, ϵ, and $\overline{\mathsf{T}}$

Detailed values are not provided here as attendance is not considered in this example (see step 2.5 below)

Step 2: Probabilistic Matching:

Step 2.1: The swarm individuals specialization encoding, $\boldsymbol{S}_i$: $i = 1, 2, \dots, a$; $\boldsymbol{S}_i \in \mathcal{R}^{1\times T}$ is defined in Table (5.1), for $a = 20$.

Step 2.2: Computing the confidence levels of the current task's class detection that are leveraged by the swarm's individuals, given by Eq. (4.7):

$$\hat{\varphi}_{R_i} = \boldsymbol{S}_i \widehat{\boldsymbol{P}}_T$$

$$\hat{\varphi}_{R_1} = \boldsymbol{S_1} * \widehat{\boldsymbol{P}}_T = 0.86;\ \hat{\varphi}_{R_2} = \boldsymbol{S_2} * \widehat{\boldsymbol{P}}_T = 0.86$$

$$\hat{\varphi}_{R_3} = \boldsymbol{S_3} * \widehat{\boldsymbol{P}}_T = 0.86;\ \hat{\varphi}_{R_4} = \boldsymbol{S_4} * \widehat{\boldsymbol{P}}_T = 0.86$$

And, for Robots' IDs $i = 5{:}\,20 \rightarrow \hat{\varphi}_{R_i} = 0$ because these robots are not specialized in this detected task, T_{F_1}, as indicated in T_{F_1} column of Table 5.1, then as a result they cannot leverage the input from $\widehat{\boldsymbol{P}}_T$ and $\boldsymbol{S}_i \widehat{\boldsymbol{P}}_T = 0$.

Step 2.3: Computing the maximum expected collective score of the swarm's individuals, given by Eq. (4.10):

$$\varphi_{R_i} = \boldsymbol{S}_i \boldsymbol{p_{max}}$$

$\boldsymbol{p_{max}}$ is the class conditional probability of the feature X_k in its predefined class C_k, which is usually selected to equal 1 (i.e. a perfectly reliable recognition) for every predefined feature in this test case, therefore Eq. (4.11) resulted in:

$$\boldsymbol{p_{max}} = [1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1]^{\boldsymbol{t}} \in \mathcal{R}^{T\times 1}$$

And $\boldsymbol{S}_i$'s are defined in Table 5.1, these could result in:

$$\varphi_{R_1} = \boldsymbol{S}_1\boldsymbol{p_{max}} = 2;\ \varphi_{R_2} = \boldsymbol{S}_2\boldsymbol{p_{max}} = 2;\ \varphi_{R_3} = \boldsymbol{S}_3\boldsymbol{p_{max}} = 2;$$
$$\varphi_{R_4} = \boldsymbol{S}_4\boldsymbol{p_{max}} = 3;\ \varphi_{R_5} = \boldsymbol{S}_5\boldsymbol{p_{max}} = 2;\ \varphi_{R_6} = \boldsymbol{S}_6\boldsymbol{p_{max}} = 2;$$
$$\varphi_{R_7} = \boldsymbol{S}_7\boldsymbol{p_{max}} = 2;\ \varphi_{R_8} = \boldsymbol{S}_8\boldsymbol{p_{max}} = 2;\ \varphi_{R_9} = \boldsymbol{S}_9\boldsymbol{p_{max}} = 2;$$
$$\varphi_{R_{10}} = \boldsymbol{S}_{10}\boldsymbol{p_{max}} = 2;\ \varphi_{R_{11}} = \boldsymbol{S}_{11}\boldsymbol{p_{max}} = 2;\ \varphi_{R_{12}} = \boldsymbol{S}_{12}\boldsymbol{p_{max}} = 2;$$
$$\varphi_{R_{13}} = \boldsymbol{S}_{13}\boldsymbol{p_{max}} = 2;\ \varphi_{R_{14}} = \boldsymbol{S}_{14}\boldsymbol{p_{max}} = 2;\ \varphi_{R_{15}} = \boldsymbol{S}_{15}\boldsymbol{p_{max}} = 2;$$
$$\varphi_{R_{16}} = \boldsymbol{S}_{16}\boldsymbol{p_{max}} = 2;\ \varphi_{R_{17}} = \boldsymbol{S}_{17}\boldsymbol{p_{max}} = 2;\ \varphi_{R_{18}} = \boldsymbol{S}_{18}\boldsymbol{p_{max}} = 3;$$
$$\varphi_{R_{19}} = \boldsymbol{S}_{19}\boldsymbol{p_{max}} = 2;\ \varphi_{R_{20}} = \boldsymbol{S}_{20}\boldsymbol{p_{max}} = 1;$$

Step 2.4: Computing the swarm's specialty matching probabilistic matrix, $\boldsymbol{Q} \in \mathcal{R}^{20\times 20}$, defined in Eq. (4.9):

$$\boldsymbol{Q} = \boldsymbol{dig}\left[\frac{0.86}{2}\ \frac{0.86}{2}\ \frac{0.86}{2}\ \frac{0.86}{3}\ \frac{0}{2}\ \frac{0}{2}\ \frac{0}{2}\ \frac{0}{2}\ \frac{0}{2}\ \frac{0}{2}\ \frac{0}{2}\ \frac{0}{2}\ \frac{0}{2}\ \frac{0}{2}\ \frac{0}{2}\ \frac{0}{2}\ \frac{0}{2}\ \frac{0}{2}\ \frac{0}{3}\ \frac{0}{2}\ \frac{0}{1}\right]$$
$$= \boldsymbol{dig}[0.43\ \ 0.43\ \ 0.43\ 0.29\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0]$$

Step 2.5: The agents' attendance matrix, $\boldsymbol{\vartheta}_{A_{tt}} \in \mathcal{R}^{20\times 20}$, is defined in Eqs. (4.13) and (4.14), the agents' attendance does not contribute in the computation of the task-agent matching probability, Eq. (4.16), because $\mathfrak{p} = 1$.

Step 2.6: Computing the swarm's a specialty fitting probabilities vector, Eq. (4.16):

$$\boldsymbol{\Psi} = (\mathfrak{p}\mathbf{Q} + (1-\mathfrak{p})\boldsymbol{\vartheta}_{Att})\boldsymbol{\vartheta}_{Av}$$
$$\Rightarrow \boldsymbol{\Psi} = ((1)\mathbf{Q} + (1-1)\boldsymbol{\vartheta}_{Att})\boldsymbol{\vartheta}_{Av}$$
$$\Rightarrow \boldsymbol{\Psi} = [0.43\ \ 0.43\ \ 0.43\ \ 0.29\ \ 0\ \ 0\ \ 0\ \ \cdots\ \ \cdots\ \ \cdots\ \ 0]^t$$

Step 3: Qualified responders coordination:

Step 3.1: Coordinating the qualified agents that achieve the MFT, η , Eq. (4.18):

$$\boldsymbol{\Psi}_{MFT} = [0.43\ \ 0.43\ \ 0.43\ \ 0\ \ 0\ \ 0\ \ 0\ \ \cdots\ \cdots\ \cdots\ \ 0]^t$$

Step 3.2a: Selecting the **most qualified** agent that achieve the MFT, Eq. (4.20): In this case, max operator extracts the ID of the agent with the highest fitting probability. If more than one agent share the same fitting score and the latter is maximum then max operator extracts the ID of the first highest probability stored in, $\boldsymbol{\Psi}_{MFT}$, vector:

$$\emptyset_{BEST\ RESPONDER\ INDEX=[\ R_1\ ]}$$

*this case is considered in this example (Table 5.3)

Step 3.2b: Selecting **all qualified** agents that achieve the MFT, Eq. (4.18): In this case, an iterative loop runs inside the vector, $\boldsymbol{\Psi}_{MFT}$, in Eq. (4.18) and extracts the IDs of all agents with fitting probabilities that are higher than η:

$$\emptyset_{BEST\ RESPONDERS\ INDEXES=[\ R_1\ \ R_2\ \ R_3]}$$

*this case is provided here just for illustrating the case when the application requires that all of the qualified agents need to be assigned.

These results are indicated in Table 5.3, columns 4 and 8 from the left of binary encoding case, it is also shown in Figure 5.1b.

Step 4: Repeating the process:

The process is automatically repeated starting from step 1, upon a new detected task with updated $\widehat{\boldsymbol{P}}_T$ and $\boldsymbol{\vartheta}_{Av}$.

5.3.C1.2 Numerical Example (Modulated encoding case)

This example provides the mathematical details of computing the fitting probabilities of 5.3C1 sample case, presented in Figure 5.1c and Table 5.3. This example is detailing the numerical processing of the proposed scheme, which shows how the probabilities of agents R_1, R_3 and R_4 are obtained at the output starting from a target recognition confidence level of 0.86 at the input.

Step 0: System parameters and initialization:

Step 0.1: $\eta = 0.3$, initial value of MFT, (it also can be initialized by a human operator).

$\mathfrak{p} = 1$, $r_{task} = 2$ design parameters, embedded by the system designer.

Step1: Inputs:

Step 1.1: Target object recognition confidence level Eq. (4.8), and number of targets is $T = 14$;

$$\widehat{\boldsymbol{P}}_T = [0.86\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]^t \in \mathcal{R}^{T\times 1}$$

Step 1.2: The agents' availability vector, $\boldsymbol{\vartheta}_{Av} \in \mathcal{R}^{a\times 1}, a = 20$ robot, is defined based on the current internal status of each robot, Eq. (4.12):

$$\boldsymbol{\vartheta}_{Av} = [1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1]^t \in \mathcal{R}^{a\times 1}$$

Step 1.3: d_i the Euclidean distance Eq. (4.15) between the current location (x_i, y_i) of the robot R_i and the location of the estimated target (x_T, y_T), v_t target velocity and $v_{i_{max}}$ the maximum velocity of the robot, all are provided by the sensing system.

Step 1.4: The robotic agents attendance parameters are generated based on the logic that is given by Eq. (4.14), and used to evaluate the agents attendance, given by Eq. (4.13). These parameters are:

Υ, ϵ, and $\overline{\mathsf{T}}$

Detailed values are not provided here as attendance is not considered in this example (see step 2.5 below)

Step 2: Probabilistic Matching:

Step 2.1: The swarm individuals specialization encoding, $\boldsymbol{S}_i$: $i = 1, 2, \ldots, a$; $\boldsymbol{S}_i \in \mathcal{R}^{1\times T}$is defined in Table (5.2), for $a = 20$.

Step 2.2: Computing the confidence levels of the current task's class detection that are leveraged by the swarm's individuals, specializations encoding, Table 5.2, given by Eq. (4.7):

$$\hat{\varphi}_{R_i} = \boldsymbol{S}_i \widehat{\boldsymbol{P}}_T$$

$$\hat{\varphi}_{R_1} = \boldsymbol{S_1} * \widehat{\boldsymbol{P}}_T = 0.69;\ \hat{\varphi}_{R_2} = \boldsymbol{S_2} * \widehat{\boldsymbol{P}}_T = 0.26$$

$$\hat{\varphi}_{R_3} = \boldsymbol{S_3} * \widehat{\boldsymbol{P}}_T = 0.52; \hat{\varphi}_{R_4} = \boldsymbol{S_4} * \widehat{\boldsymbol{P}}_T = 0.34$$

And, for Robots' IDs $i = 5{:}20 \rightarrow \hat{\varphi}_{R_i} = 0$ because these robots are not specialized in this detected task, T_{F_1}, as indicated in T_{F_1} column of Table 5.2, then as a result they cannot leverage the input from $\widehat{\boldsymbol{P}}_T$ and $\boldsymbol{S}_i\widehat{\boldsymbol{P}}_T = 0$.

Step 2.3: Computing the maximum expected collective score of the swarm's individuals, given by Eq. (4.10):

$$\varphi_{R_i} = \boldsymbol{S}_i\boldsymbol{p_{max}}$$

$\boldsymbol{p_{max}}$ is the class conditional probability of the feature X_k in its predefined class C_k, which is usually selected to equal 1 (i.e. a perfectly reliable recognition) for every predefined feature in this test case, therefore Eq. (4.11) resulted in:

$$\boldsymbol{p_{max}} = [1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1]^{\boldsymbol{t}} \in \mathcal{R}^{T\times 1}$$

And $\boldsymbol{S}_i$'s are defined in Table 5.2, these could result in:

$$\varphi_{R_1} = \boldsymbol{S}_1\boldsymbol{p_{max}} = 1;\ \varphi_{R_2} = \boldsymbol{S}_2\boldsymbol{p_{max}} = 1;\ \varphi_{R_3} = \boldsymbol{S}_3\boldsymbol{p_{max}} = 1;$$
$$\varphi_{R_4} = \boldsymbol{S}_4\boldsymbol{p_{max}} = 1;\ \varphi_{R_5} = \boldsymbol{S}_5\boldsymbol{p_{max}} = 1;\ \varphi_{R_6} = \boldsymbol{S}_6\boldsymbol{p_{max}} = 1;$$
$$\varphi_{R_7} = \boldsymbol{S}_7\boldsymbol{p_{max}} = 1;\ \varphi_{R_8} = \boldsymbol{S}_8\boldsymbol{p_{max}} = 1;\ \varphi_{R_9} = \boldsymbol{S}_9\boldsymbol{p_{max}} = 1;$$
$$\varphi_{R_{10}} = \boldsymbol{S}_{10}\boldsymbol{p_{max}} = 1;\ \varphi_{R_{11}} = \boldsymbol{S}_{11}\boldsymbol{p_{max}} = 1;\ \varphi_{R_{12}} = \boldsymbol{S}_{12}\boldsymbol{p_{max}} = 1;$$
$$\varphi_{R_{13}} = \boldsymbol{S}_{13}\boldsymbol{p_{max}} = 1;\ \varphi_{R_{14}} = \boldsymbol{S}_{14}\boldsymbol{p_{max}} = 1;\ \varphi_{R_{15}} = \boldsymbol{S}_{15}\boldsymbol{p_{max}} = 1;$$
$$\varphi_{R_{16}} = \boldsymbol{S}_{16}\boldsymbol{p_{max}} = 1;\ \varphi_{R_{17}} = \boldsymbol{S}_{17}\boldsymbol{p_{max}} = 1;\ \varphi_{R_{18}} = \boldsymbol{S}_{18}\boldsymbol{p_{max}} = 1;$$
$$\varphi_{R_{19}} = \boldsymbol{S}_{19}\boldsymbol{p_{max}} = 1;\ \varphi_{R_{20}} = \boldsymbol{S}_{20}\boldsymbol{p_{max}} = 1;$$

Step 2.4: Computing the swarm's specialty matching probabilistic matrix, $\boldsymbol{Q} \in \mathcal{R}^{20\times 20}$, defined in Eq. (4.9):

$$\boldsymbol{Q} = \boldsymbol{dig}\left[\frac{0.69}{1}\ \frac{0.26}{1}\ \frac{0.52}{1}\ \frac{0.34}{1}\ \frac{0}{1}\ \frac{0}{1}\ \frac{0}{1}\ \frac{0}{1}\ \frac{0}{1}\ \frac{0}{1}\ \frac{0}{1}\ \frac{0}{1}\ \frac{0}{1}\ \frac{0}{1}\ \frac{0}{1}\ \frac{0}{1}\ \frac{0}{1}\ \frac{0}{1}\ \frac{0}{1}\ \frac{0}{1}\right]$$
$$= \boldsymbol{dig}[0.69\quad 0.26\quad 0.52\ 0.34\quad 0\quad 0\quad 0\quad 0\quad 0\quad 0\quad 0\quad 0\quad 0\quad 0\quad 0\quad 0\quad 0\quad 0\ 0\ 0]$$

Step 2.5: The agents' attendance matrix, $\boldsymbol{\vartheta}_{A_{tt}} \in \mathcal{R}^{20\times 20}$, is defined in Eqs. (4.13) and (4.14), the agents' attendance does not contribute in the computation of the task-agent matching probability, Eq. (4.16), because $\text{ɒ} = 1$.

Step 2.6: Computing the swarm's a specialty fitting probabilities vector, Eq. (4.16):

$$\boldsymbol{\Psi} = (\text{ɒ}\mathbf{Q} + (1-\text{ɒ})\boldsymbol{\vartheta}_{Att})\boldsymbol{\vartheta}_{Av}$$
$$\Rightarrow \boldsymbol{\Psi} = ((1)\mathbf{Q} + (1-1)\boldsymbol{\vartheta}_{Att})\boldsymbol{\vartheta}_{Av}$$
$$\Rightarrow \boldsymbol{\Psi} = [0.69\quad 0.26\quad 0.52\quad 0.34\quad 0\quad 0\quad 0\quad \cdots\quad \cdots\quad \cdots\quad 0]^t$$

Step 3: Qualified responders coordination:

Step 3.1: Coordinating the qualified agents that achieve the MFT, η , Eq. (4.18):

$$\boldsymbol{\Psi}_{MFT} = [0.69\quad 0\quad 0.52\quad 0.34\quad 0\quad 0\quad 0\quad \cdots\ \cdots\ \cdots\quad 0]^t$$

Step 3.2a: Selecting the **most qualified** agent that achieves the MFT, Eq. (4.20): In this case, max operator extracts the ID of the agent with the highest fitting probability in, $\boldsymbol{\Psi}_{MFT}$, vector:

$$\emptyset_{BEST\ RESPONDER\ INDEX=[\ R_1\ ]}$$

*this case is considered in this example (Table 5.3)

Step 3.2b: Selecting **all qualified** agents that achieve the MFT, Eq. (4.18): In this case, an iterative loop runs inside the vector, $\boldsymbol{\Psi}_{MFT}$, in Eq. (4.18) and extracts the IDs of all agents with fitting probabilities that are higher than η:

$$\emptyset_{BEST\ RESPONDERS\ INDEXES=[\ R_1\ R_2\ R_3]}$$

*this case is provided here just for illustrating the case that when the application requires that all the qualified agents need to be assigned.

These results are indicated in Table 5.3, columns 5 and 9 from the left of modulated encoding case, it is also shown in Figure 5.1c.

Step 4: Repeating the process:

The process is automatically repeated starting from step 1, upon a new detected task with updated $\widehat{\boldsymbol{P}}_T$ and $\boldsymbol{\vartheta}_{Av}$.

5.3.C3.1 Numerical Example (Binary encoding case)

This example provides the mathematical details of computing the fitting probabilities of 5.3.C.3, presented in Figure 5.4d and Table 5.6, (third column from the right). This example is detailing the numerical processing of the proposed scheme, which shows how the agents, R_{12}, is assigned to respond to the task, T_{E_2}, when the agent, R_{11}, is not available "withdrawn".

Step 0: System parameters and initialization:

Step 0.1: $\eta = 0.3$, initial value of MFT, (it also can be initialized by a human operator).

$\mathfrak{p} = 1$, $r_{task} = 2$ design parameters, embedded by the system designer.

Step1: Inputs:

Step 1.1: Target object recognition confidence level Eq. (4.8), and number of targets is $T = 14$;

$$\widehat{\boldsymbol{P}}_T = [0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0.77\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0]^t\ \in \mathcal{R}^{T\times 1}$$

Step 1.2: The agents' availability vector, $\boldsymbol{\vartheta}_{Av} \in \mathcal{R}^{a\times 1}, a = 20$ robot, is defined based on the current internal status of each robot, Eq. (4.12):

$$\boldsymbol{\vartheta}_{Av} = [1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 0\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ ]^t \in \mathcal{R}^{a\times 1}$$

Step 1.3: d_i the Euclidean distance Eq. (4.15) between the current location (x_i, y_i) of the robot R_i and the location of the estimated target (x_T, y_T), v_t target velocity and $v_{i_{max}}$ the maximum velocity of the robot, all are provided by the sensing system.

Step 1.4: The robotic agents attendance parameters are generated based on the logic that is given by Eq. (4.14), and used to evaluate the agents attendance, given by Eq. (4.13). These parameters are:

Υ, ϵ, and $\overline{\mathsf{T}}$

Detailed values are not provided here as attendance is not considered in this example (see step 2.5 below)

Step 2: Probabilistic Matching:

Step 2.1: The swarm individuals specialization encoding, $\boldsymbol{S}_i$: $i = 1, 2, \dots, a$; $\boldsymbol{S}_i \in \mathcal{R}^{1\times T}$is defined in Table (5.1), for $a = 20$.

Step 2.2: Computing the confidence levels of the current task's class detection that are leveraged by the swarm's individuals, given by Eq. (4.7):

$$\hat{\varphi}_{R_i} = \boldsymbol{S}_i \widehat{\boldsymbol{P}}_T$$

For Robots' IDs $i = 1{:}\,10 \rightarrow \hat{\varphi}_{R_i} = 0$ because these robots are not specialized in this detected task, T_{E_2}, as indicated in T_{E_2}column of Table 5.1, then as a result they cannot leverage the input from $\widehat{\boldsymbol{P}}_T$ and $\boldsymbol{S}_i \widehat{\boldsymbol{P}}_T = 0$.

$$\hat{\varphi}_{R_{10}} = \boldsymbol{S}_{11} * \widehat{\boldsymbol{P}}_T = 0.77;\ \hat{\varphi}_{R_2} = \boldsymbol{S}_{12} * \widehat{\boldsymbol{P}}_T = 0.77;\ \hat{\varphi}_{R_{13}} = \boldsymbol{S}_{\boldsymbol{3}} * \widehat{\boldsymbol{P}}_T = 0.77;$$

And, for Robots' IDs $i = 14{:}\,20 \rightarrow \hat{\varphi}_{R_i} = 0$ because these robots are not specialized in this detected task, T_{E_2}, as indicated in T_{E_2}column of Table 5.1, then as a result they cannot leverage the input from $\widehat{\boldsymbol{P}}_T$ and $\boldsymbol{S}_i \widehat{\boldsymbol{P}}_T = 0$.

Step 2.3: Computing the maximum expected collective score of the swarm's individuals, given by Eq. (4.10):

$$\varphi_{R_i} = \boldsymbol{S}_i \boldsymbol{p}_{\boldsymbol{max}}$$

$\boldsymbol{p}_{\boldsymbol{max}}$ is the class conditional probability of the feature X_k in its predefined class C_k, which is usually selected to equal 1 (i.e. a perfectly reliable recognition) for every predefined feature in this test case, therefore Eq. (4.11) resulted in:

$$\boldsymbol{p}_{\boldsymbol{max}} = [1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1]^{\boldsymbol{t}} \in \mathcal{R}^{T\times 1}$$

And $\boldsymbol{S}_i$'s are defined in Table 5.1, these could result in:

$$\varphi_{R_1} = \boldsymbol{S}_1 \boldsymbol{p}_{\boldsymbol{max}} = 2;\ \varphi_{R_2} = \boldsymbol{S}_2 \boldsymbol{p}_{\boldsymbol{max}} = 2;\ \varphi_{R_3} = \boldsymbol{S}_3 \boldsymbol{p}_{\boldsymbol{max}} = 2;$$
$$\varphi_{R_4} = \boldsymbol{S}_4 \boldsymbol{p}_{\boldsymbol{max}} = 3;\ \varphi_{R_5} = \boldsymbol{S}_5 \boldsymbol{p}_{\boldsymbol{max}} = 2;\ \varphi_{R_6} = \boldsymbol{S}_6 \boldsymbol{p}_{\boldsymbol{max}} = 2;$$
$$\varphi_{R_7} = \boldsymbol{S}_7 \boldsymbol{p}_{\boldsymbol{max}} = 2;\ \varphi_{R_8} = \boldsymbol{S}_8 \boldsymbol{p}_{\boldsymbol{max}} = 2;\ \varphi_{R_9} = \boldsymbol{S}_9 \boldsymbol{p}_{\boldsymbol{max}} = 2;$$
$$\varphi_{R_{10}} = \boldsymbol{S}_{10} \boldsymbol{p}_{\boldsymbol{max}} = 2;\ \varphi_{R_{11}} = \boldsymbol{S}_{11} \boldsymbol{p}_{\boldsymbol{max}} = 2;\ \varphi_{R_{12}} = \boldsymbol{S}_{12} \boldsymbol{p}_{\boldsymbol{max}} = 2;$$
$$\varphi_{R_{13}} = \boldsymbol{S}_{13} \boldsymbol{p}_{\boldsymbol{max}} = 2;\ \varphi_{R_{14}} = \boldsymbol{S}_{14} \boldsymbol{p}_{\boldsymbol{max}} = 2;\ \varphi_{R_{15}} = \boldsymbol{S}_{15} \boldsymbol{p}_{\boldsymbol{max}} = 2;$$
$$\varphi_{R_{16}} = \boldsymbol{S}_{16} \boldsymbol{p}_{\boldsymbol{max}} = 2;\ \varphi_{R_{17}} = \boldsymbol{S}_{17} \boldsymbol{p}_{\boldsymbol{max}} = 2;\ \varphi_{R_{18}} = \boldsymbol{S}_{18} \boldsymbol{p}_{\boldsymbol{max}} = 3;$$
$$\varphi_{R_{19}} = \boldsymbol{S}_{19} \boldsymbol{p}_{\boldsymbol{max}} = 2;\ \varphi_{R_{20}} = \boldsymbol{S}_{20} \boldsymbol{p}_{\boldsymbol{max}} = 1;$$

Step 2.4: Computing the swarm's specialty matching probabilistic matrix, $\boldsymbol{Q} \in \mathcal{R}^{20\times 20}$, defined in Eq. (4.9):

$$\boldsymbol{Q} = \boldsymbol{dig}\left[\frac{0}{2}\ \frac{0}{2}\ \frac{0}{2}\ \frac{0}{3}\ \frac{0}{2}\ \frac{0}{2}\ \frac{0}{2}\ \frac{0}{2}\ \frac{0}{2}\ \frac{0}{2}\ \frac{0.77}{2}\ \frac{0.77}{2}\ \frac{0.77}{2}\ \frac{0}{2}\ \frac{0}{2}\ \frac{0}{2}\ \frac{0}{2}\ \frac{0}{3}\ \frac{0}{2}\ \frac{0}{1}\right]$$
$$= \boldsymbol{dig}[0\ \ 0\ \ 0\ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ 0.38\ \ 0.38\ \ 0.38\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ 0\ 0]$$

Step 2.5: The agents' attendance matrix, $\boldsymbol{\vartheta}_{A_{tt}} \in \mathcal{R}^{20\times 20}$, is defined in Eqs. (4.13) and (4.14), the agents' attendance does not contribute in the computation of the task-agent matching probability, Eq. (4.16), because $\mathfrak{p} = 1$.

Step 2.6: Computing the swarm's a specialty fitting probabilities vector, Eq. (4.16):

$$\boldsymbol{\Psi} = (\mathfrak{p}\mathbf{Q} + (1-\mathfrak{p})\boldsymbol{\vartheta}_{Att})\boldsymbol{\vartheta}_{Av}$$
$$\Rightarrow \boldsymbol{\Psi} = ((1)\mathbf{Q} + (1-1)\boldsymbol{\vartheta}_{Att})\boldsymbol{\vartheta}_{Av}$$
$$\Rightarrow \boldsymbol{\Psi} = [0\ \ 0\ \ 0\ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ 0.38\ \ 0.38\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ 0\ 0]^t$$

Step 3: Qualified responders coordination:

Step 3.1: Coordinating the qualified agents that achieve the MFT, η , Eq. (4.18):

$$\boldsymbol{\Psi}_{MFT} = [0\ \ 0\ \ 0\ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ 0.38\ \ 0.38\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ 0\ 0]^t$$

Step 3.2a: Selecting the **most qualified** agent that achieve the MFT, Eq. (4.20): In this case, max operator extracts the ID of the agent with the highest fitting probability. If more than one agent share the same fitting score and the

latter is maximum then max operator extracts the ID of the agent that has first highest probability stored in, $\boldsymbol{\Psi}_{MFT}$, vector:

$$\emptyset_{BEST\ RESPONDER\ INDEX=[\ R_{12}\ ]}$$

*this case is considered in this example (Table 5.6)

Step 3.2b: Selecting **all qualified** agents that achieve the MFT, Eq. (4.18): In this case, an iterative loop runs inside the vector, $\boldsymbol{\Psi}_{MFT}$, in Eq. (4.18) and extracts the IDs of all agents with fitting probabilities that are higher than η:

$$\emptyset_{BEST\ RESPONDERS\ INDEXES=[\ R_{12}\ \ R_{13}]}$$

*this case is provided here just for illustrating the case that when the application requires that all the qualified agents need to be assigned.

These results are indicated in Table 5.6, of binary encoding case (third column from the right), its also shown in Figure 5.4d.

Step 4: Repeating the process:

The process is automatically repeated starting from step 1, upon a new detected task with updated $\widehat{\boldsymbol{P}}_T$ and $\boldsymbol{\vartheta}_{Av}$.

5.3.C3.2 Numerical Example (Modulated encoding case)

This example provides the mathematical details of computing the fitting probabilities of 5.3.C3, presented in Figure 5.5d and Table 5.6 (second column from the right). This example is detailing the numerical processing of the proposed scheme, which shows how the agents, R_{12}, is assigned to the task, T_{E_2}, when the most specialized agent, R_{13}, is not available "withdrawn".

Step 0: System parameters and initialization:

Step 0.1: $\eta = 0.3$, initial value of MFT, (it also can be initialized by a human operator).

$\mathfrak{p} = 1$, $r_{task} = 2$ design parameters, embedded by the system designer.

Step1: Inputs:

Step 1.1: Target object recognition confidence level Eq. (4.8), and number of targets is $T = 14$;

$$\widehat{\boldsymbol{P}}_T = [0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0.77\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0]^t \in \mathcal{R}^{T\times 1}$$

Step 1.2: The agents' availability vector, $\boldsymbol{\vartheta}_{Av} \in \mathcal{R}^{a\times 1}, a = 20$ robot, is defined based on the current internal status of each robot, Eq. (4.12):

$$\boldsymbol{\vartheta}_{Av} = [1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 0\ 1\ 1\ 1\ 1\ 1\ 1\ ]^t \in \mathcal{R}^{a\times 1}$$

Step 1.3: d_i the Euclidean distance Eq. (4.15) between the current location (x_i, y_i) of the robot R_i and the location of the estimated target (x_T, y_T), v_t target velocity and $v_{i_{max}}$ the maximum velocity of the robot, all are provided by the sensing system.

Step 1.4: The robotic agents attendance parameters are generated based on the logic that is given by Eq. (4.14), and used to evaluate the agents attendance, given by Eq. (4.13). These parameters are:

Υ, ϵ, and $\overline{\mathsf{T}}$

Detailed values are not provided here as attendance is not considered in this example (see step 2.5 below)

Step 2: Probabilistic Matching:

Step 2.1: The swarm individuals specialization encoding, $\boldsymbol{S}_i$: $i = 1, 2, \dots, a$; $\boldsymbol{S}_i \in \mathcal{R}^{1\times T}$is defined in Table (5.2), for $a = 20$.

Step 2.2: Computing the confidence levels of the current task's class detection that are leveraged by the swarm's individuals, given by Eq. (4.7):

$$\hat{\varphi}_{R_i} = \boldsymbol{S}_i\widehat{\boldsymbol{P}}_T$$

For Robots' IDs $i = 1{:}\,10 \rightarrow \hat{\varphi}_{R_i} = 0$ because these robots are not specialized in this detected task, T_{E_2}, as indicated in T_{E_2}column of Table 5.2, then as a result they cannot leverage the input from $\widehat{\boldsymbol{P}}_T$ and $\boldsymbol{S}_i\widehat{\boldsymbol{P}}_T = 0$.

$$\hat{\varphi}_{R_{10}} = \boldsymbol{S}_{11} * \widehat{\boldsymbol{P}}_T = 0.23;\ \hat{\varphi}_{R_2} = \boldsymbol{S}_{12} * \widehat{\boldsymbol{P}}_T = 0.48;\ \hat{\varphi}_{R_{13}} = \boldsymbol{S_3} * \widehat{\boldsymbol{P}}_T = 0.62;$$

And, for Robots' IDs $i = 14{:}\,20 \rightarrow \hat{\varphi}_{R_i} = 0$ because these robots are not specialized in this detected task, T_{E_2}, as indicated in T_{E_2}column of Table 5.2, then as a result they cannot leverage the input from $\widehat{\boldsymbol{P}}_T$ and $\boldsymbol{S}_i\widehat{\boldsymbol{P}}_T = 0$.

Step 2.3: Computing the maximum expected collective score of the swarm's individuals, given by Eq. (4.10):

$$\varphi_{R_i} = \boldsymbol{S}_i\boldsymbol{p_{max}}$$

$\boldsymbol{p_{max}}$ is the class conditional probability of the feature X_k in its predefined class C_k, which is usually selected to equal 1 (i.e. a perfectly reliable recognition) for every predefined feature in this test case, therefore Eq. (4.11) resulted in:

$$\boldsymbol{p_{max}} = [1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1]^{\boldsymbol{t}} \in \mathcal{R}^{T\times 1}$$

And $\boldsymbol{S}_i$'s are defined in Table 5.2, these could result in:

$$\varphi_{R_1} = \boldsymbol{S}_1\boldsymbol{p_{max}} = 1;\ \varphi_{R_2} = \boldsymbol{S}_2\boldsymbol{p_{max}} = 1;\ \varphi_{R_3} = \boldsymbol{S}_3\boldsymbol{p_{max}} = 1;$$
$$\varphi_{R_4} = \boldsymbol{S}_4\boldsymbol{p_{max}} = 1;\ \varphi_{R_5} = \boldsymbol{S}_5\boldsymbol{p_{max}} = 1;\ \varphi_{R_6} = \boldsymbol{S}_6\boldsymbol{p_{max}} = 1;$$
$$\varphi_{R_7} = \boldsymbol{S}_7\boldsymbol{p_{max}} = 1;\ \varphi_{R_8} = \boldsymbol{S}_8\boldsymbol{p_{max}} = 1;\ \varphi_{R_9} = \boldsymbol{S}_9\boldsymbol{p_{max}} = 1;$$
$$\varphi_{R_{10}} = \boldsymbol{S}_{10}\boldsymbol{p_{max}} = 1;\ \varphi_{R_{11}} = \boldsymbol{S}_{11}\boldsymbol{p_{max}} = 1;\ \varphi_{R_{12}} = \boldsymbol{S}_{12}\boldsymbol{p_{max}} = 1;$$
$$\varphi_{R_{13}} = \boldsymbol{S}_{13}\boldsymbol{p_{max}} = 1;\ \varphi_{R_{14}} = \boldsymbol{S}_{14}\boldsymbol{p_{max}} = 1;\ \varphi_{R_{15}} = \boldsymbol{S}_{15}\boldsymbol{p_{max}} = 1;$$
$$\varphi_{R_{16}} = \boldsymbol{S}_{16}\boldsymbol{p_{max}} = 1;\ \varphi_{R_{17}} = \boldsymbol{S}_{17}\boldsymbol{p_{max}} = 1;\ \varphi_{R_{18}} = \boldsymbol{S}_{18}\boldsymbol{p_{max}} = 1;$$
$$\varphi_{R_{19}} = \boldsymbol{S}_{19}\boldsymbol{p_{max}} = 1;\ \varphi_{R_{20}} = \boldsymbol{S}_{20}\boldsymbol{p_{max}} = 1;$$

Step 2.4: Computing the swarm's specialty matching probabilistic matrix, $\boldsymbol{Q} \in \mathcal{R}^{20\times 20}$**,** defined in Eq. (4.9):

$$\boldsymbol{Q} = \boldsymbol{dig}[0\quad 0\quad 0\ 0\quad 0\quad 0\quad 0\quad 0\quad 0\quad 0\quad 0.23\quad 0.48\quad 0.62\quad 0\quad 0\quad 0\quad 0\quad 0\ 0\ 0]$$

Step 2.5: The agents' attendance matrix, $\boldsymbol{\vartheta}_{A_{tt}} \in \mathcal{R}^{20\times 20}$, is defined in Eqs. (4.13) and (4.14), the agents' attendance does not contribute in the computation of the task-agent matching probability, Eq. (4.16), because $\mathfrak{p} = 1$.

Step 2.6: Computing the swarm's a specialty fitting probabilities vector, Eq. (4.16):

$$\boldsymbol{\Psi} = \left(\mathfrak{p}\mathbf{Q} + (1-\mathfrak{p})\boldsymbol{\vartheta}_{Att}\right)\boldsymbol{\vartheta}_{Av}$$
$$\Rightarrow \boldsymbol{\Psi} = ((1)\mathbf{Q} + (1-1)\boldsymbol{\vartheta}_{Att})\boldsymbol{\vartheta}_{Av}$$
$$\Rightarrow \boldsymbol{\Psi} = [0\quad 0\quad 0\ 0\quad 0\quad 0\quad 0\quad 0\quad 0\quad 0\quad 0.23\quad 0.48\quad 0\quad 0\quad 0\quad 0\quad 0\quad 0\ 0\ 0]^t$$

Step 3: Qualified responders coordination:

Step 3.1: Coordinating the qualified agents that achieve the MFT, η , Eq. (4.18):

$$\boldsymbol{\Psi}_{MFT} = [0 \quad 0 \quad 0 \; 0 \quad 0 \quad 0 \; 0 \quad 0 \quad 0 \; 0 \quad 0 \quad 0 \; 0.48 \quad 0 \quad 0 \; 0 \quad 0 \quad 0 \; 0 \; 0]^t$$

Step 3.2a: Selecting the **most qualified** agent that achieves the MFT, Eq. (4.20): In this case, max operator extracts the ID of the agent with the highest fitting probability. If more than one agent share the same fitting score and the latter is maximum then max operator extracts the ID of the agent that has first highest probability in, $\boldsymbol{\Psi}_{MFT}$, vector:

$$\emptyset_{BEST\ RESPONDER\ INDEX=[\ R_{12}\ ]}$$

Step 3.2b: Selecting **all qualified** agents that achieve the MFT, Eq. (4.18): In this case, an iterative loop runs inside the vector, $\boldsymbol{\Psi}_{MFT}$, in Eq. (4.18) and extracts the IDs of all agents with fitting probabilities that are higher than η:

$$\emptyset_{BEST\ RESPONDERS\ INDEXES=[\ R_{12}]}$$

*this case is provided here just for illustrating the case that when the application requires that all the qualified agents need to be assigned.

These results are indicated in Table 5.6, (second column from the right) its also shown in Figure 5.5d.

Step 4: Repeating the process:

The process is automatically repeated starting from step 1, upon a new detected task with updated $\widehat{\boldsymbol{P}}_T$ and $\boldsymbol{\vartheta}_{Av}$.

5.3.C4. Numerical Example (Modulated encoding case)

This example provides the mathematical details of computing the fitting probabilities of 5.3C4 sample case, presented in Figure 5.7b and Table 5.14, (fourth column from the right). This example is detailing the numerical processing of the proposed scheme with activated attendance, which shows how the probabilities of agents R_1, R_1, R_3 and R_4 are obtained at the output starting from a target recognition confidence level of 0.86 at the input.

Step 0: System parameters and initialization:

Step 0.1: $\eta = 0.3$, initial value of MFT, (it also can be initialized by a human operator).

$\mathfrak{p} = 1 - (\eta - 0.1) = 0.71$, $r_{task} = 15\ km$ design parameters, embedded by the system designer.

Step1: Inputs:

Step 1.1: Target object recognition confidence level Eq. (4.8), and number of targets is $T = 14$;

$$\widehat{\boldsymbol{P}}_T = [0.86 \; 0 \; 0 \; 0 \; 0 \; 0 \; 0 \; 0 \; 0 \; 0 \; 0 \; 0 \; 0 \; 0]^t \in \mathcal{R}^{T\times 1}$$

Step 1.2: The agents' availability vector, $\boldsymbol{\vartheta}_{Av} \in \mathcal{R}^{a\times 1}, a = 20$ robot, is defined based on the current internal status of each robot, Eq. (4.12):

$$\boldsymbol{\vartheta}_{Av} = [1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ ]^t \in \mathcal{R}^{a\times 1}$$

Step 1.3: d_i the Euclidean distance Eq. (4.15) between the current location (x_i, y_i) of the robot R_i and the location of the estimated target (x_T, y_T), v_t target velocity and $v_{i_{max}}$ the maximum velocity of the robot, all are provided by the sensing system.

Step 1.4: The robotic agents attendance parameters are generated based on the logic that is given by Eq. (4.14), and used to evaluate the agents' attendance, given by Eq. (4.13). These parameters are introduced based on simulated sensor's data considering static target:

$$\epsilon = 0, \text{ and } \overline{\mathrm{T}} = 1.$$

Detailed values of Υ are provided (see step 2.5 below)

Step 2: Probabilistic Matching:

Step 2.1: The swarm individuals specialization encoding, $\boldsymbol{S}_i$: $i = 1, 2, \dots, a$; $\boldsymbol{S}_i \in \mathcal{R}^{1\times T}$is defined in Table (5.2), for $a = 20$.

Step 2.2: Computing the confidence levels of the current task's class detection that are leveraged by the swarm's individuals, specializations encoding, Table 5.2, given by Eq. (4.7):

$$\hat{\varphi}_{R_i} = \boldsymbol{S}_i \widehat{\boldsymbol{P}}_T$$

$$\hat{\varphi}_{R_1} = \boldsymbol{S_1} * \widehat{\boldsymbol{P}}_T = 0.69; \hat{\varphi}_{R_2} = \boldsymbol{S_2} * \widehat{\boldsymbol{P}}_T = 0.26$$

$$\hat{\varphi}_{R_3} = \boldsymbol{S_3} * \widehat{\boldsymbol{P}}_T = 0.52; \hat{\varphi}_{R_4} = \boldsymbol{S_4} * \widehat{\boldsymbol{P}}_T = 0.34$$

And, for Robots' IDs $i = 5{:}\,20 \rightarrow \hat{\varphi}_{R_i} = 0$ because these robots are not specialized in this detected task, T_{F_1}, as indicated in T_{F_1}column of Table 5.2, then as a result they cannot leverage the input from $\widehat{\boldsymbol{P}}_T$ and $\boldsymbol{S}_i\widehat{\boldsymbol{P}}_T = 0$.

Step 2.3: Computing the maximum expected collective score of the swarm's individuals, given by Eq. (4.10):

$$\varphi_{R_i} = \boldsymbol{S}_i \boldsymbol{p_{max}}$$

$\boldsymbol{p_{max}}$ is the class conditional probability of the feature X_k in its predefined class C_k, which is usually selected to equal 1 (i.e. a perfectly reliable recognition) for every predefined feature in this test case, therefore Eq. (4.11) resulted in:

$$\boldsymbol{p_{max}} = [1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1]^{\boldsymbol{t}} \in \mathcal{R}^{T\times 1}$$

And $\boldsymbol{S}_i$'s are defined in Table 5.2, these could result in:

$$\varphi_{R_1} = \boldsymbol{S}_1\boldsymbol{p_{max}} = 1;\ \varphi_{R_2} = \boldsymbol{S}_2\boldsymbol{p_{max}} = 1;\ \varphi_{R_3} = \boldsymbol{S}_3\boldsymbol{p_{max}} = 1;$$

$$\varphi_{R_4} = \boldsymbol{S}_4\boldsymbol{p_{max}} = 1;\ \varphi_{R_5} = \boldsymbol{S}_5\boldsymbol{p_{max}} = 1;\ \varphi_{R_6} = \boldsymbol{S}_6\boldsymbol{p_{max}} = 1;$$

$$\varphi_{R_7} = \boldsymbol{S}_7\boldsymbol{p_{max}} = 1; \varphi_{R_8} = \boldsymbol{S}_8\boldsymbol{p_{max}} = 1; \varphi_{R_9} = \boldsymbol{S}_9\boldsymbol{p_{max}} = 1;$$

$$\varphi_{R_{10}} = \boldsymbol{S}_{10}\boldsymbol{p_{max}} = 1; \varphi_{R_{11}} = \boldsymbol{S}_{11}\boldsymbol{p_{max}} = 1; \varphi_{R_{12}} = \boldsymbol{S}_{12}\boldsymbol{p_{max}} = 1;$$

$$\varphi_{R_{13}} = \boldsymbol{S}_{13}\boldsymbol{p_{max}} = 1; \varphi_{R_{14}} = \boldsymbol{S}_{14}\boldsymbol{p_{max}} = 1; \varphi_{R_{15}} = \boldsymbol{S}_{15}\boldsymbol{p_{max}} = 1;$$

$$\varphi_{R_{16}} = \boldsymbol{S}_{16}\boldsymbol{p_{max}} = 1; \varphi_{R_{17}} = \boldsymbol{S}_{17}\boldsymbol{p_{max}} = 1; \varphi_{R_{18}} = \boldsymbol{S}_{18}\boldsymbol{p_{max}} = 1;$$

$$\varphi_{R_{19}} = \boldsymbol{S}_{19}\boldsymbol{p_{max}} = 1; \varphi_{R_{20}} = \boldsymbol{S}_{20}\boldsymbol{p_{max}} = 1;$$

Step 2.4: Computing the swarm's specialty matching probabilistic matrix, $\boldsymbol{Q} \in \mathcal{R}^{20\times 20}$**,** defined in Eq. (4.9):

$$\boldsymbol{Q} = \boldsymbol{dig}\left[\frac{0.69}{1}\ \frac{0.26}{1}\ \frac{0.52}{1}\ \frac{0.34}{1}\ \frac{0}{1}\ \frac{0}{1}\ \frac{0}{1}\ \frac{0}{1}\ \frac{0}{1}\ \frac{0}{1}\ \frac{0}{1}\ \frac{0}{1}\ \frac{0}{1}\ \frac{0}{1}\ \frac{0}{1}\ \frac{0}{1}\ \frac{0}{1}\ \frac{0}{1}\ \frac{0}{1}\ \frac{0}{1}\ \frac{0}{1}\right]$$

$$= \boldsymbol{dig}[0.69\ \ 0.26\ \ 0.52\ 0.34\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0]$$

Step 2.5: The agents' attendance matrix, $\boldsymbol{\vartheta}_{A_{tt}} \in \mathcal{R}^{20\times 20}$, is defined in Eqs. (4.13) and (4.14), the agents' attendance contribution in boosting the task-agent matching probability, Eq. (4.16), weighted by $(1 - \mathfrak{p}) = 0.29$.

The distances d_i's between the current location of each robotic agent R_i and the target location are introduced based on simulated sensor's data considering using Eq. (4.15), $r_{task} = 15km$, given that all agents are available:

$$d_1 = 78 > r_{task} \rightarrow \Upsilon_1 = 1;$$
$$; d_2 = 14.1 < r_{task} \rightarrow \Upsilon_2 = 0\ ;$$
$$d_3 = 23.9 > r_{task} \rightarrow \Upsilon_3 = 1;$$
$$d_4 = 25.4 > r_{task} \rightarrow \Upsilon_4 = 1;$$

Then $\boldsymbol{\vartheta}_{A_{tt_i}}$ can be computed using Eq. (4.14)

$$\boldsymbol{\vartheta}_{A_{tt_1}} = \frac{1\left(0\left[\frac{v_1}{v_{max}}\right] + \left[\frac{15}{78}\right]^1\right)}{2^0} = 0.19;$$

$$\boldsymbol{\vartheta}_{A_{tt_2}} = \frac{1\left(0\left[\frac{v_2}{v_{max}}\right] + \left[\frac{15}{14.1}\right]^0\right)}{2^0} = 1;$$

$$\boldsymbol{\vartheta}_{A_{tt_3}} = \frac{1\left(0\left[\frac{v_3}{v_{max}}\right] + \left[\frac{15}{23.9}\right]^1\right)}{2^0} = 0.63;$$

$$\boldsymbol{\vartheta}_{A_{tt_4}} = \frac{1\left(0\left[\frac{v_4}{v_{max}}\right] + \left[\frac{15}{25.4}\right]^1\right)}{2^0} = 0.59;$$

For $\boldsymbol{\vartheta}_{A_{tt_i}}: i = 5{:}\,20$, these values are not calculated here in this example in the purpose to avoid making this example very long, because these agents are not specialized in this task and these valued are not of our interest here in this example, but their values are brought from the simulated trial.

$$\boldsymbol{\vartheta}_{A_{tt}} = dig[0.19\ \ 1\ \ 0.63\ \ 0.59\ \ 0.19\ \ 0.19\ \ 0.24\ \ 0.19\ \ 0.18\ \ 0.17\ \ 0.20\ \ 0.17\ \ 0.19\ \ 0.17\ \ 0.12\ \ 0.25\ \ 0.24\ \ 0.27\ \ 0.13\ \ 0.12]$$

Step 2.6: Computing the swarm's a specialty fitting probabilities vector, Eq. (4.16):

$$\boldsymbol{\Psi} = (\mathfrak{p}\mathbf{Q} + (1-\mathfrak{p})\boldsymbol{\vartheta}_{Att})\boldsymbol{\vartheta}_{Av}$$
$$\Rightarrow \boldsymbol{\Psi} = ((1)\mathbf{Q} + (1-1)\boldsymbol{\vartheta}_{Att})\boldsymbol{\vartheta}_{Av}$$
$$\Rightarrow \boldsymbol{\Psi} = [0.53 \quad 0.47 \quad 0.54 \quad 0.41 \quad 0 \quad 0 \quad 0 \quad \cdots \quad \cdots \quad \cdots \quad 0]^t$$

Step 3: Qualified responders coordination:

Step 3.1: Coordinating the qualified agents that achieve the MFT, η , Eq. (4.18):

$$\boldsymbol{\Psi}_{MFT} = [0.53 \quad 0.47 \quad 0.54 \quad 0.41 \quad 0 \quad 0 \quad 0 \quad \cdots \cdots \quad \cdots \quad 0]^t$$

Step 3.2a: Selecting the **most qualified** agent that achieves the MFT, Eq. (4.20): In this case, max operator extracts the ID of the agent with the highest fitting probability in, $\boldsymbol{\Psi}_{MFT}$, vector:

$$\emptyset_{BEST\ RESPONDER\ INDEX=[\ R_3\ ]}$$

Step 3.2b: Selecting **all qualified** agents that achieve the MFT, Eq. (4.18): In this case, an iterative loop runs inside the vector, $\boldsymbol{\Psi}_{MFT}$**,** in Eq. (4.18) and extracts the IDs of all agents with fitting probabilities that are higher than η:

$$\emptyset_{BEST\ RESPONDERS\ INDEXES=[\ R_1\ \ R_2\ \ R_3\ \ R_4]}$$

*this case is provided here just for illustrating the case that when the application requires that all the qualified agents need to be assigned.

These results are indicated in Table 5.14, (fourth column from the right) , its also shown in Figure 5.7.b.

Step 4: Repeating the process:

The process is automatically repeated starting from step 1, upon a new detected task with updated $\widehat{\boldsymbol{P}}_T$ and $\boldsymbol{\vartheta}_{Av}$.

5.3.C5. Numerical Example (Sequential Tasks Allocation):

This example provides the logical details of computing the fitting probabilities of 5.3.C5 sample case, presented in Table 5.17. This example is detailing the procedure of the sequential task allocation process when more than one task are detected in the same time, which shows how multiple agents are sequentially assigned to multiple tasks, the latter being detected simultaneously. In this example, the logic is detailed whereas the process is simulated, given that the procedure inside the loop is similar to the process that is detailed in the previous example. The results of this example are introduced in Table 5.17.

Step 0: System parameters and initialization:

Step 0.1: $\eta = 0.3$, initial value of MFT, (it also can be initialized by a human operator).

$\mathfrak{p} = 1$, $r_{task} = 15$ design parameters, embedded by the system designer.

Step1: System Inputs:

Step 1.1: Simultaneous target object recognition confidence levels Eq. (4.8), and number of targets is $T = 14$;

$$\widehat{\boldsymbol{P}}_{T_{SIM}} = [0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0.86\ 0.81\ 0\ 0]^t \in \mathcal{R}^{T\times 1}$$

*Note: $\widehat{\boldsymbol{P}}_{T_{SIM}}$ here is $\widehat{\boldsymbol{P}}_T$ itself, just written here as $\widehat{\boldsymbol{P}}_{T_{SIM}}$ to refer to a simultaneous detection of two tasks.

Step 1.2: The agents' availability vector, $\boldsymbol{\vartheta}_{Av} \in \mathcal{R}^{a\times 1}, a = 20$ robot, is defined based on the current internal status of each robot, Eq. (4.12):

$$\boldsymbol{\vartheta}_{Av} = [1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 0\ 1\ 1\ 1\ 1\ 1]^t \in \mathcal{R}^{a\times 1}$$

Step 1.3: d_i the Euclidean distance Eq. (4.15) between the current location (x_i, y_i) of the robot R_i and the location of the estimated target (x_T, y_T), v_t target velocity and $v_{i_{max}}$ the maximum velocity of the robot, all are provided by the sensing system.

Step 1.4: initialize responders Counter:

$K = 0$; # Responders counter

Algorithm (Sequential Task Allocation)

Inputs: $\{\widehat{\boldsymbol{P}}_{T_{SIM}}, \boldsymbol{\vartheta}_{Av,}\ K\}$

$$\widehat{\boldsymbol{P}}_T = [0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]^t \in \mathcal{R}^{T\times 1}$$

For i in $\widehat{\boldsymbol{P}}_{T_{SIM}}$;

If $\widehat{\boldsymbol{P}}_{T_{SIM}}(i)>0$;

Then $\widehat{\boldsymbol{P}}_T(i) = \widehat{\boldsymbol{P}}_{T_{SIM}}(i)$;

Step 2: Probabilistic Matching: (logic is provided and the process is simulated)

Step 2.1: The swarm individuals specialization encoding, $\boldsymbol{S}_i$: $i = 1, 2, \dots, a$;
$\boldsymbol{S}_i \in \mathcal{R}^{1\times T}$ is defined in Table (5.2), for $a = 20$.

Step 2.2: Computing the confidence levels of the current task's class detection that are leveraged by the swarm's individuals, given by Eq. (4.7):

$$\hat{\varphi}_{R_i} = \boldsymbol{S}_i \widehat{\boldsymbol{P}}_T : i = 1\ to\ a$$

Step 2.3: Computing the maximum expected collective score of the swarm's individuals, given by Eq. (4.10): $\varphi_{R_i} = \boldsymbol{S}_i \boldsymbol{p}_{\boldsymbol{max}}$

$\boldsymbol{p}_{\boldsymbol{max}}$ is the class conditional probability of the feature X_k in its predefined class C_k, which is usually selected to equal 1 (i.e. a perfectly reliable recognition) for every predefined feature, in this test case, therefore Eq. (4.11) resulted in:

$$\boldsymbol{p}_{\boldsymbol{max}} = [1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1]^{\boldsymbol{t}} \in \mathcal{R}^{T\times 1}$$

And $\boldsymbol{S}_i$'s are defined in Table 5.2, these could result in:

$$\varphi_{R_i} = \boldsymbol{S}_i \boldsymbol{p}_{\boldsymbol{max}} = 1;\ i = 1\ to\ a;$$

Step 2.4: Computing the swarm's specialty matching probabilistic matrix, $\boldsymbol{Q} \in \mathcal{R}^{20\times 20}$, defined in Eq. (4.9):

$$\boldsymbol{Q} = \boldsymbol{dig}\left[\frac{\hat{\varphi}_{R_1}}{\varphi_{R_1}} \quad \cdots \quad \cdots \quad \frac{\hat{\varphi}_{R_a}}{\varphi_{R_a}}\right]$$

Step 2.5: The agents' attendance matrix, $\boldsymbol{\vartheta}_{A_{tt}} \in \mathcal{R}^{20\times 20}$, is defined in Eqs. (4.13) and (4.14), the agents' attendance does not contribute in the computation of the task-agent matching probability, Eq. (4.16), because $\mathfrak{p} = 1$.

Step 2.6: Computing the swarm's a specialty fitting probabilities vector, Eq. (4.16):

$$\boldsymbol{\Psi} = (\mathfrak{p}\mathbf{Q} + (1-\mathfrak{p})\boldsymbol{\vartheta}_{Att})\boldsymbol{\vartheta}_{Av}$$
$$\Rightarrow \boldsymbol{\Psi} = [\Psi_1 \quad \cdots \quad \cdots \quad \Psi_a]^t$$

Step 3: Responders coordination: (The logic is provided and the process is simulated)

Step 3.1: Coordinating the qualified agents that achieve the MFT, η , Eq. (4.18):

$$\boldsymbol{\Psi}_{MFT_i} = \begin{cases} \Psi_i & : \Psi_i \geq \eta \\ 0 & : \Psi_i < \eta \end{cases}$$

Step 3.2a: Selecting the **most qualified** agent that achieves the MFT, Eq. (4.20):

In this case, max operator extracts the ID of the agent with the highest fitting probability.

$[Score \quad ID] = max\ \{\boldsymbol{\Psi}_{MFT}\}$; # This returns the score and the index of the most qualified agent.

Step 3.2b: Update the dynamic availability state

$\boldsymbol{\vartheta}_{Av}(ID) = 0$; # This step stores zero for the availability state of the selected agent.

*this case is considered in this example (Table 5.17)

Step 3.3: Responders indexing:

$$Responder_K = R_{ID}$$

End

$RespondersList(K) = [\ Responder_K]$; # This vector returns the sequential responders.

$K = K + 1$;

End;

Return $RespondersList$

Step 4: Repeating the process:

The process is automatically repeated starting from step 1, with updated $\widehat{\boldsymbol{P}}_{T_{SIM}}$ and $\boldsymbol{\vartheta}_{Av}$.

5.3.C6 Numerical Example (Complex Tasks)

This example details the numerical processing of 5.3.C6 sample case, Example C6-1.

Step 0: System parameters and initialization:

Step 0.1: $\eta = 0.3$, initial value of MFT, (it also can be initialized by a human operator).

$\mathfrak{p} = 1$, $r_{task} = 4$ design parameters, embedded by the system designer.

Step1: Inputs:

Step 1.1: Target object recognition confidence level Eq. (4.8), and number of target classes is $T = 5$;

$$\widehat{\boldsymbol{P}}_T = [0 \;\; 0 \;\; 0.68 \;\; 0 \;\; 0.63]^t \in \mathcal{R}^{T\times 1}$$

Step 1.2: The agents' availability vector, $\boldsymbol{\vartheta}_{Av} \in \mathcal{R}^{a\times 1}, a = 8$ robot, is defined based on the current internal status of each robot, Eq. (4.12):

$$\boldsymbol{\vartheta}_{Av} = [1\ 1\ 1\ 1\ 1\ 1\ ]^t \in \mathcal{R}^{a\times 1}$$

Step 1.3: d_i the Euclidean distance Eq. (4.15) between the current location (x_i, y_i) of the robot R_i and the location of the estimated target (x_T, y_T), v_t target velocity and $v_{i_{max}}$ the maximum velocity of the robot, all are provided by the sensing system.

Step 1.4: The robotic agents attendance parameters are generated based on the logic that is given by Eq. (4.14), and used to evaluate the agents attendance, given by Eq. (4.13). These parameters are:

Υ, ϵ, and $\overline{\mathsf{T}}$

Detailed values are not provided here as attendance is not considered in this example (see step 2.5 below)

Step 2a: Probabilistic Matching:

Step 2a.1: The swarm individuals specialization encoding, $\boldsymbol{S}_i$: $i = 1, 2, \dots, a$; $\boldsymbol{S}_i \in \mathcal{R}^{1\times T}$ is defined in Table 5.20, for $a = 8$.

Step 2a.2: Computing the confidence levels of the current task's class detection that are leveraged by the swarm's individuals, given by Eq. (4.7):

$$\hat{\varphi}_{R_i} = \boldsymbol{S}_i \widehat{\boldsymbol{P}}_T$$

$$\hat{\varphi}_{R_1} = \boldsymbol{S_1} * \widehat{\boldsymbol{P}}_T = 0;\ \hat{\varphi}_{R_2} = \boldsymbol{S_2} * \widehat{\boldsymbol{P}}_T = 0.63$$

$$\hat{\varphi}_{R_3} = \boldsymbol{S_3} * \widehat{\boldsymbol{P}}_T = 0;\ \hat{\varphi}_{R_4} = \boldsymbol{S_4} * \widehat{\boldsymbol{P}}_T = 0.63$$

$$\hat{\varphi}_{R_5} = \boldsymbol{S_5} * \widehat{\boldsymbol{P}}_T = 0.68;\ \hat{\varphi}_{R_6} = \boldsymbol{S_6} * \widehat{\boldsymbol{P}}_T = 1.31;$$

$$\hat{\varphi}_{R_7} = \boldsymbol{S_7} * \widehat{\boldsymbol{P}}_T = 1.31;\ \hat{\varphi}_{R_8} = \boldsymbol{S_8} * \widehat{\boldsymbol{P}}_T = 0.63;$$

Step 2a.3: Computing the maximum expected collective score of the swarm's individuals, given by Eq. (4.10):

$$\varphi_{R_i} = \boldsymbol{S}_i \boldsymbol{p_{max}}$$

$\boldsymbol{p_{max}}$ is the class conditional probability of the feature X_k in its predefined class C_k, which is usually selected to equal 1 (i.e. a perfectly reliable recognition) for every predefined feature in this test case, therefore Eq. (4.11) resulted in:

$$\boldsymbol{p_{max}} = [1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1]^{\boldsymbol{t}} \in \mathcal{R}^{T\times 1}$$

And $\boldsymbol{S}_i$'s are defined in Table 5.20, these could result in:

$$\varphi_{R_1} = \boldsymbol{S}_1 \boldsymbol{p_{max}} = 2;\ \varphi_{R_2} = \boldsymbol{S}_2 \boldsymbol{p_{max}} = 2;\ \varphi_{R_3} = \boldsymbol{S}_3 \boldsymbol{p_{max}} = 2;$$

$$\varphi_{R_4} = \boldsymbol{S}_4 \boldsymbol{p_{max}} = 2;\ \varphi_{R_5} = \boldsymbol{S}_5 \boldsymbol{p_{max}} = 2;\ \varphi_{R_6} = \boldsymbol{S}_6 \boldsymbol{p_{max}} = 2;$$

$$\varphi_{R_7} = \boldsymbol{S}_7\boldsymbol{p_{max}} = 3;\ \varphi_{R_8} = \boldsymbol{S}_8\boldsymbol{p_{max}} = 3;$$

Step 2a.4: Computing the swarm's specialty matching probabilistic matrix, $\boldsymbol{Q} \in \mathcal{R}^{20\times 20}$, defined in Eq. (4.9):

$$\boldsymbol{Q} = \boldsymbol{dig}\left[\frac{0}{2}\ \ \frac{0.63}{2}\ \ \frac{0}{2}\ \ \frac{0.63}{2}\ \ \frac{0.68}{2}\ \ \frac{1.31}{2}\ \ \frac{1.31}{3}\ \ \frac{0.63}{3}\right]$$
$$= \boldsymbol{dig}[0\ \ 0.31\ \ 0\ \ 0.31\ \ 0.34\ \ 0.65\ \ 0.43\ \ 0.21]$$

Step 2a.5: The agents' attendance matrix, $\boldsymbol{\vartheta}_{A_{tt}} \in \mathcal{R}^{8\times 8}$, is defined in Eqs. (4.13) and (4.14), the agents' attendance does not contribute in the computation of the task-agent matching probability, Eq. (4.16), because $\mathfrak{p} = 1$.

Step 2a.6: Computing the swarm's a specialty fitting probabilities vector, Eq. (4.16):

$$\boldsymbol{\Psi} = (\mathfrak{p}\mathbf{Q} + (1-\mathfrak{p})\boldsymbol{\vartheta}_{Att})\boldsymbol{\vartheta}_{Av}$$
$$\Rightarrow \boldsymbol{\Psi} = ((1)\mathbf{Q} + (1-1)\boldsymbol{\vartheta}_{Att})\boldsymbol{\vartheta}_{Av}$$
$$\Rightarrow \boldsymbol{\Psi} = [0\ \ 0.31\ \ 0\ \ 0.31\ \ 0.34\ \ 0.65\ \ 0.43\ \ 0.21]^t$$

Step 2b: Prioritization scheme (This step is running in parallel with Step 2a)

Step 2b.1: The mission's primary and secondary requirements are pre-initialized and embedded in the primary selector by the systems designer and set to "ON" or "OFF" based on the application demand.
Primary Selector set to:

$$[Off\ \ Off\ \ Off\ \ Off\ \ ON];$$

Then the resulted primary encoder, Eq.(4.21):

$$\boldsymbol{B}_\rho = diag\left[0\ \ 0\ \ 0\ \ 0\ \ \widehat{\boldsymbol{P}}_{T_5}\right];\ \boldsymbol{B}_\rho \in \mathcal{R}^{T\times T}$$
$$\boldsymbol{B}_\rho = diag[0\ \ 0\ \ 0\ \ 0\ \ 0.63]$$

Step 2b.2: To associate a priority with the robots that possess primary capabilities, the categorized specialization vector of each robot, $\boldsymbol{S}_{\rho_i} \in \mathcal{R}^{1\times T}$, is calculated using Eq. (4.22): (Given that the individuals specializations encoding, $\boldsymbol{S}_i$, is defied in step 2a.1 above, linked in Figure 4.3).

$$\boldsymbol{S}_{\rho_i} = \boldsymbol{S}_i\boldsymbol{B}_\rho$$
$$\boldsymbol{S}_{\rho_1} = \boldsymbol{S}_1\boldsymbol{B}_\rho = [0\ \ 0\ \ 0\ \ 0\ \ 0];\ \boldsymbol{S}_{\rho_2} = \boldsymbol{S}_2\boldsymbol{B}_\rho = [0\ \ 0\ \ 0\ \ 0\ \ 0.63];$$
$$\boldsymbol{S}_{\rho_3} = \boldsymbol{S}_3\boldsymbol{B}_\rho = [0\ \ 0\ \ 0\ \ 0\ \ 0];\ \boldsymbol{S}_{\rho_4} = \boldsymbol{S}_4\boldsymbol{B}_\rho = [0\ \ 0\ \ 0\ \ 0\ \ 0.63];$$
$$\boldsymbol{S}_{\rho_5} = \boldsymbol{S}_5\boldsymbol{B}_\rho = [0\ \ 0\ \ 0\ \ 0\ \ 0];\ \boldsymbol{S}_{\rho_6} = \boldsymbol{S}_6\boldsymbol{B}_\rho = [0\ \ 0\ \ 0\ \ 0\ \ 0.63];$$
$$\boldsymbol{S}_{\rho_7} = \boldsymbol{S}_7\boldsymbol{B}_\rho = [0\ \ 0\ \ 0\ \ 0\ \ 0.63];\ \boldsymbol{S}_{\rho_8} = \boldsymbol{S}_8\boldsymbol{B}_\rho = [0\ \ 0\ \ 0\ \ 0\ \ 0.63];$$

Step 2b.3: Calculating the swarm's primary probability, $\boldsymbol{Q}_\rho \in \mathcal{R}^{1\times a}$, Eq. (4.23), based on $\hat{\varphi}_{\rho_i} = \sum \boldsymbol{S}_{\rho_i}$, and $\varphi_\rho = Trsc\{\boldsymbol{B}_\rho\} + \kappa$, given that κ is a very small constant used to avoid the division by zero and to calibrate the detected confidence level on the primary class:

$$\kappa = 0.05 \Rightarrow\ \varphi_\rho = Trsc\{\boldsymbol{B}_\rho\} + \kappa = 0.68;$$
$$\hat{\varphi}_{\rho_1} = 0;\ \hat{\varphi}_{\rho_2} = 0.63;\ \hat{\varphi}_{\rho_3} = 0;\ \hat{\varphi}_{\rho_4} = 0.63;\ \hat{\varphi}_{\rho_5} = 0;\ \hat{\varphi}_{\rho_6} = 0.63;\ \hat{\varphi}_{\rho_7} = 0.63;\ \hat{\varphi}_{\rho_8} = 0.63;$$
$$\Rightarrow \boldsymbol{Q}_\rho = \left[\frac{\hat{\varphi}_{\rho_i}}{\varphi_\rho} : i = 1, \dots, a\right] = \frac{1}{0.68}[0\ \ 0.63\ \ 0\ \ 0.63\ \ 0\ \ 0.63\ \ 0.63\ \ 0.63]$$
$$\Rightarrow \boldsymbol{Q}_\rho = [0\ \ 0.92\ \ 0\ \ 0.92\ \ 0\ \ 0.92\ \ 0.92\ \ 0.92]$$

Step 2b.4: Algorithm 4.1: Primary Capabilities Prioritization
Input: $\left[Q_{\rho_i}, i\right] = (Q_\rho(i))$; i in $\{1, \dots, a\}$; $\text{PRIM}_{min} = \frac{1}{1+2\kappa} = \frac{1}{1+2(0.05)} = 0.90$
Output: $\boldsymbol{R}_\rho = diag[0\ 1\ 0\ 1\ 0\ 1\ 1\ 1] \in \mathcal{R}^{a\times a}$;
(The choices of κ and $\text{PRIM}_{\min}$ are discussed after this example).

Step 3: Qualified responders coordination:

Step 3.1: Coordinating the qualified agents that achieve the MFT, η , Eq. (4.18): (this results from **Step 2a.6**)

$$\boldsymbol{\Psi}_{MFT} = [0 \quad 0.31 \quad 0 \quad 0.31 \quad 0.34 \quad 0.65 \quad 0.43 \quad 0]^t \in \mathcal{R}^{a\times 1}$$

Step 3.2: Activating the prioritization layer to filter the agents allocation priorities:

Due to a detection of a primary class, the term $Trac(\boldsymbol{B}_\rho)$ introduces a none zero value. If this value is ≥ 0.5, therefore the system can trust it, as a considerable confidence level of a detected primary class, and activates the prioritization scheme to filter the agents task allocation priorities. In this example:

$Trac(\boldsymbol{B}_\rho) = 0.63$, then the switch of Figure 4.3 activates the prioritization scheme.

As indicated in Figure 4.3, if the switch is active, then Eq. (4.24) is considered:

$$\boldsymbol{h_\rho} = \boldsymbol{R_\rho \Psi_{MFT}} = (diag[0 \quad 1 \quad 0 \quad 1 \quad 0 \quad 1 \quad 1 \quad 1]) * [0 \;\; 0.31 \;\; 0 \;\; 0.31 \;\; 0.34 \;\; 0.65 \;\; 0.43 \;\; 0]^t$$

$$\Rightarrow \boldsymbol{h_\rho} = [\mathbf{0} \;\; 0.31 \;\; \mathbf{0} \;\; 0.31 \;\; \mathbf{0} \;\; 0.65 \;\; 0.43 \;\; 0] \text{ (these three } \mathbf{bold} \text{ zeros are the suspended scores)}$$

Step 3.2a: Selecting the **most qualified** agent that achieves the MFT, Eq. (4.25): In this case, max operator extracts the ID of the agent with the highest fitting probability. If more than one agent share the same fitting score and the latter is maximum then max operator extracts the ID of the agent that has first highest probability in, $\boldsymbol{h_\rho}$, vector:

$$\emptyset_{BEST\ RESPONDER\ INDEX=[\ R_6\ ]}$$

Step 3.2b: Selecting **all qualified** agents that achieve the MFT, Eq. (4.24): In this case, an iterative loop runs inside the vector, $\boldsymbol{\Psi}_{MFT}$, in Eq. (4.18) and extracts the IDs of all agents with fitting probabilities that are higher than η:

$$\emptyset_{BEST\ RESPONDERS\ INDEXES=[\ R_2\ \ R_4\ \ R_6\ \ R_7]}$$

*this case is provided here just for illustrating the case that when the application requires that all the qualified agents need to be assigned.

These results are indicated in Table 5.24, (2nd columns from the right) of binary encoding case, its also shown in Figure 5.20c.

Step 4: Repeating the process:

The process is automatically repeated starting from step 1, upon a new detected task with updated $\widehat{\boldsymbol{P}}_T$ and $\boldsymbol{\vartheta}_{Av}$.

Calibrating κ and PRIM$_{min}$:

To ensure the stability of the proposed prioritization scheme, two constants are considered. That are κ and PRIM$_{min}$. κ is a small constant. The value of this constant is selected to do dual role: 1) to avoid the division by zero in Eq. (4.23). 2) To compute PRIM$_{min} = \frac{1}{1+2\kappa}$ and balance between the considerable level of the trust on the detected primary classes and the minimum primary probabilities, that can result in 1's diagonal elements in $\boldsymbol{R}_\rho$. This calibration is studied for all expected confidence levels of the target object detection in the case that one primary class is assumed to be detected, $\widehat{\boldsymbol{P}}_{T_{Prim}}$. Given that $\hat{\varphi}_{\rho_i} = \sum \boldsymbol{S}_{\rho_i}$ and if the robot is qualified with the primary capability, then:

$$\hat{\varphi}_{\rho_i} = \sum \boldsymbol{S}_{\rho_i} = \hat{P}_{T_{Prim}}, \text{ and } \varphi_\rho = Trsc\{\boldsymbol{B}_\rho\} + \kappa = \widehat{\boldsymbol{P}}_{T_{Prim}} + \kappa$$

From Eq. (4.23), we have:

$$Q_{\rho_i} = \frac{\hat{\varphi}_{\rho_i}}{\varphi_\rho}$$

Table A.1 Constants calibrations

$\widehat{\boldsymbol{P}}_{T_{Prim}}$	$\kappa = 0.01$	PRIM_{min}	$\kappa = 0.05$	PRIM_{min}	$\kappa = 0.1$	PRIM_{min}	$\kappa = 0.2$	PRIM_{min}
	Q_{ρ_i}		Q_{ρ_i}		Q_{ρ_i}		Q_{ρ_i}	
0.30	0.96	0.98	0.85	0.90	0.75	0.83	0.60	0.71
0.40	0.97		0.88		0.80		0.66	
0.50	0.98		0.90		0.83		0.71	
0.60	0.98		0.92		0.85		0.75	
0.70	0.98		0.93		0.87		0.77	
0.80	0.98		0.94		0.88		0.80	
0.90	0.99		0.95		0.90		0.81	

*Green range of the detected primary class is the range that is to be trusted by the prioritization scheme, which will result in 1's diagonal values of the matrix $\boldsymbol{R}_{\rho}$.

Based on the observation of the calibration results in Table A.1 a fixed formula of $\text{PRIM}_{min} = \frac{1}{1+2\kappa}$ is adopted and the value of the constant κ can be selected by the system designer in this range: $\kappa \in (0.01,\ 0.05,\ 0.1,\ 0.2]$. This design assumes that the low conference level: ($\widehat{\boldsymbol{P}}_{T_{Prim}} < 0.50$) on the primary class detection is not trustable and the prioritization scheme should not trust it. Higher values of κ can be more trustable because the difference between the blocked level and the trusted level is higher.

Appendix B

6.2.C1 Numerical Example

This example provides the mathematical details of computing the fitting probabilities of 6.2.C1 sample case, presented in Figure 6.1 and Table 6.3, Chapter 6. This example details the numerical processing of the proposed scheme, which shows how the probability of agents R_1, R_2, R_3, and R_4 is obtained at the output to equal 0.48 starting from a target recognition confidence level of 0.96 at the input.

Step 0: System initialization:

Step 0.1: $\eta = 0.4$, initial value of MFT (it also can be initialized by a human operator)

$\mathfrak{p} = 1$, $r_{task} = 2$ design parameters, embedded by the system designer

Step 1: Inputs:

Step 1.1: Target object recognition confidence level Eq. (6.1):

$$\widehat{\boldsymbol{P}}_T = [0, \quad 0.96, \quad 0, \quad 0, \quad 0]^T$$

Step 1.2: The agents' availability vector, $\boldsymbol{\vartheta}_{Av} \in \mathcal{R}^{a\times 1}$, is defined based on the current internal status of each robot, Eq. (4.12):

$$\boldsymbol{\vartheta}_{Av} = [1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1]^T$$

Step 1.3: d_i the Euclidean distance Eq. (4.15) between the current location (x_i, y_i) of the robot R_i and the location of the estimated target (x_T, y_T), v_t target velocity and $v_{i_{max}}$ the maximum velocity of the robot are all provided by the sensing system.

Step 1.4: The robotic agents' attendance parameters are generated based on the logic that is given by Eq. (4.14), and these are used to evaluate the agents' attendance, given by Eq. (4.13). These parameters are:

Υ, ϵ, and $\overline{\mathsf{T}}$

Detailed values are not provided here as attendance is not considered in this example (see step 2.5 below).

Step 2: Probabilistic matching:

Step 2.1: The swarm individuals' specialization coding, $\boldsymbol{S}_i$: $i = 1, 2, \ldots, a$; is defined in Table (6.2), for $a = 7$.

Step 2.2: Computing the confidence levels of the current task features' detection that is subjected to the swarm's individuals, given by Eq. (4.7):

$$\hat{\varphi}_{R_i} = \boldsymbol{S}_i \widehat{\boldsymbol{P}}_T$$

$$\hat{\varphi}_{R_1} = [1 \quad 1 \quad 0 \quad 0 \quad 0] * \begin{bmatrix} 0 \\ 0.96 \\ 0 \\ 0 \\ 0 \end{bmatrix} = 0.96; \ \hat{\varphi}_{R_2} = [0 \quad 1 \quad 1 \quad 0 \quad 0] * \begin{bmatrix} 0 \\ 0.96 \\ 0 \\ 0 \\ 0 \end{bmatrix} = 0.96$$

$$\hat{\varphi}_{R_3} = [0 \quad 1 \quad 0 \quad 1 \quad 0] * \begin{bmatrix} 0 \\ 0.96 \\ 0 \\ 0 \\ 0 \end{bmatrix} = 0.96; \hat{\varphi}_{R_4} = [0 \quad 1 \quad 0 \quad 0 \quad 1] * \begin{bmatrix} 0 \\ 0.96 \\ 0 \\ 0 \\ 0 \end{bmatrix} = 0.96$$

$$\hat{\varphi}_{R_5} = [0 \quad 0 \quad 1 \quad 0 \quad 0] * \begin{bmatrix} 0 \\ 0.96 \\ 0 \\ 0 \\ 0 \end{bmatrix} = 0; \hat{\varphi}_{R_6} = [0 \quad 0 \quad 0 \quad 1 \quad 0] * \begin{bmatrix} 0 \\ 0.96 \\ 0 \\ 0 \\ 0 \end{bmatrix} = 0$$

$$\hat{\varphi}_{R_7} = [0 \quad 0 \quad 0 \quad 0 \quad 1] * \begin{bmatrix} 0 \\ 0.96 \\ 0 \\ 0 \\ 0 \end{bmatrix} = 0$$

Step 2.3: Computing the maximum expected collective score of the swarm's individuals, given by Eq. (4.10):

$$\varphi_{R_i} = \boldsymbol{S}_i \boldsymbol{p}_{\boldsymbol{max}}$$

$\boldsymbol{p}_{\boldsymbol{max}}$ is the class conditional probability of the feature X_k in its predefined class C_k, which is usually selected to equal 1 (i.e. a perfectly reliable recognition) for every predefined feature in this test case, which results in:

$$\varphi_{R_1} = [1 \quad 1 \quad 0 \quad 0 \quad 0] * \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = 2; \varphi_{R_2} = [0 \quad 1 \quad 1 \quad 0 \quad 0] * \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = 2; \varphi_{R_3} = [0 \quad 1 \quad 0 \quad 1 \quad 0] * \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = 2;$$

$$\varphi_{R_4} = [0 \quad 1 \quad 0 \quad 0 \quad 1] * \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = 2; \varphi_{R_5} = [0 \quad 0 \quad 1 \quad 0 \quad 0] * \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = 1; \varphi_{R_6} = [0 \quad 0 \quad 0 \quad 1 \quad 0] * \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = 1;$$

$$\varphi_{R_7} = [0 \quad 0 \quad 0 \quad 0 \quad 1] * \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = 1;$$

Step 2.4: Computing the swarm's specialty matching probabilistic matrix, $\boldsymbol{Q} \in \mathcal{R}^{7\times7}$, defined in Eq. (4.9):

$$\boldsymbol{Q} = \begin{bmatrix} 0.48 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.48 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.48 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.48 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Step 2.5: The agents' attendance matrix, $\boldsymbol{\vartheta}_{A_{tt}} \in \mathcal{R}^{7\times7}$, is defined in Eqs. (4.13) and (4.14). The agents' attendance does not contribute to the computation of the task-agent matching probability, Eq. (4.16), because $\mathfrak{p} = 1$.

Step 2.6: Computing the swarm's specialty fitting probabilities vector, Eq. (4.16):

$$\boldsymbol{\Psi} = (\mathfrak{p}\mathbf{Q} + (1 - \mathfrak{p})\boldsymbol{\vartheta}_{Att})\boldsymbol{\vartheta}_{Av}$$

$$\Rightarrow \boldsymbol{\Psi} = ((1)\mathbf{Q} + (1 - 1)\boldsymbol{\vartheta}_{Att})\boldsymbol{\vartheta}_{Av}$$

$$\Rightarrow \boldsymbol{\Psi} = [0.48 \quad 0.48 \quad 0.48 \quad 0.48 \quad 0 \quad 0 \quad 0]^{\mathrm{T}}$$

Step 3: Qualified responders' coordination:

Step 3.1: Coordinating the qualified agents that achieve the MFT, η, Eq. (4.18):

$$\boldsymbol{\Psi}_{MFT} = [0.48 \quad 0.48 \quad 0.48 \quad 0.48 \quad 0 \quad 0 \quad 0]^{\mathrm{T}}$$

Step 3.2: Selecting **all qualified** agents that achieve the MFT, Eq. (4.18): In this case, an iterative loop runs inside the vector (4.18) within a length of the $\boldsymbol{\Psi}_{MFT}$ vector and extracts the IDs of all agents with fitting probabilities that are higher than η:

$$\emptyset_{BEST\ RESPONDERS,\ INDEXES=[\ R_1\ R_2\ R_3\ R_4]}$$

These results are indicated in Table 6.3 and Figure 6.1b.

Step 4: Repeating the process:

The process is automatically repeated starting from step 1 upon a new detected task with updated $\widehat{\boldsymbol{P}}_T$ and $\boldsymbol{\vartheta}_{Av}$.