POLITECNICO DI MILANO
Corso di Laurea in Ingegneria Informatica
Dipartimento di Elettronica e Informazione

Environment Classification: an Empirical Study of the Response of a Robot Swarm to Three Different Decision-Making Rules

IRIDIA
Institute de Recherches Interdisciplinaires
et de Développementes en Intelligence Artificielle
Université Libre de Bruxelles

Relatore: Prof. Andrea Bonarini
Prof. Marco Dorigo

Correlatore: Ing. Gabriele Valentini
Ing. Andreagiovanni Reina
Ing. Anthony Antoun

Tesi di Laurea di:
Davide Brambilla, matricola 804985

Anno Accademico 2014-2015

A Giulia

Summary

Swarm robotics inspects the class of systems whereby a large number of robots interact in a *self-organized* and *decentralized* way in order to collectively reach a certain goal. Our work is better contextualized into a subcategory of swarm robotics, called *collective decision making.* In collective decision-making problems the swarm is place in front of a set (discrete or continuous) of mutually exclusive alternatives. The general goal of a collective decision is to have every robot (or a large majority) of the swarm agreeing toward one of the options, usually the one which maximizes a certain performance of the system (e.g. the covered area, the execution time of an action). In this thesis we present a self-organizing, decentralized, general and portable solution to a novel scenario called *environment classification.* In environment classification, an homogeneous swarm of autonomous robots has to classify the environment by the resources that it contains. To consider the problem correctly solved, every robot of the swarm has to agree toward the most available resource in the environment. The goal of this thesis is to give an empirical analysis of the dynamics of the swarm when three decision rules are applied in the decision-making process, in terms of accuracy of the solution and time required to reach the consensus. The decision rules are the weighted voter model, the direct comparison, and the majority rule. The comparison has been perform with both physic-based simulation experiments and experiments with a swarm of real robots.

Acknowledgement

I want to thank all the people who helped me in this year doing my thesis. First of all I want to thank Prof. Andrea Bonarini to gave me this opportunity to make this experience abroad. Moreover I want to thank him to have been always really into the work, advising me with interest.

I want to thank Prof. Marco Dorigo to hosted me in Iridia and to gave me the chance to do this wonderful experience.

I want to thank my three correlators, Gabri, Anthony and Gio, to drove me through this year.

I want to thank all the guys from Iridia to have been always close to me making me feel the welcome.

The guys of the erasmus program and the guys from the residence who lived with me. Merci mec, gracias papu.

Vorrei ringraziare tutti i miei amici, quelli d'infanzia e quelli che ho conosciuto pi tardi per essermi stati sempre vicino e anche quelli che non ci sono pi.

Un grazie speciale lo voglio dire a Gabri, che mi sempre stato molto vicino ed ha spesso messo me davanti a tutto il resto. Grazie mille Gabri, non dimenticher mai quello che hai fatto per me.

Vorrei ringraziare la mia famiglia per avermi dato tanto nella vita. La mia mamma, che con i suoi occhi mi rassicura sempre. Mio pap, ti voglio bene boss. Mio fratello, mia nonna ed i miei zii, Carlo e Claudia, a cui devo un sacco di cose.

Vorrei ringraziare Lucio, Grazia, Pippo, e Gabri, che sono stati la mia seconda famiglia.

Il grazie pi grosso va alla persona pi importante della mia vita. Una persona che mi ha preso per mano quando avevo l'et di 16 anni, e non sapevo neanche cosa ci facessi su questa terra. Che mi sempre stata vicina, qualunque fosse la mia scelta. E che vorrei avere accanto per sempre, perch in fondo ancora non so che cosa ci faccio su questa terra.

Contents

Chapter 1

Introduction

> *"Great things are done by*
> *a series of small things*
> *brought together."*
>
> ***Vincent Van Gogh***

1.1 Swarm Robotics and Collective Decision Making

The area of interest of this thesis is *swarm robotics*, a branch of robotics that takes as source of inspiration some examples of collective behaviours present in nature. *Swarm robotics* inspects the class of systems whereby a large number of robots interact in a *self-organized* and *decentralized* way in order to collectively reach a certain goal. In a self-organized system every agent is separated from the others, that is, every agent is behaving independently from the other agent's state. After the starting steps, agents begin to create some kind of relations (connections) with the others. The parts-separated system become hence a parts-joined system [3]. Decentralized means instead that the swarm does not have robots with the role of coordinating the other robots. In swarm robotics, every robot has only a local information deriving both from the environment and from the neighbours. This is a big difference with the centralized systems, where the central robot has a global knowledge about the system. The coordination between robots derives from the processing of the information that every robot collects during the execution. Information deriving from the communication with the neighbours and from the exploration of the environment are pooled and processed by every robot following a determined strategy that brings the system to a collective behaviour.

The number of robots involved in the process represents the *swarm size*. Effects of varying the swarm size have been widely studied in literature. Larger swarm sizes are usually preferable since it would imply higher level of redundancy and parallelism, that is, higher redundancy and robustness. The purpose of swarm robotics is to create systems with three characteristic properties: *flexibility*, *scalability*, and *robustness*. These characteristics aim, respectively, to build a swarm: 1) able to adapt itself to the changes of the environment during the time; 2) that is correctly working even increasing or decreasing the number of its components; and 3) that is fault tolerant to eventual individual failures of the robots. These features make swarm robotics particularly feasible for a wide range of real-world applications: dangerous tasks (e.g., demining, radioactive-garbage collection, difficult-environment exploration [9], [86]), situations with an unknown environment, or situations where the conditions of the environment are rapidly changing (e.g., oil-leakage [86]). A very complete overall review on swarm robotics has been done by Brambilla et al. [9], and we will go deeply in the explanation of swarm robotics in the following chapter (2).

Our work is better contextualized into a sub-category of swarm robotics, called *collective decision making*. Collective decision-making problems are widely studied in swarm robotics. In such kind of problems the swarm is place in front of a set (discrete or continuous) of mutually exclusive alternatives. The general goal of a collective decision is to have every robot (or a large majority) of the swarm agreeing toward one of the options, usually the one that maximizes a certain performance of the system (e.g. the covered area, the execution time of an action).

A parallel can be found in nature: social insects are simple cognitive agents able to take individual decisions. They are just informed about some local information, for example on the surrounding environment or the status of the neighbour elements [98]. Through direct or indirect communication [9] the group of insects is able to reach a final state where every individual has taken the same choice. The individual decision of an element (either a robot or an insect) is the result of the process of gathering information from the environment. Instead, *collective decisions in swarm robotics* are emerging from the self-organization process of the robots and the decentralized nature of the group. Usually the collective decision-making process is composed by the phase of exploration, in order to gather information, and the information pooling. After all the information have been collected, every single robot has to take a decision basing on them. Through numerous local communication among the robots and with the environment and without a centralized control a collective decision can be reached [12], [100].

Two big subclasses of collective decision making are agreement (or consensus achievement) and specialization [9]. In agreement the desired outcome is that every robot, or a large majority of the robots, is converging, after the execution, on the same option among the set of possible one. In specialization, instead, the robots should distribute themselves on a set of possible tasks that must be executed. The most common example of specialization is task allocation, that is how to allocate the robots to a set of known tasks in order to maximize the performance of the system. An example is the cleaning of one room: let us suppose that, in order to clean a room, two tasks must be achieved: the first step is to remove all the object on the floor while the second is to distribute the robots on the floor and clean the destined area. The collective decision-making problem concerns the allocation of these tasks among the robots in a way that optimize the cleaning of the room.

A particular case of collective decision making is called best-of-n, and are problems characterized by a discrete set of opinions that the swarm has to discriminate. The set of alternatives is characterized by the presence of a single option that is the best one, i.e., the one that maximizes some metric of the problem. Usually every option has an associated quality, and the best option is the one with the highest quality.

1.2 Motivations and Contributions of the Thesis

In this thesis we analyse the behaviour of a swarm of robots facing a best-of-n decision-making problem in a never studied scenario. Starting from the work of Valentini et al. [103], [101] we proposed a solution for a new scenario, focusing on the dynamics of the behaviour of the swarm under the application of three different strategies of decision. In this problem, a swarm has to *classify* the environment by the different *resources* it contains. The goal of the swarm is to discover which is the most available resource that can be found in the environment. Analysing our scenario we can easily identify the key factors characterizing the best-of-n decision-making problems: the discrete set of alternatives is represented by the resources in the environment, while the best option that the swarm has to desirable choice is the most available one. Every agent of the swarm is following the same behaviour described by a probabilistic finite-state machine and is eventually applying, in the decision-making process, the same decision rule. The main goal of this thesis is to analyse, from a quantitatively perspective, the behaviour of the swarm when are applied three well-known decision-making rules: *weighted voter model*, *direct comparison*, and *majority rule*. The object of our work is to track the two variables better describing the performance of the solution, that are the *consensus time* and the *exit probability*. The first one is the time needed by the swarm to solve the problem while the second one is the accuracy of the solution in terms of correctness of the solution.

Weighted voter model and majority rule have already been treated in literature by other works [103], [101], [110], [54], [64], [104]. We introduced a never studied decision rule that is the direct comparison. This approach is using more information with respect than the other two rules that are completely self-organizing and are not leaning on the exchange of information. We decided to introduce direct comparison as control strategy in order to understand in which conditions is preferable to use a self-organized approach and when is better to use an extensive exchange of information.

Our main goal was to give a complete analysis of the swarms' behaviour under the application of three strategies in order to solve the same problem. In the process of build up the comparison between the strategies we made research works that can be useful to the rest of the community. The innovative contributions of this research are:

- We gave a decentralized, self-organizing, general, and portable solution to a non-studied scenario of a binary best-of-n decision-making problem. The main innovation is the scenario, that is never been exploited before;

- We made a comparison between three different strategies in the same scenario, focusing on the variables that describe the dynamics of the system. We made an analysis of the well-known speed versus accuracy problem of the three decision rules, identifying the conditions in which each decision rule is more advantageous to be used with respect to the others;

- We conducted extensive real-robots experiments comparing three different decision rules. Real-robots experiments are a definitive test-bench: in real-robot experiments the situation is not the ideal one that is used in simulations. It entails that the results obtained from the experiments done with simulation tools can be different from the ones made using real robots. We analysed the behaviour of the swarm composed by real robots when the different strategies have been applied in order to solve the same problem in the same scenario;

- We studied the behaviour of the swarm using the direct comparison, that is a strategy requiring an higher quantity of information exchanged. This decision rule has never been applied to a swarm of autonomous robots solving a collective decision-making problem. We decided to introduce it because it is quite different than the other two decision rules used. We wanted to test the direct comparison as a control strategy, to see how.

This scenario still has a lot of extensions that can be studied in the future. We recall that, in our scenario, the swarm has to discriminate the resources present in the environment. We analysed a scenario where there were two resources in the environment, reducing then the problem to a *binary* best-of-n decision-making problem. In the future it would be interesting to study the case where the resources in the environment are more than only 2, extending the cardinality of the set of alternatives. Moreover, in our solution every robot is behaving in the same way, following the same decision rule. Another extension of the problem could be the analysis of the behaviour of the swarm when different decision ruless are applied to different portions of the swarm: what could be the effect of applying, for example, the weighted voter model to one half of the swarm and the majority rule to the other half? Would it speed-up the consensus time? And what about the accuracy of the decisions?

1.3 Structure of the Thesis

The thesis is structured as follow.

In **Chapter 2** (State of the Art) we reviewed the state of the art interesting this thesis work. We started by introducing, defining, and describing the swarm robotics field giving indications about the related works. After that we completely explain the collective decision making, going deeply in the details of the works that are directly linked to this thesis. This chapter has been thought to introduce the reader to the swarm robotics, giving information about this general field before to go deeper into the details of collective decision making, the subcategory better describing our problem.

In **Chapter 3** (Environment Classification) we defined our problem, describing the scenario, the solution that we have proposed, and the finite state machine describing the behaviour of the robots. In this introduction we have just briefly introduced the environment classification, while in chapter 3 it will be fully described.

In **Chapter 4** (Simulation Experiments) we showed the experiments that we have done in the simulation phase, showing the motivations that pushed us to do each experiment and discussing about the obtained results.

In **Chapter 5** (Real Robots Experiments) we spread out the real-robots experiments, discussing about the experiments done in order to set-up the experimental environment and about the results obtained by the experiments.

In the conclusion chapter we summarized what we have done and which are the results of the experiments.

Chapter 2

State of the Art

In this chapter we are going to introduce and define the Swarm Robotics as branch of robotics. In order to do that we will discuss the origins of swarm robotics and the influence of the observation of some biological systems as source of inspiration. We will explain the main characteristics defining swarm robotics, and we will list some works done up to now in this area. Finally we will discuss about the points that are still lacking in this research field.

After that we are going to focus on the sub-branch of swarm robotics' called collective decision making, that is the sub-area where this thesis occurs. We will explain the works done in collective decision making, classifying them by the nature of the set of their alternatives that can be discrete or continuous.

2.1 Swarm Robotics

Swarm intelligence is the discipline that studies the *collective* and *intelligent* behaviour of a group of entities, both animals or robots, emerging from the local interactions between *simple* individuals and between the individuals and the environment [16].

Swarm robotics is the application of swarm intelligence to multi-robot systems [86]. Swarm robotics studies the design of groups of cooperative robots working together without any external infrastructure or any form of centralized control [20]. The ideal outcome of a swarm robotics system is a collective behaviour that performs as desired in order to find the solution of a specific problem [9],[86]. Swarm robotics has been developed after the study of self-organized behaviours present in nature, performed by social animals. Examples of natural swarm behaviours are some kind of eusocial insects, as ants and termites, honeybees, cockroaches [8]. Other examples of collective behaviours are easily observable in fish schools [45] and bird flocks [5].

The main characteristics of collective behaviours in nature are *flexibility* to environmental changes, *robustness* to individual robot failures, and *scalability* with the size of the swarm [11]. These characteristics are exactly what is desirable to have in a swarm robotics system [86]. These properties are the result of a really simple behaviour followed by the entities of the self-organized swarm and of the *local interaction* among robots and between robots and the environment surrounding them.

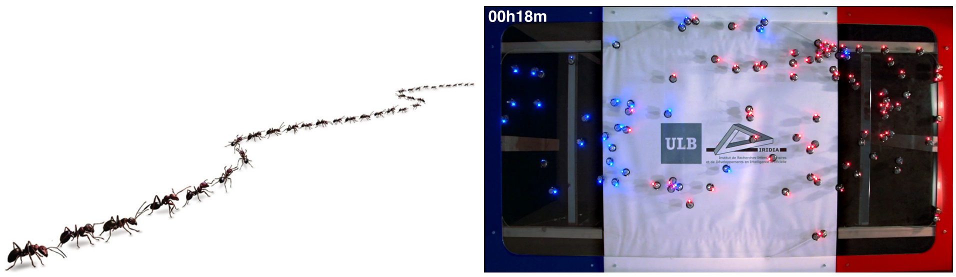
The range of applications of swarm robotics is really wide: dangerous tasks (e.g., demining, radioactive-garbage collection, difficult-environment exploration [9], [86]), situations with an unknown environment or situations where the conditions are rapidly changing (e.g., oil-leakage [86]).

Up to now, no engineering approaches for the design of swarm systems have been defined: swarm robotics is still such an art, where the researcher has to put his own capabilities without a predefined approach. An engineering way to define, design, realize, verify and maintain a swarm is still lacking. The most common way to design a swarm of robots is bottom-up, that is, starting from the design of a single robot behaviour the engineer tries to reach the desired behaviour of the whole swarm by trial and error [15]. Some top-down approaches have been proposed [9], [109], [83]. However, no real-world application has been implemented using swarm robotics. Possible reasons for the absence of swarm robotics applications are for example, hardware limitations of the current robots, the lack of an engineering approach to design and validate the swarm, and the uncertainly of the outcome of the swarm [9].

2.1.1 Origins and Characteristics

Initially the term *swarm intelligence* was used to indicate a particular class of cellular robotic systems [7] and was not a simple concept to define: the term *intelligence behaviour* in this context was defined as the capacity of producing a desired outcome in a non-predictable way. The term swarm intelligence assumes the meaning of a group of non-intelligent cellular robots producing an intelligent (i.e., desired and not predictable output) outcome [7].

Swarm robotics is the application of swarm intelligence to multi-robot systems [86], [19]. Several definitions of swarm robotics have been defined.



(a) (b)

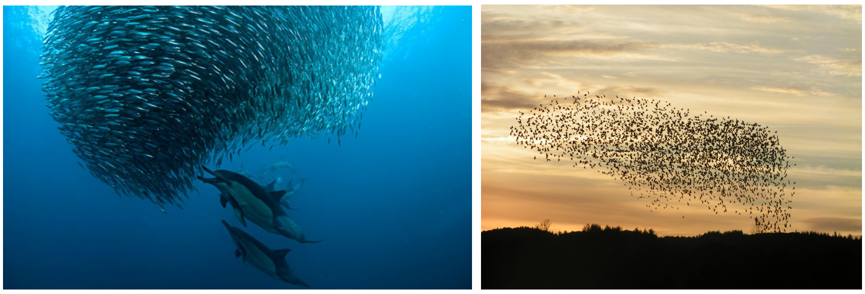
Figure 2.1: (a): Train of ants (https://www.proofpest.com/michigan-ant-indentificationnorthvillemichigan/); (b): Collective decision making in robotics bio-inspired by nest-site selection in honeybees colonies, (G. Valentini et al. [101])

For example, in [86] is defined as: *"the study of how large number of relatively simple physically embodied agents can be designed such that a desired collective behavior emerges from the local interactions among agents and between the agents and the environment"*, while [17], [20] define it as: *"the study of how to design groups of robots that operate without relying on any external infrastructure or on any form of centralized control"*. The ideal outcome of a swarm robotics system is a collective behaviour that performs as desired in order to find the solution of a specific problem [9],[86].

Let us define two concepts that will be useful in the rest of the writing: *swarm level* and *individual level*. Swarm level, or *macroscopic* level, is the high-level point of view of the system. If we speak about macroscopic level we are referring to the properties or the behaviours of the whole swarm as unique identity. Individual level, or *microscopic* level, is the single individual point of view, that is the characteristics of the single individuals and their interactions.

Usually, if a system is taken with a swarm-level approach then the modelling approach is a top-down one: the design starts from the desired properties and behaviours of the swarm, usually through ordinary differential equations or other mathematical models. Otherwise, if the approach is individual-level, the swarm is designed following a bottom-up method: the designers start modelling the single-robot behaviour in order to reach some desired properties of the whole swarm.

Ross Ashby, in his treatment about self-organization [3], gives two meanings to this concept. It says that a self-organized system starts with every agent separated from each other, that is, every agent is behaving independently from the other agent's state. After the starting steps, agents begin to create some kind of relations (connections) with the others. The parts-



(a) (b)

Figure 2.2: Examples of collective behaviour in nature: (a). School of fish aggregating to defence from aggressions (www.archives.deccanchronicle.com); (b). Groups of birds flocking (https://en.wikipedia.org)

separated system become hence a parts-joined system. The second meaning of self-organization given by Ashby is still referring to the first one (i.e. from parts-separated to parts-joined system) but it adds a factor to this definition. The system is not just changing from unorganized to organized but from bad organized to correctly organized.

One of the most studied self-organized natural behaviour is from ant colonies and perfectly represents an example of the concept of synergy just discussed: an ant alone cannot achieve complex tasks, but a colony of ants can construct nests, carrying food and so on. One of the most incredible behaviour of ants colonies is the selection of the closest source of food with respect to the nest and the shortest path to reach it [48]. In order to achieve this task, every ant in the colony releases a pheromone while walking to the food source, and decide which pheromone trail to follow based on other pheromones released on the floor. This principle has been exploited for an important optimization algorithm for the solution of computational problems, reducible to the shortest path finding in a graph (Ant Colony Optimization, by Dorigo, [18]), totally inspired by swarm intelligence.

Other insects having swarm behaviours are, as said, the honeybees. An interesting behaviour of the honeybees colonies is the process called *swarming*, where the queen and a part of the colony move from one nest to a new one. The process follows the nest site selection, where the swarm really follow a collective decision making process in order to find the best nest between the different alternatives [72], [108]. An application of this behaviour to a robot scenario has been presented in [106], [107], [105], [101]. A swarm of 100 simple robots was put in an environment with two available nests to explore. The goal was to decide which one was the best to move in. The behaviour followed by the swarm was inspired by the house chasing behaviour of honeybees. Nest-site selection is particularly important for this thesis work, we will discuss it better later.

Not only insects have this kind of autonomous and self-organizing behaviours but also more complex animals, as fishes and birds. These two categories are visibly acting in a collectively way, just think about the image we have of flock of birds flying in big and coordinated groups, or fishes in a schools swimming all together in order to defence against predator and in order to accomplish travelling and foraging processes [78], [71], [23].

Despite the synchronized operations of social animals the main characteristic of a swarm is the non-centralized form of coordination of the group and the local interactions among group members and between group members and the environment surrounding them. Even without a centralized control the swarm keeps three fundamental system-level properties desirable for swarm robotics too: *flexibility*, *robustness* and *scalability.* A key concept we want to highlight in this introduction of swarm robotics is that this research field is aiming to give to the designed systems these three characteristics.

Robustness: A robust swarm is able to keep operating in the desired way even if some robots fail, although with lower performance [86]. This property is mainly ensured by four key elements characterizing a swarm: *redundancy*, *decentralized coordination*, *simplicity of the robots* and *distribuited* sensing among the robots.

Flexibility: A system has the property to be flexible if it is able, without changing the algorithm at individual level, to maintain the same swarm-level behaviour even in presence of some environmental changing. Moreover, the single robot has to be able to *dynamically relocating* itself to different tasks to adapt to the specific environment and operating conditions [65], [9].

Scalability: This property has been defined, in literature, quite in a common way. Scalability refers to the property for a swarm of keep performing with the same output with a different *swarm sizes* [86]. The swarm size is the *cardinality* of the swarm, that is, the number of individuals (robots in robotic swarms, or animals in natural swarms) involved in the process. Both the swarm-level and the individual-level behaviour should not change adding or removing elements [9]. Swarm performance should show graceful degradation: even adding elements the swarm performance should keep growing until a bottleneck point, where communication and coordination between robots are too complex to be managed and the system fails.

Flexibility, scalability, and robustness are not enough to correctly distinguish swarm robotics research from other kind of multi-robots ones (e.g., collective robotics [59], [52], robot-colonies [2]). A swarm robotics system can be more precisely defined with other important features of the swarm and of the single individuals. The desired shape is a swarm with a *large swarm size*, made of *autonomous* and relatively *inefficient* robots with *local capabilities* of sensing and communication.

We want to underline that the sensing and communication capabilities of each robot must be locally situated. Every robot should be able to communicate with the neighbours without leaning on a global communicational channel. If the robots communicate and sense locally, both these aspects would be distributed in the environment, leaving the properties of scalability and robustness [86], [65], [9].

2.1.2 Overview of Swarm Robotics

Sahin et al. [86] gives examples of *possible* real-world applications of swarm robotics, subdividing the tasks basing on properties matching with swarm robotics ones. Brambilla et al. [9] instead subdivides the literature of swarm robotics in the exhibited collective behaviours (i.e. Spatially-organizing, navigation, and collective decision making, that will be discussed in the next paragraphs), giving examples of already done works. We will follow this guideline to draw a short review of these works.

Spatially-organizing behaviours are those situations where the components of the swarm have to spatially dispose themselves (and optionally objects) in the environment following a defined pattern or configuration, as for example put all the elements of the swarm in a delimited region of the environment (*aggregation*) [92], [35], [99] or in determined patterns (*pattern formation*) [4], [93], [91], [26].

In *Navigation behaviours* the robots have to find a way to navigate the robots from a point to another one, in order to transport objects or to explore the environment in a determined way [94], [73], [68], [67], [40], [24], [46].

Collective decision making, which we focus in this thesis (2.2), treats the problems where the robots have to influence the other members of the swarm in order to reach a final unanimous decision. The two main subcategories of collective decision making problems are agreement (on an opinion) and task allocation. Collective decision making, being the area of interest of this thesis, will be deeply discussed in the next section (2.2). Here we just list some works referred to the main subcategories of the collective decision making: [37], [27], [42], [101], [43], [70], [64], [110], [106], [51], [60], [76], [77], [112].

2.1.3 Open Challenges of Swarm Robotics

Swarm robotics has not yet been adopted in real world problems due to some limitations. Some steps in technology, either at hardware and designing/modelling/analysing level, must be done to export swarm robotics in real-world applications.

The large number of robots involved in swarm robotics systems require a cheap and small robots. The *lack of dedicated hardware* for this kind of robots is still an open issues that is going hopefully to be improved with the improving of technology. Indeed, the production of the device needed for swarm robotics is still at research level and no mass production of suitable robots with the integration of mechanical, sensors and actuators has been made yet, even if technology is improving a lot opening a way to swarm robotics [86]. Power consumption is another hardware limit: actual robots do not allow swarm robotics to perform for long periods of time. If we image a situation where a big environment has to be cleaned we realize that, in real world application, it is more than credible that a swarm robotics task is requiring a long time to be executed.

Design and modelling of a swarm of robots is the second crucial factor. Up to now no engineering way to design and model a swarm have been studied. The most used approach is a kind of trial and error one, where designer work on single robots behaviour, taking care of their interactions and the interaction with the environment. Even though some top-down approach has been proposed (e.g., [59] [55]) and some approach to tackle the micro-macro link problem (2.1.1) [82] has been studied, is still missing an engineering and standardized approach to face the following phases of the design:

- Modelling and specify requirements of the system as a whole;
- Design and realize the system, including every desired property and outcome. The lack in designing, as said, is about top-down design. Even some methods have been proposed (Brambilla et al. [9] Section 2.1) there are still limitations in the generalization of this process. Proposed methods tends to require a knowledge of the domain. A new tool for the realization of macroscopic level designed systems seems to has just been developed by Pinciroli et al. [74]. Their simulator is a tool allowing to both design the system from a microscopic level or

from a macroscopic one;

- Verification and validation of the system: there are still no guarantees on the outcome of the system. Liveness (property of a system to show the desired outcome) and safety (property of the system to do not show a not-desired outcome) analysis are still not strongly studied [111];
- Maintainance of the system;

Limitations are also due to the absence of a valid way for a simple human-swarm interaction: researchers are still looking for a way to let humans communicate with the swarms in order, for example, to coordinate effectively and control the swarm once it started to operate. This issue still needs to be studied and studied deeper, but some preliminary studies have been conducted to help humans to cooperate with swarms. McLurkin et al. [62] proposed a system communicating to the engineer through LEDs and sounds. Podevjin [79] proposed a method to give commands to the swarm using the Microsoft Kinect system.

For a complete discussion about swarm robotics issues, from a designing point of view, refer to Barca et al. [6].

2.2 Collective Decision Making

From an high level perspective, collective decision making is that category of swarm robotics' problems where the swarm has to reach a general consensus on some option in a set of possible ones. The consensus is reached when every element of the swarm (or in some cases, the *large* majority of the elements of the swarm) is preferring the same option in the set of alternatives. The emerging behaviour is a collective choice that can be, for example, which is the shortest path for the robots to reach a defined location, or which is the best place where to collect a certain resource in the environment.

An interesting parallel to the collective decision-making problems as defined in robotics' literature can be found in nature: social insects are simple individual cognitive agents able to take individual decisions. They are just informed about some local information, for example on the surrounding environment or the status of the neighbour elements [98]. Through direct or indirect communication [9] the group of insects is able to reach a final state where every individual has taken the same choice. An example of perfect intelligent-collective decision can be found in Tereshko et al. [98], where the authors inspect how a swarm of honeybees can always find a collective decision about the source of food to select through indirect communication (e.g.

waggle dance, pheromone trail laying, stridulation), even if the environment is wide and rapidly changing over the time.

The individual decision of an element (either a robot or an insect) is the result of the process of gathering information from the environment. Instead, collective decisions in swarm robotics (and in groups in general) are emerging from the self-organization process of the robots. Usually the collective decision-making process is composed by the phase of exploration, in order to gather information, and the information pooling. After all the information has been collected, every single robot has to take a decision basing on them. Through numerous local communication among the robots and with the environment and without a centralized control a collective decision can be reached [12], [100].

Two big subclasses of collective decision making are agreement (or consensus achievement) and specialization [9]. In agreement the desired outcome is that every robot, or a large majority of the robots, is converging, after the execution, on the same option among the set of possible one. In specialization, instead, the robots should distribute themselves on a set of possible tasks that must be executed. The most common example of specialization is task allocation, that is how to allocate the robots to a set of known tasks in order to maximize the performance of the system. An example is the cleaning of one room: let us suppose that, in order to clean a room, two tasks must be achieved: the first step is to remove all the object on the floor while the second is to distribute the robots on the floor and clean the destined area. The collective decision-making problem concerns the allocation of these tasks among the robots in a way that optimize the cleaning of the room.

Agreement has a wide area of application. It indeed concerns the agreement of the whole swarm on a single decision, that can be of every type. Examples are findable in navigation problems, where the group has to decide which direction to follow. It is a collective decision-making problem among a continuous set of alternatives (i.e. the infinite directions that can be followed), as in flocking problems.

The possible alternatives are called options, and could be of different types, depending from the problem: the possible nests to discriminate and choose, the set of different resources that the robots have to pick up present in the environment, the locations where to perform some task, the direction to follow for the swarm etc. [69]. The set of options could be both continuous or discrete. In this case, the problem is called best-of-n decision making problem, where n represents the cardinality of the set of alternatives. The problem treated in this thesis is falling in this subcategory: the swarm has to decide one option between a discrete set of alternatives.

Every option usually has an associated *quality*, that is the attractiveness of the relative option. Each robot has to evaluate the quality of the options in order to get its own opinion about the best option and to communicate it to the other elements of the swarm. The qualities could be easily measured or not, they could be spread in all the environment or concentrated in certain locations. A possible example to clarify the concept of quality is the following: Wessnitzer et al. [110] proposed a best-of-n decision-making problem where the swarm has to " chase" two moving targets. The swarm has firstly to decide which target to chase first, and then move to chase it. They proposed two versions of the behaviour, changing the collective decision part of the problem: the selection of which target chase first. In the first case the swarm has to chase the closest target, while in the second one, the majority rule is applied on the components of the swarm to select the first target to chase.

The set of possible options is then composed by the two targets while the set of associated qualities varies in the two cases. In the first one the quality is a physical and measurable value: the distance of the target from the nest. In the second case the quality is not physical and is not physically measurable. Indeed, it is represented by the number of robots voting for the associated opinion (target).

The goal for this kind of problem is to have, after the execution of the experiment, every robot (or the majority) of the swarm converging toward the same opinion, possibly the one that maximizes some measure of the system. One of the biggest problems in a decentralized system making a best-of-n decision is that each robot has just a partial information about the system and it opinions. It requires strategies to make the robots communicate , spreading the information, and applying some algorithms to select one opinion.

Examples of this kind of class in literature can be find in foraging (in Gutierrez et al. [42] the swarm has to discriminate two foraging areas in order to understand the closest one), nest-site selection (in Valentini et al. [101] the goal of the swarm is to decide which one will be the new nest site between two alternatives, characterized by a quality), or again aggregation (in Francesca et al. [27]).

Best-of-n is a subclass of collective decision-making problems because of the set of opinions: the set of opinions must be discrete and there must be one opinion that is better than the other. These characteristics make, for example, flocking not included in best-of-n decision making, until is not casted to a discrete set of opinion. Indeed, the possible directions are not a finite set and, moreover, there are no opinions better than the other.

Reina et al. [81] proposed a cognitive design pattern for a collective decision-making problem for a decentralized swarm of self-organizing robots.

2.2.1 Overview of Collective Decision Making

Studies on collective decision-making processes has largely considered *pheromone-lying* and *pheromone-following* to tackle and solve collective decision-making problems such as the selection of the shortest path [64].

Pheromones are chemical signals that organisms as ants release on the ground in order to communicate with other organisms. The use of pheromones in artificial systems has been implemented with both real chemical substances [85], [29], [30] and with virtualization. Engineers simulated pheromones in different ways: by projecting images on the floor, in order to emulate a pheromone trail [96], [36], [44], or by exchanging messages [13]. Another way to use a pheromone-like strategy has been studied by modifying the environment: in some cases [57], [53], RFID tags were put in the environment in order to be wrote or read by the robots, hence simulating the pheromones trails. Another study of simulation of pheromones is represented by the use of a paint covered floor that glows if irradiated by ultraviolet LEDs [61]. Finally, some works represent pheromones by actual robots [73], [66], [67], [22].

Pheromones simulations, as they have been developed so far, have important limitations: chemical traces require complex specific sensors that make the robots expensive and less reliable; projecting lights requires controlled conditions and is, consequently, not adaptable to unknown environment; using robots has "pheromones" is not robust: a robot might eventually fail and it would be critic for the system. Furthermore the use of specialized robot (hence, more complex robots) would play against the simplicity required in a swarm robotics system; modifications of the environment (e.g., RFID and special-painted floor) are cheap solutions but require the modification of the environment before the experiment. This is not always possible to be done.

A large part of collective behaviours are characterized by a quality-dependent decision-making problem. Indeed, collective behaviours as shortest path finding or collective transport need as a prerequisite to solve a collective decision-making process and choose the alternative to exploit. Solutions proposed for the problems falling in this category are usually characterized by two basic behavioural phases: a process for *quality-based discrimination* of the alternatives and the decision-making process [101] [12].

A problem with a discrete set of options needs a strategy in order to solve the so called best-of-n decision-making problem and to find the most valued option in the set. This distributed process relies on the handling of the information gathered from the environment about the quality of the alternatives in order to influence the whole swarm (or a majority of it) toward the best opinion. This process, called *modulation of the positive feedback* [38], is based on the amplification or inhibition of the period of time in which robots take part into the decision-making process by spreading their opinion for a duration proportional to the opinions' quality estimated.

Previously studied algorithms are strongly related to the environment, in the sense that the modulation of positive feedback process uses methods that are domain specific and hence difficult to transfer on other scenarios. Moreover, the modulation of positive feedback can be direct or indirect; in direct solutions robots are communicating directly with each others and eventually apply a decision rule in order to take the decision, while in indirect ones the robots are communicating through the environment. In direct modulation the robots are directly modifying the positive feedback: it could be for example that they amplify or shrink the period of time in which they spread their opinion. In indirect modulation instead the robots' behaviour does not change, but the spread of the opinions is modulated by the environment composition. Finally there are cases where the modulation is still quality dependent, but the proposed algorithm has been using abstracted qualities for the opinions, making the solution portable in several cases of best-of-n problems [101].

Other collective decision-making problems are those where the set of options is continuous and the options are equally-valued. In these problems the consensus achievement process does not require a quality-based discrimination process.

In this chapter we are going to split the studies made in literature in this way:

- Systems with discrete set of opinions;
- Systems with continuous set of opinions;

2.2.1.1 Discrete Decision-Making Systems

Collective decision-making problems are often characterized by a finite number of alternatives, called options. Often (but not always) problems characterized by a finite number of options that are characterized by qualities. They are called best-of-n decision-making problems. In these kind of problems the goal is to have all the robots (or a large majority) of the swarm converging to the opinion with the highest quality.

We are going to study in this section the works relative to the problem with a discrete set of alternatives, describing some works adopting the direct form of modulation of the positive feedback and some other utilizing the indirect one.

Montes et al., 2011 [64]: in this paper, the authors defined a collective decision-making strategy by building on the work of Krapivsky and Redner [50]. The problem can be casted to a best-of-n decision-making problem: choose the best opinion in a set of two possible ones. More specifically, the problem to be solved by the swarm is the well-known double bridge problem (Goss et al. [39]). In this problem, the swarm has to choose the shortest path between two available paths that connect a pair of locations without measuring time or distance.

Following their algorithm, robots repeatedly apply the majority rule on small teams of three robots. Lambiotte et al. [54] first studied the concept of latency applied to majority rule. Latency is a period of time in which the robots can not be influenced by the others. Montes et al. introduced the concept of *differential latency*, that is, a case where the duration of the latency period is different for the two different opinions. The latency period is associated to the execution time of the actions from the robots. With this assumption, opinions could be associated to different latency periods.

The algorithm is simple: the robots have to repeatedly go from the starting point to the goal point through the selected path. Once in the starting point, the robots are in not-latent state and form k teams of 3 agents. Every agent in the team has its own opinion and broadcast it locally in the starting point. Concurrently every robot of the team collects the other opinions and apply majority rule, assuming the most shared one and transitioning to the latent state for the period of time associated to that action. After the application of the majority rule, every component of the team has the same opinion and the robots can execute the selected action (going to the goal point passing through the selected path). Once the action has been executed the robots pass to the not-latency state and the process can restart.

A further analysis on majority rule with differential latency applied to swarm of robots performing collective decision-making has been done in Valentini et al. [104]. They analysed the previously done works with homogeneous Markov chains with finite state space, with the aim to demonstrate that the system is absorbing [49], [87].

A. Brutschy et al., 2012 [10]: Brutschy et al. took inspiration from the

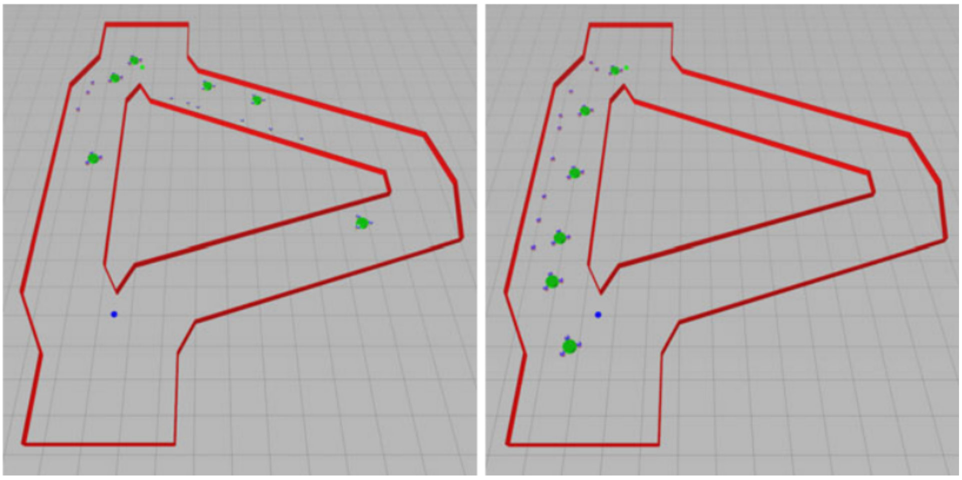


Figure 2.3: Double-bridge problem scenario: on the left the swarm is running the experiment, while on the right the swarm has chosen the shortest path. By **Montes et al.: Majority-rule opinion dynamics with differential latency: a mechanism for self-organized collective decision-making**.

work of Amé et al. [1], as done by Campo et al. [64]. The decision rule used by the individuals to drive the collective decision toward the best opinion. Brutschy et al., always keeping into the account the shortest path to link two points, put some constraint in the algorithm followed by the robots of Montes et al..

The robots have to travel along the paths between the start point and the goal point. The two paths, representing the two options, have two different execution time (they emulate the execution of two different actions). Instead of applying the majority rule, the individuals apply the k-unanimity rule defined as follows: a robot has a memory window where to store the opinion of the encountered robots with a *FIFO* logic. When the window is full it erases the oldest listed opinion and store the new one. When the robot listens consequently K opinions all agreeing with each other it adopts this opinion.

Brutschy et al. introduce a constraint to the classic k-unanimity rule. Usually, the robots can listen the other opinions in every moment they meet another robot. In their work, instead, robots can exchange information only in the observation point. The observation point is the starting point and the duration of the observation state is fixed and equal for every robots, independently from their opinion. This key factor is the one that allows to the swarm to collectively reach a consensus. Since the robots committing to the best opinion are travelling along the shortest path, they will return to the starting point with an higher rate with respect to the robots choosing the suboptimal path. In this way, they are spreading their opinion an higher number of times. The probability to find robots with the best opinion in the starting point is hence higher than the one to find the robots fevering the less valued opinion.

The modulation of the positive feedback is made by the environment. The robots travelling the shortest path are spreading the opinion more often. The quality is not measured and not taken into account by robots, i.e., robots do not know anything about the environment and the path qualities.

Campo et al., 2010 [12]: In this work the researcher took as started point the work of Amé et al. [1]. Amé et al. explained how the cockroaches collectively choose a shelter where to hide. The scenario (from Amé et al.) is an environment with several shelters available for the cockroaches where to aggregate. The cockroaches aggregate in a shelter as big as what is required by the colony of cockroaches. Amé stated that the cockroaches are not only aggregating under the biggest shelter; they are looking for a shelter big enough to host all of them but not larger than what is required, in order to avoid concurrency problems with other potential cockroaches colonies and risky situations.

Campo et al. modelled the behaviour of the cockroaches in the following way. The agents are exploring the environment until a shelter is found. At this point, the agent stays in the shelter. The probability for the agent to leave the shelter is inversely proportional to the number of other agents under the shelter. Campo et al. adopted this behaviour with some adaptation. The cockroaches are substituted by artificial robots. The shelters are representing a generic resource in the environment and its surface is the capacity (availability) of the resource. Several resources with different capacities (differently with respect of Amé et al.) are placed in the environment. The total need of the swarm of robots is the sum of the surfaces of every agent. From this point we will use shelter and resource as synonyms because, as said, they can be represented in the same way.

The problem tackled by Campo et al. is such that the robots do not have enough capabilities to understand the dimension of the shelter (the capacity of the resource) or the number of robots in the shelter. They had to adapt the behaviour with an adhoc method in order to estimate the number of robots present in the resource and then calculate the probability to leave the shelter. Once arrived in the resource area they keep performing a random walk, keeping trace of the encountered robots in the area. In this way they can have an estimation of the crowd in the resource and calculate their probability to leave or not.

In this work there is no direct communication between robots. We can see the presence of the robots in the resource area as a kind of positive feedback. Staying under a shelter means, for them, that is the right resource. they represent in such a way the quality of that area: many robots mean higher quality with respect to areas with fewer robots. They are then indirectly modulating their positive feedback, through the environment. The better the resource area fits the needs of the swarm the longer time the robots are going to stay there.

Francesca et al. [27] replicated this work by using evolutionary robotics and proposed a comparison between the two works with a macroscopic model. In this work the authors make the swarm reaches the consensus using a memoryless behaviour, that is, by using only the values of the sensor relative to that time step. They use as controller a fully connected, feed-forward neural network that transforms the 12 inputs, relative to the sensors, into a two-lines output, one for each wheel.

Wessnitzer and Melhuish, 2003 [110]: In this work, they proposed a swarm performing collective decision-making in order to decide which prey to chase, before to collectively move toward the decided target. It is showing, then, a collective decision-making followed by a target chasing collective behaviour. The experiment is made of two target robots and a group of chaser robots. The target robots are moving at the double of the velocity of the predator robots. The goal of the swarm is to collectively decide which target to chase and consequently chase it. In their paper, the authors studied how three different local rules can bring the swarm to collective decision. Swarming through 1) local direction control, 2) majority rule, and 3) hormone-inspired decision making.

A *local direction control* is used to keep the swarm compact. Initially every robot is placed in a corner of the environment and has initial direction pointing to the opposite corner of the environment. After the beginning of the experiment, every robot keeps measuring the distances from the two preys, without knowing the direction to follow. The measured distances are compared step by step with the ones measured by the neighbours. If the distance of a robot to the target is smaller with respect that of the other robots then, it keeps going in the actual direction, otherwise it moves toward the direction of the robot with the minimum distance from the target.

Majority rule is used instead of collective deciding which prey to chase first. Initially, a random value is set for the preference (opinion) of each robot. The opinion can be with equal probability both 1 or 2, that means, respectively, chase first the robot target 1 or the robot target 2. At every time step, each robot evaluates the opinion of the neighbours and takes as his opinion the majority of the sensed opinions.

Hormone-inspired decision-making: it makes the swarm decide if the target has been chased and, thus, the swarms' behaviour has to change. The object of this algorithm is that the swarm should recognize when the prey has been chased and switch the behaviour to catch the other prey. The stagnation state (i.e., the state when the target has been chased and the chasing robots are not moving any more) is recognized if the estimated distance from the target is changing less than a decided value (let us call it ϵ). If a robot recognizes to be in a stagnation state for a sufficiently long period of time, it sends a message to the neighbours. This message contains the number of robots agreeing to switch to the next behaviour (hence, if a robot agrees in changing behaviour it forward the message incrementing the number of robots agreeing).

This experiment used direct communication and showed how to apply collective decision making as base for a collective behaviour. The majority rule was used to just break the symmetry and decide which object to chase first. A bio-inspired algorithm was instead used to actually choose whether to switch algorithm or not. In this algorithm the communication was actually direct among the robots. Every robot was transmitting messages spreading his own opinion adding to the opinion of the neighbours its own opinion.

Schmickl amd Crailsheim, 2008 [88]: The proposed algorithm is exploiting trophallaxis [14] in order to achieve a collective behaviour in foraging. The environment is composed by dirt particles spread on the ground and a dump area, where the dirt particles are supposed to be drop at the end of the experiment. The robots are moving toward the garbage source that is in an unknown position. Once reached it they pick up a dirt particle in order to drop it in a dump area. Trophallaxis is a form of coordination performed by social insects (specifically by bees). Transported to the robots scenario, Schmickl and Crailsheim proposed the following behaviour: robots are moving randomly in the environment performing obstacle avoidance. To emulate the thropallaxis behaviour they have an internal variable that simulates the food carried by the honeybees. If two robots are meeting there is a transfer of "virtual" food from the robot with an higher value of virtual food to the one with a lower one. Once reached the dirt source this variable is set to the maximum.

When the dirt particle is dropped down, is instead reset. In this way a gradient between the dump area and the dirt source is created. If a robot is looking for the dirt source it will scale up the gradient. Otherwise, if the robot has as target to drop down a particle, it will scale the gradient up.

Gutierrez et al., 2010 [42]: Gutierrez et al. showed an interesting collective decision-making behaviour also inspired by trophallaxis of honeybees and extended with sensorial capacities of the robots. Several food sources are spread in the environment and a decentralized swarm of autonomous robots has to find the closest one. This is an example of collective decision-making problem applied to foraging.

The robots are initially placed in the center of the scenario arena, without perceiving neither the food area nor the nest area. After the begin of the experiment the robots start performing a random walk until the food or the nest area are found. Once it happens the robots save the *location* of the areas and continuously try to go back and forth between the nest and the food area. Sometimes, it happens that the robots, due to some errors of the sensors and the actuators, or to some noise in the estimation of the location point, in this path lost itself and has to reset the saved locations. When two robots are meeting they exchange their *local* evaluation of the positions of the locations.

This peer-to-peer communication is the basis for the collective decision-making strategy. Let's assume the following as hypothesis, in order to simplify the explanation:

- Every robot calculates a value that is its own level of confidence in the food source location estimation;
- This estimation varies according to the distance travelled. If the robot travels for a long time before to reach the food source in the estimated location, then the level of confidence is lowered;
- When two robots are exchanging information the robot with a lower level of confidence adopts the location of the other robot;

This mechanism of exchange information brings the system to choose the shortest path to reach the food source. Gutierrez et al. developed a collective decision-making strategy that uses the direct modulation of positive feedback: the robots are "weighting" the importance of the exchanged information by using the level of confidence given by the travelled path. It can also be seen as an indirect way to modulate the positive feedback, because the modulation is actually given by the time needed to travel the path.

Parker and Zhang, 2009 [69]: In their work, the authors proposed a generic algorithm for nest-site selection, totally based on the algorithm followed by the honeybees and a specific type of ant in the nest-site process [56], [90], [80]. They proposed a solution for the best-of-n decision-making problem applicable to a decentralized swarm of autonomous robots. The goal of the swarm is to find, among a set of possible alternatives, the nest with the highest quality.

The proposed solution is based on local form of communication, that allows many peer-to-peer parallel communications. The exchange of information between robots is really important and is the key factor for the goal achievement. The individual behaviour of the robots of the swarm is subdivided into six states:

- *Idle*: the robots in idle state are stopped, waiting to be recruited into the process;
- *Searching*: the robots in research state are randomly exploring the environment in order to find the alternative nests;
- *Advocating*: in advocating state, robots are re-joining their team-mates and sending recruitment messages to them;
- *Researching*: robots in this state are evaluating the quality of a particular nest-site;
- *Committed*: robots sending just committed messages to encountered robots;
- *Finished*: robots which have finished the process;

The transitions between states represent the behaviour of the robots. Initially the robots can be both in the idle or in the searching state. Once they find a nest (in the searching state) they determine the quality of it and then enter the advocating state, going back to the team-mates. The task of the robots in the advocating state is to periodically send recruitment messages to other robots, trying to advise them about the estimated quality of the explored nest. Robots in the idle, searching and advocating states can be influenced by the recruitment messages. As soon as a robot in these states receives a recruitment messages it transits in the researching state, and goes to evaluate the quality of the specified site before going back to the advocating state. In order to evaluate the quality of the nest, robots send query messages to understand if the other robots are of the same idea. The more robots agree with the goodness of that nest, the more the robot is giving an high quality of that opinion.

When the robot thinks that his alternative is popular enough it moves to the committed state, and starts sending committed messages. When another robot receives such a message, two situations are possible: 1) the robot was already in commitment state, so that it doesn't do anything; or 2) the robot was not in commitment state. In this situation it transits to commitment state and reply with an acknowledge message. This is the key factor that determines the collective agreement. When a robot does not receive acknowledge messages to its commitment message for a long enough period of time it passes to a finish state and where it definitively chooses that opinion.

The primary points of this algorithm are the follows:

- No direct comparison between robots' estimates are done. Indeed, any robots can do some error in quality evaluation. It can happen that a robot is fevering the best option but have done an under-estimation of the quality, due to the noise. If this robot communicates with a robot fevering the not optimal option but with a quality still better than the other robots' quality, it can switch the opinion. However, probabilistically there is a much higher probability that the opposite way changing happens. The lack of direct comparison avoids that robots that are erroneously overestimating the quality of some alternatives can erroneously influence other robots correctly thinking about another opinion;

- Direct local communication is another crucial factor: the robots are locally and directly exchanging information about the estimation. It allows to have a shared and distributed information among the swarm;

- Direct modulation of the positive feedback: the robots are sending the recruitment messages with a ratio that is directly proportional with their quality estimation;

Reina et al, 2014 [81]: Reina et al. proposed a cognitive design pattern for collective decision making. They studied an analytical model for the nest-site selection process in a binary-choice scenario. The swarm is subdivided in three groups: uncommitted individuals, with a population of N_u robots, and individual committed to one of the two alternatives, with a population of respectively N_a and N_b robots. There are 4 types of transitions available that fully describe the behaviour of the individuals: *discovery*, *abandonment*, *recruitment* and *cross-inhibition*.

They showed their solution in a shortest path selection scenario, to ease the comprehension of the analysis they have done. The proposed scenario

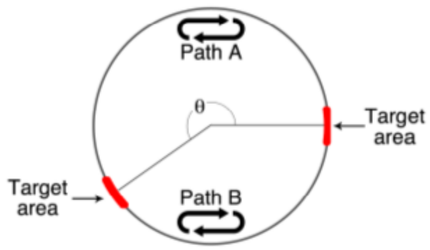


Figure 2.4: A graphical representation of the multi-agent scenario. The monodimensional environment is a circle in which the agents move on the circumference line to navigate back and forth between the two target areas. By Reina et al, 2014b: Towards a Cognitive Design Pattern for Collective Decision-Making

was a circular path with two target areas (see Figure 2.4). The robots move on the border of the circle and have to select the shortest path to travel between the two target areas. To evaluate the distances, robots have to travel the paths and estimate the distance using dead reckoning. However, due to noises in the movements, estimated positions are having cumulative errors.

Reina et al. developed an analytical pattern to follow in order to design the behaviours for the collective decision making, and proposed a solution of this specific problem by following the proposed pattern. When a robot, randomly walking, discovers the two areas it stores the local positions of the areas and commit itself to the last travelled path. The probability to commit to the shortest path is higher since it is easier to find the two points following the shortest path. Once the two points are discovered, the robot starts going from one target area to the other one by following the selected path. When the robot fails in reaching the target area, it moves to abandoned state, abandoning its commitment and passing to uncommitted state. It also erases the stored information about the position.

The interaction part is the most important one. First of all Reina et al. gave some rules to well mix the interactions. The robots can communicate only if they are in one of the two target areas. The robots, once in the target area, stay there with a probability of 0.9. If, while in the target area, a state changes then the robot goes out from it. When two robots are interacting with each other there are two possibilities. The first, one of the two robots is committed while the other is not, in this situation the uncommitted robot changes commitment to the same opinion of the encountered robot with a fixed probability (P_p) and receives the estimations about the locations of the two target areas (*recruitment*). The second possibility is that the robots are both committed but to a different opinion. In this case, with a fixed probability different to the previous one (P_σ) the robot erases its estimate and switches to uncommitted state.

Valentini et al., 2014-2015 [106], [107], [105], [101]: In these works Valentini et al. studied the well known best-of-n decision problem applied to the nest-site selection problem, taking inspiration from honeybees nest-site selection and particularly from the waggle dance performed by honeybees in order to disseminate their opinion [89]. The scenario proposed (see Figure 2.1) was a rectangular arena with three distinct areas: the two sites at the extremities and the nest, in the middle, equally distanced by the two sites.

The particularity of these works were basically three: they decided to abstract the qualities of the nest from the real features; they used a large swarm of 100 real robots using the kilobots [84] and decided to decouple the modulation of the positive feedback from the particular decision rules. The qualities of sites were represented by beacons placed under the nests area, ρ_i, and was defined as: $\rho_i \in [0, 1]$.

The behaviour of the single robot is very simple and can be represented by a four-state probabilistic finite state machine. The robots can be either in waggle dance states ($W_a or W_b$) or in surveys states ($S_a or S_b$). The waggle dance states are performed just in the nest, while the survey concerns the evaluation of the alternative sites and is therefore done in the two candidate sites. Every robot in every time step has its own opinion that can be A, if it thinks that the quality of the site A is higher or B otherwise.

The robots initially start in survey states and go in the direction of the site that they have to explore. Once there, they evaluate the quality of the site and go back to the nest, where they change to waggle dance state. In the waggle dance state, robots perform the random walk and broadcast their own opinion within a limited range. The duration of the waggle dance state is modulated by the estimated quality of the site, that is going to modulate the parameter of an exponential random variable. Before to start the new survey state, the robots pool the information shared by the neighbours and choose their new opinion applying a decision rule to the information in this pool.

In the two works Valentini et al. decided to apply two different decision rules: first they applied the weighted voter model and lately they applied the majority rule. With the weighted voter model the robots pick randomly an opinion from the pool, while with the majority rule they apply the majority rule (described before) to those opinions. The pool is composed by the information received by the neighbours robots, since the communication is locally done within a limited range.

This algorithm is the reference algorithm for this thesis work.

2.2.1.2 Continuous Decision-Making Systems

Ferrante et al., 2010a [24]: the authors proposed a novel method to tackle the problem of collective transport in presence of obstacles. A group of three mobile robots (i.e., foot-bot, described in [21]) has to transport an object from a start to a goal location in an environment where obstacles are placed. The challenge of the paper was to make the robots negotiate about the direction to follow, since the perception of the environment of every robot is heterogeneous.

The authors decomposed the collective transportation problem into three sub-tasks: go to goal, obstacle avoidance and social mediation. Each robot has a different perception of the environment and therefore they are going to have different goals. A robot directly seeing the goal will try to reach it directly, while a robot perceiving an obstacle in front of it will try to perform obstacle avoidance. Hence, the agreeing about the direction to follow must be negotiated among the robots of the group. Let us call σ_p the desired direction of the single robot and σ_s the socially mediated direction. σ_p is the general alternative of the problem and, being a direction, is a continuous set.

When a robot has no perception of anything in the environment, neither the goal nor the obstacles, it will adopt as desired direction the average of the directions sensed from the neighbours and will broadcast locally this value. When instead a robot has an information, it will not keep calculating the direction as the average of the other robots' desired directions, but it will calculate the desired direction as a consequence of the perception of the environment. If it perceives an obstacle, it will try to avoid it and it will send this information to the neighbours. The final direction of the robot is obtained by averaging the sensed information with the own direction of the robot.

The decision about the direction to follow is obtained by the social mediation between the robots. If a robot does not perceive any obstacle or goal it just follows the other robots' information. Otherwise, if it senses the goal it just advice the other robots about it. In the moment it perceives an obstacle, instead, there are two possible situations: if the robot only perceives the obstacle and not the goal it just try to avoid it setting its desired direction as the direction to follow to avoid the object. If the robot perceives both the obstacle and the goal it has to find a way to reach the goal avoiding the obstacle and it has to mediate it with the other robots of the group. In this case, it has to average the direction to the goal with the direction to avoid the obstacle with a weighting factor related to the distance from the obstacle (i.e., how urgent is to avoid it). Finally, once the desired direction of the robot is computed, the process of social mediation takes part giving as the output the needed direction for each robot.

In this experiment, the swarm has to compute the collective decision about which direction to follow. This collective decision has an infinite set of opinions (i.e., the opinions, that are the possible directions to follow) among which to choose the best one.

Ferrante et al., 2012 [25]: authors present a novel approach to solve flocking with a generic algorithm requiring low capabilities from the robots. This method is based on magnitude-dependent motion control and does not lay on external hardware, alignment control algorithms or goal direction. The flocking vector (f) of each robot, that is the direction to follow in order to keep the flocking behaviour, is composed by three components: the proximal control vector (p), that encodes the attraction and repulsion rules, the alignment vector (a), describing the alignment rule and the goal direction to follow (g).

$$f = p + a + g \tag{2.1}$$

They adapted the flocking method also taking in consideration only certain elements of the flocking vector described above:

$$f = p \tag{2.2}$$

$$f = p + g \tag{2.3}$$

$$f = p + a \tag{2.4}$$

We are going to describe the three components of the vector focusing on the way to compute each component. Proximity control is given by the sensors and it takes into account the distances from the neighbours. The algorithm is intended to let the robots keep the distance from the neighbours within a range. The robots tend to be attracted by robots that are farer than a pre-choose distance and tend to be repulsed by robots that are too close. The alignment control instead computes an estimation of the average of the orientations of the neighbours and adapt the robots' own orientation according to this value. The goal direction is instead given by the physical direction that has to be followed in order to reach the final goal.

In the paper motion control algorithms are used in order to translate the flocking control vector calculated with the above described rules into the actual linear and angular velocity of the robots. The two methods are called MDMC and MIMC. With MDMC, the forward and the angular velocity of each robot depends directly from the magnitude and the direction of the flocking control vector described before. In MIMC, instead, forward and angular speed of the robots do not fully depend from the components of the flocking control vector: only the direction of the flocking control vector is keeping into the account.

The results show that the swarm reaches an ordered state only when using MDMC. They showed experiments with a medium-size swarm and a large-size swarm. With the medium-size swarm the ordered state is reached within 700 simulated seconds, while in the large-size swarm it happens within 1500 simulated seconds. Instead, when the MIMC is used, the system never reaches the ordered state.

Chapter 3

Environment Classification

Environment classification is a specific scenario that can be casted to a best-of-n decision-making problem. In this problem, a swarm has to classify the environment by the different resources it contains. Let us recall that the best-of-n decision-making problem (discussed in 2.2) is a collective decision-making problem where a swarm of robots has to discriminate the finite set of possible options and choose the most valued one.

We studied a self-organized, general, and portable solution to such a problem for a swarm of simple and autonomous robots with a form of decentralized control. The behaviour of the individuals that we have proposed follows a simple probabilistic finite-state machine. We proposed a solution with a direct form of communication and with a direct modulation of the positive feedback based on the quality of the option estimated by the robot. Following Valentini et al. [101] we decided to decouple the modulation of the positive feedback from the application of the decision making rule. We wanted to test the implementation of the individual agents' decisions with three different decision-making rules (*weighted voter model, direct comparison, and majority rule*) and to study how does the swarm react to the three different situations.

The basic behaviour of the robots is the same independently from the decision rules. The only exception is made for the use of the direct comparison. In this case the modulation of the positive feedback is not made because the quality is already compared by the robots in the communication

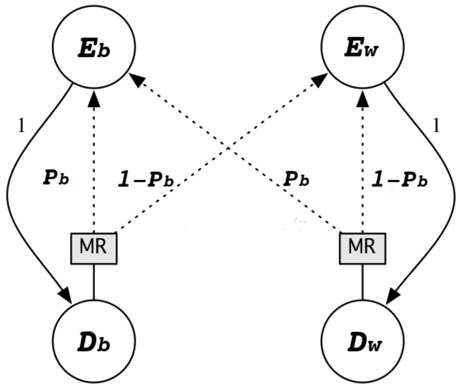


Figure 3.1: Probabilistic finite state automata describing the behaviour of the individuals. D_B and D_W are respectively representing the dissemination states of the black and white options. E_B and E_W are respectively representing the exploration states of the black and white options' qualities. P_B is the probability to pick up a black opinion from the pool and hence to pass to the dissemination state of the black opinion. 1-P_B is instead the probability to do not pass to black opinion dissemination state

phase.

3.1 Description of the Problem

Environment classification is a scenario where a decentralized swarm of autonomous simple robots has to decide which resource, among the present ones, is the most available in a closed environment. Every best-of-n problem is characterized by a set of options, each one related to a value called quality, a swarm of robots that has to solve the problem, and an environment where the problem is situated. In the environment classification problem, we can easily distinguish and identify each of those elements: we are working with a swarm of autonomous and simple robots, more specifically e-pucks, that must decide which is the resource (i.e., option) mostly available (i.e., highest quality) in the environment.

We can try to image some examples of real world applications of the environment classification: the classification of the garbage relative to an after-nuclear disaster, where a swarm of robots has the task of identify hazardous areas and clean the environment; or in human body scenario where the swarm must distinguishes areas containing cancer cells from healthy ones, or in the exploration of an extra-terrestrial scenario where the swarm needs to identify and classify the resources present on the unknown environment and decide if it suit construction.

We developed the environment classification in an experimental context, where the resources are easily understandable by the robots that we have used. The environment is a floor covered by black and white squares where the colors are representing the two resources. We can hence reduce the problem to the best-of-n decision problem with $n = 2$ options, represented by the two colors. The quality of the options is the availability of each resource, that is, the quantity of black and white cells on the floor. The solution of the problem is reached when the swarm reaches a consensus on one resource.

Briefly, the robots have to alternate two phases in order to solve the problem: in the first phase they have to explore the environment while in the second one they have to communicate their opinion about which is the best option. We will discuss better the behaviour of the robots in 3.2.

Since the resources are spread in the environment, the quality of each option is not easily estimable: each robot can, in the limited duration of the exploration state, explore only a local part of the environment. It does not allow the robots to have a global knowledge of the environment. Every time the robot finishes the exploration and dissemination state he erases all the data collected in the exploration state and has the possibility to make a new estimation of the quality. Moreover, the estimation is only a noisy quantification of the quality. The local knowledge about the environment allows the swarm to act even under different environmental conditions, giving a flexible character to the solution.

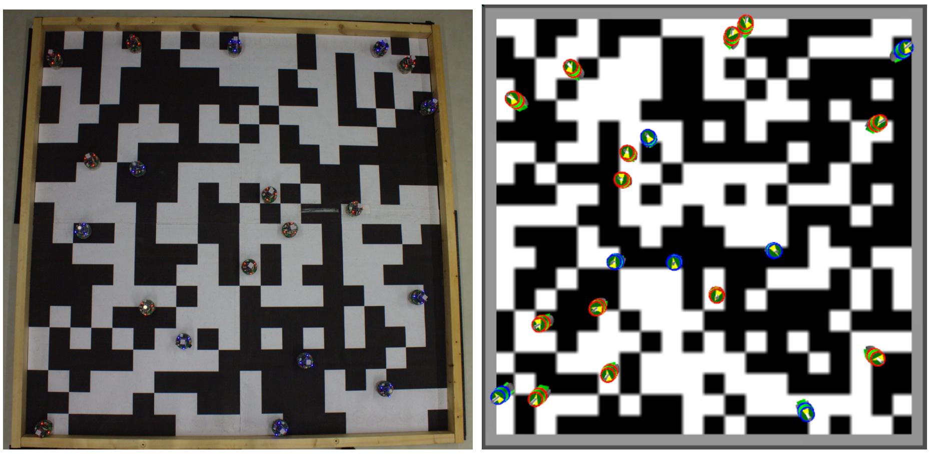
More over, the communication among the robots are direct and locally situated: every robot has a range of communication within broadcasts its own opinion. This feature, with the decentralized nature of the control, gives to the solution scalability in swarm size and robustness to eventual mistake and/or failures of some individuals.

Our solution presents an abstraction of the quality from its physical meaning (e.g., what kind of resource is the one that the robot is analysing). We assume that every robot has a sensor able to sense every resource, in our case the color under the body of the robot. The measured value returned from the robots after the exploration state is a bounded value $\in [0, 1]$. In this way the studied solution has a more general and portable nature. Generality of the solution is also supported by the fact that the set of available options is not known a-priori by robots in the swarm; each robots discovers them step by step, when encountering the different resources in the environment.

The behavioural finite state automata that we have used for the resolution of our problem is the same of Valentini et al. [101] previously developed to solve a nest-site selection problem (where the swarm had to decide which site to chose among two possible ones). Environment classification is a new scenario of a best-of-n decision-making problem. This scenario has never been exploited before and the analysis of a system in this new scenario is one of the main contributions of this thesis. The particularity of this scenario is the distribution of the quality of each resource. The resources, represented by the colours, are spread all over the floor of the environment. Differently from the other works, the qualities of the resources are not represented by a measure evaluable in a single point: for example, in a general problem of nest site selection (as [103]) the quality is directly measurable in single points, that are the candidate sites. In environment classification the distributed nature of the resource implies that every robot cannot directly measure the quality but has to make an estimation of it, by exploring the environment locally in every iteration.

We define two performance measures: *consensus time* and *exit probability*. Consensus time is the time required by the swarm to reach the global consensus, that is, every robot in the swarm has the same option. Once this state is reached no robots can change option. Due to the decision rules we are using, the only possibility is to adopt an opinion present in the information pool and, if every robot is agreeing on the same option, there is no way that in the pool there are different options. Achieving consensus does not imply that the swarm has chosen the most valued option: the swarm might erroneously converge on the wrong one. Exit probability concerns exactly this fact. It is the ratio of correctly taken decisions over the total number of trying that have been done, that depends from the initial conditions. We will discuss better these two concepts and the results obtained in chapter 4 and 5.

The goal of our work is to study the macroscopic behaviour of the swarm when three different decision-making rules are applied: weighted voter model, majority rule and direct comparison. We want to analyse, varying the initial conditions, how do the exit probability and consensus time variables change in this scenario - the well known speed vs accuracy problem ([28], [58], [72], [102], [101], [105], [97]). The comparison between the three decision rules is another innovative contribution of this work. In literature rarely have been done works about the comparison in the same scenario of three different strategies. We opted to use majority rule and weighted voter model, that are completely self-organizing strategies, and that have already been studied by some researcher in literature ([103], [101], [110], [54], [64], [104]). Direct comparison, instead, is a strategy that uses more information than the other analysed decision-making rules. We introduced direct comparison as control strategy, in order to



(a) (b)

Figure 3.2: Pictures of the environment classification scenario: (a). Swarm of 20 real-robots working on an hard decision-making problem in real experiments; (b). Swarm of 20 robots working in the scenario in simulation. Parameters of the scenario: 52% *black vs* 48% *white cells.*

evaluate the difference in performance between completely self-organizing strategies and direct comparison, to give a complete picture of the situation about when its more advantageous to use the different strategies. We first analyse the behaviour of the swarm by means of a simulator, and then, we will validate our results with real robots.

3.1.1 Scenario and Arena

We used a squared arena with a surface of 4 m^2. The arena is subdivided into 400 squares. Each square has a surface equal to 100 cm^2 and is either black or white, dependently from the initial conditions. The black and white cells are distributed in a completely random way on the floor at the beginning of each new run. The arena is enclosed by four walls that prevent the robots from leaving the experimental environment (See fig 3.2).

The initial condition characterizing the scenario were mainly: 1) the qualities of the resources, that are going to have a bearing on the difficulty of the problem; 2) the swarm size, that is going to weigh on the speed vs accuracy performances; 3) the initial number of robots fevering black or white opinions; and 4) the transition rates (σ and ρ), the parameters that define the mean time of the exploration and dissemination states.

We have studied two scenarios for what concerns the difficulty of the problem, where for difficulty of the problem we intend the difference in the qualities of the two resources. We choose to analyse two cases that we call *simple* and *difficult* scenario. In the simple case, we set a ratio between white and black cells of 2, that is, the number of the white cells was the double with respect to the ratio of the black cells. The percentages are 66% white cells vs 33% black cells. In the difficult case, we study a situation where the qualities of the two resources are closer to each other and the optimal solution is harder to discriminate. The ratio of white cells over the black ones is 0.923, that is, on the floor there is a percentage of white cells equal to 48% and a percentage of black ones of 52%. Without loss of generality we will assume from now to go on that the black cells are always more than the white ones, making the distinction between difficult and simple cases explained above.

For the initial condition of the swarm size we need to make a distinction between real-robots experiments and simulation ones. In simulation we varied the swarm size up to 100 robots. In real-robots experiments we instead choose to utilize 20 robots, due both to the robots availability of our laboratory and the physical limitations that an experiment with a largely swarm size would have entailed.

The element involved in the initial conditions is the proportion of robots favoring the opinions. An higher percentage of robots favoring the "best" option influences the macroscopic behaviour of the swarm both in term of consensus time and exit probability. In simulation, we decided to study every initial choice of this feature, while in real-robots experiment we decided to have a balanced situation adopting the 50% of the robots favoring the two opinions.

In every moment, the robots have an opinion about which alternative is the best option of the problem. In the exploration state, robots have to estimate the quality of their actual opinion. In order to do that, the robots have to understand when they are on a cell that is of the same color of their own actual opinion and keep track of the time spent on these cells. When the exploration state is finished, robots compute the proportion between the time spent on the opinions' cells and the total time spent in exploration in order to calculate the quality. This estimated quality is relative to a limited and local area. Next, robots use this value to modulate the time in which they broadcast (within a limited range) their current opinion (i.e., duration of dissemination state). The dissemination state is mainly composed by two partially overlapped phases. Initially the robots are just broadcasting the value but in the last seconds of the dissemination state the robots listen and save in a pool the information that the neighbours are communicating. In the last time step of the dissemination state, right before to go back to the exploration of the environment, the robots have to select a new opinion. The selection is made applying one of the three decision rules to the information collected in the pool.

The desired outcome is the *convergence* of the swarm on one of the two opinion, that is every robot of the swarm favoring the same opinion, possibly the most valued one (black).

It is useful to visualize the opinions of the robots during the evolution of each run in order to understand how are the dynamics of the swarm. To advise their own opinion the robots are turning on the LEDs and each opinion is settle to be represented by one color:

- *RED Leds:* if in a certain time step t a robot has RED lights on, it means that its opinion at time t is $BLACK$, hence that it thinks that the quality of the black resource is higher than the whites' one;

- *BLUE Leds:* $BLUE$ lights on, instead, means that the robot is believing that the most valued option is the white resource;

Since the colors of the LEDs are representing the opinions of the robots, thus, the color that every robot is thinking that is most available on the floor, probably you are wondering about the choice of the colors of the cells. It would be more rational to use red and blue cells for the floor and let the LEDs representing the red and blue options with, respectively, the red and blue LEDs. The reason for the choice of the floors' colors has been imposed by the capabilities of the sensors equipped on the robots we have used. Indeed, our robots are only able to sense grey-scale colors.

Additionally, we want the robots to communicate they internal state (exploration state or dissemination state). To distinguish the two cases, the robots blink the central LED when in the dissemination state, while keeping all LEDs fixed on in the exploration state.

3.1.2 Robots

For our experiment we choose to use E-pucks (fig 3.3). These robots are simple, small, open sources wheeled robots with a diameter of 7 cm designed and developed by Francesco Mondada and his team at EPFL (cole polytechnique fdrale de Lausanne), in 2006 [63]. E-pucks are equipped with a small set of default sensors: a low-resolution camera, an accelerometer, a sound sensor and 8 proximity sensors. Beside those sensors there is the possibility to add extra features to extend the capabilities of the e-pucks, as for example the Fly-Vision turret or the omnidirectional vision. We did not

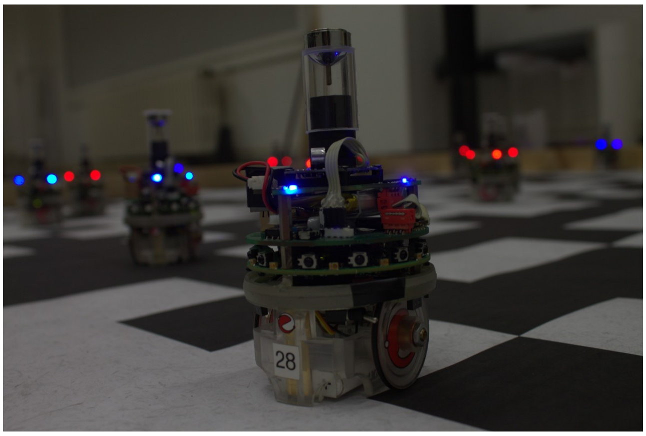


Figure 3.3: E-puck extended with range and bearing, Linux extension board and omni-directional camera. In this picture the e-puck is fevering the white opinion by turning on blue LEDs

use all the sensors available for e-pucks. In the following list there is a brief description of the used capabilities:

- 8 Infra-red proximity sensors placed all around the robot. By means of these sensors robots can perform obstacle detection. Proximity sensors return a value proportional to the distance with the detected object. Proximity sensors are also able to work as light sensors. We used these sensors in order to make the robots perform obstacle avoidance, that is stay away from the obstacle. In our case are the wall limiting the border of the arena and the other robots in the environment;

- Ground sensor: the ground sensor is composed by a PCB board which mounts three proximity sensors pointing directly the floor. We used the ground sensors to detect the color of the floor under the robot in each time step, in order to estimate the quality of the opinion;

- Range and bearing board: Gutierrez et al. [41] designed this local communication board allowing the robots to communicate within a determined range of distance and to sense at the same time both the range and the bearing of the emitter robot without the utilization of any other infrastructure or centralized control. The range of communication can be controlled through software from $0cm$ and $80cm$. On each board are mounted 12 IR emitter/receiver modules that are taking care of the sending and receiving part. Unluckily some limitations in the range part of the sensor board is present. The calculation of the range value is quite noisy and not always reliable;

- Linux extension board: it gives to the e-pucks all the characteristics of a processor running Linux, including the possibility to use an USB port to be linked to the pc or the possibility to use the Wifi network.

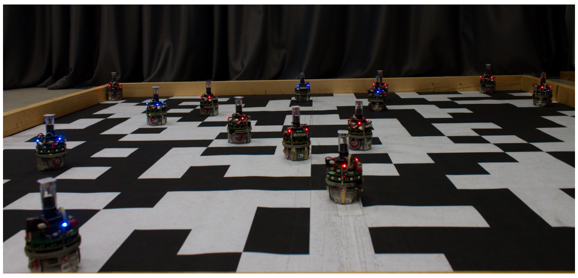


Figure 3.4: Portion of a swarm of 20 robots working on environment classification. Scenarios' parameters:
ρ_B=52%; ρ_W=48%;
Swarm size=20;
Decision Rule=majority rule;

The e-puck hardware and software are fully open source so that a low-level access to every electronic device is possible. The robot's battery can stand for up to 45 minutes approximately, but the performance of some sensor (as for example the Range and Bearing) after about 25 minutes decreases drastically. We therefore decided to run at maximum our experiments for a maximum of 20 minutes before to changing the batteries for a new run.

3.2 Behavioural Finite State Automata

At a high level, the best-of-n decision making problem is composed by a group of N agents trying to decide for the most valued alternative in a set of n possible ones:

$$\{a_1, a_2, ..., a_n\}.$$

Where a_i are the possible options of the problem. Each alternative has an associated quality:

$$\{\rho_1, \rho_2, ..., \rho_n\}.$$

Where ρ_i is the quality relative to the i-th alternatives. Every robot has, at every moment, an opinion about the best alternative:

$$\{r_1(t) = a_i, r_2(t) = a_j, ..., r_m(t) = a_z\}.$$

Where $r_i(t)$ is the opinion associated to the i-th robot at the time step t.

In our case the set of n alternatives is composed by only two elements, that are the black cells and the white ones:

$$a_1 = BLACK; a_2 = WHITE.$$

The associated qualities are the percentages of cells present on the floor of the two colors. From now we will call ρ_B and ρ_W respectively the quality of the black resource and the quality of the white one and we always consider that $\rho_B > \rho_W$, without any loss of generality. We have already made the two distinction between simple and difficult case:

$$\text{Simple Case: } \rho_B = 66\%; \rho_W = 34\%$$

while

Difficult Case: $\rho_B = 66\%; \rho_W = 34\%$.

A problem of collective decision-making is considered *successfully* solved only if two conditions are satisfied:

1. The opinion of every robot is the same:

$$\{r_1(t) = BLACK, r_2(t) = BLACK, ..., r_m(t) = BLACK\}$$

or

$$\{r_1(t) = WHITE, r_2(t) = WHITE, ..., r_m(t) = WHITE\}.$$

   Once this state is reached the consensus has been achieved. In this situation no robot can change idea since, even continuing the run, the information collected are all agreeing on the same opinion of the robot;

2. The opinion chosen from the swarm is the most valued one. It means that the quality associated to this opinion is the highest among the ones in the set of possible opinions. With our hypothesis ($\rho_B > \rho_W$) the solution for the environment classification problem is that every robot of the swarm is fevering Black:

$$\{r_1(t) = BLACK, r_2(t) = BLACK, ..., r_m(t) = BLACK\}.$$

To solve the environment classification problem every robot follows a behaviour that describe by a Finite State Automata. The behaviour of the robots is very simple and is modelled by statistical rules. There are two states in the FSA, representing the two phases of the robot's behaviour. In the first state, the *exploration state*, the robot has to examine the floor in order to estimate the quality of its current opinion. The duration of the exploration state is defined by a random variable exponentially distributed. This concept will be defined and better explained later. Once the time defined for the exploration state is expired the robot passes to the second state, the *dissemination state*. The goal of the robot in the dissemination state is to influence as many robot as possible with its own opinion. It is done by randomly walking in the environment while broadcasting, within a limited range, its own opinion. During both exploration and dissemination state the robots have to perform a random walk in order to inspect and spread the opinion in a completely random way.

In the exploration state the robot keeps track of the time spent on the cells coloured by the color of its opinion. The quality is hence calculated by a simple division between the time spent on these cells over the total time of the exploration state. We can therefore call white exploration (E_w) state and black exploration state (E_b) the states relative to the explorations of the different qualities, even if the behaviour in the two state is the same with the only difference of the resource that must be examined. The same distinction can be made for the dissemination state: in (D_w) the robots will advertise the white opinion while in (D_b) the black one.

The behaviour has been thought to make the robots diffusing their own opinion for a period of time directly proportional to the quality estimated in the exploration state. The weighting of the duration of the dissemination state is called modulation of the positive feedback. We are going to describe deeply the two state later. We will speak with more details about the duration of the two states and about the tasks that the robots have to do in each state.

During all the behaviour, independently if it is in dissemination state or in exploration state, the robot is performing a random walk. During the performance of this task the robots are moving straight for an exponentially distributed period of time, before to turn for a uniformly distributed time. The parameters of the two distribution have been set in order to make the robots turn for a much shorter period of time with respect of the period of time in which they are going forward.

3.2.1 Exploration State

The exploration state, as introduced in 3.2, is the starting point of the behave of every robot. In this phase the robots perform a random walk [47] and obstacle avoidance, in order to prevent the collision with the walls and with the other robots.

The obstacle avoidance is implemented using the proximity sensors mounted on the e-pucks. These sensors are returning two values: 1) a value bounded in [0,1], representing the distance of the obstacle; and 2) the bearing of the obstacle. If the robot is sensing a robot closer than a threshold (empirically determined), it sets its own wheels velocity in order to turn on the spot and go in the opposite direction with respect of the obstacle.

While walking in the environment, the robots check the color of the floor under them using the ground sensor, incrementing a counter variable for every time step spent on a cell of the same color of the robot' opinion. Let us call T_B and T_W the time spent respectively on a black and on a white floor and T_{EXP} the total duration of the exploration state. Considering a generic robot i, if $\mathrm{r}_i(t) = a_1 = BLACK$ then he will keep trace only of T_B, otherwise he will keep trace of T_W. The quality is finally calculated as

$$Eb \Longrightarrow \rho_B = \frac{T_B}{T_{EXP}}$$

and

$$Ew \Longrightarrow \rho_W = \frac{T_W}{T_{EXP}} \ .$$

The duration of the exponential state follows a random exponential distribution with mean σ. As in [103], it has been decided to adopt an exponential random distributed period of time because the memoryless property of this distribution that simplifies the mathematics modelling done by the authors.

$$T_{EXPLORATION} = Exponential(\sigma)$$

This parameter has been selected to ensure to the robots to have a reasonable estimation of the quality but also to have a sufficient level of noise because we are interested to study the situation of poor estimation of the quality. With high noise, we expect to see some behavioural discrepancies in the performance of the swarm with the three decision rules adopted, in terms of exit probability and consensus time described in the last section.

Our expectation is that the exit probability is directly proportional to the σ value, hence to the time spent in exploration state, and consensus time inversely proportional to that. The quality estimation with high values of σ will be more accurate, producing higher dissemination times for the robots fevering the right opinion. The effects of the parameter σ on the behaviour of the swarm will be better explained in 4.3 and 4.2.3.

After the exploration state, whether E_b or E_w, the robot will pass with probability one in the associated dissemination state: D_b in the first case and D_b in the second one (Figure 3.1).

3.2.2 Dissemination State

The dissemination state is composed by three subtasks: the broadcasting of the current robot opinion about the best option; the recording of the information broadcasted from the neighbours; and the actual decision-making process, that is the application of the decision-making rule over the collected information.

The broadcasting of the current robot opinion is made in every moment for the whole duration of the dissemination state. Concurrently, the listening is performed only in the last $3s$ of the state to avoid time-correlations of the collected opinions. The decision-rule application is instead an instantaneous operation: right before to change state the robots apply the decision-making rule in order to select only one of the collected.

In parallel to the above discussed tasks, is always performed the robot random walk. This is a key factor for the success of the strategy. The random walk in the dissemination state has the objective to keep the robots spatially well-mixed (i.e. randomly distributed in the environment), in order to avoid the fragmentation of the opinions (e.g., the formation of clusters of robots with the same opinion). The well-mixing property of the swarm in this state is an influencing factor of the efficiency of the strategy. If the swarm is spatially well-mixed then the strategy is efficient and reliable. The more the swarm is far to be well-mixed, the slower the decision-making process is and the lower is the efficient of the strategy [101].

For the duration of the dissemination state we opted to use an exponentially random distributed time, as in the exploration state, with different mean. The key factor of the proposed solution is the direct modulation of this time period. Every robot uses the quality estimated in the exploration state in order to modulate the duration of the dissemination state, thus the period of time in which the opinion is broadcasted. This introduces a positive feedback that pushes the robots to broadcast more the best option. The mean parameter (ζ) of the exponentially distributed variable representing the dissemination time is given by g weighted by the quality estimated, as described before

$$\zeta = \rho_i \cdot g \ .$$

Where g is a parameter empirically set by the designers. We decided to use a value of g equal to the mean of the exploration state σ. We recall that we analysed three decision rules (weighted voter model, direct comparison, and majority rule). When the applied decision rule is the direct comparison, the modulator factor is not the quality estimated in the exploration but is the quality associated to the best valued opinion: 0.52 in case of difficult scenario and 0.66 in case of simple one. In this case we decided to do not modulate the positive feedback because the direct comparison is already using the estimated quality in the picking up process. We will discuss about it in 3.2.3.3.

The effect of the modulation of positive feedback is that the most valued opinion will be probabilistically broadcasted for a longer time and thus the most valued opinion will be broadcasted more, with higher probability to influence other robots in the swarm. The modulation of the positive feedback introduces, over the time, a bias of the robots in favor of the most valued option bringing the system to the right consensus.

We have already introduced the concept of the listening phase of the dissemination state. We want that every robot listens the other neighbours opinion for an equal period of time in order to have more or less the same number of neighbours. Moreover, we desire that the robots use, for the selection of the new opinion, only a set composed by recently sent information. Indeed, we do not want to risk that the robots works on information not-up-to-date (i.e., opinions of robots that have already changed over the time). For these reasons we choose to limit the listening time to $3s$. The final total time of the dissemination state will be the random exponential value plus the constant listening time at the end of the state:

$$T_{DISS} = Exponential(\rho_i \cdot g) + 3s \ .$$

After having applied the decision rule the robot will switch state, passing to the exploration state. It can happens both that the robot changes opinion (i.e., from having a black opinion to have a white one or vice-versa) with a certain probability or that the opinion stays the same. Defining as P_b the probability to pick up a black favoring robots' opinion, and $1 - P_b$ the probability to pick up a white favoring robots':

- *Black exploration state (Eb)* with probability P_b;
- *White exploration state (Ew)* with probability P_b;

A particular focus of this work is to study the dynamics of different decision rules. The three decision rules that we are going to be coupled to the modulation of the positive feedback, biasing the choice of the swarm toward the best option.

- *Weighted voter model:* This decision rule is the simplest one. The robot chooses an opinion in a completely random way between the set of collected ones and blindly trust it, adopting that opinion for the next exploration state;
- *Direct comparison:* As in the weighted voter model the robot chooses a random opinion from the set of received ones; but instead of blindly trusting it the robot compares the received quality and changes idea only if this is higher than its current opinion;
- *Majority rule:* With the majority rule, the decision making process evaluates the whole pool of collected opinion. The new opinion adopted by the robots is the most numerous one;

Independently from the decision rule used, the robot is performing the same behaviour described by the FSA presented in the previous section. We will analyse the three rules in terms of speed in taking decision, accuracy of the decision and computational complexity of the used algorithm.

Parameters that can affect the quality of the decision are the number of robots that are initially fevering the right opinion and the quality of the right opinion. Clearly the bigger is the number of robots starting with the right opinion, the higher will be the probability to take the right decision independently from the decision rule applied. The same, if there is a big difference between the number of cells with the dominant color and the others then the right decision will be taken with higher probability. We will show the results of these variation later

3.2.3 Decision Rules

3.2.3.1 Weighted Voter Model

In weighted voter model [102] every robot chooses randomly an opinion in the pool and blindly adopts it as its new favourite opinion. The weighted voter model is the simplest decision rule that we tested. In our model every robot stores at most 2 messages, thus two opinions and randomly chooses one of them. picking-up the most valued option.

3.2.3.2 Majority Rule

Majority rule is one of the most studied decision rules and it has been previously applied in several other experiments in swarm robotics [33, 31, 32] (see 2). This decision rule requires to choose as next opinion the opinion favoured by the majority of the neighbours.

In our scenario, the robot that applies the decision has collected up to 2 opinions from the neighbours. It adds to the collected opinions its own opinion and then proceeds to the decision-making process. The adopted opinion is the one that is most present in the set composed by the received opinion and the robots' one. If in the set of received opinions there is a tie, i.e., there is not a majority, then the robot keeps its own opinion.

3.2.3.3 Direct Comparison

Direct Comparison is the decision rule that uses more information in the decision process with respect of the other rules since it takes into account not only the opinions received but also their qualities.

As in the weighted voter model, a random opinion is taken among the received ones, but it is not directly adopted by the robot. The robot compares the quality of the selected opinion with the quality estimated in the previous exploration state and, if and only if the quality estimated is higher than the received one, then the robot switch opinion adopting the one compared with his one.

Because of the comparison of the qualities, the direct comparison rule is more susceptible to noise in the environment and to the resulting goodness of the estimations made by the robots. This fact, as we will see later, can be problematic in situations of very difficult scenarios or very unreliable robots.

We introduce the analysis of this decision rule as control rule to see the efficacy of self-organizing strategies. Therefore we decided to do not use the modulation of the positive feedback. This choice has been done also to do not unbalance the three decision rules. Otherwise it would have influenced the decision process twice: in the broadcasting process, where it would have been spread more being the most valued one, and in the comparison process. This double weighting nature the comparison between the three decision rules would have been not-reliable and not-fair. The dissemination time is thus:

$$T_{DISS} = Exponential(\rho_{Opt} \cdot g) + T_{LIST},$$

where ρ_{Opt} is the quality pertaining to the optimum opinion.

Since direct comparison rule makes use of the quality estimated by the neighbours, it is characterized by higher complexity of the algorithm and of the communication stage. In order to send also the quality, the robots have doubled the payload sent from 2 *Bytes* (the dimension needed to send only the ID and the opinion) to 4 *Bytes.* This increased payload of messages has also a bearing on the robots energy autonomy due to the power consumed during the communication. This excessive afford recalls the difficulties in the usage of the Range and Bearing of the e-pucks.

Chapter 4

Physics-Based Simulations

In this chapter we are going to present the experiments and the evaluations done with physics-based simulations. Before starting with the final experiments (i.e., the experiments involving the evaluation of the consensus time and of the exit probability) we want to have a complete overview about the main points of the experiments. The behaviour is composed by a set of sub-tasks that the robots have to achieve in order to reach the consensus. These sub-tasks, further described in Chapter 3.2, are the random walk performed with obstacle avoidance, the performance of the exploration state, and the performance of the dissemination state. In this chapter, we explain the experiments done in order to understand the functioning of the main sensors and actuators used to perform the main tasks, and finally the dynamics of the main variables describing the performance of the swarm:

- Duration of the states: as described in Chapter 3.2, the behaviour of the robots is subdivided into two states, characterized by a duration determined by an exponential random distribution (to recall: the dissemination time is an exponential randomly distributed variable to which a constant of $3s$ has been added, while the exploration state follows a pure exponential randomly distributed variable). We run some experiments in order to validate these distribution times (4.2.1);

- Neighbourhood size: this concept is strongly linked to the first two points (random walk and duration of the states) and represents the average number of messages that each robot receives in one dissemination state. This concept, that directly involves the use of the range and bearing board, and the study of the neighbourhood size in our scenario are described in 4.2.2;

- Quality estimations: the estimation of the opinions' quality is one of the central points of our system. The sensor used in order to estimate the qualities is the ground sensor. The experiments conducted in order evaluate the performance of the robots in the quality estimation processes are presented in Section 4.2.3;

- Exit probability and consensus time: the two goals of this thesis are to study the dynamics of the swarm in terms of exit probability and consensus time, two macroscopic properties of the swarm that have been introduced and explained in Section 3.1. We studied the trade off of these two variables in one simpler and one more difficult scenario, applying three decision rules and with different swarm sizes (Section 4.3). More specifically, we studied the trend of these two variables varying the initial number of robots favouring the best option (Section 4.3.1), varying the difficulty of the problem (4.3.2), and varying the value of σ (Section 4.3.3). For an explanation about these parameters refer to Chapter 3.2.1;

We will start with the description of the simulation tools and of the algorithm implemented to solve the problem and then we will proceed with the explanation of the above mentioned experiments.

4.1 Simulator and Description of the Algorithm

Simulations are a preliminary step to test a designed system in an environment safe for the robots. For the simulations, we used ARGoS [75] (*Autonomous Robots Go Swarming*), a physics-based simulator ensure flexibility and efficiency, due to its modularity and parallelism.

ARGoS has been specifically designed to design multi-agent systems, heterogeneous or not, starting from the design of the single individuals. We wrote the code of our controller and a loop function to control the entire simulation experiment. The controller determines the behaviour of each class of robots. A class is a group of robots having the same behaviour. The loop functions are functions to customize the features of the experiments that allow the designer to implement some experiment-dependent characteristics (e.g., how to start and when to finish the experiment, which data to collect in the experiment, dynamically change the environment during the execution of the experiment, interact with the objects and with the robots during

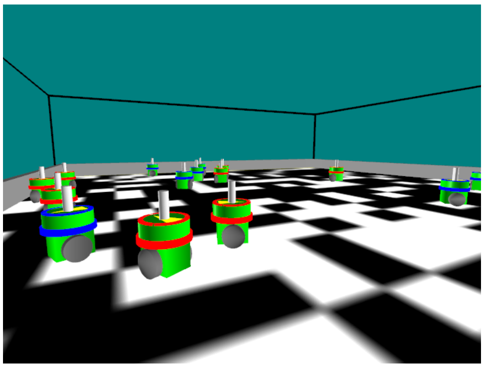


Figure 4.1: Picture taken from the run of one physics-based simulation in ARGoS 3. In the picture is visible a group of 20 robots trying to solve the environment classification problem in a simple scenario. The robots are the simulations of the e-pucks, available thanks to the plug-in developed by Garattoni et al. [34]. Parameters: Swarm Size $= 20$, *Initial Black Robots* $= 10$, *Black Quality* $= 66\%$, *White Quality* $= 52\%$.

the experiment). ARGoS also provides a graphic tool through which the designer can visually check, step by step, how the swarm behaves (Fig. 4.1 shows a picture of the simulation visualizer). One of the features provided by ARGoS is the possibility to eventually develop an ad-hoc plug-in for each kind of robot. Garattoni et al. [34] developed a plug-in to integrate the e-puck robots to ARGoS simulator and give to the designer the possibility to use the same code in both real robots and simulations. In this way, the designer can easily switch to real robot experiments.

Our program was composed by just one type of controller, since our swarm is an homogeneous and decentralized system where every robot has the same behaviour. The code of the controller is the following: when the experiment starts, every robot sets-up its internal variables, calculating the exponential variables for the dissemination and exploration time (that are decreased every time step) and resets every variable regarding the opinion and the quality. Every robot is set to be in exploration state at the beginning of the experiment.

Every time step the robots evaluate the color of the floor through the ground sensor. The ground sensor is only able to recognize the white and the black colors. The robot receives from the ground sensor a value contained in 0 and 1: all values between 0 and 0.5 mean that the ground sensor recognized a black color, values between 0.5 and 1 mean that the ground sensor recognized the white one. In the meantime the robots are performing the random walk and, through the proximity sensor avoid collisions with the other robots and the walls (obstacle avoidance). Just before switching to dissemination state (i.e., when the time to be spent in exploration state is finished) the robot calculates the quality of the explored option and the next dissemination time.

In dissemination state the robots send messages through the range and bearing device. They send a packet with the following informations:

- *Sender ID*: every robot sends its own ID, in order to allow the receiving robot to skim eventual double messages from one single robot;
- *Quality*: the quality evaluated in the last exploration state, relative to the actual opinion. This value does not need to be sent if the applied decision rule is the weighted voter model or the majority rule. When direct comparison is applied, then the robot has to compare the received quality with its own one in order to decide if to change opinion or not (Chapter 3.2.3.3);
- *Opinion*: obviously the robots send their own actual opinion;

In the last $3s$ of the dissemination state, the robot saves the incoming messages following a simple policy to select which of them to keep. The robots have been programmed in order to not save more than one message from each different robot in each dissemination state (i.e., if it gets twice the opinion of one neighbour, it only saves the last received one). Moreover, they save at maximum a number k of messages for each dissemination state, and in our case $k = 2$ (the choice of this parameter is described in Chapter 5.1.3.2). When the remaining dissemination time is finished, the robot changes opinion by applying a decision rule to the information gathered pool, and switches back to the exploration state.

Additionally, we implemented the reset function, that allows to perform more than one run with the same initial parameters setting. This function is only designed to reset all variables of the system and to bring back the whole experiment to the initial condition. The loop function collects statistics to write in the output file. The results of every run are saved in the output file as a set of rows. Each row is composed by: 1) the number of robots favouring the different options when the experiment finishes; and 2) the time steps passed since the start of the experiment when the consensus has been reached.

4.2 Preliminary Studies

4.2.1 Analysis of the Exploration and Dissemination Time Distributions

The duration of exploration and dissemination states follows an exponential random distribution (Chapter 3.2) with parameters σ and $\rho_i * g$ respectively. The mean duration of the exponential state is then σ, that we set to $10s$. The mean duration of the dissemination state is equal to the parameter g

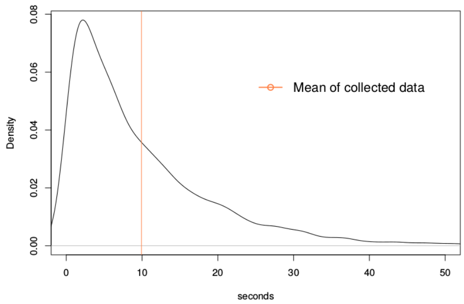
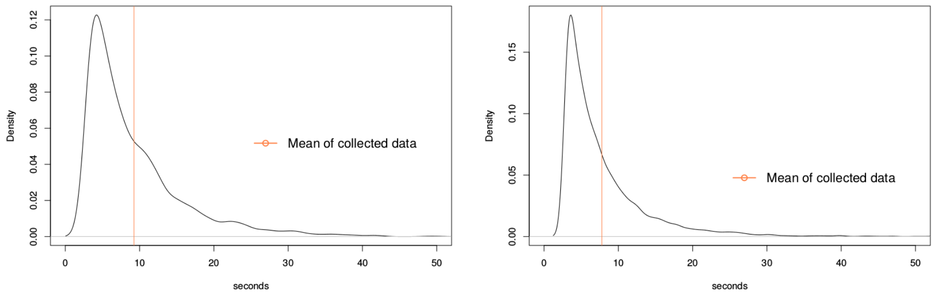


Figure 4.2: The graphs report the distribution of the exploration times, with the black curve. The vertical orange line represents the calculated mean of the duration of the exploration states. Parameters: $\sigma = 10s$, $g = 10s$.

weighted with the estimated quality. We set-up the parameter g to be always equal to σ. More precisely, we want to recall that the dissemination state is an exponential randomly distributed variable at which has been added a constant of $3s$, in order to avoid the generation of too short dissemination times. This distribution is hence a combination of the exponential and of the constant.

To test the trends of the two distributions, we have collected the duration of the states of every robot. We performed experiments both in the difficult and in the simple scenario. The distribution of the exploration times is the same in the two scenarios since it only depends on the parameter σ (not dependent from the environment). The distribution of the dissemination state, instead, directly depends by the quality estimated (i.e., from the difficulty of the problem). We collected data relative to the durations of the states. For both the exploration state and the dissemination state, we recorded the duration of every single iteration of all the robots, independently from their current opinion. Hence, we recorded one big set of data containing all durations of the three state analysed for the experiments (exploration state, dissemination state in hard scenario, dissemination state in simple scenario). The graphs report the densities of the set of data recorded. Fig. 4.2 reports the density of the distribution of the exploration state times. Fig. 4.3(b) and 4.3(a) show the distributions of the dissemination states in, respectively, the difficult scenario and the simple one.

In Fig. 4.2, we see that, as expected, the exploration time distribution has a mean (represented by the vertical orange line) equal to $10s$, as the parameter σ. In Fig. 4.3(b) and 4.3(a) is shown that the dissemination state times still follow an exponential distribution with a mean close to $10s$. However, in this case, the weighting factors slightly shift the mean to lower values. We recall that the mean of this distribution is weighted with the quality estimated with $\rho_i \in (0, 1)$. For this reason, the mean of the distribution of the dissemination states is always lower than $10s$. Moreover,



(a) (b)

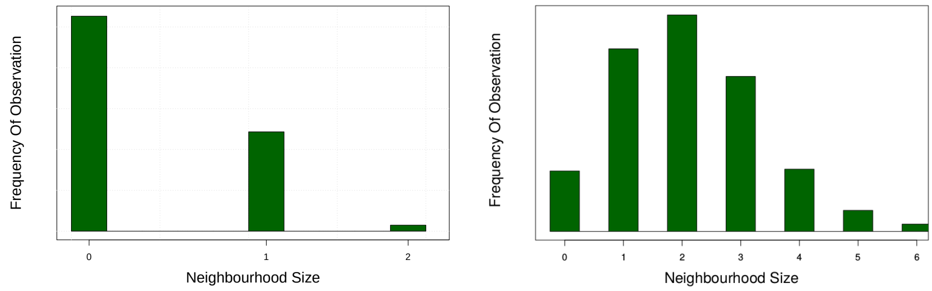
Figure 4.3: In the graphs are reported, with the black curve, the distributions of the dissemination times. The vertical orange lines represent the calculated mean of the durations of the dissemination states. Parameters: $\sigma = 10s$, $g = 10s$, *Swarm Size* $= 20Robots$. *a) Dissemination time distributions in the simple scenario,* $\rho_{Black} = 66$; *b) Dissemination time distributions in the difficult scenario,* $\rho_{Black} = 52$;

the qualities estimated in the two scenarios are different: in the first scenario the black option is favoured and its quality is equal to 66%; in the second scenario the black option is still favoured, but with a lower quality, that is, 52%. The duration of the dissemination state is determined both by the time spent in dissemination state for the black robots and for the white ones. In this experiment, the number of black robots is going to be higher than the number of white ones, since the black option is the best one. For these reasons, the average dissemination time in the simple case is higher than the average dissemination time in the hard scenario. Fig. 4.3(a) and 4.3(b) show this difference. Even if the two distributions have a similar dynamic, the mean of the distribution in the simple scenario is closer to 10 than the distribution in the difficult scenario.

4.2.2 Study of Neighbourhood Size

Our first idea was to make the robots communicate within a controlled range, that is three times the e-puck diameter ($21cm$). Unluckily, after the first tests on the real-robot sensors and actuators performance, we figured out that the range and bearing (the board that must take care of the communication phase) was not allowing us to control the range of communication because of its unreliability and the high noise in its usage.

For this reason, we chose to adopt another strategy, that is, to control the maximum number of saved messages of the robots at the software-level. To decide how many packets to save at maximum in order to have a situation



(a) (b)

Figure 4.4: Graphs reporting the neighbourhood sizes with a communication range of $21\ cm$*. The green histograms represent the frequency at which the correspondent value of the X-Axis has been observed. A higher histogram bar means a more probable observable value. Parameters: Range Of Communication* $= 0.21\ cm$*;* $\rho_{Black} = 66\%$*;* $\rho_{White} = 34\%$*; Initial Black Robots* $= 50\%$ *of the swarm size;* $\sigma = 10s$*;* $g = 10s$*. a) Experiments performed with a swarm size of 20 Robots; b) Experiments performed with a swarm size of 100 Robots.*

as close as possible to the ideal one, we determined the *neighbourhood size* in each dissemination state. For neighbourhood size, we intend the number of incoming packets received by one robot during one whole dissemination state. We then performed a series of experiments which output was the number of messages listened in each dissemination state.

We set the configuration parameters to the ideal condition:

- Limited range of communication of $21cm$. Limiting the range of communication is allowed and reliable with the simulations tool;
- We choose the simple scenario because it is the case in which the robots are receiving more values. Indeed, as shown in 4.3(b), due to the higher evaluation of the qualities, the dissemination time is higher in this scenario than in the difficult scenario. Hence, more messages are exchanged;
- We adopted a swarm with a size of 20 and 100 robots;

We ran a set of experiments which output was a file containing the number of received messages by each robot in every dissemination, for the duration of the experiment. The data is plotted in Fig. 4.4 by means of an histograms graph. The histograms represent the frequency of each neighbourhood size. From these histograms emerges the difference between the two situations with different swarm sizes. In 4.4(a) are shown the neighbourhood sizes when using a swarm of 20 robots. In this case, one robot *never* receives more than two messages per dissemination state. For this reason, we have chosen to put a limit of 2 messages per dissemination state, and we also used this limit in the real-robot experiments (where the neighbourhood size was quite higher, see Fig. 4.4). Moreover, the number of times in which the neighbourhood size is zero (i.e., the robot has not received messages in that state) is the majority of the cases. Consequently, the robots only rarely receive two messages. Fig. 4.4(b) reports the neighbourhood sizes obtained using a robot of 100 robots in physics-based simulations. Watching the resulting histogram-graph, we notice that the robots receives up to 6 messages for dissemination state. In this case, the maximum value is registered for two robots as neighbourhood size.

4.2.3 Preliminary study of quality estimation procedure

As mentioned above, the estimation of the opinions' quality is a factor that strongly influences the accuracy and the decision time needed to reach a consensus by the swarm. The noise on the quality estimations alters the performance of the swarm when applying different decision rules. It influences mainly two aspects of the behaviour. The first aspect is the determination of the dissemination state. As largely discussed, the dissemination state time mean is weighted by the quality estimation of the current opinion. To an underestimation or overestimation of the quality corresponds a wrong definition of the dissemination state time, thus an under-broadcasting or an over-broadcasting of the opinion. The other mainly affected point is the comparison of the qualities done when using the direct comparison strategy (Chapter 3.2.3.3). In this strategy, the qualities are compared before deciding whether to change opinion or not. We conducted several experiments in order to understand how noisy is the estimation of the quality by the robots.

Rationally, we are expecting that for an infinite period of time spent in exploration state (i.e., for really high values of σ) the robots perfectly evaluate the qualities, without errors. On the other hand, if the time spent in exploration is really low, the estimations will be extremely noisy. In the case in which a robot can explore the environment for only one second, the probability to see only one resource in that second of exploration is really high. It means that, if the resource is the one that the robot favourites, then the estimation of the quality will be close to 1, otherwise it will be close to 0.

Moreover, larger swarm sizes could involve higher interferences rate between robots, that can be a factor affecting the estimations as well. In order to see the effects of σ and of the swarm size, we have conducted the following

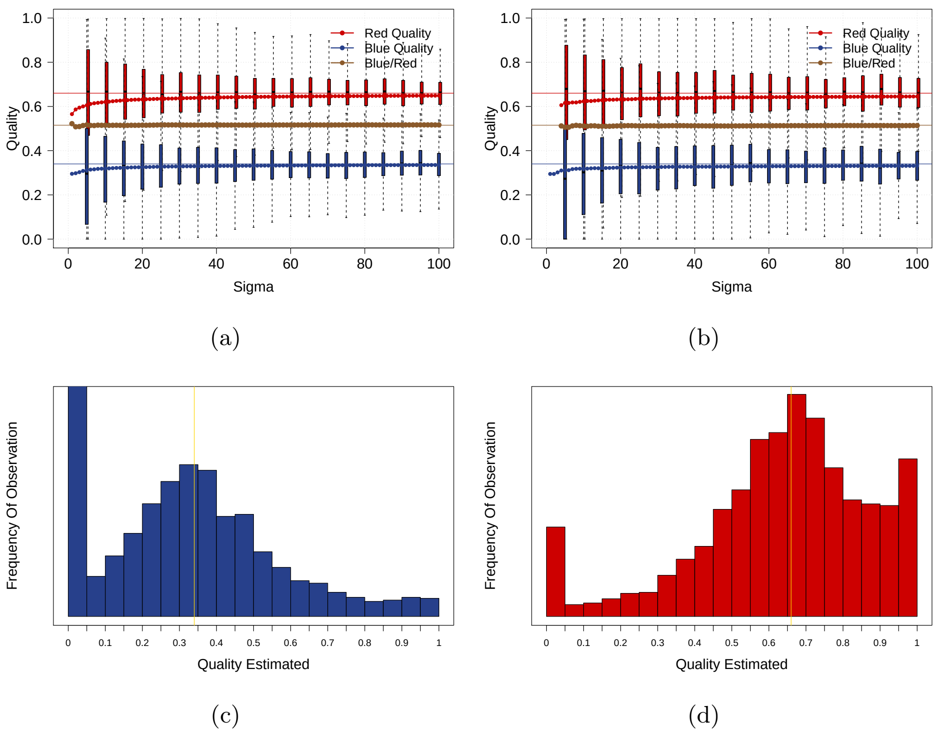


Figure 4.5: Simple scenario: Graphics relative to the quality estimation in a simple scenario. The plots (a) and (b) show the quality estimation of the floor with, respectively, 1 robot and with 100 robot in the arena. The red (blue) boxplots are the overall representations of the estimations for the associated value of σ on the X-Axis when analysing the black (white) option. The red (blue) points are the mean of the estimated quality of the black (red) option for every discrete value of σ. The brown points are the ratio between the black and the white estimations. The horizontal lines placed, respectively, at 0.34, 0.66, 0.5 are the expected estimations of the two qualities (white and black) and the correct ratio ($\frac{0.66}{0.34}$). Parameters: $g = 0s$, $\sigma \in \{1, 2, \ldots, 100\}$, $\rho_{Black} = 66\%$, $\rho_{White} = 34\%$; The plots (c) and (d) represent the distribution of the estimated quality by the black and the white robots. The distribution is represented with histograms. Red (Blue) histograms represent the measurements of the black (white) option. The yellow vertical line is the actual quality of the option. Parameters: $g = 0s$, $\sigma = 10s$, $\rho_{Black} = 66\%$, $\rho_{White} = 34\%$;

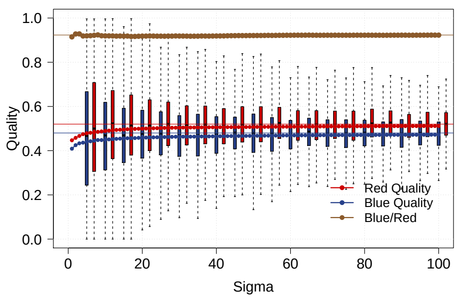
tests:

- *Simple scenario:*
  - Difficulty: $\rho_{Black} = 66\%$, $\rho_{White} = 34\%$;
  - Swarm size: 1 Robot (4.5(a)), 100 Robots (4.5(b));
  - $\sigma \in \{1, 2, \ldots, 100\}$;
  - $g = 0s$;
- *Difficult scenario:*
  - Difficulty: $\rho_{Black} = 52\%$, $\rho_{White} = 48\%$;
  - Swarm size: 1 Robot (4.6(a)), 100 Robots (4.6(b));
  - $\sigma \in \{1, 2, \ldots, 100\}$;
  - $g = 0s$;

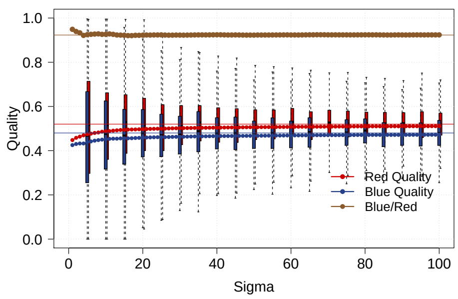
The controller used for the experiments aiming to understand the quality estimation of the robots is the same as the one described in Chapter 3.2. We collected 10000 estimations for each point of the graph, hence for each couple of configuration of σ and swarm size.

Fig. 4.5(a) and 4.5(b) show the graphics of the quality estimation performed by one single robot in the arena, with swarm sizes of, respectively, 1 and 100 robots. The trend of the mean of the quality estimations is indicated by the red and blue points and it is shown to get closer to the real options' quality when σ increases. With really low values of σ the estimation is poor. we can for example notice that for $\sigma = 1$ the quality estimation of the red option is lower than 0.6. The boxplots show that the quality estimation variances decrease when σ decreases. It means that by increasing σ, the values measured are closer to the mean. These results are explainable by the fact that if a robot has more time to explore the environment then the estimation will be better. The lower is the time, the poorer is the estimation of the quality. Comparing the results obtained with swarm size = 1 and with swarm size = 100, we notice that even if the mean is similar and really close to the option's quality in the two cases, the boxplots report differences. With a swarm size of 1 robot, the estimations have a lower variance than in the case with a swarm size of 100 robots. This can be explained by the movement interferences due to the presence of a high number of robots.

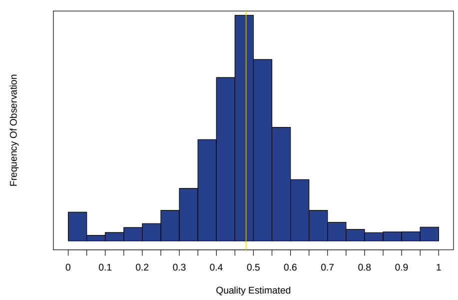
The graphics in Fig. 4.5(c) and in Fig. 4.5(d) show the frequencies of the estimation of each qualities. The results are plotted by using histograms. Each histogram represents the estimations falling within a range of qualities



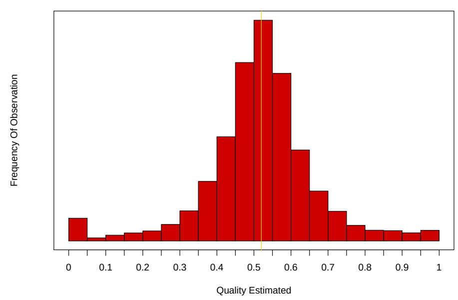
(a)



(b)



(c)



(d)

Figure 4.6: Difficult scenario: Graphics relative to the quality estimation in a difficult scenario. The plots (a) and (b) show the quality estimation of the floor with, respectively, 1 robot and with 100 robot in the arena. The red (blue) boxplots are the overall representations of the estimations for the associated value of σ on the X-Axis when analysing the black (white) option. The red (blue) points are the mean of the estimated quality of the black (red) option for every discrete value of σ. The brown points are the ratio between the black and the white estimations. The horizontal lines placed, respectively, at 0.48, 0.52, 0.923 are the expected estimations of the two qualities (white and black) and the correct ratio ($\frac{0.48}{0.52}$). Parameters: $g = 0s$, $\sigma \in \{1, 2, \ldots, 100\}$, $\rho_{Black} = 52\%$, $\rho_{White} = 48\%$; The plots (c) and (d) represent the distribution of the estimated quality by the black and the white robots. The distribution is represented by means of histograms. Red (Blue) histograms represent the measurements of the black (white) option. The yellow vertical line is the actual quality of the option. Parameters: $g = 0s$, $\sigma = 10s$, $\rho_{Black} = 52\%$, $\rho_{White} = 48\%$;

(each range is large 0.05). The graphs refer to the experiments done with $\sigma = 10s$ in the simple scenario. It means that the exploration time is relatively low. The robots often measure the limit values (0 and 1). Indeed, in Fig. 4.5(c) we can see that there is a strong majority of estimations ended with the measurement of a quality between 0 and 0.05. On the other hand, in the graph shown in Fig. 4.5(d) a high frequency of quality estimations ended with a value between 0.95 and 1. This is due to the short time available for the exploration: the unbalanced situation in the number of cells makes robots overestimating the black option, while underestimating the white one. Both graphs are characterized by a high frequency of quality estimations ended with the correct value (i.e., 0.66 for the black surveys and 0.34 for the white surveys). For the black quality, the frequency of estimations progressively increases from zero to the right option quality, with the exception already discussed for the range between 0 and 0.5. After the correct option quality, the frequencies decrease until reaching the other limit value, one, where another peak is present. The graph relative to the white quality estimation's frequencies is symmetric to the one for the black quality estimation.

Fig. 4.6(a) and 4.6(b) show the graphs relative to the estimations of the quality in the difficult scenario, with swarm sizes of 1 robot and 100 robots. The considerations for this graphs are similar to the ones for the simple case. When increasing the value of σ, the behaviour of the two quality estimations is similar to the one discussed above. Unlike previously, the graphs in Fig. 4.6(c) and 4.6(d) show a bell-shaped trading with a maximum in the range of the actual current quality of the two options. The high frequency of quality estimations ended with 0 is shown in Fig. 4.5(c) and with 1 is shown in 4.5(d) are not present anymore. This fact is due to the more balanced situation between the two qualities: since the number of cells is similar, it is difficult for the robot to explore only cells of the same color, even in a limited time.

4.3 Exit Probability And Consensus Time

The goal of our work is to draw a comparison between three different decision-making rules (weighted voter model, majority rule, and direct comparison) in the environment classification problem. The comparison is made in terms of exit probability and consensus time, two variables largely discussed in Chapter 3. We analyse the dynamics of these two variables scaling the main influencing factor of the problem: the difficulty ($\rho_{black} vs \rho white$), the number of robots initially favouring the best option, and the value of σ.

In the following sections, we show the results of these studies.

4.3.1 Varying Initial Number of Black Robots

The initial number of robots favouring the black option, that is the best one as we defined in Chapter 3, is probably the most influencing variable for the dynamics of the swarm. An important information is the time to reach a solution and the accuracy of the solution varying this variable. With this aim, we tested in simulation the simple and the difficult scenarios with a swarm size of 20 robots and 100 robots. For each initial condition (i.e., number of correctly favouring robots), we performed 1000 runs. The output of the experiments is a text file containing, for each run, a row with the number of black robots and white robots after the end of the experiment, and the time needed to reach the consensus. Using the results in these files, we plotted the dynamics of the two variables (Fig. 4.7 and 4.8).

The graphs in Fig. 4.7 show the dynamics of the system in the simple scenario. Fig. 4.7(a) and 4.7(b) show the performance of a swarm composed by 20 robots, while in Fig. 4.7(c) and 4.7(d) the robots involved in the experiments were 100. The graphs show a rougher shape than with 20 robots, due to the higher number of points. Let us analyse first the consensus time: from the graphs 4.7(a) and 4.7(c), the time required to reach the consensus is higher when a higher number of robots is involved. We can see how the time required by the majority rule is less affected by the swarm size than the other strategies, remaining in both cases the fastest strategy. Overall, the weighted voter model is the slowest. But when number of initial robots favouring black is really lower than the swarm size, the direct comparison takes more time to reach a consensus than the other strategies. All the strategies are characterized by an increase of the consensus time from zero to a certain point, before starting to decrease to zero. Before these points, the solution is easily reachable, since the initial number of black robots is really low and the solution is quickly going towards the wrong option. After these points, the solution starts to be correct, but is difficult to reach. Obviously, an higher number of initial black robots speeds up the reaching of the solution.

We started discussing about the easiness of reaching a solution. It directly introduces us to the exit probability. We can easily see from 4.7(d) and 4.7(b) how the exit probability monotonically increases with the increase of the initial number of black robots. The comparison between them shows how the curves are accentuated with the increase of the swarm size. Looking at the majority rule curve, we notice that it approaches a step curve

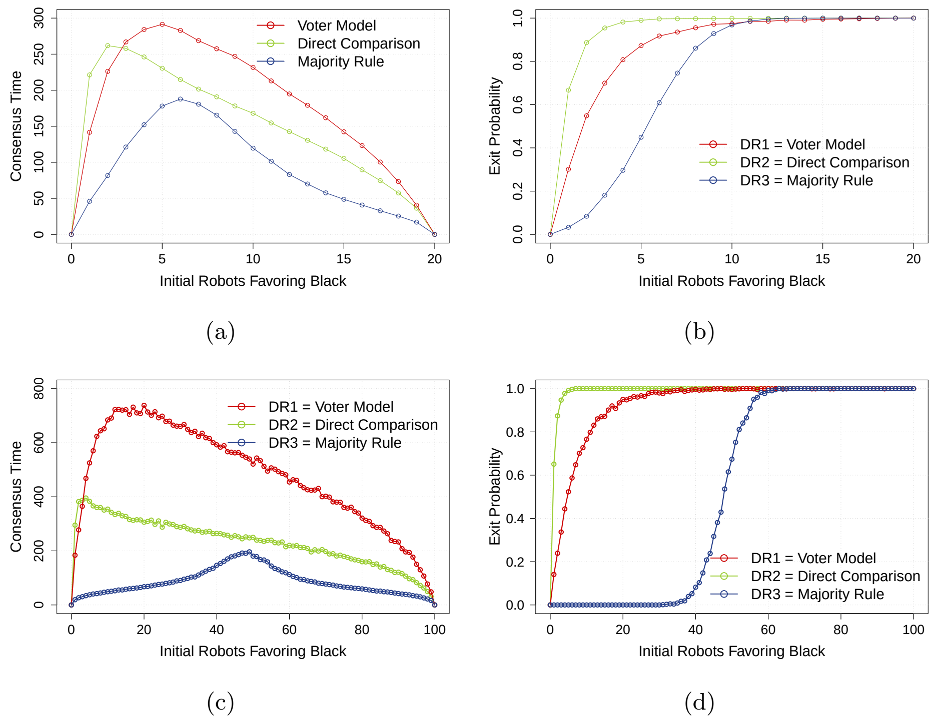


Figure 4.7: In the graphs are shown the dynamics of the exit probability and of the consensus time scaling the initial number of robots favouring the black option in the simple scenario obtained in simulation. We used for the experiments a swarm of 20 robots and a swarm of 100 robots. The points represent the average of the consensus time (or of the exit probability) obtained with 1000 runs with the same number of initial robots favouring the black option. The three colors represent the different decision rule used: red = weighted voter model, green = majority rule, blue = majority rule. Parameters: Number of initial black robots=$(1, 2, \ldots, 100)$*,* $\rho_{black} = 0.66\%$*,* $\rho_{white} = 0.34\%$*,* $\sigma = 10s$*,* $g = 10s$*. a) Consensus time obtained with a swarm size = 20 robots; b) Exit probability obtained with a swarm size = 20 robots; c) Consensus time obtained with a swarm size = 100 robots; d) Exit probability obtained with a swarm size = 100 robots.*

with the center corresponding to an initial number of black favouring robots of 50% of the swarm, and more precisely, with a swarm size of 100 robots the exit probability is 0.5 when the initial number of robots favouring the best option is 47. In *such a simple scenario*, also the exit probability for the direct comparison also approaches a step curve in zero.

What is really evident looking at the graphs reported in Fig. 4.8(d) is the incredibly long time required by direct comparison. The graph shows the consensus time in the difficult scenario with 100 robots. The direct comparison is extremely lower than all the other strategies and, moreover, is extremely lower than the direct comparison in all the other situations. Fig. 4.8(d) (the detail panel) shows the consensus times functions relative to the majority rule and the weighted voter model. As reasonable thinking, the voter model takes more time in the difficult scenario than in the simple one. The majority rule, instead, takes approximatively the same time. Another difference between the consensus times in the difficult scenario with respect to the simple scenario is the points of maximum, where the required time starts to decrease. Indeed, the shape of the curves still follows the same trend but the maximum points (i.e., the point where the initial conditions make the problem simpler to solve) are shifted closer to the 50% of the swarm size. It is due to the more difficult nature of the problem.

Analyzing the exit probabilities, we notice other important characteristics. In Fig. 4.8(b), we can see that all decision rules are less accurate, but the majority rule which has a similar exit probability. The direct comparison is instead the most disadvantaged by the higher difficulty: in the simple case, with 20 robots, direct comparison was correctly solving the problem with a probability of 100% in presence of 5 initial black robots. The difficult case, instead, is never reaching the 100%. The voter model, with the increase of the difficult of the problem, approaches a straight line starting from 0 and reaching 1. Another factor that is easily noticeable is the much higher noise present in the curves of the direct comparison, in the difficult problem. It is due to the unreliability of the direct comparison under high level of noise (i.e., the more difficult of the problem).

4.3.2 Varying Problem Difficulty

As seen in 4.3.1, the difficulty of the problem is a fundamental factor for the dynamics of the swarm in some cases. To better analyze how it influences the exit probability and the consensus time, we performed more extensive experiments spanning the two qualities. Using the two usual swarm sizes, that are 20 and 100 robots, we spanned all the qualities from 52% to 66%.

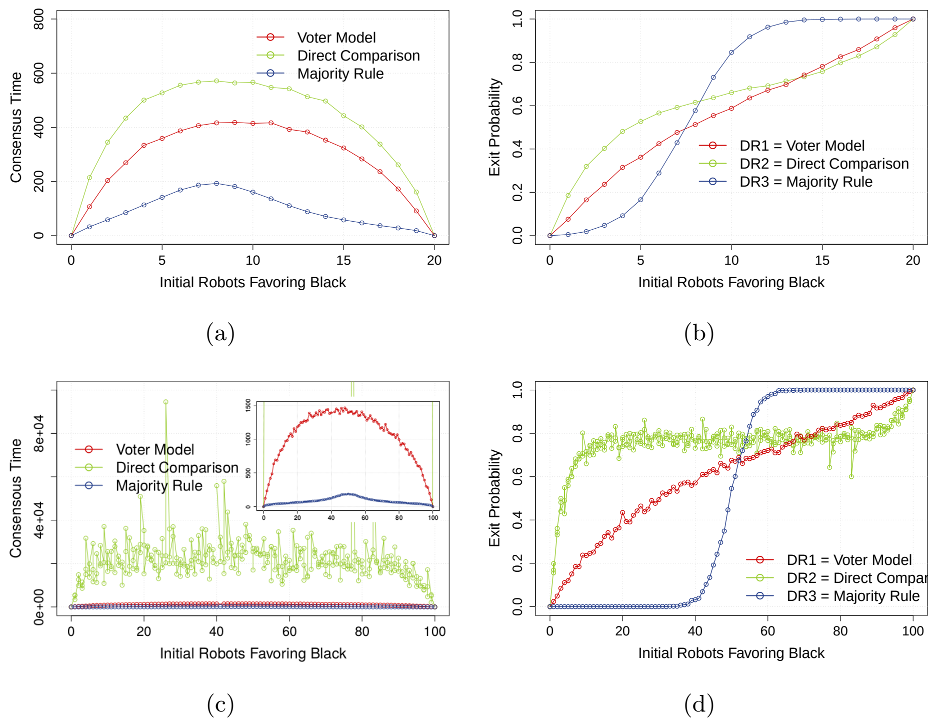


Figure 4.8: The graphs show the dynamics of the exit probability and of the consensus time scaling the initial number of robots favouring the black option in the difficult scenario obtained in simulation. We used a swarm of 20 robots and a swarm of 100 robots. The points represent the average of the consensus time (or of the exit probability) obtained with 1000 runs with the same number of initial robots favouring the black option. The three colors represent the different decision rule used: red = weighted voter model, green = majority rule, blue = majority rule. Parameters: Number of initial black robots=$(1, 2, \ldots, 100)$ $\rho_{black} = 0.52\%$, $\rho_{white} = 0.48\%$, $\sigma = 10s$, $g = 10s$. *a) Consensus time obtained with a swarm size = 20 robots; b) Exit probability obtained with a swarm size = 20 robots; c) Consensus time obtained with a swarm size = 100 robots. In the square in the center of the main graphs is shown the zoomed detail of the curves. In the main graph, it is obvious how the direct comparison takes more time than the other rules, but the behaviour of majority rule and weighted voter model is not obvious. For this reason we decided to zoom them; d) Exit probability obtained with a swarm size = 100 robots.*

We decided to keep σ equal to 10s, to keep a high level of noise for the experiments.

From the graphs in 4.9(c) and 4.9(a), it is evident how the behaviour of the consensus time is to decrease with the decrease of the difficulty of the problem, independently from the used swarm size. The way of decreasing is different in the two cases. As already seen in 4.8(c), for a high level of noise and a high number of robots involved in the decision-making process, the direct comparison takes a really long time to reach the consensus and with a high noise. The majority rule has a consensus time with a rather invariant trend. The consensus time taken by this strategy is constant, independently of the difficulty of the problem. On the other hand, the weighted voter model is faster as the difficulty decreases.

For the exit probability, Figures 4.9(d) and 4.9(b) show that all the strategies have an increasing trend with the decrease of the difficulties. In order to analyze these graphs, one must keep in mind the graphs about the exit probability presented in the Sections 4.8 and 4.7. For the majority rule, in graphs (4.9(b) and 4.9(d)), we can immediately notice that the exit probability. However, in the previous section (Section 4.3.1) we calculated that the point where the exit probability is 0.5 for the majority rule corresponds to 47 initial robots favouring black, for the simple scenario, and 50 initial black robots for the difficult scenario. Our results, here, do not match with the ones of the previous works made about majority rule (Valentini et al. [101], Montes et al. [64]). For this reason we decided to make a further analysis that is presented in **??**. In their work, for problem with the same difficulty as our simple problem, the step of convergence corresponds to a lower number of robots initially favouring the best option. Weighted voter model and direct comparison are instead behaving similarly, increasing with the decrease of the problems' difficulty. With these initial conditions (i.e., with the swarm size equally parted in black and white favouring robots), direct comparison is more accurate than weighted voter model. The fastest strategy is still the majority rule.

Overall, these graphs highlights the really long consensus time required by the direct comparison in situations of high noise, that is, with a high difficulty of the problem. The behaviours of the weighted voter model and of the direct comparison strategies get better both in terms of exit probability and in consensus time, with the decrease of the problem's difficulty. Moreover, we see how weighted voter model and direct comparison gain in accuracy to solve the same problems when the size of the used swarm increases, even if the required time is higher.

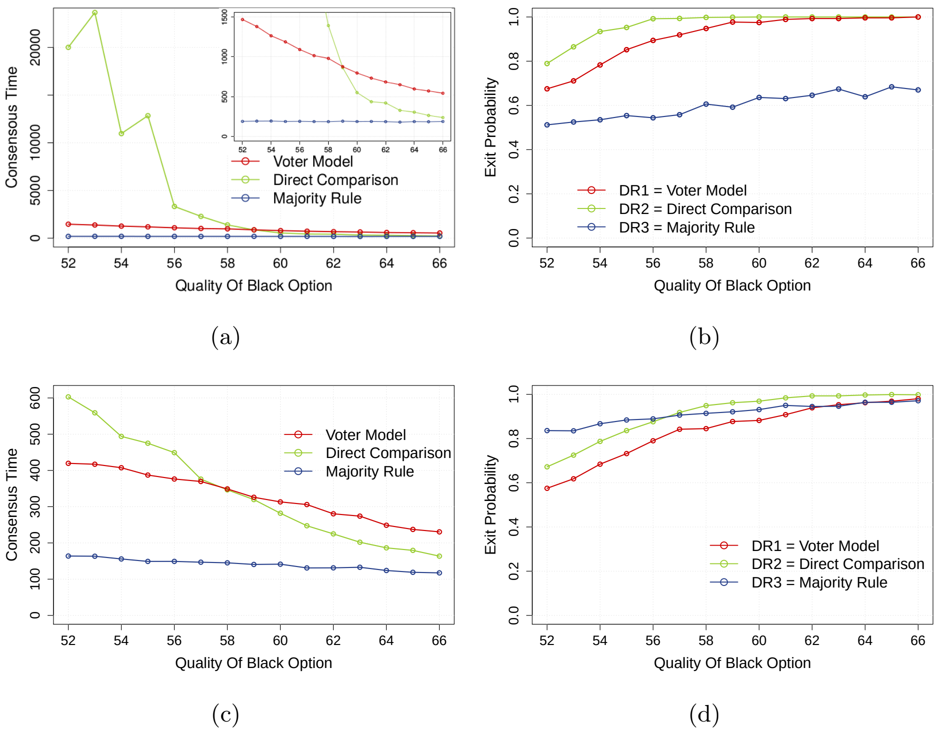


Figure 4.9: The graphs show the dynamics of the exit probability and of the consensus time scaling the difficulty of the problem obtained in simulation. We used for the experiments a swarms of 20 robots and a swarm of 100 robots. The points represent the average of the consensus time (or of the exit probability) obtained with 1000 with 50% of black robots. The three colors represent the different decision rule used: red = weighted voter model, green = majority rule, blue = majority rule. Parameters: $\sigma = 10s$, $\rho_{black} =$(52, 53, ..., 66)%, $\rho_{white} = 100 - rho_{black}$, $g = 10s$. a) Consensus time obtained with a swarm size = 100 robots, initial black robots = 50, initial white robots = 50. In the square in the center of the main graphs is shown the zoomed detail of the curves. In the main graph is evident how the direct comparison takes more than the other rules, but the behaviour of majority rule and weighted voter model is not evident. For this reason we decided to zoom them; b) Exit probability obtained with a swarm size = 100 robots, initial black robots = 50, initial white robots = 50; c) Consensus time obtained with a swarm size = 20 robots, initial black robots = 10, initial white robots = 10; d)Exit probability obtained with a swarm size = 20 robots, initial black robots = 10, initial white robots = 10.

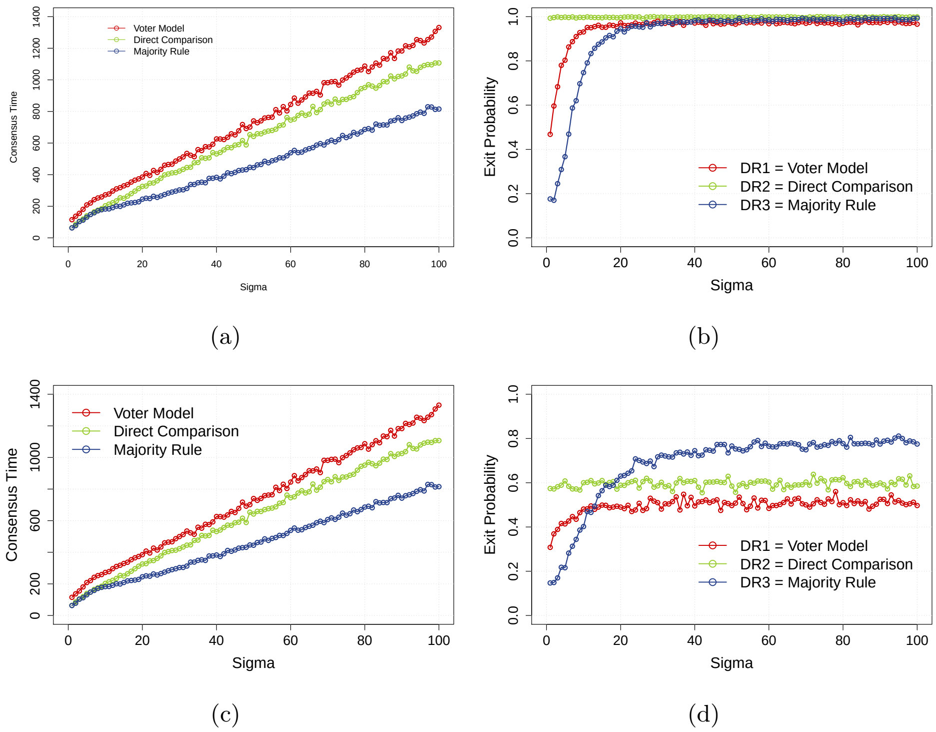


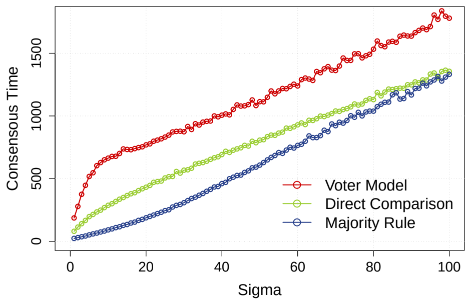
Figure 4.10: The graphs show the dynamics of the exit probability and consensus time for a range of values of σ obtained in physics-based simulations, with a swarm size of 100 robots. The points represent the average of the consensus time (or of the exit probability) obtained with 1000 runs with the same difficulty. The three colors represent the different decision rule used: red = weighted voter model, green = majority rule, blue = majority rule. Parameters: $\sigma =$(1, 2, ..., 100)%, $g = \sigma$, black robots = 30% of the swarm, white robots = 70% of the swarm. a) Consensus time obtained with a swarm size of 20 robots, $rho_{black} = 66\%$, $\rho_{white} = 100 - rho_{black}$; b) Exit probability obtained with a swarm size of 20 robots, $rho_{black} = 66\%$, $\rho_{white} = 100 - rho_{black}$; c) Consensus time obtained with a swarm size of 20 robots, $rho_{black} = 52\%$, $\rho_{white} = 100 - rho_{black}$; d)Exit probability obtained with a swarm size of 20 robots, $rho_{black} = 52\%$, $\rho_{white} = 100 - rho_{black}$.

4.3.3 Varying Exploration Time

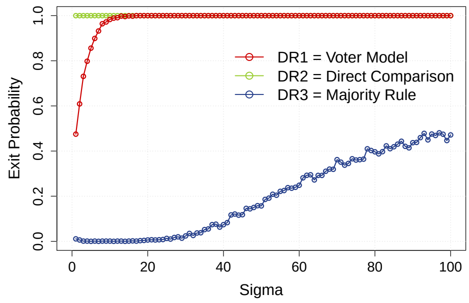
The exploration time is represented by its mean, σ, throughout this thesis. We performed experiments with fixed initial condition. We decided to test the case in which the proportion of initial black robots is only 30% (for the results obtained scaling the initial number of black robots, refer to Fig. 4.7, 4.8, and 4.9). We tested the two usual scenarios, the simple and the difficult one.

We start with the analysis of the consensus time in the four situations (20 and 100 robots and the two scenarios: Fig. 4.10(a), 4.10(c), 4.11(a), 4.11(c)). The speed of the strategies has been studied in the previous experiments (Sec. 4.3.1). In the simple scenario, the majority rule is the fastest while the weighted voter model the slowest. In the difficult scenario, the majority rule is still the fastest but the weighted voter model is faster than the direct comparison. Moreover, we can see how the direct comparison is highly dependent from the noise. In the difficult scenario, with the usage of a swarm composed by 100 robots, the behaviour of the direct comparison is emblematic. We can notice the extremely big discrepancy between the consensus time obtained with low values of σ and with high ones. As mentioned previously, σ determines the noise in the quality evaluations. High σ values leads to a robust estimation of the qualities. If the quality estimation is noisy and the qualities of the two opinions are close, then the estimations can easily be inverted (i.e., the best option estimated less than the worst option). A high number of comparisons of the (noisily estimated) qualities, determined by the large swarm size, implies a high number of mistakes and thus a long time to reach the consensus. The direct comparison is the only strategy that is going faster with the increase of the value of σ.

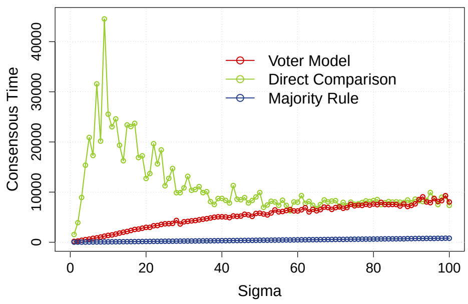
Concerning the exit probability (Fig. 4.11(d), 4.10(d), 4.11(b) 4.10(b)) we notice a quite uniform trend for the direct comparison and the weighted voter model. As expected, the exit probabilities for direct comparison and weighted voter model are higher in the simple case than in the difficult one. However, the trend in the two cases is the same: the exit probability of the direct comparison is constant, while the one for the weighted voter model is uniformly growing until a certain point, before a constant trend. Moreover, we see how increasing the swarm size favours the accuracy and slows down the time needed to reach the consensus for all the strategies. The majority rule, that has been shown to be highly dependent on the initial proportion of black robots, increases its accuracy when increasing of σ. From Fig. 4.11(d) and 4.10(d), we notice also the importance of the swarm size for this strategy. Indeed, σ has no influence on the exit probability (that is always equal to 0)



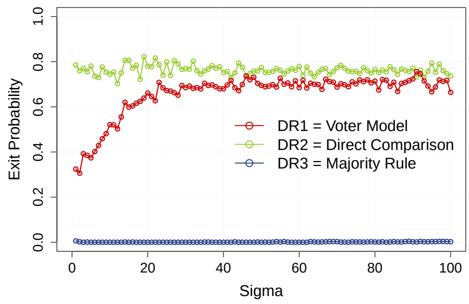
(a)



(b)



(c)



(d)

Figure 4.11: The graphs show the dynamics of the exit probability and consensus time for a range of values of σ obtained in physics-based simulations, with a swarm size of 100 robots. The points represent the average of the consensus time (or of the exit probability) obtained with 1000 runs with the same difficulty. The three colors represent the different decision rule used: red = weighted voter model, green = majority rule, blue = majority rule. Parameters: $\sigma =(1, 2, \ldots, 100)$%, $g = \sigma$, black robots = 30% of the swarm, white robots = 70% of the swarm. a)Consensus time obtained with a swarm size of 100 robots, $rho_{black} = 66\%$, $\rho_{white} = 100 - rho_{black}$; b) Exit probability obtained with a swarm size of 100 robots, $rho_{black} = 66\%$, $\rho_{white} = 100 - rho_{black}$; c) Consensus time obtained with a swarm size of 100 robots, $rho_{black} = 52\%$, $\rho_{white} = 100 - rho_{black}$; d)Exit probability obtained with a swarm size of 100 robots, $rho_{black} = 52\%$, $\rho_{white} = 100 - rho_{black}$.

when a swarm of 100 robots is used (recall that we are in the situation where there is *only* 30% of initial black robots and, with $\rho_{black} = 52\%$, in this point the majority rule has always exit probability equals to 0, Fig. 4.8(d)). This is not true for the swarms with 20 robots (Fig. 4.8(b)). In this situation, the increase of σ actually increases the accuracy of the strategy.

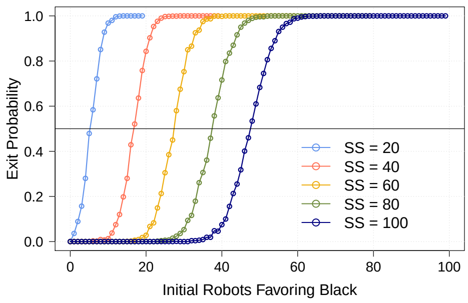
4.4 Additional Analysis of Exit Probability for Majority rule

As previously touched on (Section 4.3.3), we noticed a difference in the behaviour of the exit probability when using the majority rule between our work and the works of Valentini et al. [101], and Montes et al. [64]. Both in our study and in their studies, the majority rule approaches a step-shaped curve when using a larger swarm size. More specifically, Valentini et al. showed that when the difficulty of the problem is similar to our simple scenario (i.e., $\rho_{black} = 66\%$ vs. $\rho_{white} = 34\%$), the exit probability approaches a step function and the center of the step (i.e., the point where the exit probability is 0.5) corresponds, on the X-Axis, to an initial condition where approximately the 30% of the swarm favours the best option. The difference is that, in our scenario, with the same conditions, the exit probability for the majority rule is equal to 0.5 when the initial robots favouring the best option is 47.

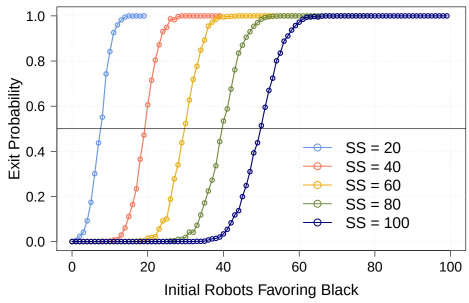
One explanation of this result can be found in the swarm size. A higher number of robots in the same environment causes high interferences and a high rate of collisions. The dissemination time (recall that it is determined also by the weighting factor, g) is an important factor since with low dissemination times and high collisions, the well-mixing of the swarm can be not ensured. In this section, we present the additional experiments performed in order to understand and explain the anomalous behaviour of the exit probability with the majority rule. For this purpose, we are going to freely vary the parameters in a different way than previously.

First, we want to show the trend of the exit probability with the same parameter that we have used in Section 4.3.1 but using different swarm sizes. More precisely, we show the exit probability for swarm sizes of 20, 40, 60, 80 and 100 robots. We present the results of these experiments both in the simple and in the difficult scenario respectively in Fig. 4.12(a) and Fig. 4.12(b). From these graphs, we can see how, in both scenarios, the exit probability approaches a step curve when the swarm size becomes larger.

The other result that emerges is the difference between the exit prob-



(a)



(b)

Figure 4.12: In the graphs are shown the dynamics of the exit probability varying all the initial conditions about the initial robots favouring the best option with different swarm sizes. The points represent the exit probability obtained with 1000 runs in the same condition. Every curve represent a different swarm sizes: light blue=20 *Robots, red*=40 *Robots, gold*=60 *Robots, dark green*=80 *Robots, dark blue*=100 *Robots. Parameters:* $\sigma = 10$, $g = \sigma$, *black robots* = $(1, 2, \ldots,$*swarm size), white robots = swarm size - black robots. a) Simple scenario; b) Difficult scenario;*

ability values in the two scenarios. The curves are shifted to the right in the more difficult scenario, with respect to the simpler one. In the difficult scenario (4.12(b)) we see that, even when increasing the swarm size, the resulting exit probability is equal to 0.5 when the initial proportion of black robots is around the 50% of the swarm. For the simpler scenario, instead, the points where the exit probability is equal to 0.5 (indicated by the intersection between the black horizontal row with the coloured curves) are generally characterized by a lower number of initial black robots. We already showed (Fig. 4.3.1) that, for a swarm size of 100 robots, in the simple case the exit probability is equal to 0.5 when the initial number of black robots is 47 (0.47%). From the curve representing the experiments done with a swarm size of 20 robots, instead, the exit probability is equal to 0.5 when the initial black robots are 5 (0.25%). This value is important because it shows that, for smaller swarm sizes, the results obtained are consistent with the results of by Valentini et al. and Montes et al. ([101], [64]). Moreover, the hypothesis that the collisions affect the well-mixing properties of the swarm favouring the clustering of the opinions is confirmed.

In view of these results, we decided to test the effects on the exit probability when using a longer dissemination state, in the two scenarios (simpler and more difficult). For this purpose, we therefore performed experiments using a swarm size of 100 robots, with $\sigma = 10s$, and applying the majority rule. We decided to use a parameter $g = 100s$, because in this way the

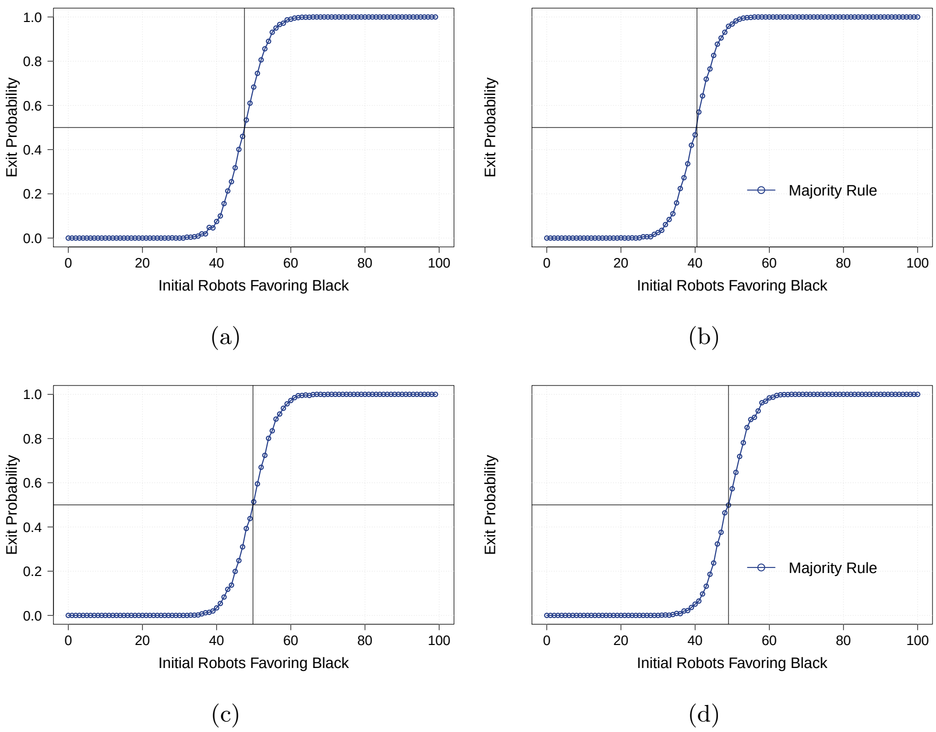


Figure 4.13: In the graphs are shown the dynamics of the exit probability. In the picture on the left side, are reported the results obtained with the experiments described in 4.3.1. On the right side, are reported the results of the experiments with exactly the same parameters but with g=100. The X-Axis represent the initial number of robots favouring the black option. On the Y-Axis there is the exit probability with the respective initial conditions. The points represent the exit probability obtained with 1000 runs in the same condition. The horizontal black lines are lines to show where the exit probability is 0.5. They have been drawn only for better visualize that point. The vertical black lines are the intersection between the curves and the horizontal lines. They indicate which is the initial condition of robots favouring the best option to obtain an exit probability of 0.5. Parameters: $\sigma = 10s$, black robots = $(1, 2, \ldots, 100)$, white robots = 100 - black robots, decision rule = majority rule. a) simple scenario, $g = 10s$, exit probability = 0.5 when initial black robots = 47; b) simple scenario, $g = 100s$, exit probability = 0.5 when initial black robots = 40; c) difficult scenario, $g = 10s$, exit probability = 0.5 when initial black robots = 50; d) difficult scenario, $g = 100s$, exit probability = 0.5 when initial black robots = 49.

robots have the possibility to listen to the neighbours, therefore avoiding the clustering of the opinion (recall that g is the parameter that has to be weighted to define the mean of the dissemination state). By setting g=100s, we obtain dissemination times ten times longer in average. In the other experiments, we always used g=10s. This choice was made because of the battery limitations of the real robots. With lower values of g, the consensus time is shorter and the robots do not occur in limitations due to the batteries. With higher values of g, we expect that the robots have more time to mix with other robots, receiving opinions of more robots in different zones of the environment. The considered opinions will only be the last 2 received in the duration of the state (Section 4.2.2).

Fig. 4.13 shows the results of the exit probabilities using the majority rule both in the cases where g=10s and g=100s, in order to help the reader in the comparison between the two cases. Moreover, we show the results of the exit probabilities both in the difficult and in the simple scenario, respectively in Fig. 4.13(c), 4.13(d) and Fig. 4.13(a), 4.13(b). First, we analyse the difficult scenario. We notice that in the two graphs the center of the step curves (i.e., the point where the exit probability is 0.5) is really close. The calculated initial number of black robots needed to get an exit probability equal to 0.5 (indicated by the intersection between the two black lines and the curves) is 49 when g=100s (Fig. 4.13(d)) and 50 when g=10s (Fig. 4.13(c)).

In the simpler scenario, instead, the difference is bigger. It is because the larger gap between the qualities of the two options. Indeed, in the simpler scenario, the modulation of positive feedback is stronger than in the difficult one. Moreover, we see how, when increasing the value of g, the modulation of the positive feedback has a stronger influence: the center point of the step curve is shifted to the left when a longer dissemination time is used. The graphs in Fig. 4.13(a) and Fig. 4.13(b) show that the center point corresponds to an initial number of black robots equal to 47, in the case of using g=10s, while it is equal to 40.5 when $g = 100s$. This shows the effect of the modulation of the positive feedback on higher times. This result is consistent with the works of Valentini et al. [101] and Montes et al. [64].

4.5 Overall Considerations

We briefly sumarize the results in order to give a more ordered idea of how the decision rules affect the dynamics of the system.

The *weighted voter model* is the slowest strategy when the problem is simple and the situation is not noisy.The parameter σ influences the strategy but does not compromise it: the times remain more or less uniform and the exit probability is always quite high.

The *direct comparison* rule is accurate in every situation. The exit probability is almost always the highest. The very big drawback of this strategy are mainly two: the heavy quantity of information exchanged and the extreme dependence on the environment. Difficult problems or high levels of noise in the quality estimations (i.e., low values of σ) results in an extremely long time to reach a consensus. Moreover, increasing the swarm size increase in a noisy way the consensus time.

The *majority rule* is the fastest strategy. Increasing the swarm size results in an exit probability function that approaches a step curve. This confirms the findings of Valentini et al. [101] and by Montes et al. [64]. However, when the swarm size is large, the majority rule needs to have a longer dissemination time in order to ensure an efficient modulation of the positive feedback. In Section 4.3.2 we saw how with g=10s the difference, using the majority rule, between the simple and the difficult scenario is really weak. In Section 4.4 we demonstrated that a higher dissemination time is required to ensure an efficient modulation of the positive feedback and to avoid the clustering of the opinion.

Another result that is commune to all the strategies is the increasing of the accuracy with the increasing of the swarm size.

Chapter 5

Real-Robot Experiments

We performed several real-robot experiments using e-pucks (See 3.1.2, [63]) in an experimental environment with the aim of validating the results obtained in simulation. More precisely, we showed the results in terms of consensus time and exit probability for three decision rules and in scenarios with different difficulties. The goal of these experiments was to test weighted voter model, direct comparison and majority rule in real-world conditions.

First, we conducted experiments aimed at understanding the capacities of perception and actuation of the robots (thus, to understand the performance of sensors and actuators). Second, we focused on the comparison of the results in terms of speed and accuracy of the solution. To ensure the correctness of the validation work, we conducted experiments in an environment that is as close as possible to that of the simulation studies.

5.1 Arena and Experimental Setup

The number of experiments performed in simulation was impossible to replicate in real-robots experiments, due to both availability and time limitations. The size of the swarm is limited by the number of robots available in our laboratory, that is equal to 20. Moreover, the time and effort required for a real-robot experiment is dramatically higher than required in simulations. Finally, simulations are processed by a cluster that can run thousands of parallel executions, while real-robot experiments cannot be parallelized. These are the reasons that pushed us to focus to a certain situation, with fixed initial conditions, and to run a lower number of runs than the ones made in simulations.

Let us recall that the goal of our swarm is to find which resource is the most available in the environment, where resources are represented by different colors on the floor. The floor of our environment is a chess-like arena with randomly disposed cells and with an unbalanced number of black and white cells. The percentage of black cells represents the difficulty of the problem.

The starting point for the real-robot experiments was to choose how the robots had to detect the color they were passing over. We chose to use the ground sensor to perceive the floor. E-pucks real sensors have physical limitations. The biggest limitation of the ground sensor (3.1.2) is that this sensor is only able to recognize grey-scale colors, reducing the range of possible colors for our floor to three: black, white, and grey. Nevertheless, using the ground sensor is the most adaptable and portable solution since, in a real-world case, the swarm cannot fully rely on an external infrastructure but has to operate with the equipped sensors.

Another option would have been to virtualize a coloured floor by means of the tracking system [95] available in our laboratory. We chose not to use this solution because it would have reduced the portability of our system, limiting its use in a controlled environment.

We opted for the real-sensor equipped on the e-pucks for the following reasons:

- The limitations of this solution were not influencing our work: we only needed to study a case of binary best-of-n decision-making problem (i.e., with two resources, black and white);
- The use of this solution in a possible real-world application is more credible than the other proposed;

.

5.1.1 Experimental Environment

The environment set up has been made as close as possible to the environment used in simulations. The first step was to physically set-up the arena. In order to replicate the floor analyzed in simulations we composed a paper ground of 2 $m * 2$ m size. For this purpose, we printed 4 sheets of heavy coated paper glued together. Each sheet was $1m^2$ size and represented one quadrant of the arena. The configuration of the squares on the floor was copied from a simulation test, and the squares were randomly disposed in

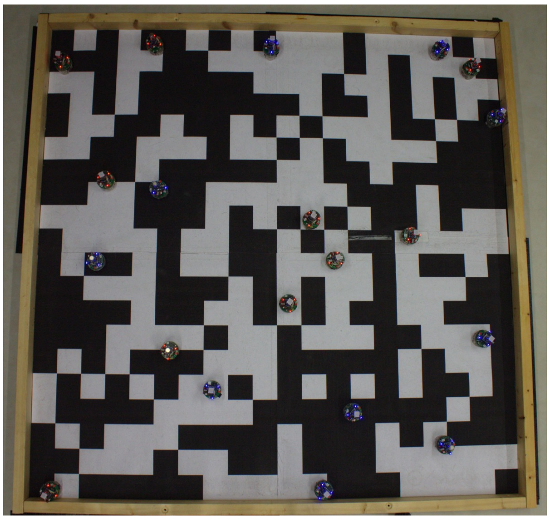


Figure 5.1: Picture of the swarm of 20 e-pucks running an experiment in the real-robots experiments environment

a grid. After that, we placed 4 wood sticks to enclose the environment, in order to avoid the robots to accidentally leave the arena (Fig. 5.1).

We recorded every experiment with the Iridia Tracking System [95] and we used the Wifi network to communicate with the the robots. The communication was peer-to-peer, that is, we were sending the messages and retrieving the saved information from each single robot. As in 2.1.3 a macroscopic human-swarm interaction is still lacking.

In order to reduce noise to its minimum, we calibrated every sensor of the robots. The performance of the sensors depends on the environmental conditions. We calibrated the sensors under a controlled condition of light brightness, that were never changed during all the experiments.

The output of the experiments consists of a text file for each robot. During the experiments, every robot was saving its own opinion at each time step. This allowed usto recover the evolution in the time of the robot states after the experiments.

Moreover, the e-pucks were visually displaying their actual opinion to the eventual observers through coloured LEDs. The black opinion was signaled through turning on the red LEDs, while the white opinion by turning on the blue LEDs (see Figure 5.2). The robots were signaling if they were in dissemination state by blinking all the LEDs that were turned on.

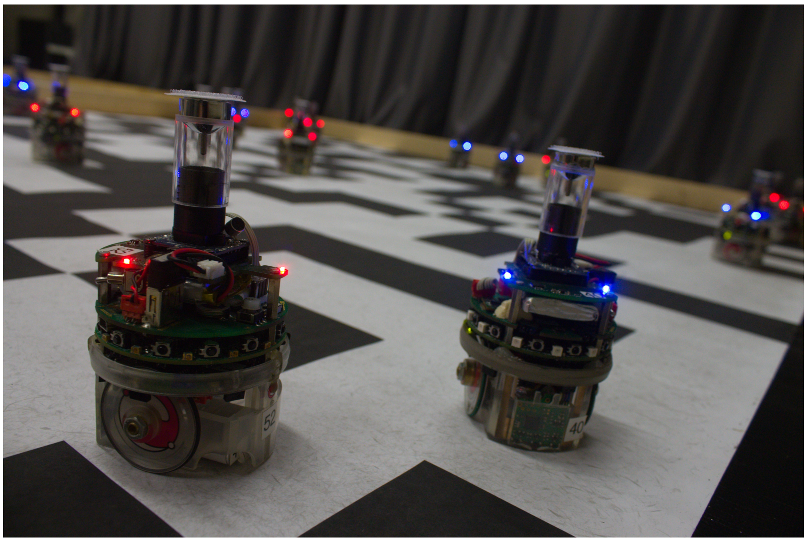


Figure 5.2: Blue and red blinking robots in a real-robots experiments: in the picture there is a robot with the blue LED on and a robot with the red LED on. They are advising the human being watching the experiment about their own opinion. In this way the designer knows when the experiment can be considered finished (when the swarm is entire showing the same color). In the background is possible to see a portion of the rest of the swarm, the coloured floor and the borders of the arena.

5.1.2 Choice of Initial Conditions

The sensitive parameters affecting the swarm-level behaviour are mainly three:

- Difficulty of the problem: we had to determine the proportion of the two resources place in the arena. In simulations, we tested the swarm's behaviour in two situations: one with relatively simple discrimination problem (66% of black cells vs 34% of white ones) and one more difficult problem (52% of black cells vs 48% of white ones). To ease the comparison with real-robot experiments, we prepared two arenas, one for the simple and one for the difficult scenarios;

- Initial fraction of robots favoring the best option: this is one of the most influential elements for the exit probability and consensus time variables. The safest choice is to use 50% of initial red-thinking robots and 50% of blue ones, because in this way the initial situation of the swarm is unbalanced;

The other parameters have been set in order to be as close as possible to the parameters used in simulations.

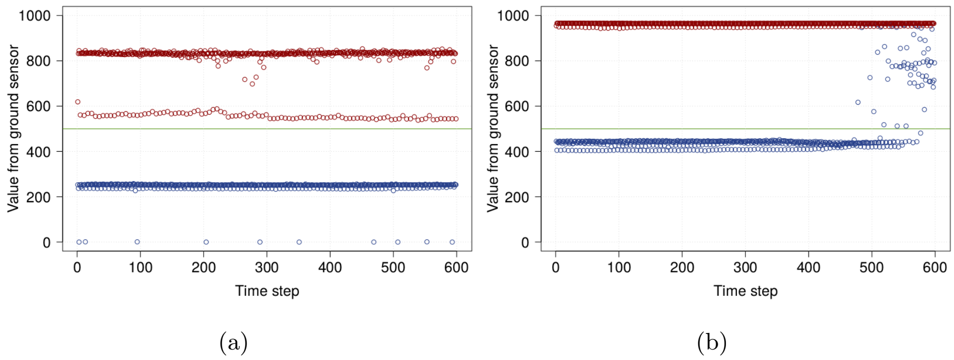


Figure 5.3: Returned data from the ground sensor during the execution of the test. $1\ Timestep = \frac{1}{10}\ sec.$ *The blue circles are the values returned by the sensor in the white floor test, the red circles are the value returned by the sensor in the black floor test. The green horizontal line represents the threshold value (*500*) chosen for our experiments.*
(a): Data relative to the calibration of the robot 44; (b): Data relative to the calibration of the robot 45;

5.1.3 Sensor Performance

To be certain about the correctness of the results and the reliability of the performance of the robots, we performed experiments to calibrate and analyze the robot sensors (we recall that the behaviour of each robot is subdivided into exploration state and dissemination state). The exploration state is characterized by the sensing of the floor in order to explore the environment. The dissemination state instead is characterized by the communication between robots that broadcast their opinion. Two sensors are mainly involved in these two states: the ground sensor (3.1.2), in order to distinguish the white cells from the black cells; and the range and bearing (3.1.2), used to make the robots communicate with their neighbours.

5.1.3.1 Ground Sensor

At each time step, the robot determines the color of the ground using its *ground sensor*. This sensor returns, every time step, a value between zero and 1000: if the ground is white then the returned value will be (ideally) 1000. Otherwise, if the floor is black, the value will ideally be zero. The e-pucks are endowed with three ground sensors, one in the center and two on the sides. We only used the central ground sensor in all the experiments. In this way, when the robot is between two or more cells the returned value will refer to the cell where the center of the robot is.

A real sensor the measurement is affected by numerous factors as the light intensity, the type of material of the floor. To have a good idea of the average estimation and to understand if we could have fixed a credible threshold to parse the returned value to black or white, we decided to put every robot on a black floor firstly and then on a white one and to analyze the data collected by the sensor. The results of the two tests are plotted in the same graph (Fig.5.3a,b). As shown in the graphs, the sensor is not extremely noisy. A reasonable and reliable threshold for the distinction between the black survey and the white survey is 500, as indicated in the graph by an horizontal green line.

Fig.5.3a shows the results of the calibration relative to the robot 44, while in Fig.5.3b shows th ones relative to the robot 45. We notice that every robot identifies the black and white with a certain value, that can depends both from the environment condition or from the robots' hardware. The ground sensor of the robot 44 returns values with a mean close to, approximatively, 200 and 800, when the robot is placed respectively on a black and on a white floor. The values returned by the ground sensor of robot 45, instead, have a mean around 1000 and 400, respectively for the white and the black.

We see that the sensor seems to return values that are distributed around two average values, for the white case (600 and 800)(Figure 5.3a). However, even in presence of the described noise, the sensor still returns values higher than 500 for the white floor and the outcome is not compromised. The noise is probably due to the fact that white paper has an higher transparency than the black one, where the robot returns values with a lower level of noise.

For the robot 45 instead, the characteristics of the noise are different. This robot' sensor is more cable to recognizing the white color, returning values really close to 1000. Additionally, the black color test is approximatively reliable and is characterized by relatively low variance. The noise, here, comes out after 500 time steps (when the calibration is going to finish). This error is probably due to interferences in the experiment that compromised the last part of the calibration. We reported this graph (Fig. 5.3b) in order to show how little changes in the environment conditions (e.g., the level of the light).

We ran this experiment using 8 robots and retrieved the data from each experiment. With these results, we validated the use of a threshold at 500 to reliably determine the color of the floor. We decided to report in this thesis the graphs relative to two experiments: the one done using the robot 44, that is one of the experiments run without errors, and the experiment done with the robot 45, that had a malfunctioning in the last time steps.

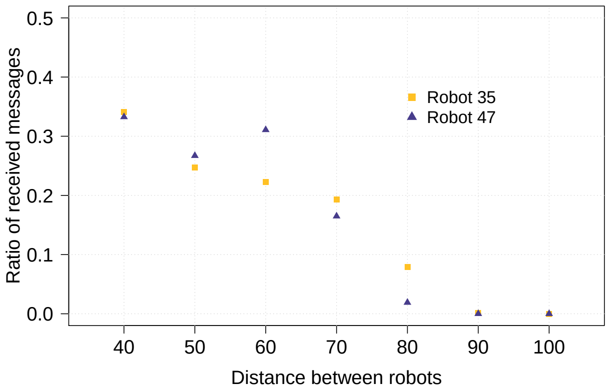


Figure 5.4: Ratio of the received messages over the sent ones when varying the distance between the two robots used for the experiment. On the X-Axis we report the distance between the robots, expressed in cm. We conducted one experiment for each measure of distance using the two robots. We report on the Y-Axes the ratio between the number of total sent messages during the experiment and the number of the received ones by the two robots. The yellow squares represent the reception ratio of the robot number 35, while the blue triangles the reception ratio of the robot number 47.

5.1.3.2 Range and Bearing

The robots exchange messages through the range and bearing board that sends and receives messages locally through infra-red communication. The range and bearing allows the robots to get a measure of both the range and the bearing of the robot that sent the received messages.

The communication is a crucial point in our strategy. We performed preliminary analysis about the quality and the quantity of messages received by the robots before the final experiments.

First, we analyzed the range of communication of the sensor. It occurred to be impossible to fix a credible range of communication with the range and bearing. It was therefore critical to know the number of packages are dropped in relation to the distance between two robots communicating by using the standard threshold of the robots.

For this purpose, we put two robots at a fixed distance. The two robots were sending messages and listening incoming messages while rotating on themselves. In this way, we could have a good estimation of the number of dropped messages as a function of the distance between the two robots. We repeated the same test placing the robots at different distances: from $1m$ to $40cm$ with a step of $10cm$.

The graph in Fig. 5.4 shows the relationship between the received mes-

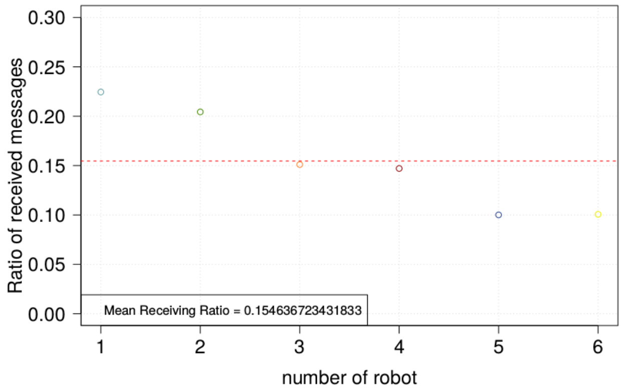


Figure 5.5: Ratio of the received messages over the sent ones relative to six robots. On the X-Axes there are the robots used for the experiments. On the Y-Axes there is the ratio between the number of total sent messages during the experiment and the number of the received ones by the two robots. Each coloured circle represents the receiving ratio of one robot. We run every experiment using two robots. The couple that worked together are: (Rob 1 vs Rob 3); (Rob 2 vs Rob 4); (Rob 5 vs Rob 6). The red dotted horizontal line represents the mean of the receiving ratio of all the experiments, that is indicated in the box at the bottom left corner of the graph.

sages and the sending distance. On the X-axes are reported the distances $(40, 50, \ldots, 100cm)$ between the robots during the experiments. In every experiment, the number of sent messages was known since the robots were sending a message in each time step. For every distance, we kept track of the received messages from the two robots and we calculated the ratio of the received messages over the total number of messages sent. In the graph, these values are reported with a blue triangle and with a yellow square, respectively for the robot number 47 and for the robot number 35. The ratio monotonically decreases with the increase of the distances. The only exception is made for the received messages by the robot 47 when placed $60cm$ far from the other robot. In this case, the ratio was higher than when the robot has been placed $50cm$ far from the other. It is however reasonable to assume that the maximum distance for the robots to receive a solid (i.e., higher than the 10% of the total sent messages) number of messages is $0.7m$. Above this distance the number of messages is not relevant. The ratio of received messages at 0.7 m is around the 20%.

The second experiment on the range and bearing board was similar to the first one, but we decided to test it in an enclosed arena with the robots moving. The border of the arena can mirror the messages, creating different dynamics in the range and bearing communication. Moreover, differently from the previous experiment, the robots are moving and are thus modifying the communication performance. We placed two robots performing random walk and obstacle avoidance (the same performed by the controller used for our final experiments, 3.2) in an arena with dimension $0.40m * 0.40m$, enclosed by four wood sticks. We repeated the same experiment with three different couple of robots (robots 1 and 3, robots 2 and 4, robots 5 and 6).

Fig. 5.5 shows the results of the experiments. We present in this graph the results of the communication between each couple of robots.

The results of this experiment confirmed us that, even when moving in an enclosed environment (with the above discussed size), the robots could successfully exchange approximatively 20% of the messages. The success rate of the robots is not uniform, as in the previous experiments. In this case the robots are receiving a different number of messages, even if in the same experiment, while in the previous experiment the ratio was approximatively the same for the robots involved. We notice that the robot 1, that was involved in the experiment with the robot 3, had a success rate around 0.225 while the robot 3 had a success rate around 0.15, thus closer to the mean.

After estimating the range of communication between two robots in a controlled situations, we wanted to understand how does the communication work between robots in our particular scenario: in an arena $(2 * 2)m^2$ size, with twenty robots walking randomly and sending messages for an exponential period of time (dissemination state 3.2).

In the dissemination state every robot sends messages for a time controlled by an exponential random distribution weighted with the quality of the opinion favoured by the robot. The parameter of the exponential distribution is

$$DiffTime = \rho_i * g + l;$$

where rho_i is the quality of the fevering opinion estimated in the previous exploration state, g is the mean time that must be weighted with the quality and l is a fixed period in which every robot, besides to sending messages, also listen the opinions of the neighbours. l is the final part of the dissemination state, i.e., the listening task occurs in the last l seconds of the dissemination state.

The goal of this experiment was to select a feasible listen time (l). The constraint on the listening time was introduced in order to avoid the listening of obsolete opinions (i.e., we do not want to take into accounts the opinions from the robots that have already change the opinion). Listening just in

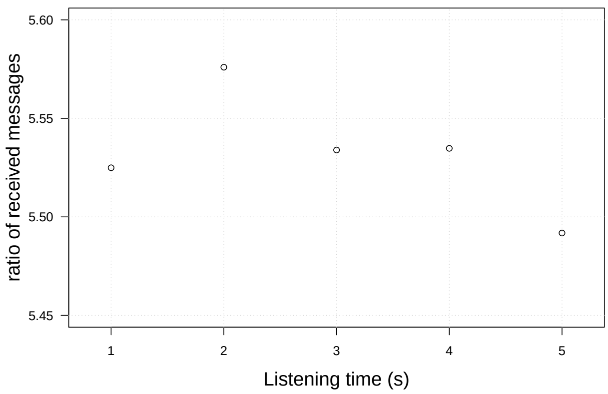


Figure 5.6: Neighbourhood size varying listening time (l). On the X-axes is present the duration of the listening period (in seconds). On the Y-axes there is the average number of received messages by all the robots during all the dissemination states (i.e., the neighbourhood size).

the last l seconds, the probability of listening obsolete values is minimized. We performed preliminary experiments in order to see how the listening time is influencing, in experiments with real robots, the number of different messages received by every robot in one dissemination state, that is, the neighbourhood size. We recorded the messages received from every robot in the same way we save them in the definitive experiment. The policy of recording of the message is the following:

- The robots skim off repeated messages: every robot takes only *one* message in consideration from each robot in every dissemination state;
- The robots throw away the "default" values: the range and bearing is always sending messages, even when no values have been set to be send (for example the robots send messages even in the exploration state). In order to avoid the reception of these messages, we set a default value that the robots recognize as default value, to be throw away;

Every experiment was characterized by a duration of $10minutes$ and by a different listening time. We varied the duration of the listening time l from $5s$ down to $1s$, with a step of $1s$.

The graph in Fig. 5.6 shows the overall average of the incoming messages of every robot for every dissemination state for all the duration of the experiment. It emerges that, considering a maximum listening time of $5s$, the average number of received messages from every robot is more or less constant (5 messages received). Of course this result could change if we consider an infinite listening time. As previously explained we take in consideration only a limited listening time in order to keep the property of receiving new messages, avoiding the listening of obsolete information.

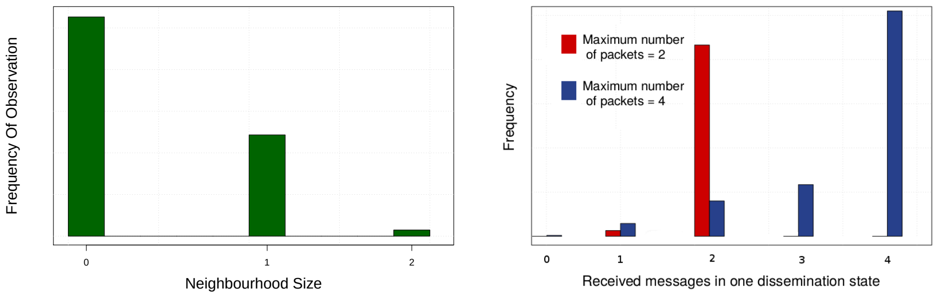
As a rule of thumb, we set the range of communication of $21cm$, that is, three times the diameter of the e-puck. In this way, the number of exchanged messages would have been limited and in some way controlled. In Fig. 5.7(a), this graphs emerges from experiments done in simulation and better explained in chapter 4, is possible to see the neighbourhood size obtained by the experiments with a range of $0.21cm$ without any limitations or control. From this graph we can notice that the robots *never* receive more than two messages per dissemination state.

We would like to have a situation as close as possible to the ideal one, but the limitations discussed about the range and bearing board sensibility did not allow us to limit the range. Varying the listening time did not prove effective in limiting and controlling the number of messages received. We decided to put a stronger software-level limit on it, both in simulations and in real-robots experiments.

For this purpose, we observed how varies the controlled neighbourhood size in our scenario with a k maximum number of incoming messages. The robots listen every incoming messages but save only the last k values. In this way are ensured both the limitation on the number of incoming messages, and the recentness of the information. We choose to test:

- $k = 2$;
- $k = 4$;

The results of this test are shown in Fig. 5.7(b). The histograms show the frequency of the dissemination states characterized by n incoming messages (where $n \in [0, 1, \ldots, 4]$) in the two cases where we limited the incoming messages to 2 and 4, respectively with the blue and red colors. In the two cases, the column relative to the maximum value (2 and 4) are the highest because when are listened more than k messages then the number of incoming messages is limited to k. According with the results of Fig. 5.6 is evident that there are, in a large majority of the cases, more than 2 incoming messages in real-robots experiments. Indeed, in average, the robots receive 5 messages per dissemination state. This fact is supported by the number of time in which have been recorded 4 (or more) incoming messages. Is then clear that if we limit the number of incoming messages to 4 then we would



(a) (b)

Figure 5.7: Neighbourhood size limiting number of incoming messages to 2 *and* 4

have had a strong discrepancy with the simulations case. Looking to Fig. 5.6, we see that the incoming messages are always less then 2. For this reason we choose to adopt the policy to save only 2 messages per dissemination state. There is, although, a difference that cannot be avoided: in simulation the majority of the times the number of recorded messages are 0 or 1. That is not true in real-robots experiments where, due to the impossibility to fix a range of communication, we cannot adjust this value. In this case the number of incoming messages is, for a large majority of the time, 2.

5.2 Analysis of Exit Probability and Consensus Time

We studied the environment classification with two scenario, one simpler and one more difficult. In the two scenarios the ratio between the black and white qualities qualities are, respectively, 0.5151 and 0.923, that is:

- Simple scenario: 66% black quality and 34% white quality;
- Difficult scenario: 52% black quality and 48% white quality;

For each scenario we have conducted 45 runs, 15 for each decision rule for a total of 90 runs. We are going to show the results obtained from the experiments in both the cases and subdividing for each case the three decision rules.

The output of each slot of runs (i.e. the 15 runs for each decision rule in each scenario) is synthesized by two graphs (all the graphs are reported in Fig. 5.8 and in Fig. 5.9): in one graph we have plotted the trading of each run, representing with the color blue the number of white fevering robots and with the color red the black ones; in the second graph we have plotted the boxplots of the same trading of the whole set of runs.

5.2.1 Simple Scenario

In a simple-scenario set-up the number of the black cells (the most valued option) is the double with respect of the white ones. In this situation the gap between the two qualities is relatively high and the level of noise in their evaluation is really low. Generally speaking, without going deeper in the decision rules results, in this scenario we expect that the swarm will often choose the best option. Both the exit probability will be high and the needed time to reach a consensus relatively low.

Simple scenario set-up:

- Arena: squared arena of $2m * 2m$ composed by a coloured floor as described in 3.2;
- Difficulty: 66% of black cells versus 34% of white cells;
- Swarm: group of 20 e-pucks, as described in 3.1.2, composed by an equal number of initially favoring black and white robots;
- Application of the three decision rules: direct comparison, majority rule, and weighted vote model;
- $\sigma = 10s$;

The results of the experiments done in the simple scenario are shown in Fig. 5.8 and will be discussed in 5.3.

5.2.2 Difficult Scenario

In the difficult scenario the gap between the two qualities is really small and the level of noise in their evaluation is higher with respect to the other case. In this scenario we are expecting that the choice of the swarm will be more variable. The swarm will probably mistake more often and the average consensus time will be much higher with respect to the first scenario.

To summarize this scenario we list the features of the experiment:

- Arena: squared arena of $(2 * 2)m^2$ composed by a coloured floor as described in 3.2;
- Difficulty: 52% of black cells versus 48% of white cells;

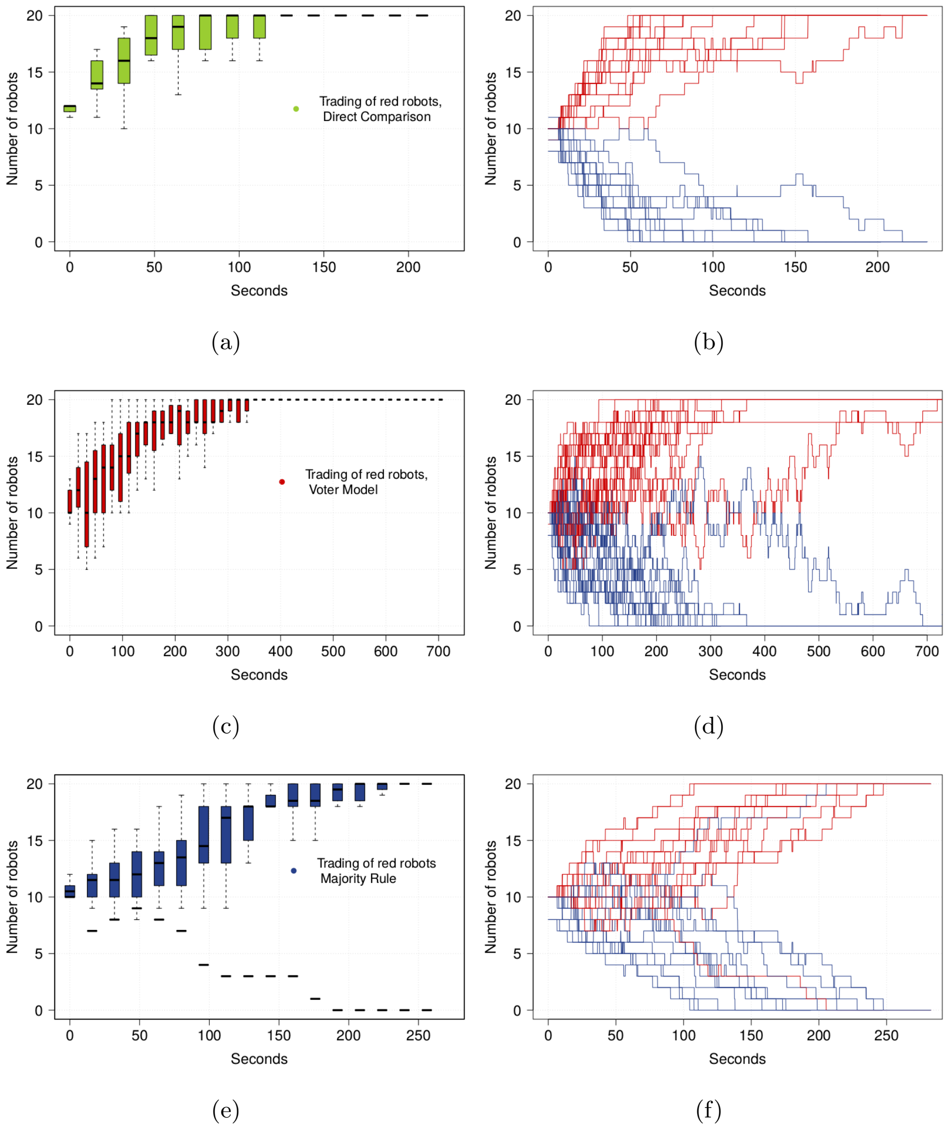


Figure 5.8: Graphs deriving from the real-robots experiment in the simple scenario. The graphs show the trend of the robots' states during the execution of the experiments. In the graphs in the left column are shown the boxplots relative to the overall experiments done with every decision rule. In the graphs in the right column is individually plot the trand of every run. The blue lines represent the number of robots favoring the white option in every time step. The red lines represent the number of robots favoring the black opinion in every time. $1\ Timestep = \frac{1}{10}\ sec.$ *a) Boxplot of runs with direct comparison; b) Single runs with direct comparison; c) Boxplot of runs with weighted voter model; d) Single runs with weighted voter model; e) Boxplot of runs with majority rule; f) Single runs with majority rule. Parameters: Swarm size* $= 20$; *Red robots* $= 50\%oftheswarmsize$; *Blue robots* $=50\%oftheswarmsize$; *Difficulty:* $\rho_{Black} = 66\%,\ \rho_{White} = 34\%$; $\sigma = 10s$; $g = 10s$;

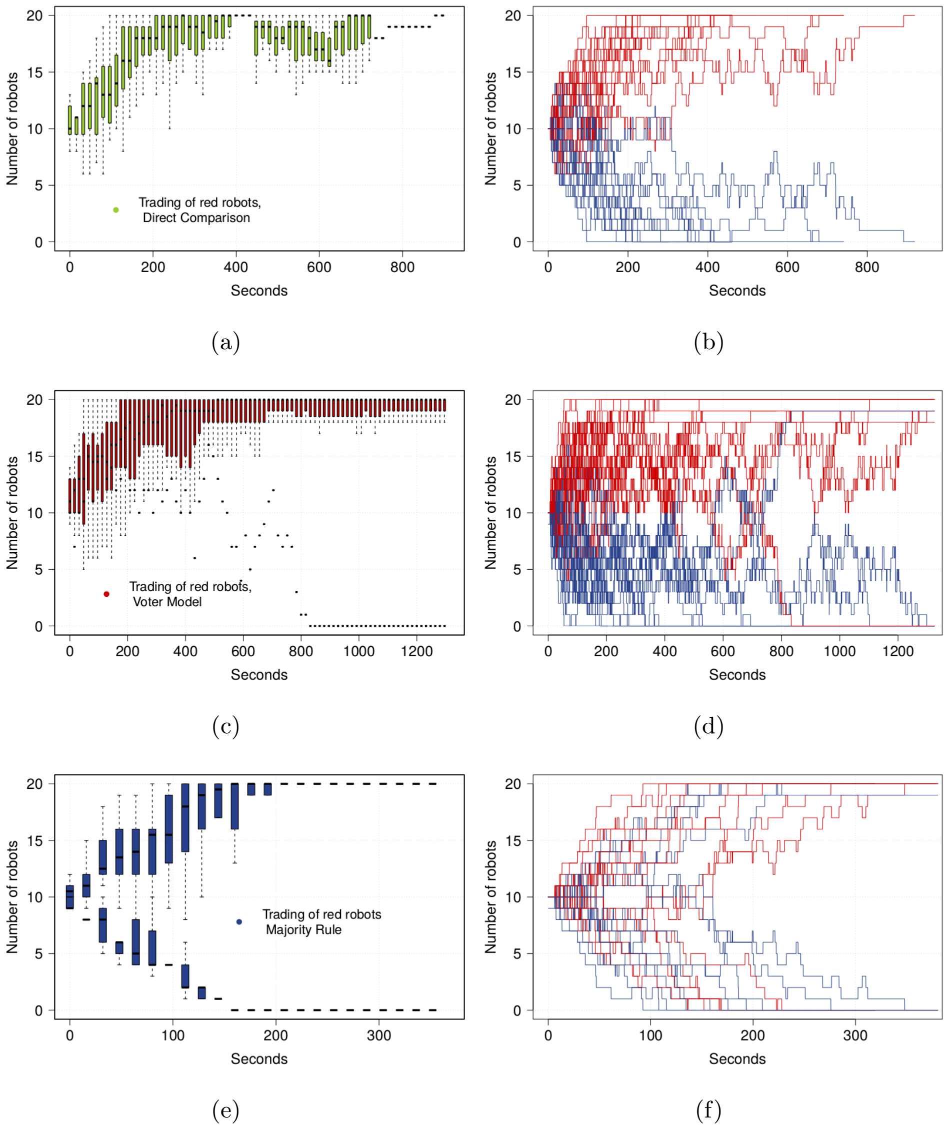


Figure 5.9: Graphs deriving from the real-robots experiment in the difficult scenario. The graphs show the trend of the robots' states during the execution of the experiments. In the graphs in the left column are shown the boxplots relative to the overall experiments done with every decision rule. In the graphs in the right column is individually plot the trand of every run. The blue lines represent the number of robots favoring the white option in every time step. The red lines represent the number of robots favoring the black opinion in every time. $1\ Timestep = \frac{1}{10}\ sec.$ *a) Boxplot of runs with direct comparison; b) Single runs with direct comparison; c) Boxplot of runs with weighted voter model; d) Single runs with weighted voter model; e) Boxplot of runs with majority rule; f) Single runs with majority rule. Parameters: Swarm size* $= 20$; *Red robots* $= 50\% of the swarm size$; *Blue robots* $=50\% of the swarm size$; *Difficulty:* $\rho_{Black} = 52\%,\ \rho_{White} = 48\%$; $\sigma = 10s$; $g = 10s$;

- Swarm: group of 20 e-pucks, as described in 3.1.2, composed by an equal number of initially fevering black and white robots;
- Application of the three decision rules: direct comparison, majority rule, and weighted vote model;
- $\sigma = 10$;

The results of the experiments done in the difficult scenario are shown in Fig. 5.9 and will be discussed in 5.3.

5.3 Overall Consideration

In the graphs reported in Fig. 5.8 and Fig. 5.9 are shown the results of the experiments done with real robots in the, respectively, the simple and the hard decision-making problem. The results have been shown in two forms: in one type of graphs (Fig. 5.2.1(b),(d),(f) and Fig. 5.2.2(b),(d),(f)) the run are plot individually, showing the trading of the number of robots favoring the two options in every time steps; in the other graphs (Fig. 5.8(a),(c),(e) and Fig. 5.9(a),(c),(e) are reported the boxplot relative to the overall runs for each decision rule. In the graphs on the right, the red lines represent the number of robots favoring the black option, while the blue lines the robots favoring the white option.

We tested the three decision rules applied to two scenarios, a simpler one and a more difficult one. The main difference between the two scenarios is the distance between the qualities of the two resources: in one situation the gap is really low (only 4%) while in the other is bigger (32%). This factor influences the direct comparison more, that is, the decision rule that seems to reflect differences in the behavioural dynamics the most. Indeed the estimation of the quality in this decision rule assumes a central role.

In direct comparison, the quality is directly used to compare the two opinions in the decision-making process. This is the main factor influencing the switching of the opinion of the robot. A very accurate estimation of the quality ensures a correct behaviour of the robots in the case of small gap in terms of difference of qualities. Otherwise, the best valued opinion could be erroneously estimated as worst than the not-best-valued one, implying the not correct decision of the robot.

From the 5.8(a),(b) and 5.9(a),(b) is easy to see the difference in the behaviour of this decision rule. We have done only 15 runs in real-robots experiments, but every run ended with the swarm taking the right choice, both in the simple and in the difficult case. The big differences are twofold: the consensus time and the oscillatory behaviour of the swarm.

In the simple scenario the maximum consensus time observed is close to $220s$, while in the difficult scenario is higher than $800s$ ($> 400\%$). The trend of the red robots and of the blue robots is clean and rather monotonic, in the simple scenario, where the robots favoring the black opinion is uniformly and almost monotonically growing in all the runs. This feature is not essentially different in the difficult scenario: before reaching the point where the swarm is monotonically changing opinion the robots keep switching from black fevering back to white fevering and vice-versa. This behaviour is directly linked to the longer time to reach the consensus and is due to the over or under estimation of the qualities, as described in the first paragraph.

Figures 5.8(c),(d) and 5.9(c),(d) show the results about the application of the weighted voter model. This decision rule is the one that takes longer than all the other in both the scenarios. In the first case the maximum time needed to reach the consensus is about $700s$, while in the second one is close to $1200s$. However, even incrementing the difficulty of the problem, the increasing of the consensus time is lower than in direct comparison: the difficult scenario takes less than 50% more in the difficult scenario (while in direct comparison takes 400% of the time more).

The accuracy of the decision rule is quite high: in the simple scenario all the runs exited with the right choice, while in the difficult one only one run was erroneous. The trading of the changing robots is pretty the same in the two scenarios: considering Figures 5.8(d) and 5.9(d), is possible to notice that the character of the changing swarm is not so monotonic: the oscillation is characteristic for this decision rule.

Indeed, the pool of collected information is composed in the following way: when the situation is balanced between number of black and white robots the probability to have, in the pool the probability to have black or white opinions is around 50%. Since a robot randomly picks up an opinion and blindly trust on it, it switches continuously from one opinion to another one. The correctness of the decision rule is ensured by the fact that for a long time the best opinion will be spread more.

In 5.8(e),(f) and 5.9(e),(f) are shown the results about the majority rule. This is the most studied decision rule in literature and is evaluating the overall pool of collected information: the most present one opinion is the one that the robot will adopt in the next state. First of all we must notice how the swarm, in the two scenarios, takes more or less the same time to reach the consensus. This fact suggests us that this decision rule is quite independent from the difficulty of the problem.

Even the oscillatory behaviour of the switching robots is similar: in both the cases the robots are “ confused ” until, more or less, the half of the overall execution time, after that the trade becomes monotonic. It is due to the fact that the number of robots fevering one opinion is quite higher than the other faction.

However, the accuracy of the swarm is quite low: in both the cases there are runs where the swarm is taking the wrong decision. Even in the simple scenario one run is failing, while in the difficult scenario there are much more runs where the swarm mistakes.

Overall, the majority rule is the fastest decision rule but the less accurate, while the voter model is the slowest. From the data we have, the direct comparison has resulted to be the most accurate since it never failed, but we do not have enough data to ensure it. Moreover we want to recall that the information exchanged in the direct comparison are the double than the information exchanged in the other strategies. Is possible to see how, while the weighted voter model and the majority rule are keeping the same trade in both the scenarios, the direct comparison is starting to have a much stronger oscillation when noise is introduced. It can be said also for the consensus time: while in presence of higher level of noise, with majority rule and with voter model the swarm takes *more or less* the same time to reach the consensus, the direct comparison is taking 4 times more.

It suggests us that direct comparison is very quality-dependent and that, if the level of noise increases this decision rule leak in integrity. Majority rule and weighted voter model, instead, are more self-organizing and flexible since the behaviour, even increasing the level of difficulty of the problem, remains quite constant.

Chapter 6

Conclusions

Swarm robotics is a relatively new approach to the coordination of large groups of simple robots aiming to achieve together a complex task. A particular sub-category of problems of swarm robotics is called collective decision making. In collective decision-making problems, all the robots of the swarm have to agree toward the same option chosen among a set of possible alternatives, that is usually the one that maximizes a metric of the problem. Examples of collective decision making are the subdivision of a set of tasks among the robots of the swarm, or the selection of the best alternative among a set of possible ones. In this thesis, we presented a self-organizing, decentralized, general, and portable solution to the environment classification problem, a best-of-n decision-making problem. We tackled environment classification as a binary best-of-n decision-making problem. Furthermore, we analysed the behaviour of the swarm by applying three different decision-making rules (weighted voter model, majority rule, and direct comparison) in terms of accuracy and speed of the solution. This has been done both with physics-based simulation and with a swarm of real robots. In this Chapter, we summarize the contributions emerged during the course of this work, we give an overview of the obtained results, and we provide directions for future lines of research.

6.1 Results and Contributions of the Thesis

In this thesis, we gave an empirical comparison between three different decision rules applied to a decentralized swarm of autonomous robots in order to solve the environment classification problem. The environment classification, that is a best-of-n decision-making problem where the swarm has to classify the environment by the resource that is most present in it. The correct solution for the swarm is to converge on a decision for which is the most available resource. The swarm's behaviour can be described with the exit probability, i.e., the accuracy of the solutions, and the consensus time, i.e., the time required by the swarm to reach a consensus. We studied the trends of these variables varying the parameters of the problem: the initial number of robots favoring the black option (i.e., the best option), the exploration time that is directly correlated to the accuracy of the quality estimation, and the difficulty of the problem. We showed which decision rule is better to apply in each different situation, such as in presence of higher or lower levels of noise, with a difficult problem or with a simpler one. Moreover we validated the comparison between the three strategy with a real-robots swarm.

The main contributions are the following:

- We gave a self-organizing, decentralized and portable solution to a new problem called environment classification. Environment classification is a scenario of the best-of-n decision-making problem. We studied it with a binary set of alternatives;

- We made an extensive comparison of three different decision rules applied to the environment classification. We analysed the dynamics of the swarm under a wide range of different initial conditions. We tested the behaviour of the swarm both in simpler and in more difficult conditions. We showed how the studied decision rules can tolerate high levels of noise and maintain good performance in presence of it (i.e., weighted voter model) while others are completely depending on the absence of noise (i.e., the direct comparison);

- We made the analysis for the direct comparison, a decision rule never studied in this field. It is characterized by an heavy exchange of information and by the direct comparison of the estimated qualities in order to select the new opinion;

- We performed extensive experiments with real robots in a real world environment, comparing the effects of the three strategies. A comparison between different strategies using a real robots swarm is an approach that aim to validate the comparison obtained with physics-based simulations and to see which are the effects in a real-world conditions.

With our work we showed the advantages of using the different strategies. From the results emerges that the weighted voter model is the more reliable (i.e., good accuracy with a reasonable consensus time) decision rule in presence of high noise and difficult problems. The weighted voter model has been showed to be a slow strategy. However the performance of the swarm when this strategy is applied does not worsen so much with the increasing of the difficult of the problem or of the noise. We show that the direct comparison is extremely influenced by the accuracy of the quality estimations, hence by the duration of the exploration and of the problem's difficulty. This decision rule, that is exchanging double of the information exchanged by the others two, is really accurate even in noisy and difficult condition. A price to pay to have accurate solutions in difficult and noisy condition is the *extremely* high time required to reach a consensus in a difficult scenario with an high number of robots, making this decision rule not adoptable in these conditions. The majority rule is, for every initial condition, the fastest strategy. Moreover, the majority rule takes a time similar both in simple and in difficult scenarios, even being of course slower for difficult problems. However, majority rule is the less accurate one and the one more influenced by the initial number of robots favoring the best option, but it is the fastest.

An overall result is the improvement of the performance in terms of accuracy of all the strategy with the increasing of the swarm size. This result is consistent with the principles of swarm robotics. However, the consensus times with larger swarm sizes is, inevitably, higher than in case of smaller swarm sizes. The number of communication between robots is higher, and the robots to be "convinced" is higher.

6.2 Future Lines Of Research

Our study concerns a binary best-of-n decision-making problem. In our scenario the floor is fully covered by cells of two colors, representing an abstraction of two resources to be classified. The problem is extendible to the case with $n > 2$ options. The analysis of a scenario with more than two resources would require an effort in order to analyse the effects of having an initial number of robots favoring the difference options. The space of initial condition would grow combinatorily as a function of the initial robots favoring the different options.

Another situation that we did not take into account is the distribution of the resources in the environment. In our case we uniformly distributed the resource in the environment. A question that can provide a new spark for a future work is how the swarm could react to the studied strategy in a scenario where the resources are distributed in a determined way on the environment. What can happen if the resources are clustered in different zones of the floor? What about, for example, if all the black cells are placed in a rounded surface in the middle of the arena?

We studied the solution for an homogeneous swarm, where each robot acts in the same way and applies the same decision rule. A different kind of work could be the one of compare these results with the results of a solution with an heterogeneous swarm where the robots act in the same way but is applying a different decision rule. For example, the combination of apply the majority rule to the half of the swarm and the weighted voter model to the other half, would speed-up the performance obtained by the voter model? Or would improve the accuracy of the majority rule?

Bibliography

[1] Jean-Marc Ame, Colette Rivault, and Jean-Louis Deneubourg. Cockroach aggregation based on strain odour recognition. *Animal behaviour*, 68(4):793–801, 2004.

[2] Ronald C Arkin and George Bekey. *Robot colonies*. Springer Science & Business Media, 2013.

[3] W Ross Ashby. Principles of the self-organizing system. In *Facets of Systems Science*, pages 521–536. Springer, 1991.

[4] Erkin Bahceci, Onur Soysal, and Erol Sahin. A review: Pattern formation and adaptation in multi-robot systems. *Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-03-43*, 2003.

[5] Michele Ballerini, Nicola Cabibbo, Raphael Candelier, Andrea Cavagna, Evaristo Cisbani, Irene Giardina, Alberto Orlandi, Giorgio Parisi, Andrea Procaccini, Massimiliano Viale, and Vladimir Zdravkovic. Empirical investigation of starling flocks: a benchmark study in collective animal behaviour. *Animal Behaviour*, 76(1):201 – 215, 2008.

[6] Jan Carlo Barca and Y Ahmet Sekercioglu. Swarm robotics reviewed. *Robotica*, 31(03):345–359, 2013.

[7] Gerardo Beni. From swarm intelligence to swarm robotics. In Erol Şahin and William M. Spears, editors, *Swarm Robotics*, volume 3342 of *Lecture Notes in Computer Science*, pages 1–9. Springer, 2005.

[8] Eric Bonabeau, Marco Dorigo, and Guy Theraulaz. *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press, New York, 1999.

[9] Manuele Brambilla, Eliseo Ferrante, Mauro Birattari, and Marco Dorigo. Swarm robotics: a review from the swarm engineering perspective. *Swarm Intelligence*, 7(1):1–41, 2013.

[10] A. Brutschy, A. Scheidler, E. Ferrante, M. Dorigo, and M. Birattari. “Can ants inspire robots?” Self-organized decision making in robotic swarms. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 4272–4273. IEEE Press, 2012.

[11] Scott Camazine, Jean-Louis Deneubourg, Nigel R. Franks, James Sneyd, Guy Theraulaz, and Eric Bonabeau. *Self-Organization in Biological Systems*. Princeton University Press, Princeton, NJ, 2001.

[12] A. Campo, S. Garnier, O. Dédriche, M. Zekkri, and M. Dorigo. Self-organized discrimination of resources. *PLoS ONE*, 6(5):e19888, 2010.

[13] Alexandre Campo, Álvaro Gutiérrez, Shervin Nouyan, Carlo Pinciroli, Valentin Longchamp, Simon Garnier, and Marco Dorigo. Artificial pheromone for path selection by a foraging swarm of robots. *Biological cybernetics*, 103(5):339–352, 2010.

[14] Karl Crailsheim. Trophallactic interactions in the adult honeybee. *Apis mellifera*, pages 97–112, 1998.

[15] Valentino Crespi, Aram Galstyan, and Kristina Lerman. Top-down vs bottom-up methodologies in multi-agent system design. *Autonomous Robots*, 24(3):303–313, 2008.

[16] M. Dorigo and M. Birattari. Swarm intelligence. 2(9):1462, 2007.

[17] M. Dorigo, M. Birattari, and M. Brambilla. Swarm robotics. *Scholarpedia*, 9(1):1463, 2014.

[18] M. Dorigo and T. Stützle. *Ant Colony Optimization*. A Bradford book. BRADFORD BOOK, 2004.

[19] Marco Dorigo. Swarm robotics: The coordination of robots via swarm intelligence principles. In Mike Hinchey, Anastasia Pagnoni, FranzJ. Rammig, and Hartmut Schmeck, editors, *Biologically-Inspired Collaborative Computing*, volume 268 of *IFIP The International Federation for Information Processing*, pages 1–1. Springer US, 2008.

[20] Marco Dorigo and Erol Şahin. Swarm robotics. *Autonomous Robots*, pages 111–113, 2004.

[21] Marco Dorigo, Dario Floreano, Luca M Gambardella, Francesco Mondada, Stefano Nolfi, Tarek Baaboura, Mauro Birattari, Michael Bonani, Manuele Brambilla, Arne Brutschy, et al. Swarmanoid: a novel concept for the study of heterogeneous robotic swarms. *Robotics & Automation Magazine, IEEE*, 20(4):60–71, 2013.

[22] F. Ducatelle, G. A. Di Caro, and L. M. Gambardella. Principles and applications of swarm intelligence for adaptive routing in telecommunications networks. *Swarm Intelligence*, 4(3):173–198, 2010.

[23] JolyonJ. Faria, JohnR.G. Dyer, RomainO. Clment, IainD. Couzin, Natalie Holt, AshleyJ.W. Ward, Dean Waters, and Jens Krause. A novel method for investigating the collective behaviour of fish: introducing robofish. *Behavioral Ecology and Sociobiology*, 64(8):1211–1218, 2010.

[24] Eliseo Ferrante, Manuele Brambilla, Mauro Birattari, and Marco Dorigo. Socially-mediated negotiation for obstacle avoidance in collective transport. In *Distributed Autonomous Robotic Systems*, pages 571–583. Springer, 2013.

[25] Eliseo Ferrante, Ali Emre Turgut, Cristián Huepe, Alessandro Stranieri, Carlo Pinciroli, and Marco Dorigo. Self-organized flocking with a mobile robot swarm: a novel motion control method. *Adaptive Behavior*, page 1059712312462248, 2012.

[26] Paola Flocchini, Giuseppe Prencipe, Nicola Santoro, and Peter Widmayer. Arbitrary pattern formation by asynchronous, anonymous, oblivious robots. *Theoretical Computer Science*, 407(1):412–447, 2008.

[27] Gianpiero Francesca, Manuele Brambilla, Vito Trianni, Marco Dorigo, and Mauro Birattari. Analysing an evolved robotic behaviour using a biological model of collegial decision making. In *From Animals to Animats 12*, volume 7426 of *Lecture Notes in Computer Science*, pages 381–390. Springer, 2012.

[28] Nigel R. Franks, Anna Dornhaus, Jon P. Fitzsimmons, and Martin Stevens. Speed versus accuracy in collective decision making. *Proc. R. Soc. Lond. B*, 270:2457–2463, 2003.

[29] Ryusuke Fujisawa, Shigeto Dobata, Daisuke Kubota, Hikaru Imamura, and Fumitoshi Matsuno. Dependency by concentration of pheromone trail for multiple robots. In *Ant Colony Optimization and Swarm Intelligence*, pages 283–290. Springer, 2008.

[30] Ryusuke Fujisawa, Hikaru Imamura, Takashi Hashimoto, and Fumitoshi Matsuno. Communication using pheromone field for multiple robots. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 1391–1396. IEEE, 2008.

[31] Serge Galam. Majority rule, hierarchical structures, and democratic totalitarianism: A statistical approach. *Journal of Mathematical Psychology*, 30(4):426–434, 1986.

[32] Serge Galam. Real space renormalization group and totalitarian paradox of majority rule voting. *Physica A: Statistical Mechanics and its Applications*, 285(1–2):66–76, 2000.

[33] Serge Galam. Sociophysics: A review of Galam models. *International Journal of Modern Physics C*, 19(03):409–440, 2008.

[34] L. Garattoni, G. Francesca, A. Brutschy, C. Pinciroli, and M. Birattari. Software infrastructure for e-puck (and tam). Technical Report TR/IRIDIA/2015-004, IRIDIA, Université Libre de Bruxelles, Brussels, Belgium, July 2015.

[35] S Garnier, C Jost, J Gautrais, M Asadpour, G Caprari, R Jeanson, A Grimal, and G Theraulaz. The embodiment of cockroach aggregation behavior in a group of micro-robots. *Artificial Life*, 14(4):387–408, Oct 2008.

[36] S. Garnier, F. Tache, M. Combe, A. Grimal, and G. Theraulaz. Alice in pheromone land: An experimental setup for the study of ant-like robots. In *Swarm Intelligence Symposium, 2007. SIS 2007. IEEE*, pages 37–44, April 2007.

[37] Simon Garnier, Jacques Gautrais, Masoud Asadpour, Christian Jost, and Guy Theraulaz. Self-organized aggregation triggers collective decision making in a group of cockroach-like robots. *Adaptive Behavior*, 17(2):109–133, 2009.

[38] Simon Garnier, Jacques Gautrais, and Guy Theraulaz. The biological principles of swarm intelligence. *Swarm Intelligence*, 1(1):3–31, 2007.

[39] S. Goss, S. Aron, J. L. Deneubourg, and J. M. Pasteels. Self-organized shortcuts in the argentine ant. *Naturwissenschaften*, 76(12):579–581, 1989.

[40] Roderich Groβ and Marco Dorigo. Evolution of solitary and group transport behaviors for autonomous robots capable of self-assembling. *Adaptive Behavior*, 16(5):285–305, 2008.

[41] A. Gutierrez, A. Campo, M. Dorigo, J. Donate, F. Monasterio-Huelin, and L. Magdalena. Open e-puck range x00026; bearing miniaturized board for local communication in swarm robotics. In *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pages 3111–3116, May 2009.

[42] Álvaro Gutiérrez, Alexandre Campo, Félix Monasterio-Huelin, Luis Magdalena, and Marco Dorigo. Collective decision-making based on social odometry. *Neural Computing and Applications*, 19(6):807–823, 2010.

[43] dmM. Halsz, Yanting Liang, M.Ani Hsieh, and Hong-Jian Lai. Emergence of specialization in a swarm of robots. In Alcherio Martinoli, Francesco Mondada, Nikolaus Correll, Grgory Mermoud, Magnus Egerstedt, M. Ani Hsieh, Lynne E. Parker, and Kasper Sty, editors, *Distributed Autonomous Robotic Systems*, volume 83 of *Springer Tracts in Advanced Robotics*, pages 403–416. Springer Berlin Heidelberg, 2013.

[44] H. Hamann, M. Szymanski, and Heinz Worn. Orientation in a trail network by exploiting its geometry for swarm robotics. In *Swarm Intelligence Symposium, 2007. SIS 2007. IEEE*, pages 310–315, April 2007.

[45] D. J. Hoare, J. Krause, N. Peuhkuri, and J.-G. J. Godin. Body size and shoaling in fish. *Journal of Fish Biology*, 57(6):1351–1366, 2000.

[46] Andrew Howard, Maja J Matarić, and Gaurav S Sukhatme. Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem. In *Distributed Autonomous Robotic Systems 5*, pages 299–308. Springer, 2002.

[47] Nicolas E. Humphries, Henri Weimerskirch, Nuno Queiroz, Emily J. Southall, and David W. Sims. Foraging success of biological Lévy flights recorded in situ. *Proceedings of the National Academy of Sciences*, 109(19):7169–7174, 2012.

[48] Duncan E. Jackson, Steven J. Martin, Francis L. W. Ratnieks, and Mike Holcombe. Spatial and temporal variation in pheromone composition of ant foraging trails. *Behavioral Ecology*, 18(2):444–450, 2007.

[49] John G. Kemeny and J. Laurie Snell. *Finite Markov Chains*. Springer-Verlag New York, 1976.

[50] P. L. Krapivsky and S. Redner. Dynamics of majority rule in two-state interacting spin systems. *Phys. Rev. Lett.*, 90:238701, Jun 2003.

[51] Michael J. B. Krieger, Jean-Bernard Billeter, and Laurent Keller. Ant-like task allocation and recruitment in cooperative robots. *Nature*, 406:992–995, Aug 2000.

[52] C Ronald Kube and Hong Zhang. Collective robotics: From social insects to robots. *Adaptive behavior*, 2(2):189–218, 1993.

[53] Daisuke Kurabayashi et al. Realization of an artificial pheromone system in random data carriers using rfid tags for autonomous navigation. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 2288–2293. IEEE, 2009.

[54] Renaud Lambiotte, Jari Saramäki, and Vincent D. Blondel. Dynamics of latent voters. *Phys. Rev. E*, 79:046107, Apr 2009.

[55] Kristina Lerman and Aram Galstyan. Mathematical model of foraging in a group of robots: Effect of interference. *Autonomous Robots*, 13(2):127–141, 2002.

[56] E. Mallon, S. Pratt, and N. Franks. Individual and collective decision-making during nest site selection by the ant leptothorax albipennis. *Behavioral Ecology and Sociobiology*, 50(4):352–359, 2001.

[57] Marco Mamei and Franco Zambonelli. Physical deployment of digital pheromones through rfid technology. In *Swarm Intelligence Symposium, 2005. SIS 2005. Proceedings 2005 IEEE*, pages 281–288. IEEE, 2005.

[58] James A.R Marshall, Rafal Bogacz, Anna Dornhaus, Robert P̃lanqué, Tim Kovacs, and Nigel R Franks. On optimal decision-making in brains and social insect colonies. *Journal of The Royal Society Interface*, 2009.

[59] Alcherio Martinoli. *Swarm intelligence in autonomous collective robotics: From tools to the analysis and synthesis of distributed control strategies*. PhD thesis, Citeseer, 1999.

[60] Alcherio Martinoli, Kjerstin Easton, and William Agassounon. Modeling swarm robotic systems: a case study in collaborative distributed manipulation. *The International Journal of Robotics Research*, 23(4-5):415–436, 2004.

[61] Ralf Mayet, Jonathan Roberz, Thomas Schmickl, and Karl Crailsheim. Antbots: A feasible visual emulation of pheromone trails for swarm robots. In Marco Dorigo, Mauro Birattari, GianniA. Di Caro, Ren Doursat, AndriesP. Engelbrecht, Dario Floreano, LucaMaria Gambardella, Roderich Gro, Erol ahin, Hiroki Sayama, and Thomas Sttzle, editors, *Swarm Intelligence*, volume 6234 of *Lecture Notes in Computer Science*, pages 84–94. Springer Berlin Heidelberg, 2010.

[62] James McLurkin, Jennifer Smith, James Frankel, David Sotkowitz, David Blau, and Brian Schmidt. Speaking swarmish: Human-Robot interface design for large swarms of autonomous mobile robots. March 2006.

[63] Francesco Mondada, Michael Bonani, Xavier Raemy, James Pugh, Christopher Cianci, Adam Klaptocz, Stephane Magnenat, Jean-Christophe Zufferey, Dario Floreano, and Alcherio Martinoli. The e-puck, a robot designed for education in engineering. In *Proceedings of the 9th conference on autonomous robot systems and competitions*, volume 1, pages 59–65. IPCB: Instituto Politécnico de Castelo Branco, 2009.

[64] Marco Montes de Oca, Eliseo Ferrante, Alexander Scheidler, Carlo Pinciroli, Mauro Birattari, and Marco Dorigo. Majority-rule opinion dynamics with differential latency: a mechanism for self-organized collective decision-making. *Swarm Intelligence*, 5:305–327, 2011.

[65] Iñaki Navarro and Fernando Matía. An introduction to swarm robotics. *ISRN Robotics*, 2013, 2012.

[66] Shervin Nouyan, Alexandre Campo, and Marco Dorigo. Path formation in a robot swarm – self-organized strategies to find your way home, 2008.

[67] Shervin Nouyan, Roderich Groß, Michael Bonani, Francesco Mondada, and Marco Dorigo. Teamwork in self-organized robot colonies. *Evolutionary Computation, IEEE Transactions on*, 13(4):695–711, 2009.

[68] KeithJ. OHara and TuckerR. Balch. Pervasive sensor-less networks for cooperative multi-robot tasks. In Rachid Alami, Raja Chatila,

and Hajime Asama, editors, *Distributed Autonomous Robotic Systems 6*, pages 305–314. Springer Japan, 2007.

[69] Chris A C Parker and Hong Zhang. Cooperative decision-making in decentralized multiple-robot systems: The best-of-n problem. *IEEE/ASME Transactions on Mechatronics*, 14(2):240–251, 2009.

[70] Chris A C Parker and Hong Zhang. Biologically inspired collective comparisons by robotic swarms. *The International Journal of Robotics Research*, 30(5):524–535, 2011.

[71] Julia K. Parrish, Steven V. Viscido, and Daniel Grnbaum. Self-organized fish schools: An examination of emergent properties. *The Biological Bulletin*, 202(3):296–305, 2002.

[72] Kevin M. Passino and Thomas D. Seeley. Modeling and analysis of nest-site selection by honeybee swarms: the speed and accuracy trade-off. *Behavioral Ecology and Sociobiology*, 59(3):427–442, 2006.

[73] David Payton, Mike Daily, Regina Estowski, Mike Howard, and Craig Lee. Pheromone robotics. *Auton. Robots*, 11(3):319–324, November 2001.

[74] Carlo Pinciroli, Adam Lee-Brown, and Giovanni Beltrame. Buzz: An extensible programming language for self-organizing heterogeneous robot swarms. *IEEE Transactions on Robotics*, 2015. Submitted.

[75] Carlo Pinciroli, Vito Trianni, Rehan O'Grady, Giovanni Pini, Arne Brutschy, Manuele Brambilla, Nithin Mathews, Eliseo Ferrante, Gianni Caro, Frederick Ducatelle, Mauro Birattari, Luca Maria Gambardella, and Marco Dorigo. ARGoS: a modular, parallel, multi-engine simulator for multi-robot systems. *Swarm Intelligence*, 6(4):271–295, 2012.

[76] Giovanni Pini, Arne Brutschy, Mauro Birattari, and Marco Dorigo. Interference reduction through task partitioning in a robotic swarm. In *Sixth International Conference on Informatics in Control, Automation and Robotics–ICINCO*, pages 52–59, 2009.

[77] Giovanni Pini, Arne Brutschy, Marco Frison, Andrea Roli, Marco Dorigo, and Mauro Birattari. Task partitioning in swarms of robots: An adaptive method for strategy selection. *Swarm Intelligence*, 5(3-4):283–304, 2011.

[78] TonyJ. Pitcher. Functions of shoaling behaviour in teleosts. In TonyJ. Pitcher, editor, *The Behaviour of Teleost Fishes*, pages 294–337. Springer US, 1986.

[79] Gaëtan Podevijn, Rehan OGrady, and Marco Dorigo. Self-organised feedback in human swarm interaction. In *Proceedings of the workshop on robot feedback in human-robot interaction: how to make a robot readable for a human interaction partner (Ro-Man 2012)*, 2012.

[80] Stephen C. Pratt. Quorum sensing by encounter rates in the ant temnothorax albipennis. *Behavioral Ecology*, 16(2):488–496, 2005.

[81] Andreagiovanni Reina, Marco Dorigo, and Vito Trianni. Towards a cognitive design pattern for collective decision-making. In Marco Dorigo, Mauro Birattari, Simon Garnier, Heiko Hamann, Marco Montes de Oca, Christine Solnon, and Thomas Stützle, editors, *Swarm Intelligence*, volume 8667 of *LNCS*, pages 194–205. Springer, 2014.

[82] Andreagiovanni Reina, Roman Miletitch, Marco Dorigo, and Vito Trianni. A quantitative micro-macro link for collective decision: the shortest path discovery/selection example. *Swarm Intelligence*, 2015. in press.

[83] Michael Rubenstein, Christian Ahler, Nick Hoff, Adrian Cabrera, and Radhika Nagpal. Kilobot: A low cost robot with scalable operations designed for collective behaviors. *Robotics and Autonomous Systems*, 62(7):966–975, 2014.

[84] Michael Rubenstein, Adrian Cabrera, Justin Werfel, Golnaz Habibi, James McLurkin, and Radhika Nagpal. Collective transport of complex objects by simple robots: Theory and experiments. In *Proceedings of the 2013 International Conference on Autonomous Agents and Multi-agent Systems*, AAMAS '13, pages 47–54. IFAAMAS, 2013.

[85] R.Andrew Russell. Ant trails - an example for robots to follow? In *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, volume 4, pages 2698–2703 vol.4, 1999.

[86] Erol Şahin. Swarm robotics: From sources of inspiration to domains of application. In Erol Şahin and William Spears, editors, *Swarm Robotics*, volume 3342 of *Lecture Notes in Computer Science*, pages 10–20. Springer, 2005.

[87] Alexander Scheidler. Dynamics of majority rule with differential latencies. *Phys. Rev. E*, 83:031116, 2011.

[88] T. Schmickl and K. Crailsheim. Trophallaxis within a robotic swarm: bio-inspired communication among robots in a swarm. *Autonomous Robots*, 25(1-2):171–188, 2008.

[89] Thomas D. Seeley. *Honeybee Democracy*. Princeton University Press, 2010.

[90] Thomas D. Seeley and P. Kirk Visscher. Group decision making in nest-site selection by honey bees. *Apidologie*, 35(2):101–116, 2004.

[91] Brian Shucker and John K Bennett. Scalable control of distributed robotic macrosensors. In *Distributed Autonomous Robotic Systems 6*, pages 379–388. Springer, 2007.

[92] Onur Soysal and Erol Şahin. A macroscopic model for self-organized aggregation in swarm robotic systems. In *Swarm robotics*, pages 27–42. Springer, 2007.

[93] William M Spears, Diana F Spears, Jerry C Hamann, and Rodney Heil. Distributed, physics-based control of swarms of vehicles. *Autonomous Robots*, 17(2-3):137–162, 2004.

[94] Timothy Stirling and Dario Floreano. Energy efficient swarm deployment for search in unknown environments. In *Swarm Intelligence*, pages 562–563. Springer, 2010.

[95] A. Stranieri, A.E. Turgut, M. Salvaro, L. Garattoni, G. Francesca, A. Reina, M. Dorigo, and M. Birattari. Iridia's arena tracking system. Technical Report TR/IRIDIA/2013-013, IRIDIA, Université Libre de Bruxelles, Brussels, Belgium, January 2013.

[96] K. Sugawara, T. Kazama, and T. Watanabe. Foraging behavior of interacting robots with virtual pheromone. In *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 3, pages 3074–3079 vol.3, Sept 2004.

[97] David J.T Sumpter and Stephen C Pratt. Quorum responses and consensus decision making. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 364(1518):743–753, 2009.

[98] Valery Tereshko and Andreas Loengarov. Collective decision making in honey-bee foraging dynamics. *Computing and Information Systems*, 9(3):1, 2005.

[99] Vito Trianni, Christos Ampatzis, Anders Lyhne Christensen, Elio Tuci, Marco Dorigo, and Stefano Nolfi. From solitary to collective behaviours: Decision making and cooperation. 4648:575–584, 2007.

[100] Vito Trianni and Marco Dorigo. Emergent collective decisions in a swarm of robots. In *IEEE Swarm Intelligence Symposium, SIS 2005*, pages 241–248, June 2005.

[101] G. Valentini, E. Ferrante, H. Hamann, and M. Dorigo. Collective decision with 100 kilobots speed vs accuracy in binary discrimination problems. Technical Report TR/IRIDIA/2015-005, IRIDIA, Université Libre de Bruxelles, Brussels, Belgium, July 2015.

[102] G. Valentini, H. Hamann, and M. Dorigo. Efficient decision-making in a self-organizing robot swarm: On the speed versus accuracy trade-off. Technical Report TR/IRIDIA/2014-013, IRIDIA, Université Libre de Bruxelles, Brussels, Belgium, September 2014.

[103] Gabriele Valentini. Self-organized collective decision-making in swarms of autonomous robots. In Alessio Lomuscio, Paul Scerri, Ana Bazzan, and Michael Huhns, editors, *Proceedings of the 13th Int. Conf. on Autonomous Agents and Multiagent Systems*, AAMAS '14, pages 1703–1704. International Foundation for Autonomous Agents and Multiagent Systems, 2014.

[104] Gabriele Valentini, Mauro Birattari, and Marco Dorigo. Majority rule with differential latency: An absorbing Markov chain to model consensus. In *Proceedings of the European Conference on Complex Systems 2012*, Springer Proceedings in Complexity, pages 651–658. Springer, 2013.

[105] Gabriele Valentini, Heiko Hamann, and Marco Dorigo. Efficient decision-making in a self-organizing robot swarm: On the speed versus accuracy trade-off. In Rafael Bordini, Edith Elkind, Gerhard Weiss, and Pinar Yolum, editors, *Proceedings of the 14th Int. Conf. on Autonomous Agents and Multiagent Systems*, AAMAS '15, pages 1305–1314. IFAAMAS, 2015.

[106] Gabriele Valentini, Heiko Hamann, and Marco Dorigo. Self-organized collective decision-making in a 100-robot swarm. In *Proceedings of*

the Twenty-Ninth AAAI Conference on Artificial Intelligence, pages 4216–4217. AAAI Press, 2015.

[107] Gabriele Valentini, Heiko Hamann, and Marco Dorigo. Self-organized collective decisions in a robot swarm. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence, AI Video Competition*. AAAI Press, 2015. http://youtu.be/5lz_HnOLBW4.

[108] José D. Villa. Swarming behavior of honey bees (hymenoptera: Apidae) in southeastern louisiana. *Annals of the Entomological Society of America*, 97(1):111–116, 2004.

[109] Justin Werfel, Kirstin Petersen, and Radhika Nagpal. Designing collective behavior in a termite-inspired robot construction team. *Science*, 343(6172):754–758, 2014.

[110] Jan Wessnitzer and Chris Melhuish. Collective decision-making and behaviour transitions in distributed ad hoc wireless networks of mobile robots: Target-hunting. In *Advances in Artificial Life*, pages 893–902. Springer, 2003.

[111] Alan FT Winfield, Christopher J Harper, and Julien Nembrini. Towards dependable swarms and a new discipline of swarm engineering. In *Swarm robotics*, pages 126–142. Springer, 2005.

[112] Seung-kook Yun, Mac Schwager, and Daniela Rus. Coordinating construction of truss structures using distributed equal-mass partitioning. In *Robotics Research*, pages 607–623. Springer, 2011.