Value-sensitive decisions through pheromone trails

BY

Anna Font Llenas

Supervisor: Xu Xu

Co-supervisor: Andreagiovanni Reina

MSc. Automation, Control and Robotics

Full-time student 2017-2018

This report describes project work carried out in the Faculty of Arts, Computing, Engineering and Sciences at Sheffield Hallam University between July and March 2017/18.

The submission of the report is in accordance with the requirements for the award of the degree of MSc. Automation, Control and Robotics under the auspices of the University.

Acknowledgments

First of all, I would like to express my deep thanks to the Dr Xu Xu from the Sheffield Hallam University for orienting me and giving me support since the first day of my stay abroad. I feel absolutely grateful of her help as a teacher, course leader, supervisor and also, the most important thing, as a reliable, positive and solidary educator. I am profoundly indebted to Dr Xu Xu who made possible this incredible experience and that transferred me enough confidence to achieve my goals during this year. In addition, I would like to remark the help received to perform this dissertation, providing me many topics and the chance to work in collaboration with the University of Sheffield, recommending me to the Robotics department. I know that I can always count on her.

I want to remark the positive impact received from Dr Lyuba Alboul in the subject of Robotics this semester, helping me to increase my knowledge in the swarm behaviour field and bringing me the opportunity to work with similar robots as the Kilobots, the E-pucks. In addition, she provided interesting applicable ideas about the robots behaviour, inspiring me in some points of the thesis.

On the other hand, I would like to thank the Behavioural and Evolutionary Theory Lab team of the Robotics Department of the University of Sheffield. I feel so comfortable and honoured to have the opportunity to work in this powerful and interesting team and also, to be able to learn from these great Scientifics.

Specially, from this team, I would like to express my deepest appreciation to Dr Andreagiovanni Reina, my supervisor from the University of Sheffield, who

kept me motivated and excited all the time bringing me the energy to continue and never give up in the hard situations. He enabled me to enjoy the whole project and provided me his wide knowledge in this topic. What is more, he kept giving innovative and creative ideas while being attentive and patient with my suggestions and problems. I am really grateful because he leads this project to succeed and also formed my knowledge in this field while he makes me spend an amazing time that I would not change.

In addition, I would like to thank my colleague Salah Talamali who combines his capability of being an amiable person with the potential of being always approachable. He really helped me solve many problems related with the software and hardware in a fast way and also, gave me a large amount of tips in the robotics field.

This work will not be materialized without the encouragement of Dr James Marshall who was guiding and supervising the project from the start and also, giving innovative ideas.

Abstract

The imitation of a pheromone trail of an Ants Colony Model is implemented nowadays in a number of researches in order to explore more about the capabilities of swarm robotics. This project is presenting this behaviour using inexpensive and simple robots, the Kilobots. The main scope of this project is the challenge in implementing ants foraging using a limited number of virtual sensors in the Kilobots. Due to its simplicity, some solutions are implemented to sort the limitations of this technology. The environment is analysed in order to achieve the relation with a set of parameters such as the distance, quality and pheromone coefficients. It is implemented due to ARK technology, augmented reality in Kilobots (Reina, Cope, Nikolaidis, Marshall, & Sabo, 2017). It is a powerful tool designed especially for the Kilobots' communication.

Some experiments are performed in the ARGoS simulator (Pinciroli, et al., 2011) and in the laboratory using the real robots. Each experiment is compound for 50 to 200 robots and within a changing environment from one to four food sources and one nest, involving different qualities and different distances.

The analyses performed in this thesis determine the behaviour of these swarm robots using the powerful ants' tool, the pheromone, a chemical component that is virtually generated. The virtual pheromone enables the communication in a micro-scale scenario, achieving macro-scale behaviours. Thus, the Kilobots without global knowledge can mimic ants foraging behaviour.

The work presented is composed of three different experiments. The first one studies how the food source distance to the nest affects in the collection of food; the second one is based on the study of the food qualities, giving priority to a high quality food over a low quality one. The last experiment analyses all the variable parameters in the experiment such as the evaporation, the diffusion, the amount of pheromone, the number of robots and the number of food sources, to achieve the best relation for each scenario and to find the relation of those parameters.

Contents

1. Introduction

Swarm robotics is defined as a branch of robotics that modifies the environment in an autonomous way, as explained in (Sahin, 2015). It has local communicative and sensitive capabilities, but does not have access to global and control knowledge. Swarm robots have the ability to cooperate with each other in order to achieve complex tasks, in an efficient manner which could not be accomplished when the decisions are taken individually.

The project focuses on Kilobots (Rubenstein, Ahler, & Nagpal, 2012); an inexpensive type of swarm robots, allowing for experiments to be conducted using a large number of them. They are equipped with simple and small sensors and actuators in order to perform different tasks, imitating the behaviours of ants.

Their goal is to make decisions based on the best resource of food. Once the decision is made, they will collect the food and transport it back home. The pheromone trails are deposited from the robots and allow the other robots to follow the track. The Kilobots will be acting as a swarm following swarm's behaviour as the ones explained in (Brambilla, Ferrante, Birattari, & Dorigo, 2013), finding the most efficient way to carry different resources to home.

The project is designed using decisions based on quality targets, limited knowledge (instead of using a general knowledge) and short-path targets, in order to imitate as closely as possible the ants foraging behaviour. This will significantly improve upon the results of previous investigations, where the knowledge of the swarm robots is in excess, as in (Herianto, Sakakibara, & Kurabayashi, 2007) or where the decisions are based on only the proximity of the kilobots to the food source, disregarding the relative quality of the targets, as in (Garnier, Combe, Jost, & Theraulaz, 2013).

The overall goal of this project is to perform a food collection task using a swarm of kilobots discovering the best quality resources and investigating the best pheromone parameters to achieve an efficient foraging. Virtual sensors will be used for this purpose to provide swarm robots the basic information that ants could perceive through the environment, such as resources located nearby and the location of the nest. Due to previous work in the Augmented Reality in Kilobots, ARK (Reina, Cope, Nikolaidis, Marshall, & Sabo, 2017), the project presented will be able to be simulated in ARGoS as well as being executed in the laboratory using ARK.

This project will create an impact in the swarm robotics fields, particularly, in Ants Colony Models, as the ones explained in (Dorigo, Birattari, & Stutzle, 2006), due to the fact that it will prove how such a simple robot swarm can perform a complex task with limited knowledge, virtual sensors and swarm behaviour. Furthermore, it will analyse how the pheromone parameters could affect ants' foraging strongly. By modifying the environment and pheromone parameters in every experiment, the best conditions to collect the food for an Ants Colony will be found for each particular situation. The search of the suitable number of robots working as worker ants will also be included. The parameters analysed in this project include the evaporation, the diffusion and the quantity of the pheromone, among others.

It is true that other research work such as (Garnier, Tâcheb, Combe, Grimal, & Theraulaz, 2007) studied the correlation of some parameters but only with a maximum of ten robots. However, some interesting results have been presented in recent years. For instance, it is proven that there exists optimal parameters of foraging and that there is a dependency of the robot behaviours with the environment. With these assumptions, the work done in this thesis gain further understanding in this research direction.

This project will have a consistent structure, debating initially the different views of the researchers in the Critical literature survey section and the current useful approaches in the Relevant Theory and Analysis section. To continue, the process followed to perform this project will be explained in the Methodology section. Finally, the evaluation of the results, the outcomes, including the achievements and deficiencies, and a final conclusion will be presented, respectively, in the Results, in the Critique and in the Conclusions and further work sections.

2. Critical literature survey

2.1. Biological swarm behaviour

Biology studies such as (Ross & Matthews, 1991) and (Deutsch, Brusch, Byrne, Vries, & Herzel, 2007) reveal that there is not a real centralized coordination method in the social animals' and mainly, in the social insects' world. It has been analysed that there is a synchronised behaviour known as swarm. As investigated in (Sahin, 2015), there are three key motivations that make the swarm robotics to be developed, trying to imitate this social behaviour. One of the key motivations is the robustness, the ability to continue operating even in individual failures or during environment disturbances. The flexibility is another key motivation due to the requirement of generating satisfactory experiments in a wide range of different tasks. The last one is the scalability; in a swarm robotics environment, scalability is described as the capability to work with different sizes and configurations and measure them in terms of productivity and performance, in a particular collective system, as explained in (Kernbach, Handbook of Collective Robotics: Fundamentals and Challenges, 2013).

As defined in (Brambilla, Ferrante, Birattari, & Dorigo, 2013), the main characteristics of swarm robots are the autonomy of the robots, the constant modification of the environment due to the robotics knowledge, the local sensing and communication capabilities, the non-requirement of a global knowledge environment, the non-access to a centralized control and the interaction and cooperation between the robots to achieve a task together.

Main studies related to swarm robotics, (Kube & Bonabeau, 2000), (Chan & Kumartiwari, 2007) and (Deneubourg, et al., 1991) are using, as a reference control, the ants' behaviour in order to achieve cooperative tasks, such as taking some items from places in a coordinated way, following the traces that other robots have left, guiding the robots to a common goal, among others.

2.2. Pheromone studies

A huge amount of studies lay down that the ants' pheromone phenomenon is essential to be stablished in order to optimize the swarm behaviour of the robots. In some scientific analysis, (Purnamadjaja & Russell, 2007) and (Kazama, Sugawara, & Watanabe, 2006), is used the pheromone to guide the robots from a random walk around the environment to achieve its goal due to the following of the pheromone trail. This pheromone is previously deposited for robots that have acknowledged more information due to its previous exploration through the environment.

The discussion of which is the best substance to simulate ants' pheromone is being very controversial nowadays. Some experts agree that the use of physic elements to simulate the pheromone is the best option to achieve foraging behaviours, while others argue that virtual sensors are the best solution to perform more experiments and in an efficient way.

Chemical experiments involving cooperative ants' behaviour have been developed successfully. (Fujisawa, Dobata, Sugawara, & Matsuno, 2014) and (Wilson, 1965) use a chemical component with evaporation, diffusion, locality and reactivity characteristics in order to imitate the natural substance. It has been proof that the behaviour can be created as a completely autonomous system without external communication controls. However, this technology is being difficult to implement and it is heavily criticized by (Mayet, Roberz, Schmickl, & Crailsheim, 2010) for not being reliable enough.

On the other hand, some virtual systems in the pheromone printing are giving interesting results in the field, such as (Arvin, Yue, & Xiong, 2015). This one is opting to use a LCD screen located in the ground to create a light pheromone trace. In addition, some experiments have been performed using phosphorescent light as a floor, (Mayet, Roberz, Schmickl, & Crailsheim, 2010), proving its reliability. However, these technologies have high costs due to the purchasing of the LCD board. Other researches, as (Herianto, Sakakibara, & Kurabayashi, 2007), use RFID technology. This technology enables a low cost system developing artificial potential fields in a data carrier system. Nonetheless, the RFID technology is not quite acceptable for some Scientifics in the area due to the fact that the robot is not really autonomous and the pheromone lay is neither done and nor decided, by the robots. (Garnier, Tâcheb, Combe, Grimal, & Theraulaz, 2007) argues that the printing decision is made by an intelligent controller that have a global knowledge of the environment instead of giving intelligence to the swarm robots, as the ants work.

2.3. Robots and platforms

Different kinds of robots have been used to evaluate the ants' behaviour. For these specific tasks, it tends to use simple and small robots. There is a huge amount of emerging robots of this type, as the Colias and the extended version Colias-III, experimenting in the bio-inspired vision systems (Arvin, Yue, & Xiong, 2015). Another well-known robot is the E-puck, characterized for its simplicity, a user replaceable battery and a distance, camera, bearing, accelerometer and mic sensors (Bonani, et al., 2009). It is also used the Kilobots, remarkable for its long autonomy, up to 24h, and its facility to be programed in groups instead of programming them one by one (Rubenstein, Ahler, & Nagpal, 2012). The bots including foot-bots, eye-bots and hand-bots are also recently used in this field. It is true that its autonomy is lower than the other robots, although, they are capable to perform more tasks and in a more reliable way due to the large amount of sensors (Dorigo, et al., 2013). Alice is equipped with a camera and a distance sensor and has a size of only 2.2 cm. It is used in many researches nowadays such as navigation and map building (Caprari, Balmer, Piguet, & Siegwart, 1998). Another alternative is Jasmine, focused mainly in honeybee behaviours due to its suitable shape (Kernbach, Thenius, Kernbach, & Schmickl, 2009).

The affordability, scalability, flexibility and the non-requirement of huge spaces for small platforms and robots is creating a sharply increase in the swarm robot researches. Some experiments can be performed using different kind of technologies, for instance E-pucks and Kilobots. As discovered in (Nouyan, Campo, & Dorigo, 2007), satisfactory path formations to the goal can be implemented using E-pucks and Kilobots, sharing the same strategy. This enables to choose a robot depending on the information that it is needed to extract in each experiment.

There are different technologies to implement satisfactory swarm robotics simulations. The open source multi-robot simulators are the most used for the researchers in this field due to they are easy to install, flexible and without any cost. The well-known ARGoS is the most desired in the Scientifics world (Pinciroli, et al., 2011) outstanding for its scalability. ARGoS includes an extensible architecture, adding the option to modify functionalities to adapt any kind of swarm robot. Furthermore, it enables simultaneously multiple physics engines, enabling a transparent migration of the robots between engines. This software perfectly accomplishes the necessities of this thesis. Thus, ARGoS is the platform used. There are other alternatives less focused on this swarm behaviour such as ARTOO (Ciupa, Leitner, Oriol, & Meyer, 2008) that supports the idea of an Adaptive Random Testing and ROS that "*provides a structured communication layer above the host operating systems of a heterogeneous compute cluster*" (Quigley, et al., 2009). Also, other platforms such as FORMICA (English, et al., 2008) are just focused on ants' behaviour creating an easy and affordable program that can perform different tasks related to the real behaviour, such as take food from a place, leave pheromone, follow pheromone, among others.

Another challenge that a foraging system should overcome is the way to analyse the information of the current simulations due to the fact that in swarm robotics the experiments are performed using a high swarm size, different environment characteristic and also, a large amount of different coefficients of the pheromone. The information collected through the experiments should be merged and evaluated using big data analysis. As explained in (Chen & Zhang, 2014), there are a lot of techniques to analyse high amounts of data. There are some programs capable to extract the information from files and sorted in a manner to allow the analysis. There are some open-sources such as Ploty (Sievert, Parmer, Hocking, Chamberlain, &

Ram) that, integrating different programming languages, can create a complete analysis for many applications. Some other well-known platforms are Matlab (Sharma & Martin, 2009) and Python (Rossum & Drake, 1995). These two techniques are so developed and with a helpful service support. However, they are quite complex and due to this complexity, the computation increases developing to a low performance speed in some cases.

The R platform (Bunn & Korpela, 2014) is the technology chosen due to the fact that is mainly focused in the data processing and analysis. It is a simple tool which includes a wide range of useful functions. In addition, it is an open-source that is supported in all the operating systems.

2.4. Navigation control

In terms of the navigation behaviour, it has been discussed the different ways to follow the pheromone path, in an efficient manner to not follow the wrong decision, without wasting resources and achieving, in a reliable way, the goal. Some of the navigation control methods are based on the potential field approach to complete their tasks, while others do not implement the obstacle avoidance behaviour. For instance, in (Kim, Wang, & Shin, 2006), an experiment based on potential function approaches is evaluated. This method enables the robots to get aligned and to create a path as a swarm. In addition, it is creating attraction to the goals and repulsion to the obstacles around, achieving the aim of the experiment without crashing. However, these projects are not mainly implemented for this task due to the non-similarity with the real behaviour, the use of holonomic sensors for the robots and the lack of pheromone coefficients as evaporation and diffusion.

On the other hand, some studies (Jackson, Holcombe, & Ratnieks, 2004) and (Garnier, Combe, Jost, & Theraulaz, 2013) reveal that there is no need for the robots to create a program to learn, for instance, the presence of a bifurcation. The study (Garnier, Combe, Jost, & Theraulaz, 2013) uses a robotic model just programed to perform a correlated symmetrical random walk in an environment with few obstacles, other robots and walls. This work implements the Ant Colony Optimization algorithm to provide an effective choice selection, discerning between paths to choose the shortest one but without checking the food source qualities.

In order to create an intelligence system to reproduce the ants' behaviour some questions should be made. How ants can pick out one way or another? Are they capable to choose the better way? In some studies, as (Goss, Aron, Deneubourg, & Pasteels, 2009) and (Garnier, Guérécheau, Combe, Fourcassié, & Theraulaz, 2009), this issue has been studied. The Argentine ants were chosen to perform these experiments. It was found that the ants with a limited orientation can get through the shortest path due to the pheromone deposition of other ants. Other studies, such as (Wendt & Czaczkes, 2017), analyse the ants' self-control. The self-control is described as the capacity to opt for the large delayed reward instead of choosing the first one found. In the ants' field, it is referred to the capacity to discern between a food with high quality but far from the nest with one closer but with low quality, avoiding the consumption of low-quality sources rewards. However, the ants do not reject a slightly poor food; it will be after the analysis of the different pheromone trails when the ants will start discerning between paths and maximizing the collection in the richest food source, over time, the best collection will be achieved.

Another interesting question about the ants' behaviour is how the number of food sources affects to its conduct. It is tend to think that the reduction of errors or noise perception in the ants or robots through the environment will reduce the efficiency in the food collection; Nonetheless, (Deneubourg, Pasteels, & Varhaeghe, 1982) demonstrates that an optimal level of noise is advantageous for the ants in order to exploit a wide amount of the environment. Due to this fact, the ants will be able to discover more food sources and choose that food source with a better quality and distance trade-off. Nevertheless, this study was extremely criticised by (Nakamura & Kurumatani, 1997) due to the fact that the model was not representing the pheromone dynamics and was just formulating simple differential equations. The article (Nakamura & Kurumatani, 1997) presents an alternative mathematical model without a centralized control but assuming that the ants can sense the pheromone. It was discovered that the macro-scale of this behaviour appears due to the micro-scale behaviour of each individual ant.

Another work, comparing the micro-scale and macro-scale ants' behaviour, is (Reina, Miletitch, Dorigo, & Trianni, 2015) which performed some experiments analysing the honeybees' behaviour. A formalised pattern design has been created in order to take collective decisions to increase the understanding of the effects of spatially on the decision dynamics, comparing the microscopic with the macroscopic link.

The navigational control of the robots is playing a role in order to develop a similar ants' behaviour. There is a trade-off between the maximum allowance information that the robots can perceive with the effectivity to achieve its goal. There are different techniques that experts developed that should be investigated. In (Payton, Daily, Hoff, Howard, & Lee, 2001), a robustness control navigation system was created without providing to the swarm robots an explicit maps or models of the environment and also, without an explicit knowledge of the robot location. The improvement in the swarm field is remarkable due to the discovery that an intermediate representation between the computations and the real world is not needed.

Another particular work which explores the ants' behaviour is the (Sakiyama & Gunji, 2016). This experiment proposes self-organized patterns in foraging using hybrid navigation with momentary decisions; however, this hybrid navigation can produce uncertainty in the ants' foraging as explained in (Knaden & Wehner, 2005). On the contrary, it is proven that the robots, due to this model, can achieve decisions based on the exploitation and exploration, estimating the local pheromone gradients. The gradients are also implemented in some algorithms of swarm intelligence in order to behave as social organisms, as defined in the population-based algorithm, PSO (Marinakis & Marinaki, 2009). In (Sakiyama & Gunji, 2016), it is demonstrated how the decisions are made by following the Weber's law (Ekman, 1959). This decision is creating a linear reaction at micro-levels when pheromone is found, as explained in (von Thienen, Metzler, Choe, & Witte, 2014).

The most similar work to the one presented, it is the (Garnier, Tâcheb, Combe, Grimal, & Theraulaz, 2007) article, focused on the swarm behaviours. In this case, Alice robots are used to create a range of experiments to observe the behaviour of the robots going from the nest to home and vice versa. In this research, the simulation of the pheromone is done by light, creating light trails in the floor. This work is inspirational due to the fact that some interesting results and question have been disputed. First, it is proven that only a collective choice can take place if and only if the evaporation is not fast enough. What is more, an optimal number of robots should be chosen carefully for each environment agreeing with the previous work of (Krieger, Billeter, & Keller, 2000) that experiment with Khepera robots in order to create an ants' fiction colony for foraging and collect food.

2.5. ARK and ARGOS studies

The work presented is based on the ARK system, Augmented Reality for Kilobots, (Reina, Cope, Nikolaidis, Marshall, & Sabo, 2017). It is an alternative inexpensive and robust system, in comparison with the technologies analysed above, such as (Arvin, Yue, & Xiong, 2015), (Herianto, Sakakibara, & Kurabayashi, 2007) and (Garnier, Tâcheb, Combe, Grimal, & Theraulaz, 2007). This system enables the performance of a wide range of tasks in a swarm environment where the robots take their own decisions. These decisions are made taking into account the virtual sensors received from the ARK system.

What is more, this technology is capable to use multiple threads to save time during the execution, to perform the ID settings and to compute the synchronisation of the robots. This new technology has the capability to create any communication with the robots independently of the quantity of it and under an unknown environment due to the location of four cameras in the top.

The figure bellow extracted from (Reina, Cope, Nikolaidis, Marshall, & Sabo, 2017) describes the system architecture:

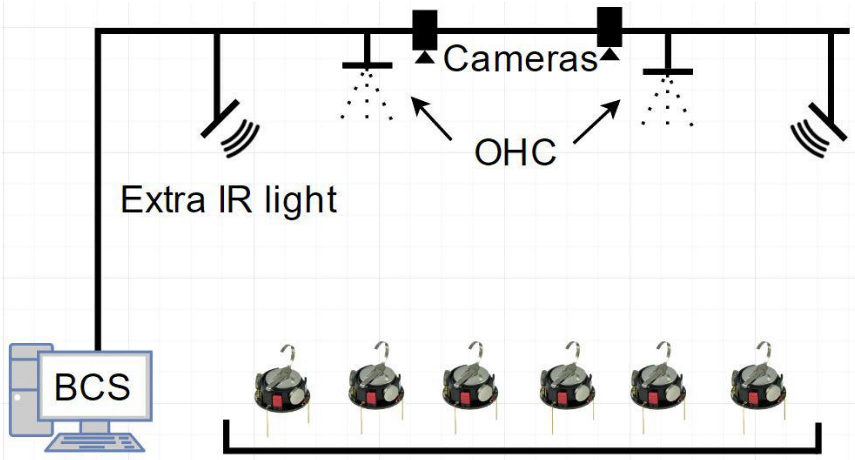


Figure 1. ARK arena hardware architecture (Source: (Reina, Cope, Nikolaidis, Marshall, & Sabo, 2017))

As shown, there is a master computer that communicates with the cameras using a binary synchronous communication (BCS). The BCS system is the one in charge to compute all the information received from the cameras, process this information and send the right message to each Kilobot via the Overhead controller (OHC), including the characteristics of the current environment. The Kilobots receive this information due to the infrared (IR) communication and send information through the LED located in the top.

The ARK system is implemented over ARGoS technology, as explained in Robots and platforms section. The code source can be found in GitHub (GitHub, 2018). This website allows a fast download and a forum for possible bugs. The structure of the files is used as a template and should not be changed, enabling an easy understanding of the whole program. The (Pinciroli, argos3-kilobot, 2017) web page is the one used to download the template which contains all the basic files, libraries and plugins. In comparison with other systems, as (Sievert, Parmer, Hocking, Chamberlain, & Ram) and (Ciupa, Leitner, Oriol, & Meyer, 2008), ARGoS is focused on the capability to allow the user to add functionalities and improvements of the open-source. In addition, it includes code examples and experiments to understand better how to program the robots in an efficient way.

3. Relevant Theory and Analysis

There are some concepts that should be understood before starting to build the code. These concepts are related with the functionalities of the robots and with the ants' behaviours.

It is important to understand how the ants walk is performed. It is clear that this is not the main specification of the project but, in order to achieve a really similar ants foraging, the movement needs to be implemented for the swarm robots as similar as possible. The ants foraging is based on a random walk and maximizing the worker ants area explored. It is known that when the area explored is increased, the throughput is maximised, as explained in (Boogert, Fawcett, & Lefebvre, 2011).

In addition, it has to bear in mind that the relation between the food sources and the collection is not linear. As confirmed in (Heck & Ghosh, 2000), an ants foraging implying an unlimited food available can cause a non-efficient collection. This effect is called the trapping syndrome and it is important to avoid it.

There are some aspects that should be analysed. For instance, it should be considered the ants' decisions in the random walk, to continue straight or turn to one side. As explained in (Pearce-Duvet, Elemans, & Feener, 2011), this decision is different for each kind of ants. It is defined a turning rate in degrees per second, an speed in millimetres per second and a turning intensity in degrees per body length in order to quantify each ant type. This study revealed that the ants have speeds between 13 to 20 mm/sec, turning rates around 70 degrees per second and turning intensity up to 16 degrees per body length.

It has to bear in mind that the main scope of the behaviour of the robots is the similarity with the real world, concretely, in the ants foraging. That is why; some biological approaches have taken into account. The principal concept of the ants' collection action is the pheromone tools to enable the swarm behaviour due to the pheromone trail, achieving an efficient collection in an ants' colony. As defined in (Morgan, 2008), the pheromone is a chemical substance that incite worker ants to follow the trails using antennae contact, the pheromone smell and their move in a jerking way. There are multiple purposes of the pheromone but, as remarkable in (Witte, Attygalle, & Meinwald, 2007); the exploitative capability of the food collection is the most used in the ants' world in front of using the pheromone as a defensive or aggressive tool.

To continue, the pheromone properties should be studied carefully in order to reproduce a virtual substance with the same behave. There are two notable properties, the evaporation (the capacity to vanish) and the diffusion (the capacity to spread out) of the pheromone. These ones have been parametrized in a model that enables a close simulation to the real behaviour, using only two simple equations. The equations determine the amount of food collected, using an extensive diffusion and a constant evaporation. There are two regions, the attracting region P(x, y, z) and the active trail T(x, y).

These ones are extracted from (Nakamura & Kurumatani, 1997) and they are going to be applied in the performance of this project, as shown below:

$$\left[\frac{\partial}{\partial t} + \gamma_{eva}\right] T(x, y) = 0$$

Equation 1. Evaporation equation

$$\left[\frac{\partial}{\partial t} - \gamma_{dif}\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}\right)\right] P(x, y, z)$$

$$= 0 \qquad\qquad if \quad z > 0$$

$$= \gamma_{eva} T\left(x, y\right) \quad if \quad z = 0$$

Equation 2. Diffusion equation

The first equation creates a constant reduction of the pheromone over the time with a biology parameter of the evaporation γ_{eva} and the active Trail T that comprise the current environment in two dimensions.

The second equation takes the biology diffusion parameter of the pheromone and multiplies this coefficient by the differential equations. Then, this parameter is subtracted by the differential time and finally, the current environment in the three dimensions is modified.

The parameter z = 0 is referred to a boundary reflection. Using both equations, the robots will be capable to create an attractive area through its way.

Once the pheromone properties are well known, the manner of following the trace should be implemented. The concept of how to create an Ant Colony Optimization is being a trend nowadays. The best technique of this optimization stills uncertain due to the large amount of theories and the use of different scenarios analysed from the researchers.

In (Dorigo, Birattari, & Stutzle, Ant Colony Optimization, 2006), a wide overview of the topic and the evolution of the foraging behaviour theories are explained. It is paying attention in the importance of expanding the experiments options changing its dynamics and stochastics aspects. This paper encourages researchers to continue investigating in this field in order to find the most approximate model to solve hard optimization problems. That is why; a large of parameters and environment variations will be implemented in this project, enabling disputed results of the experiments in the Results section.

4. Methodology

4.1. Files structure in the simulation

Before starting the simulation, some concepts should be learnt and some investigations should be done. Comparing the technologies used in Critical literature survey and making clear the goals to achieve, as explain in Introduction, the whole project should be implemented.

The first step for the simulation is to set up the ARGoS platform. This platform is an open-source and can be used in any type of computer. It has been implemented over bash code. There is some basic knowledge that should be understood in order to structure in an efficient way the different files and in order to execute the software. There are three main types of files to implement a simulation. The first file is the behaviour which has the intelligence of the robots. This file is executed as threads for each of the robots, namely, each robot will be executing the same intelligence but it will not be behaving in the same way in each time step due to the fact that the environment around will be different. Thus, as following the ants behaviour, each robot will have their own knowledge and will take each own decision. In order to receive information from the environment, the virtual sensors should be implemented. These virtual sensors are created for a unique main file called loop functions. The loop functions file is a file with the ability of reading the information from the environment, computing the information and sending the environment characteristics around the concrete robot to each one. This information should be comprised in just ten bits and should include the information that the ants would know in each case. This knowledge will be so limited due to the fact that neither a not global knowledge nor a GPS navigation system is the aim of a swarm robotics to implement an ants' colony. For instance, the locations of the food sources, its quality and the positions and the behaviours of other robots should not be known for the Kilobots. This smart system that performs all these tasks is called ARK, Augmented Reality on Kilobots. The only variables able to be known for the Kilobots are the following ones:

- Angle to Home: This project assumes that the robots have the capacity to return at home just following a virtual sensor that indicates the degrees to Home. This angle is compute in 4 bits, thus the values received will be from 0 to 7. These values will be normalized in 45 degrees each one in order to fit a computation from 0 degrees to 360 degrees. For example, a value between -22.5 and 22.5 degrees will be sent as 0, as shown:

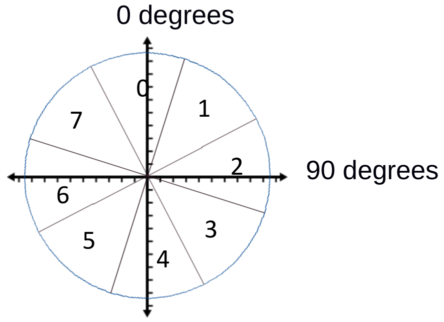


Figure 2. Angle interpretation

- At Home: This variable is a single bit to indicate to the robot if its location is within the boundaries of the nest. The real ants know this information.
- At Food: This variable is, also, a single bit. It indicates if the robot is located in a food source. When the variable is 1, food source detected, the food is collected and is brought to home printing pheromone.

- Quality: This variable has four bits. It is used to indicate the quality of the food source. This variable is just received, if and only if, the robot is located within the boundaries of a food source. Thus, the robot will be able to analyse the food, compute its richness and later print pheromone according to the value found in order to suggest to the other robots to follow or not the current trail to the food source.
- Pheromone Zones: The pheromone zones variable indicates the areas next to the robot where pheromone is found. This is the virtual sensor that represents the antennas behaviour of the ants following left to right the pheromone trail. It will be given this variable in any moment except when the quality is given. Thus, it is known if there is pheromone in the environment except when the robot already found a food source.

The loop function file will need to receive some information in order to understand the environment. This information will be the ID of the robot and its location, due to the simulated camera located in the top and also, the information of the resources in the environment. What is more, the robots will be changing its LED to communicate the current behaviour to the ARK system. This LED has three colours, one colour per behaviour:

- Red colour: It is indicating that the swarm robot is foraging, walking in a random move and searching for food.
- Green colour: The green LED is indicating that the robots are following the pheromone to the food source.
- Blue colour: Once the robots reach the food source, the LED will change from green to blue indicating that the food source is reached

and that they are carrying food. This colour will change again to green or red once the home position is reached.

The payload to send the information from ARK to the robot is shown above:

9	8	7	6	5	4	3	2	1	0
At Food[9]	At Home[8]	Pheromone Zones [7..4] Quality [7..4]				Angle To Home (4 LSB bits)			

Table 1. Payload

Other interesting files are involved in these experiments. The next one is the experiment ARGoS file, an HTML file which configures the environment. This one is the one that determines all the variables and positions in the environment and is executed to perform the experiment. In addition, the simulation cameras can be configured in this file to achieve a better visibility of the environment. Also, this file sets up the distribution of the robots and writes the random seed of the experiment. The random seed is useful to create a set of different random experiments. What is more, it is useful in order to reproduce two exactly experiments just using the same seed number. This file is also read from the loop functions file to extract the environment data and achieve a global knowledge of the experiment in order to classify and process the information to the robots. Through the command prompt the ARGoS file can be executed to perform the desired simulation. Once the ARGoS file is running, the loop function and the behaviour file are executed and the simulation takes place.

Also, a bash file is created to execute x times the experiments changing the ARGoS parameters in the file. It is useful to change the environment and perform hundreds of slightly different experiments. In addition, the bash file save the information in a set of txt files, jointly with the experiment file, to be able to analyse the data afterwards. The experiment can take quite a long time due to the fact that each exact experiment can be executed for a long period and changing the seed number to make the results reliable. In addition, the coefficients of the pheromone and the environment parameters can be combined creating a large size of experiments.

The last file created is the R file. The R program is used to evaluate the high amount the data obtained in the different files. The R file is sorting the data depending on the different variables and creating a massive file organized by tables with the different data averages. Furthermore, another R file is needed to read these tables and plot in an understandable way the results.

To summarize the information explained above, an explanation diagram has been performed in the next figure:

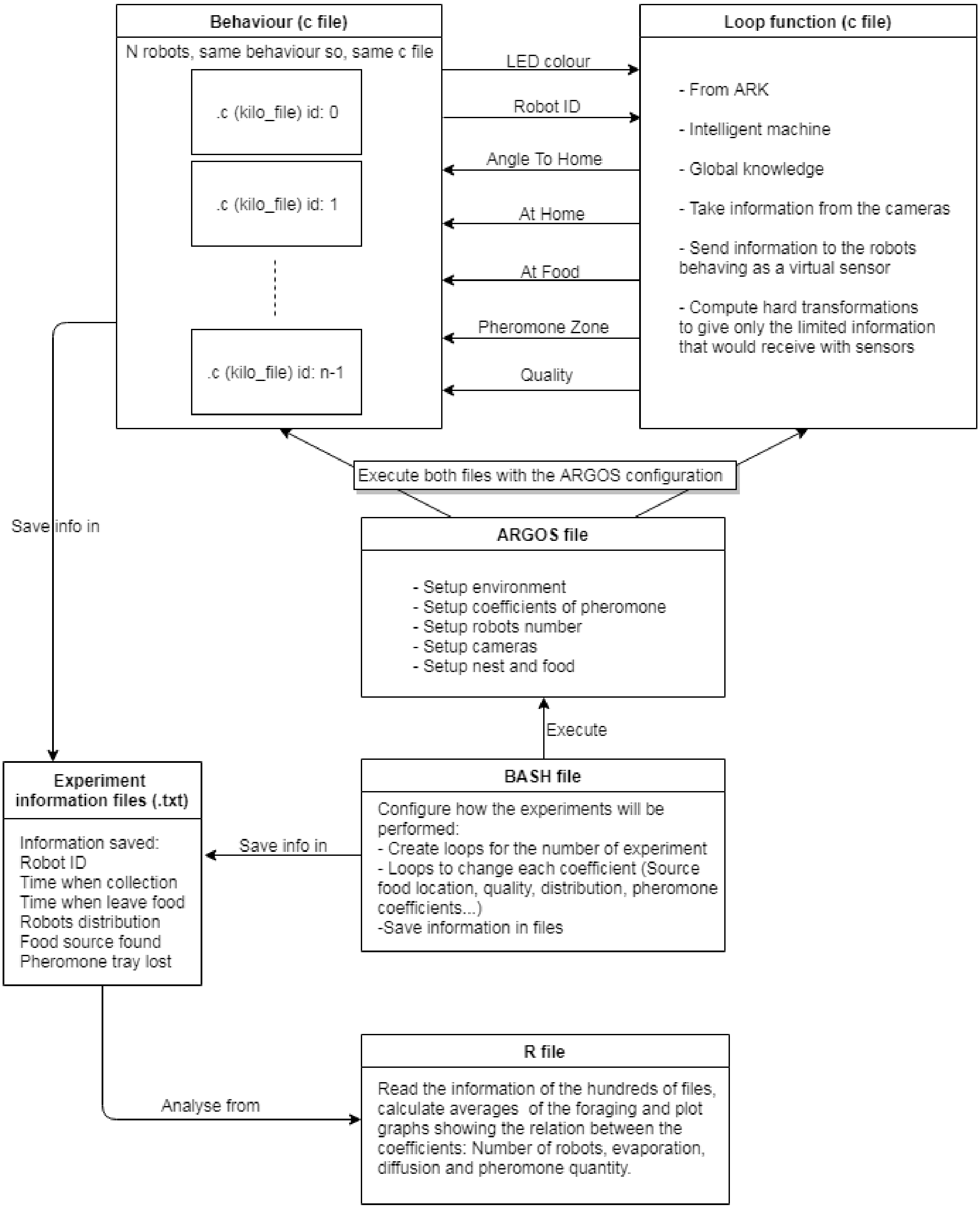


Figure 3. Files structure

In the diagram above, it is shown how each file is interacting to each other to perform a structured and efficient project. It is proven that the robot are accurately receiving similar information as the ants do, just based in what they are able to perceive through the environment.

4.2. Main behaviours

The robots have a clear and known purpose, imitate as accurately as possible the ants foraging behaviour depending on the value-sensitive of the food sources in order to achieve the best balance of quality collection in a short time.

Once the experiment is executed, independently of the food sources characteristic and independently of the number of robots, all the robots will start with the random move behaviour. Every time step they will be checking if it is received the virtual sensor of the food found. If the food is found, this will be carried to the nest. If it is not found, the robots will check if the pheromone is received. If none of those are received, the ants will be moving randomly. Once a pheromone input of the "pherozones" variable is received, it starts following the pheromone trail. The ants can get lost or can stop detecting the pheromone due to the evaporation and diffusion coefficients of the pheromone. Therefore, if the pheromone is not detected, the robots will look around searching for pheromone. If the pheromone is not detected anymore, the random move will be performed again.

Once the robots reaches home, the food will be left, the LED colour will change and the robot will turn half a round, 180 degrees, to orient themselves to the pheromone trail. The real ants perform this task in this way due to the lack of the non-global knowledge. Thus, they cannot go directly to another food source that maybe has a better quality.

In order to understand the behaviour of the robots in this system, a flowchart has been created to illustrate this performance in Figure 4.

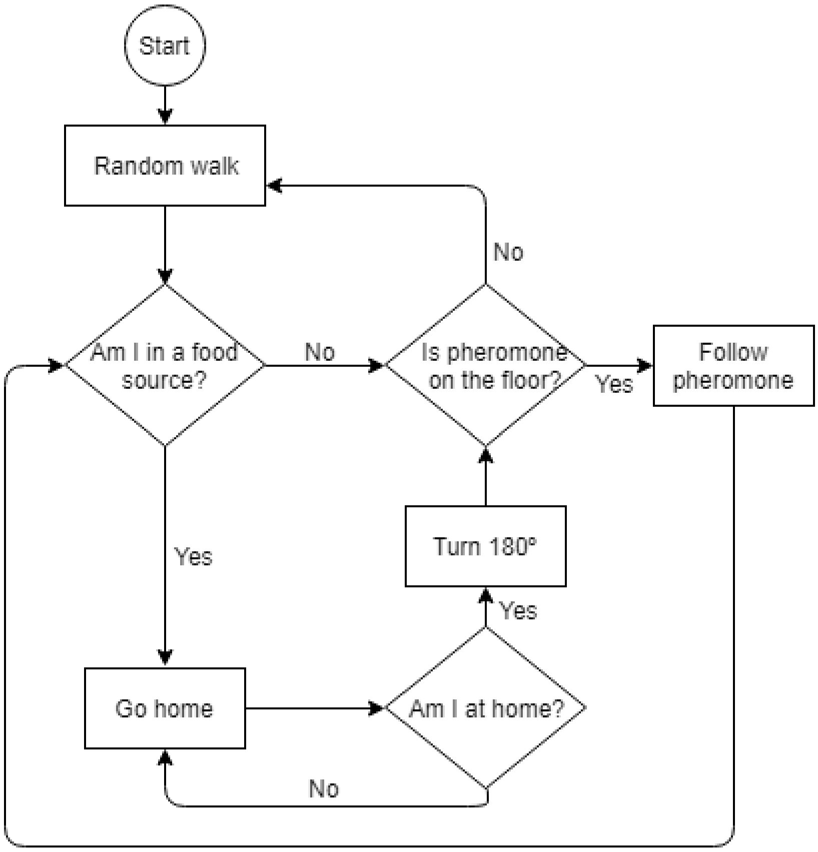


Figure 4. Flow chart of the robots behaviour

There are some main actions that should be explained in detail, as the follow of the pheromone behaviour. The real ants count on to antennas to follow the pheromone trail, in this case, a similar behaviour has been implemented. The ARK has a matrix of the floor divided by horizontal and vertical pixels among the total arena floor, creating cells. It is configured to have 150 cells per meter so, depending on the size floor; the matrix will determine its size. It is configured a visibility of the ants of 2 cm. This visibility will be performed only in the forward part of the robot, not in the backward. This means that the robots will have only 180 degrees of visibility, from -90 degrees to 90 degrees being 0 degrees the orientation of the robot. In order to perform the pheromone trail following, the loop function will send just the information of those cells that are accomplishing these characteristics. What is more, due to the fact that there is a bits limitation, the cells will be normalized to an angle comprising only 4 bits. Each bit corresponds to one area of the front part of the robot and is indicating 0 or 1 if it has any cell with pheromone. As observed, this method is creating large approximations that can be compared with the noise in the environment. This behaviour can be described as the following:

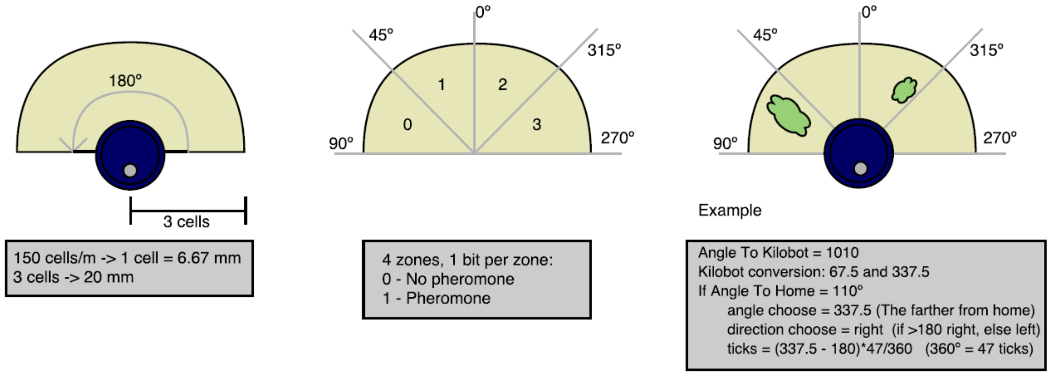


Figure 5. Pheromone zones calculation

Another important concept to bear in mind is how to change the properties of the pheromone in order to give an advice to the other robots about the food source quality. A high quality food source should have stronger pheromone than a low quality one. This difference will be observed in the quantity of the pheromone print. The quality has a range value between 0 and 10, being 0 a really low quality and 10 the best one. The constraint of this behaviour is that the loop function is the one that prints the pheromone through the virtual sensors not the robot. However, the decision of printing pheromone should be done by the robot. In order to solve this problem, the ARK is not using the quality variable to print the pheromone; this quality is sent to the robot when this is in the food source. It is processed for the robot and it is encrypted in the LED colour. The ARK prints pheromone when the Kilobots are showing a blue LED colour. That is why; the Kilobots will change the colour in order to advice the loop function to print or not the pheromone. This advice will be done as a probability depending on the quality found. For instance, a quality of 0 means a 0% of pheromone printing. Thus, the LED of the robots will be green all the time not allowing the ARK to print this pheromone. A 10 quality food source represents a pheromone printing of 100% of the time, so, the LED of the robot in this case will be always blue. This probability is calculated using a random number between 0 and 100. This random number will be compared with the quality previously multiplied per 100. If this number is lower than the quality, the blue LED will be used to print the pheromone, as shown in the next image:

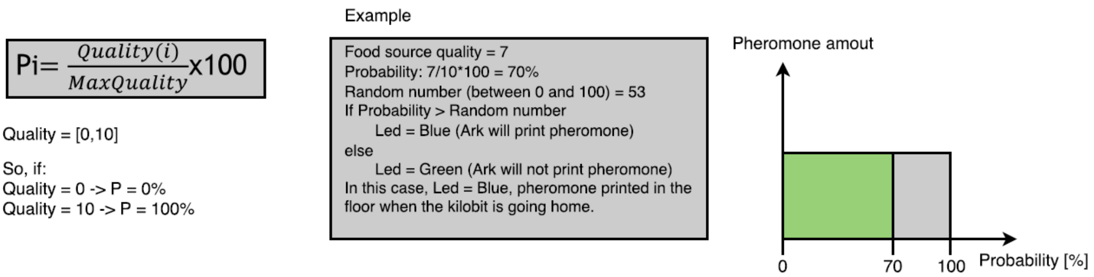


Figure 6. Calculations of the quality probability

Another challenging behaviour that should become as accurate as the ants one is the calculations of the pheromone coefficients. There are three main coefficients to bear in mind, the evaporation, the diffusion and the pheromone quantity. These coefficients can be changed at the start of each simulation or real implementation and also, during the performance of the real implementation. These three coefficients are based on the real ants' coefficients and the equations used are extracted from scientific researchers of mathematical approximation models as explained in Relevant Theory and Analysis section. In each time step these parameters should be computed for every cell of the matrix floor depending on the pheromone position. It is updated using a loop. In this loop every cell of the matrix is checked, if there is any robot with blue LED, inside the concrete cell the value of this cell will increase adding the amount previously defined. The matrix will contain the quantity of pheromone in each cell and, jointly with the pheromone coefficients, is recalculated. This matrix will be printed in the environment every 10ms. It is not done every time step because this will compromise the system. It is needed a balance of the printing time with the frames per second due to the fact that print the environment takes a long CPU time.

4.3. Experimental implementation using Kilobots

Once the simulation is fully working and similar behaviours to the real world are obtained, a simulation with swarm robots in the robotics laboratory of the University of Sheffield should be performed; corroborating the results obtained in the simulation and also finding more interesting scientific results in this field.

The files in the real experiment are organized in a similar way to the simulation. There is the behaviour one, a C++ code that is compiled alone and is converted to a .hex file. This one will be updated in the kilobots before the start of the experiment. However, the robots memory is limited. Therefore, only a maximum of 33Kb is available. Due to this fact, the code should be implemented efficiently without long variables unused.

The ARK system is implemented through a program called QT which has a huge amount of functions predetermined in order to create an interface capable to perform the experiments. This ARK over QT includes a GUI which creates a graphical interface to control the process lively enabling the identification of the Kilobots, the assignment of their IDs and the uploading of the codes. In addition, it includes the updating of the function loop created that, interacting with the GUI, can modify the parameters online while watching the virtual sensors through the screen. This powerful tool also enables the possibility to save information of the experiments and record videos to analyse all the behaviours of the robot.

The real experiment execution will follow the next steps:

1) Make the robots awake, identify each robot and assign an id.

The robots at the start are in sleepy mode in order to save battery. They are blinking in a purple colour every few seconds. Once they are reset the voltage should be checked. This voltage is identified by the LED colour of each robot. After checking the voltage, the IDs are identified from the cameras to know the position of each robot. The IDs assignment is the next step; it is done using a colours configuration. The ARK system asks to the robots to change their colours in order to identify them and assign them a unique ID, starting for 0 to the number of robots minus one.

2) Update the behaviour

The executable code .hex is updated to the robots after resetting them. In this step any command should not be used during the few seconds that the action is working due to the fact that the robots can get badly injured. Once it is finished, the robots should be reset. In this point, the robots are prepared to start the experiment.

3) Update ARK code and configure the parameters

This step consists on updating the experiment created with the GUI interface. In addition, some parameters in the interface can be changed depending on the experiment to conduct. Variables as the food and nest location, the qualities of the food, the evaporation, quantity and diffusion should be set carefully.

4) Execute the experiment

This step consists only on running the experiments. This shows through the screen the parameters configured before and it can still change some of them if it is not suitable for the experiment. For instance, if there is a need to allocate a robot in a current food source.

5) Run the robots

The last step consists on executing the robots, changing the mode to the run mode. This is so important to be performed in the last step due to the fact that if it is executed before the running execution, the relation between the current robot position and its ID can be lost.

4.4. Experiments performed

There are a wide amount of experiments that has been performed. Before the execution of them, some tests have been done, as the ones described below. First of all, it has been performed a simple test to check the important functionalities without the pheromone matrix. This test consists only in a robot following the path from the food to home evaluating the angle sent by the ARK system. Once the robot reaches the nest, the LED will change. This experiment is useful to check the accuracy of the angle given.

The next test involves two robots and a food source without a quality variation, assuming always a quality of 10. It is taking into account the pheromone matrix. One robot is located in the food and the other in the nest so, the one of the food starts printing pheromone while the one in the nest starts walking randomly. Once the first one reach home, it starts following each own trail to the food. When the second one found pheromone in the floor will perform the same behaviour. It has been used high values of quantity and diffusion and low values of evaporation to success in the collection with only two robots.

The last test includes the quality coefficient for a food source and two robots. In this last test it was proven how the difficulty to follow a pheromone trail is added and how is the pheromone just printed half of the time with a quality food source of five.

Once the functionalities are tested, some experiments can be performed in the simulation mode. The first one is a simulation with a certain amount of robots, around 50, starting from a random position and foraging for food. The food sources have the same quality, quality 10, but different distance from the nest. It has been used three food sources, one located at 0.33 m from the nest, another at 0.67 m and the last one at 1 m. In Figure 7, the main collection differences can be analysed. The collection of the food using pheromone can be shown using the ARGoS tool to run a simulation:

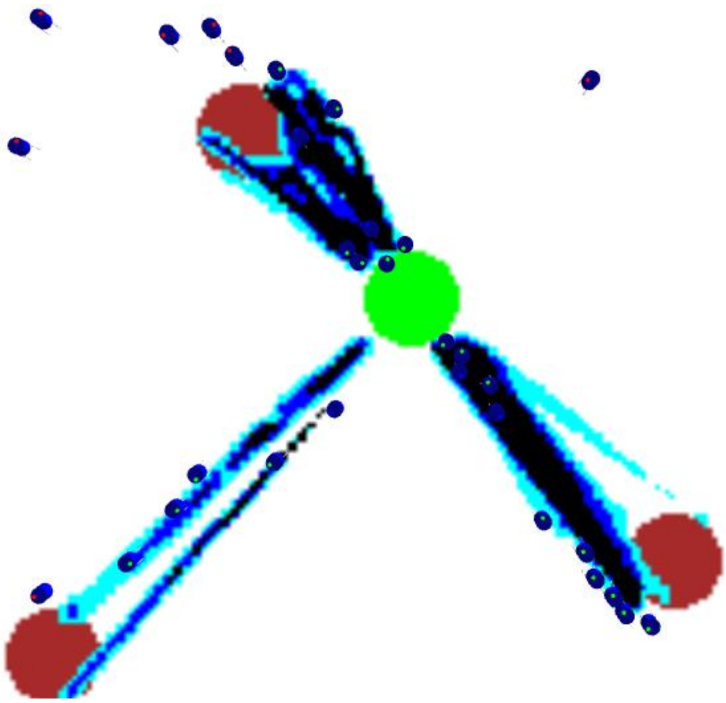


Figure 7. Distance comparison experiment

The second experiment performed includes four food sources and 50 robots. This food sources are located quite far from the nest and in the same distance from the nest because the key of this experiment is the quality variation. It is used different quality food sources, of 10, 8, 5 and 3. In the next picture a simulation of the experiment is shown:

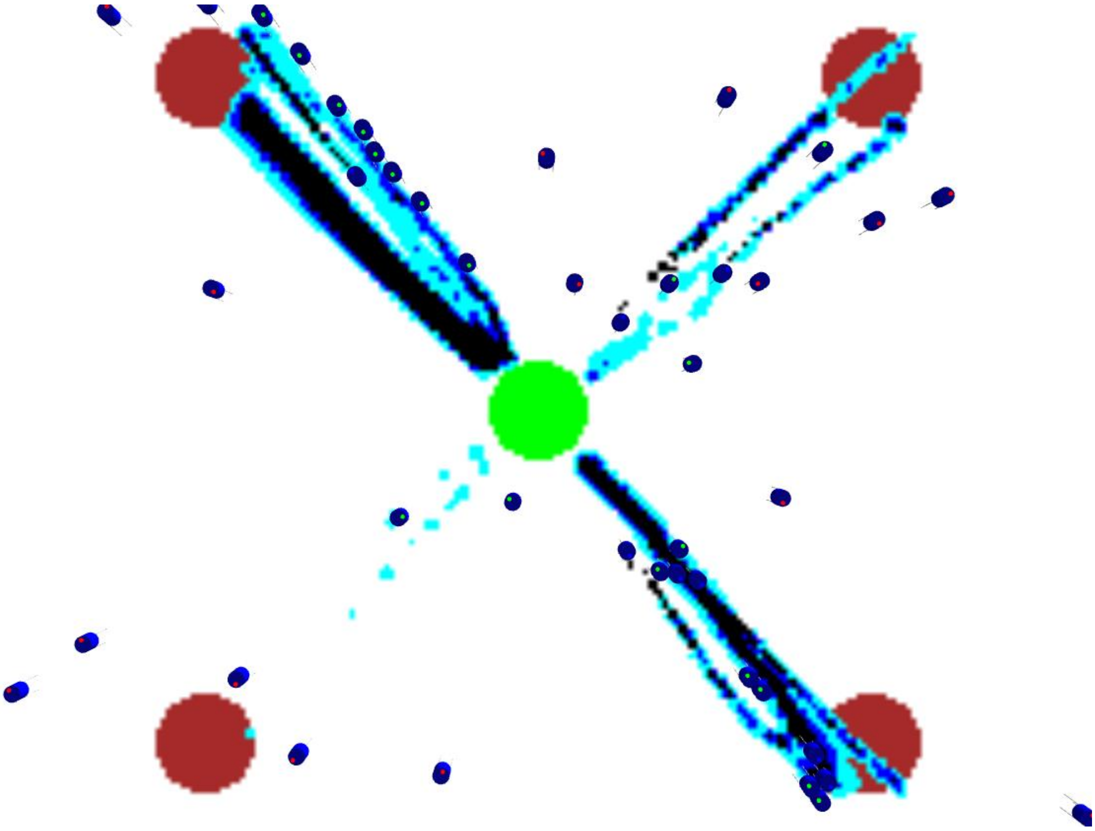


Figure 8. Quality comparison experiment

This experiment allows the analysis of how the quality affects in the pheromone printing and in the foraging. In addition, it is helpful to analyse the trade-off of the coefficients of the pheromone to collect the food but without wasting pheromone. It can be observed in the last picture how after a while the robots are not interested in the low-quality source, concretely, the food source located in the left and down of quality 3.

The last experiment performed is the most complex one. This will be essential to find the most efficient Ants Colony Model to maximise the throughput. This experiment which takes around four days to be performed includes 100 experiments for each of the following combination of values:

No. robots	Evaporation	Diffusion	Quantity	Food
50	0.01	0.3	15	1 source
100	0.03	0.6	30	2 sources
200	0.05	0.9	60	3 sources
				4 sources

Table 2. Proposed experiment options

The values are going to experiment sustainable changes in order to look for the best environment. Some coefficients of the pheromone may be increase or decrease and tested a large amount of times. In addition, the quality in this experiment is a factor that will be varied once the best performance has been chosen. Also, a wide number of robots are going to be analysed only with the final values in order to analyse the best swarm size for each environment.

This experiment is the one that enables a better interpretation of the ants' behaviour and its parameters in order to perform the most effective collection. This experiment permits to analyse how the collection changes depending on the increase or reduction of the pheromone coefficients or the robots number. Analysing the data obtained carefully, the better quantity of robots for foraging can be found for each environment. However, this value is depending also with the pheromone qualities. Due to this fact, in Swarm size analysis section, the relation between the pheromone coefficients and the robots quantity is going to be evaluated.

The environment used in this experiment is comprised for the same items of the last experiment; four food sources and the nest align in the middle. In addition, two more environments have been added in order to corroborate the choice of the best parameters analysed and compare weather or not the behaviour of the pheromone coefficient is actuating in the similar way for each environment. The next three environments will be tested:

- Environment 1: Only one food source with a quality of 10 located at 0.33m from the nest. It is interesting to check this behaviour in order to check the ants' behaviour when only one food source is available. It will allow an analysis of the overcrowding in a swarm environment and to check the efficiency of the collection depending on the number of robots:

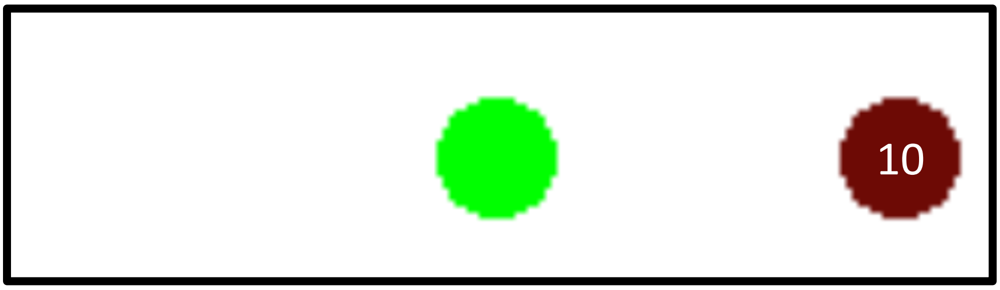


Figure 9. Environment 1

- Environment 2: The second environment is similar than the first one but adding a food source. This new source is located in the opposite site of the first food source. This second food source has half of the quality of the first one. This environment evaluates how the robots are able to discern between two food sources while maximising the throughput. What is more, it is also evaluating how the coefficients of the pheromone can be changed to obtain a better food quality priority.

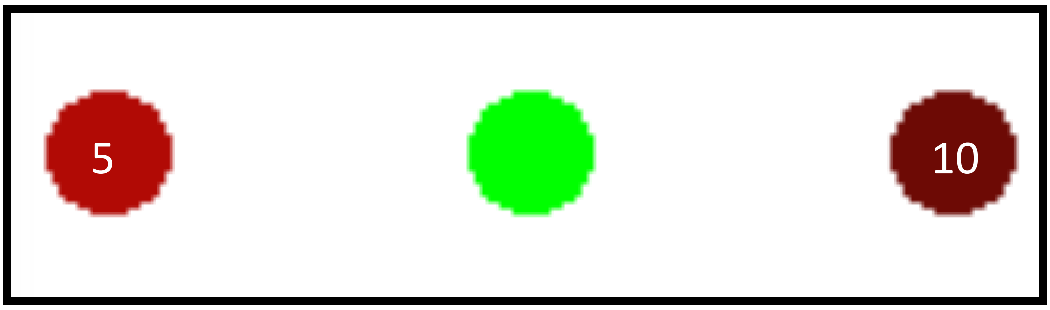


Figure 10. Environment 2

- Environment 3: This last environment is comprised with the same qualities and food sources location as the one in the Quality experiment. This environment is going to reveal how the robots can distribute and collect food while the number of food source is really high. It should be obtained which parameters of the pheromone and how many robots are needed for this environment. The current environment with each quality in white is shown in the next figure:

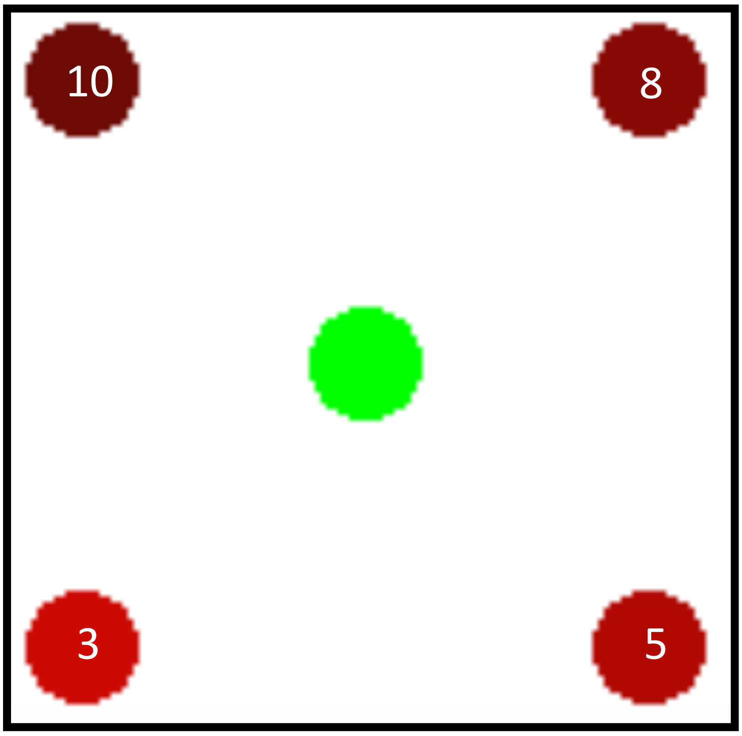


Figure 11. Experiment 3

Another interesting parameter that should be checked is the execution time of the experiment. This will be compared with different qualities and how the collection increase over the time using the parameters previously found to maximize the throughput.

All the experiments explained above are planned to be completed firstly, in simulation mode and later proven in the real robots. The simulation is going to allow more results and a better behaviour than the real implementation. Nevertheless, the real implementation is more reliable and realistic. Thus, it will have a more similar behaviour with the real ants. However, the real implementation is currently being developed.

5. Results

5.1. Distance experiment

As explained in Methodology, the first experiment performed in the simulation is the relation between the distances of the food source. This experiment will reveal which environment is the most efficient to achieve the ants' foraging.

First of all, a comparison of a medium food source from the nest and a close source from the nest is performed in order to check which is giving a large collection for the same period of time.

Analysing the collection of the food over 100 experiment with an environment comprise by two sources where source 1 (red) is located at 0.66m from the nest (medium distance location) and source 2 (blue) at 0.33m (near distance location), an overlapping histogram for both sources is shown below:

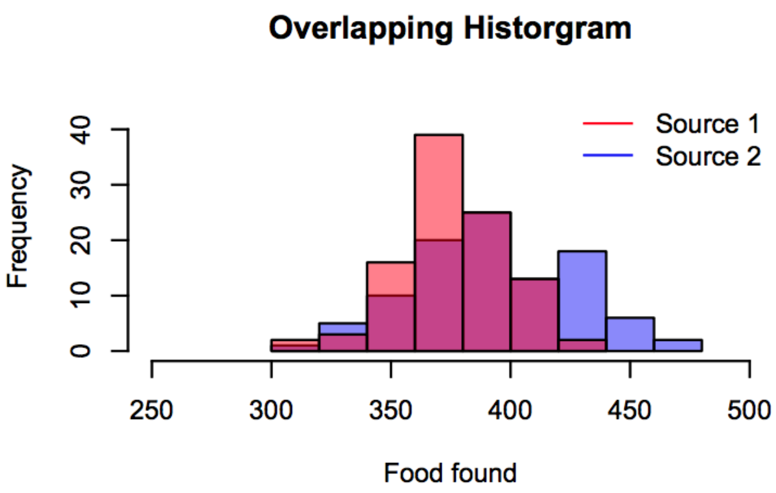
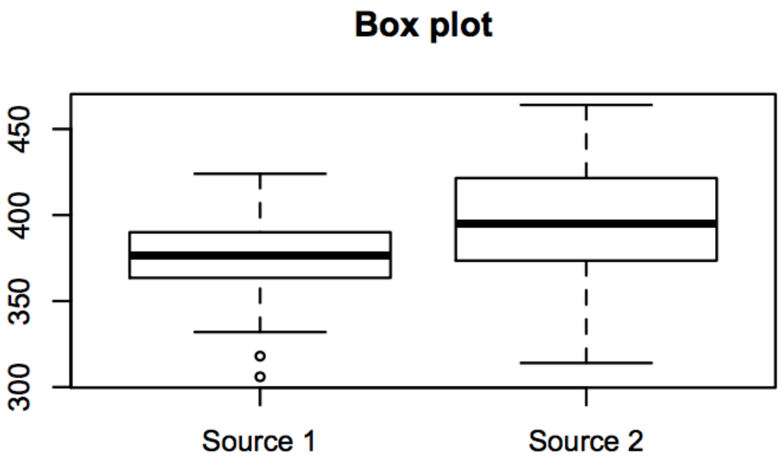


Figure 12. Histogram mid-close food

It is analysed how the collection of the food discern over the distance. In the medium location source the average collection through the 100 experiments is up to 375 whilst the closest source is around 400.



In the first instance, it can be concluded that the second source has a better performance than the first one. Analysing deeply a box plot can be plotted to approve this finding:

Figure 13. Box plot mid-close food

A box plot is a potent tool to organise the data in a visual and clear way, agreeing with (Williamson, Parker, & Kendrick, 1989). The box plot shows in a bold line the average collection and in the highest and lowest collection in the top and in the bottom respectively. What is more, the height of the rectangle shown is describing the spread of the data.

As it can be observed, the close source, source 2, has a wide range with highest and lowest values but with a middle point much bigger than the medium source. It can be conclude that the close source is performing better in most all the cases but it is not always the best solution due to the fact that its lowest point is lower than the one of the medium food source. That is why; the distribution of the robots in the environment is random which means that in some experiments the robots can be located far from the current food source. It should bear in mind that in order to perform a full analysis of the environment a high number of experiments should be execute in order to achieve reliable results.

The next comparison is performed with a far source (1m) and with a medium source (0.66m) from the nest. The results obtained using 100 experiments are the following ones:

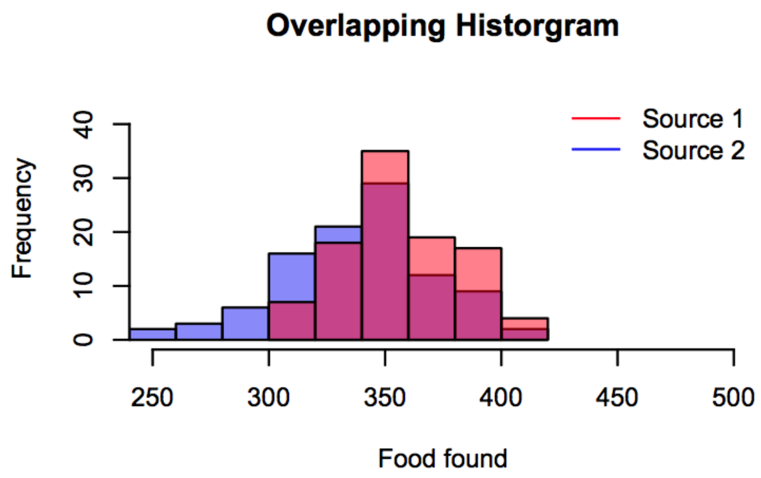


Figure 14.Histogram mid-far food

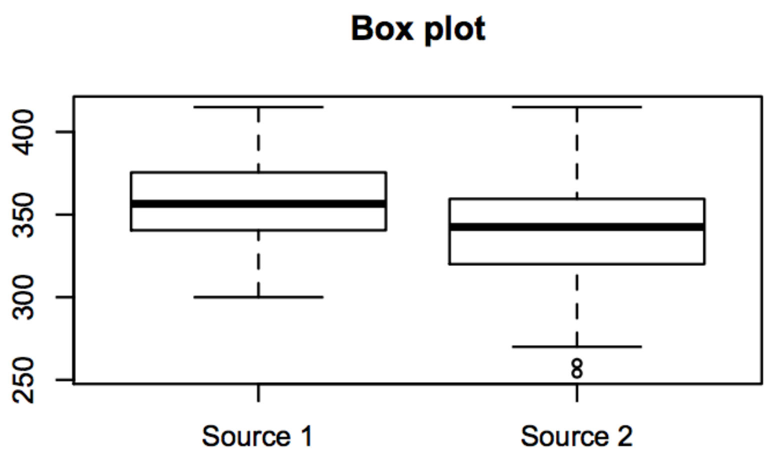


Figure 15. Box plot mid-far food

These two charts reveal that the difference of the collection of the food, taking into account the distance, stills the same, as shown above. In this case, source 1 is the medium source and source 2 is the far source. It is analysed how the medium source is achieving a lower collection with an average around 360 whilst source 2 is up to 340. In this case is shown how the collection of the medium source is reduced due to the use of a lot of resources to achieve the farthest source.

To sum up, to accomplish the best collection of food to the nest, the best food source location is the one closest to the nest. However, the food quality will affect in the ants decision. That is why; in the next section this phenomenon will be taken into account.

The distribution of the robots, once the experiment is finished, is an interesting parameter to analyse. This distribution is represented as the number of robots that are working in the collection of a specific food source when the experiment is stopped. This is delimited by a rectangular shape that comprise the way from the food to the nest.

The comparison between a close source and a medium source is the following one:

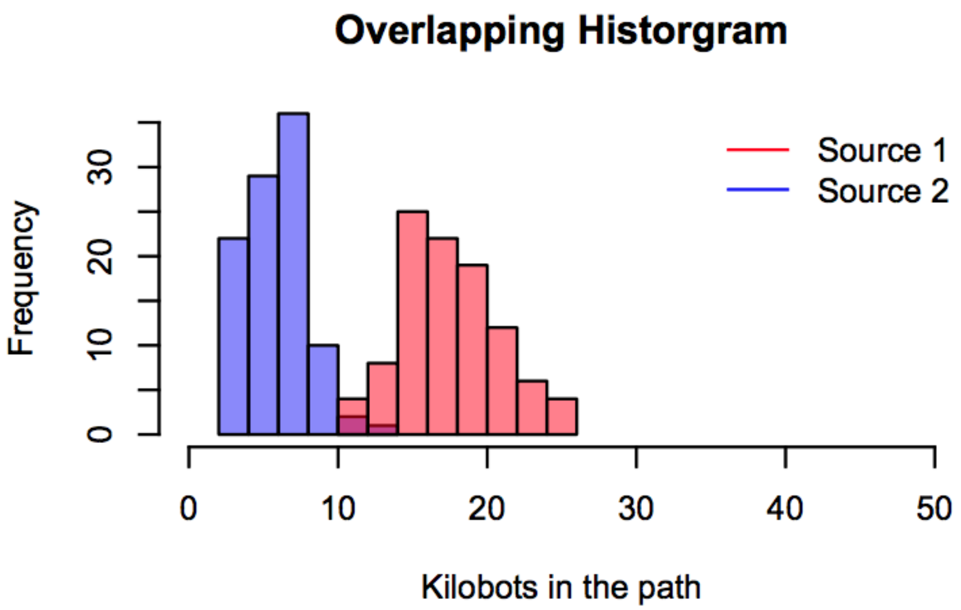


Figure 16. Distribution histogram mid-close food

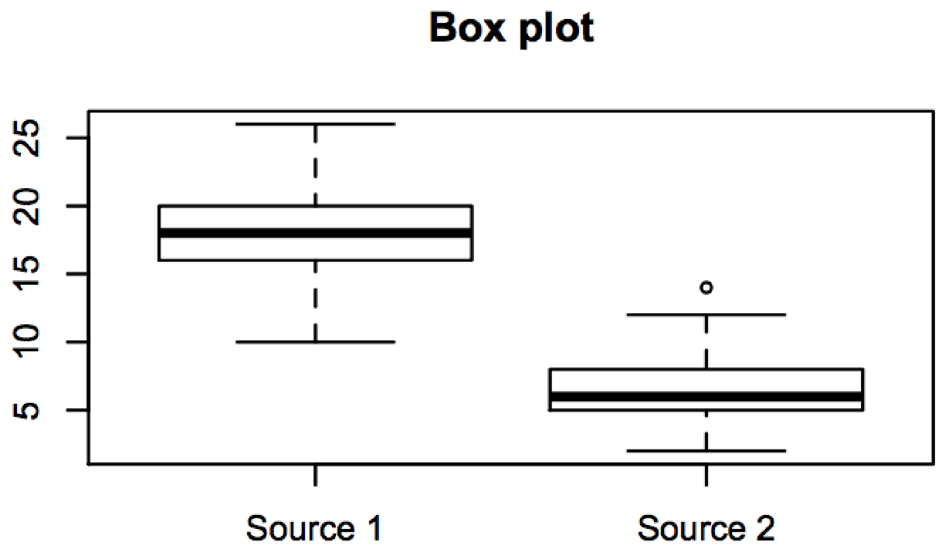


Figure 17. Distribution box plot mid-close food

The comparison between the far and the medium one is the next one:

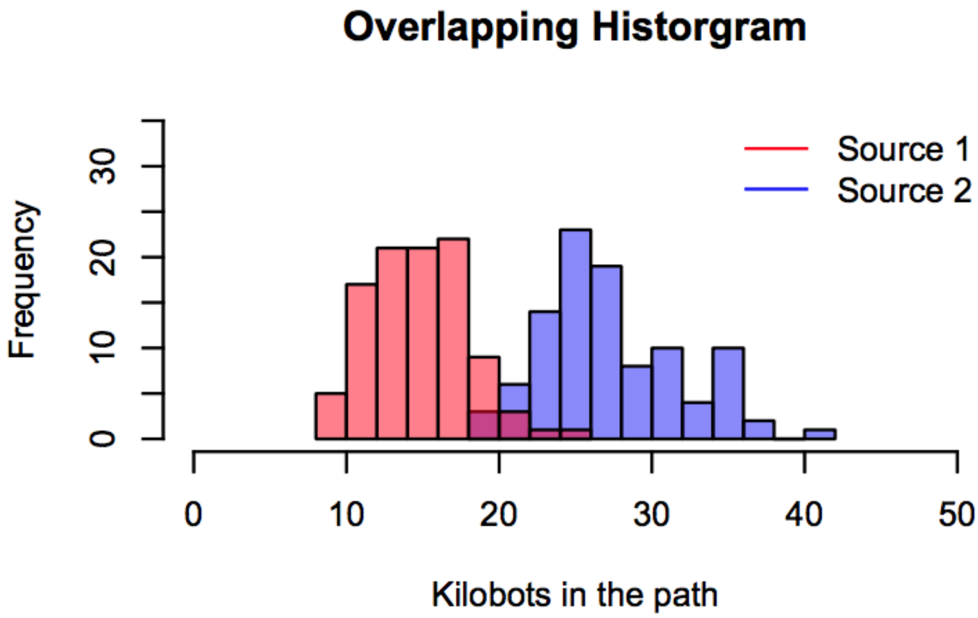


Figure 18. Distribution histogram mid-far food

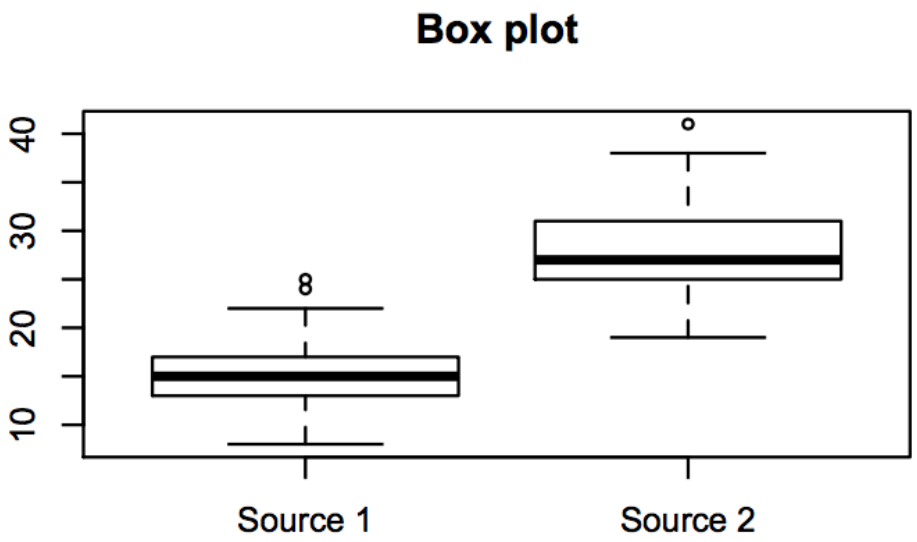


Figure 19. Distribution box plot mid-far food

The average of the experiments shown is providing the following averages: In this case, there are 6 robots for a close source, 16 robots for a medium source and 28 robots for the farthest one.

It is remarkable the results obtained in this last analysis, a higher number of robots are working in a further food sources rather than in closer ones. This fact happens because the trail of pheromone left from the robots located in a further food source is longer and it is spread in a wider distance. Due to the randomly disposition of the robots, the chance to find this trail is higher than the one for closer food sources that has a smaller area of interaction.

To conclude, a highest distance will enable a lowest collection but also, a large number of robots collecting food from it. The efficient foraging in an ants' colony, just taking into account the distance, reveals that a food source nearby spends less resources whilst enabling a fastest collection of the food.

5.2. Quality experiment

Once the distance is proven its effects in the ants' foraging, the next step is to provide a deeply analysis of the quality in the food sources.

First of all, an environment of 50 robots and 4 food sources experiment is implemented. The food sources have different qualities 10, 8, 5 and 3 respectively. After a ten minutes performance, the results obtained are the following ones:

Source number	Quality	Collections	% of collection
1	10	69	30.67%
2	8	55	24.44%
3	5	52	23.11%
4	3	49	21.78%

Table 3. Quality collections results

It can be observed how the collection number decreases with the quality due to a better quality involves a printing of pheromone every time step while a lower quality enables the printing in a linear way.

However, the percentage of collection is not following a linear progression. For instance, the percentage of collection of a quality 10 is around 30%, while the collection in source 5 is around 20%. This second one is printing half of the pheromone of the first one but the percentage of collection is not the half. That is why; some of the robots reach the food source just by chance, due to the random walk. Thus, it cannot be expected to have linearity between the collection and the quality, as shown below:

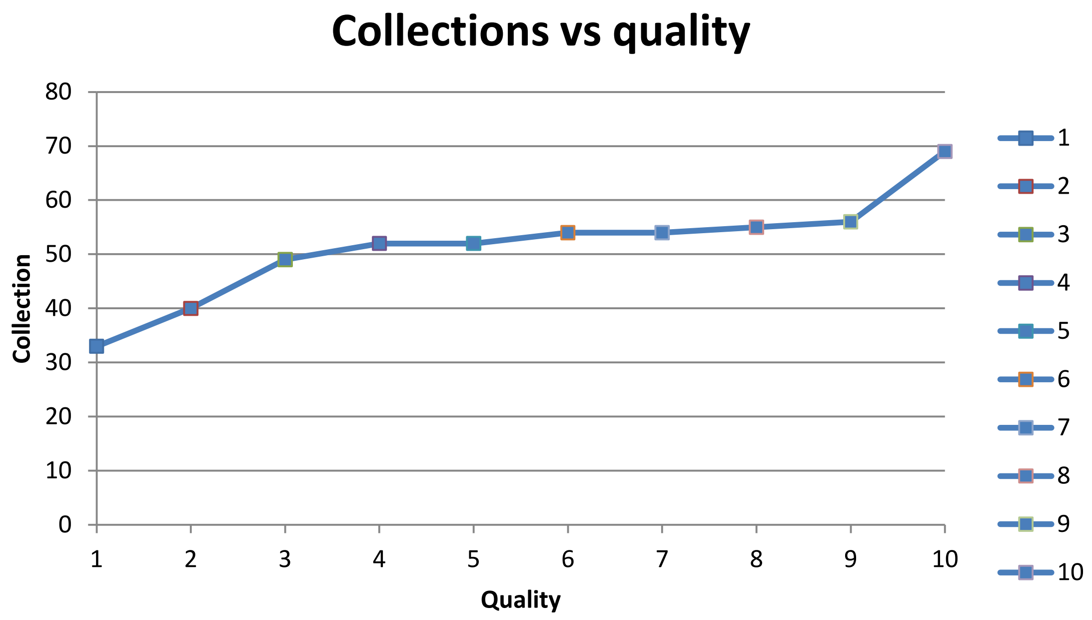


Figure 20. Collection vs quality graph

The data shown is provided by an experiment of changing each quality food source in comparison with a 10 quality food source. It is showing the non-linearity explained above. Another fact that makes the quality-collection relation nonlinear is the fact that once a quality big enough is achieved, in this case quality of 3, the ants are able to follow the pheromone path. Therefore, the ants continue collecting food from this point instead of foraging for richer food sources. Changing the diffusion, amount or evaporation pheromone parameters this quality threshold can be modified. As observed, from quality 3 to 9 the collection is almost the same which proves that the robots can follow the path without getting much lost. Despite this fact, the collection in 10 is quite higher due to the ants never get lost through the trail. However, qualities lower than 3 are achieving really low collections, mostly collecting the food because of the random move and without having the chance of following a pheromone trail.

Box plots comparing each quality with a quality of 10 can be plotted to corroborate this relation. It is created a scenario with two food source in the same distance to the nest and with 100 robots. The results obtained are the next ones where source 1 is the quality modified and source 2 is always 10:

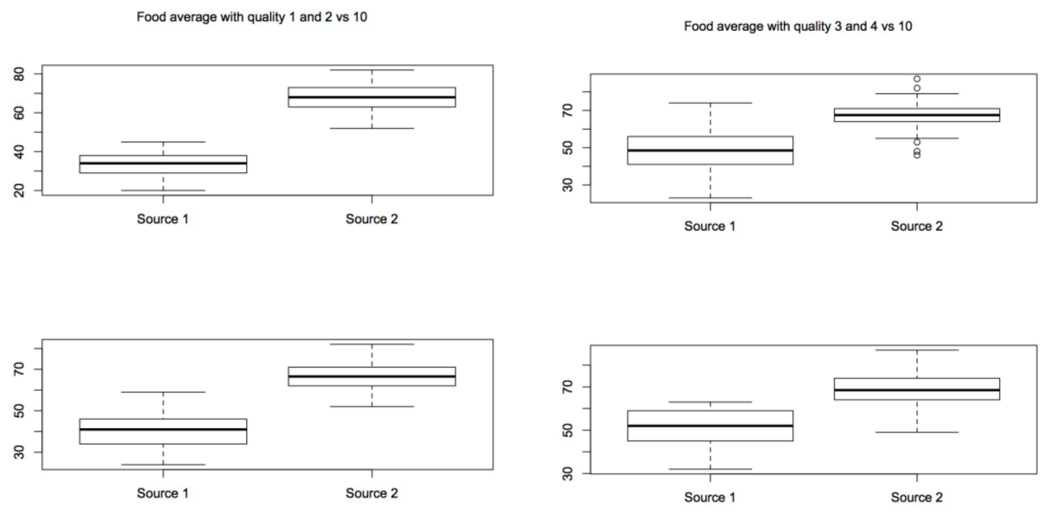


Figure 21. Box plots with 1, 2, 3 and 4 qualities, respectively vs 10 quality

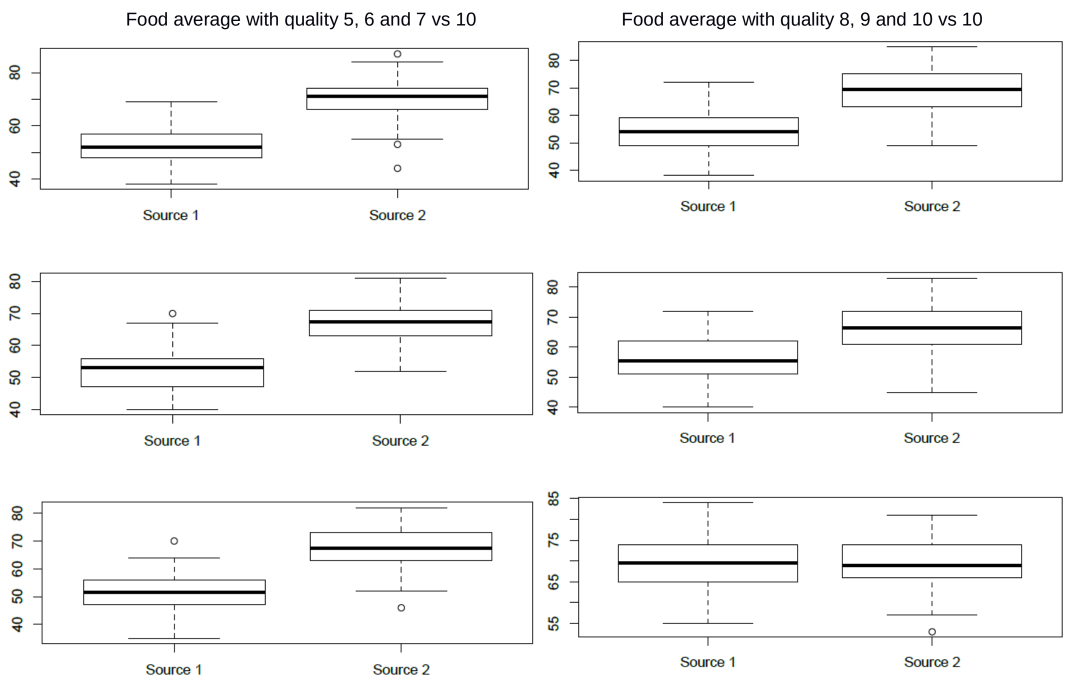


Figure 22. Box plots with 5, 6, 7, 8, 9 and 10 qualities, respectively vs 10 quality

To conclude this set of quality experiments, it can be affirmed that a fully analysis about the behaviour of the quality food source has been performed using the same environment in each quality and evaluating the changes in the collection performance.

Furthermore, the nonlinear relation of the quality of the food sources with the collection food in each source has been proven. It was found that the quality act as a threshold, allowing a more or less constant collection in the middle qualities.

In addition, a quality of 10 is always the best option due to the perfect creation of the pheromone trail not allowing the robots to get lost. For a quality above the threshold of 3, the food collection is quite poor due to the non-constant in pheromone printing.

5.3. Parameters experiment

The last experiment is composed for a large number of experiments changing the environment to observe the differences between them and to obtain the best foraging performance. This experiment is going to include changes in the quality, number of robots, distance, among others.

The parameters changed including their values are defined in Table 2. This experiment is the most difficult to obtain the results due to its complexity. It has been researched which forms of graph representation are the most suitable for this case. It has found that the heat maps plots are the most understandable form. Some researchers, such as (Hettenhausen, Lewis, & Mostaghim, 2010), have obtained interesting results about the swarm optimization using heat maps.

5.3.1. Preliminary experiments

These first experiments are going to show which parameters are more suitable to use in order to try to maximise the throughput. In addition, some interesting results about the trade-off between pheromone coefficients will be found while combining different environment with different number of food sources and different swarm size.

Using a food source located in a medium point (0.66 m) with a quality of 10 has been implemented obtaining the following collection performance:

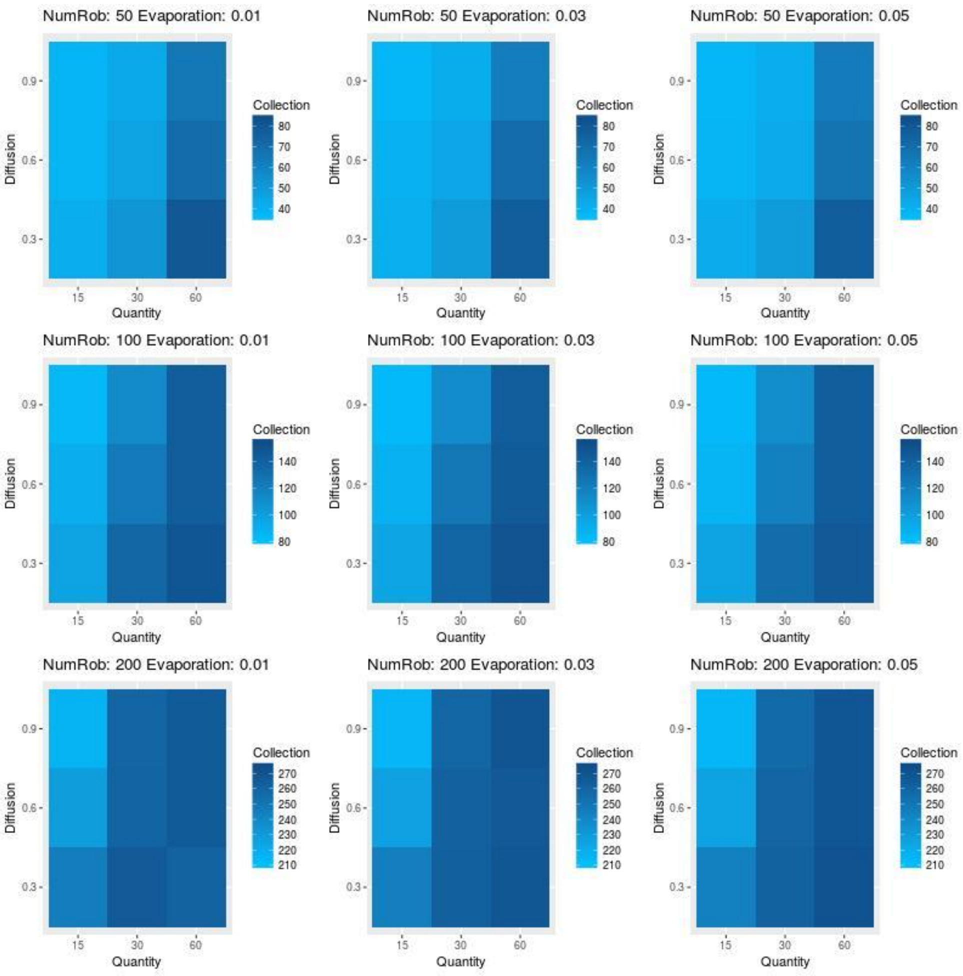


Figure 23. Heat map of the collection performance

First of all, it is clear that a higher number of robots can achieve a higher collection of the food due to the fact that there is more ants working on them and the pheromone trail crated is stronger.

Despite this fact, the relation between the increments of the robots with the collection is not linear. For instance, taking a concrete experiment of 50 robots with X collection, the collection obtained for the same environment but with 100 robots is not 2X, it is lower. This is because in the case of 100 robots the collection is not being performed in an efficient way. The environment is getting overcrowded and it is not allowing the highest performance per robot.

The table of these relations can be observed in the next figure, while the distribution implementing an experiment of 50 robots is printed in the first row of heat maps, the 100 robots in the second and the 200 robots in the last one:

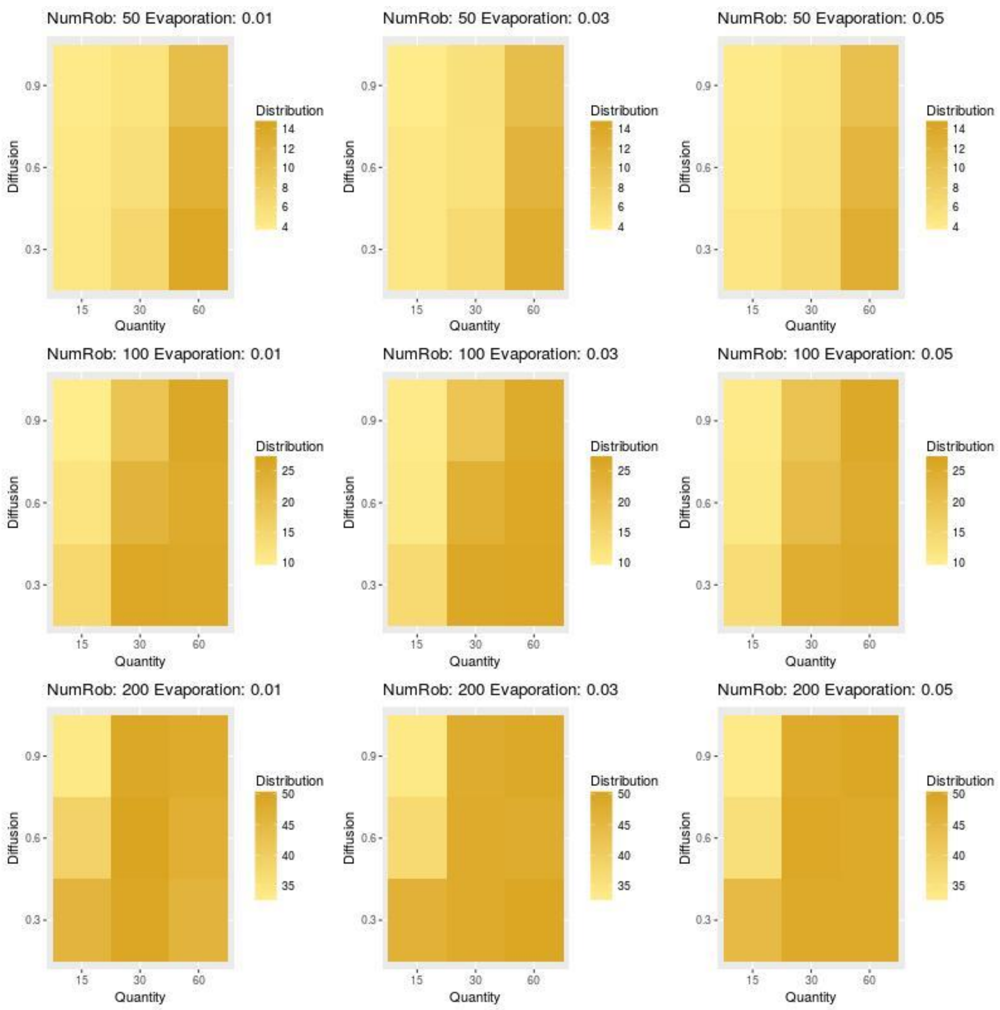


Figure 24. Heat map of the distribution performance

In each heat map is observed the pheromone quantity in the horizontal axis, the diffusion parameter is used in the vertical axis and the changes of colours is the distribution of robots, being darker a higher amount.

It can be observed that varying the pheromone parameters, the collection and distribution is experimenting big changes. There are some relations that can be observed.

Some deductions have been extracted from the plots shown above. These ones reveal that when the food source is increased, it is needed a lower amount of pheromone to achieve a better performance. A number of robots will be distributed in a better way and the collections will increase with a lower amount of pheromone with lower diffusion and higher amount. It is discovered that the distribution and the collection are providing the same behaviour.

It is also analysed that for high values of diffusion is needed an increase of the quantity of the pheromone and high number of evaporation to achieve a higher collection. This phenomenon occurs because when the pheromone is diffuses the quantity of pheromone in the initial position decreases faster to spread through the environment. Thus, a high number of the quantity is needed in order to not lose the pheromone trail. However, if the pheromone is too spread the robots will walk slowly to the food source due to the following of a wide trail area which make them not take the fastest path that is why; a high number of evaporation is needed in this case which will reduce the values further from the most efficient path.

Another analysis that should be performed is to check the increment of the number of robots among the quantity of pheromone. A table can be performed rating the collection of 50 robots to one and providing the values of the different amounts and different quantity of robots:

Amount / No. Robots	30	50	100
50	1	1	1
100	1.8, 1.9, 1.9, 1.9	1.7,1.8,1.8,1.8	1.5, 1.7, 1.7 1.8
200	3.34, 3.45, 3.7, 3.5	2.7, 3, 3, 3.2	2.3, 2.8, 2.8, 3.1

Table 4. 50 robots reduction

The different values that are asunder by commas represent one food source relations, two food source relations, three food sources relations and four food sources relations, respectively.

It can be observed that, for every food source, located in the same cell, the value relation for each environment remains almost constant. The table is revealing that the best relation is achieved with a low quantity of pheromone for 100 robots and for 200 robots in all the cases.

On the other hand, It is true that the number of collections increases with the number of robots but it has to bear in mind that the relation of increasing the number of robots with the increase of the collection is not linear due to the fact that to be linear, a collection of one in an environment by 50 robots should correspond to a collection of 4 in an environment by 200 robots. This happens because the overpopulation. The robots become less efficient after a certain amount of workers collecting food. This relation will be analysed in the Swarm size analysis section.

To continue, the relation between the pheromone amounts, taking as a reference an amount of pheromone of 30 is shown in the next table. This experiment uses the same variables shown previously but normalizing the robots when the amount 30 to 1 robot:

Amount / No. Robots	30	50	100
50	1	1.07	0.97
100	1	0.96	0.77
200	1	0.86	0.66

Table 5. Quantity of 30 reduction

In this case, it is shown an average of the different environments. It can be analysed how the amount cause an important effect in the collection. For 50 robots, the best performance found is with 50 amount of pheromone. When the number of robots is increased, the amount tends to be needed, as lower as possible. This effect corroborates the assumptions found with the heat map shown before. For instance, when the environment has 200 robots and the amount of pheromone is 100, the collection of food is reduced 34% in comparison of an amount of 30. That is why, future experiments reducing the quantity of the pheromone will be done to find the best value for the different environments.

Another interesting results extracted from this experiment is the values to find the best and the worst collections. These results are summarized in the following table:

	Food sources	No. of robots	Amount	Diffusion	Evaporation	Total value
Highest collection	1	200	30	0.3	It is not decisive	268
Lowest collection	4	50	100	0.9	It is not decisive	39
Highest distribution	1	200	30	0.3	It is not decisive	50
Lowest distribution	4	50	100	0.9	It is not decisive	5

Table 6. Best and worst performances

This table reveals that, in order to achieve the highest number of robots working in the path and the most efficient collection, the number of food source should be low; the number of robots should be high; the amount of pheromone should be small; the diffusion should be small and the evaporation is not a decisive factor.

5.3.2. Enhance parameters experiment

Due to the results obtained, new simulations should be done using lower quantities of pheromone and spreading the diffusion values. In the next case the following parameters and environments are changed:

No. of robots	50	100		200	
Evaporation	0.01	0.03		0.05	
Diffusion	0.1	0.2	0.3	0.6	0.9
Amount	15	30	60	75	90
Food sources	1 (Quality 10)	2 (Quality 10 and 5)		4 (Quality 10, 5, 8 and 3)	

Table 7. Parameters analysed

In this simulation, the number of values analysed for the diffusion and the amount coefficients are increased from three to five comprising a higher range of values. In addition, the different number of food sources analysed are three, one for only one food source in a medium distance and with quality 10. Another one with two food sources in a medium distance and qualities 5 and 10 and also, a last one with four different qualities (10, 5, 8 and 3) located all of them in a medium distance from the nest and spread in the environment.

A sort of heat maps are performed to understand how the qualities actuate using all the parameters analysed before. The food collected in each food source depends on their quality, thus, a high quality collection has more importance than a low quality collection. This total collection for each food source is weighted in a maximum of 1, according to its quality. For instance, if a food source with quality 10 has a collection of 4 items this will be counted as 4 items but if the same collection is done in a 5 quality food source the items collected will be weighted to 2. The total collection is count as:

$$\boxed{Food\ collected = \sum_{i=1}^{No.\ of\ sources} \frac{Collection_i * Quality_i}{10}}$$

Equation 3. Weighted food collection

For example, a collection of 3 in a food source of 6, a collection of 8 in a food source of 2 and a collection of 2 in a food source of 10 corresponds to the following total collection:

Food collected = 3*6/10 + 8*2/10 + 2*10/10 = 1.8 + 1.6 + 2 = 5.4 items

The information collected through the experiment has been organised and plotted using heat maps, as before. A sum of the weighted food has been done and the collection found is shown in the next page. This sum has been done for each of the environments explained in Experiments performed chapter.

- One food source:

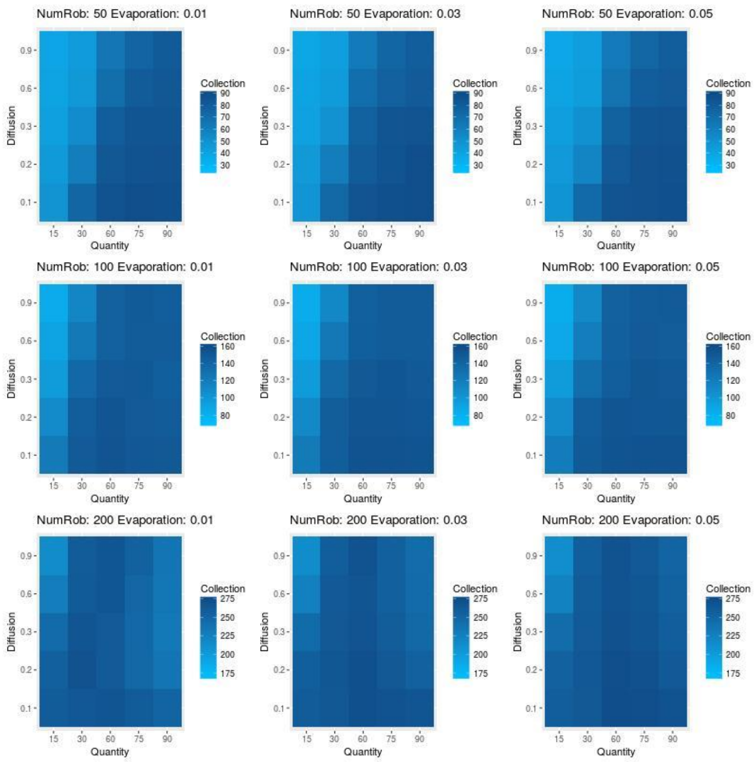


Figure 25. One food source wide values

- Two food sources

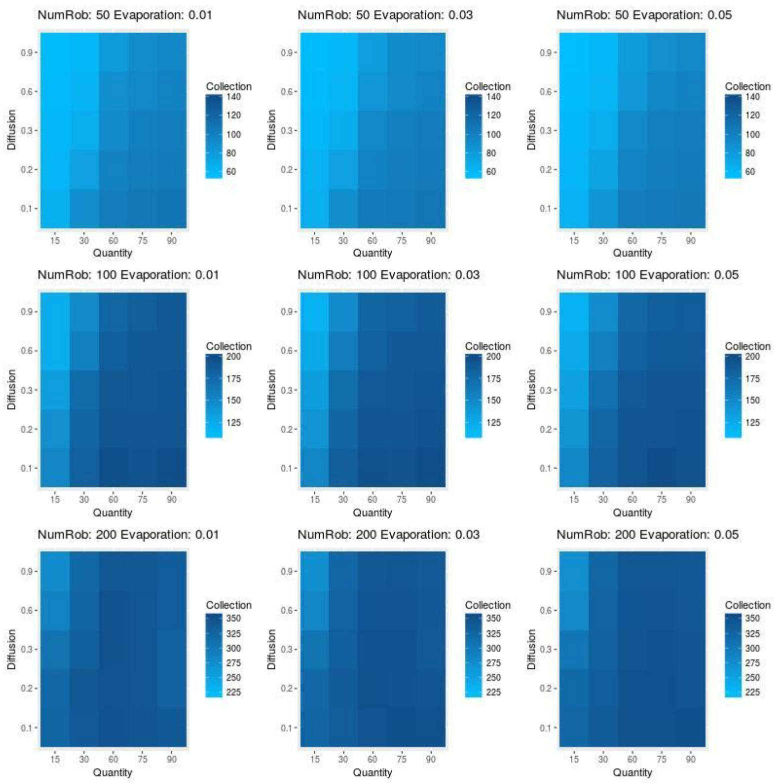


Figure 26. Two food source wide values

- Four food sources

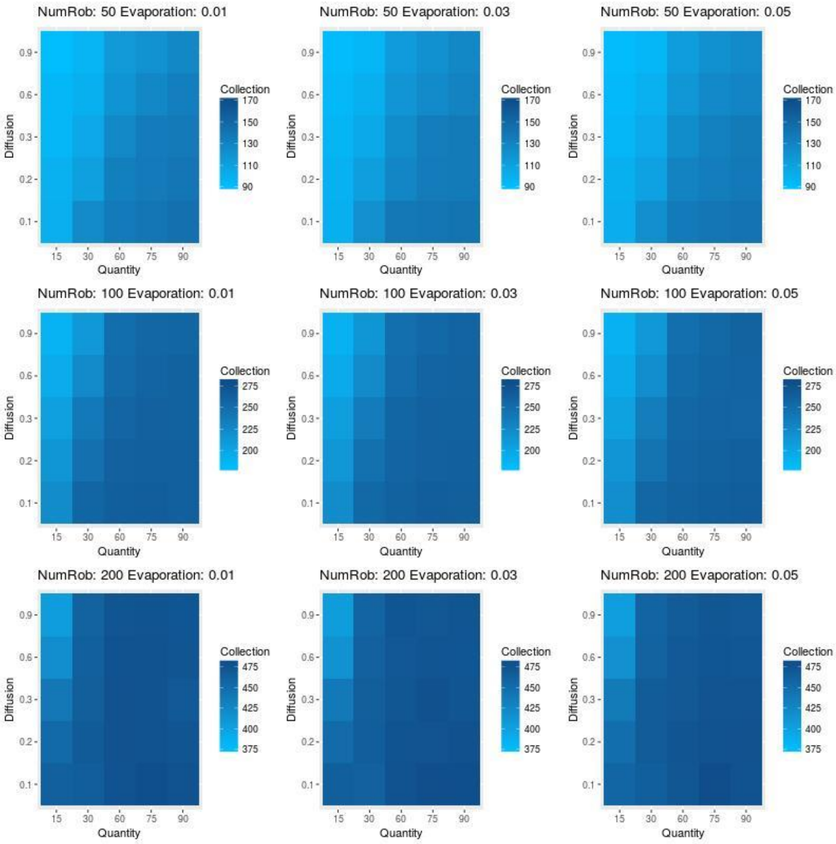


Figure 27. Four food source wide values

These plots reveal that to achieve the best collection, the evaporation used should be 0.03, the diffusion as lower as possible, 0.1, and the amount of pheromone 75. It can be evaluated that for each food source, the best parameters experiment slightly changes. For instance, for only one food source, the best amount is 60 and for four food sources the best amount is 90. That is why; a middle number has been taken. This phenomenon occurs due to the fact that when there are more food sources the robots spread more in the environment needing a higher pheromone quantity in order to keep the pheromone trail among the time.

On the other hand, the weighted food collected is not the only important value to analyse, the fact of prioritizing the highest qualities in front of the lowest ones should be consider. That is why, another plots should be done analysing the collection in the quality 10 food source divided by the total collection. This case is evaluating the best performance in order to prioritize the food source with quality of 10 over the others. This case is done for the second food source environment and the third one, as shown in the following heat maps:

- Two food sources

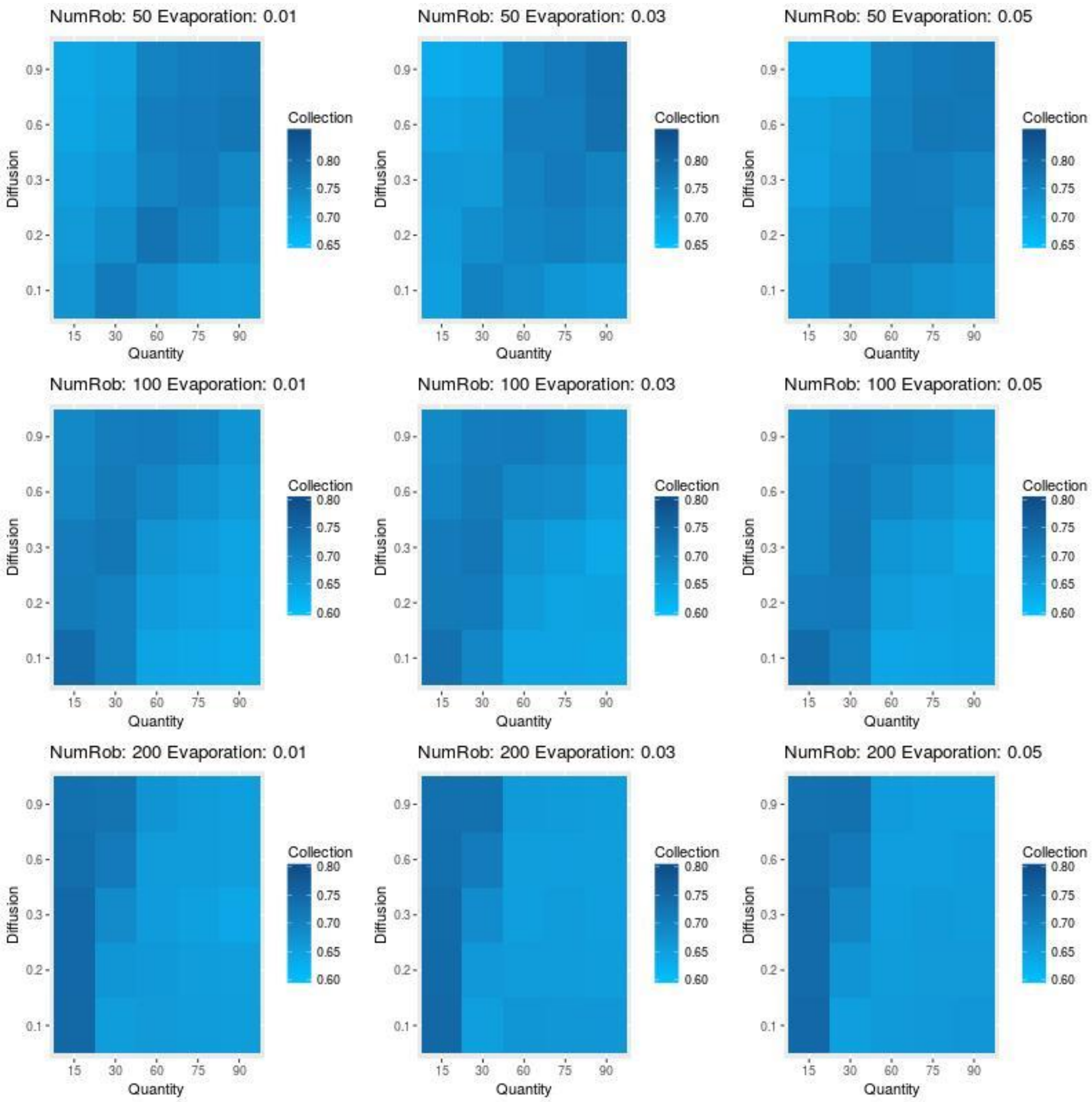


Figure 28. Ratio difference collection 2 food sources

- Four food sources

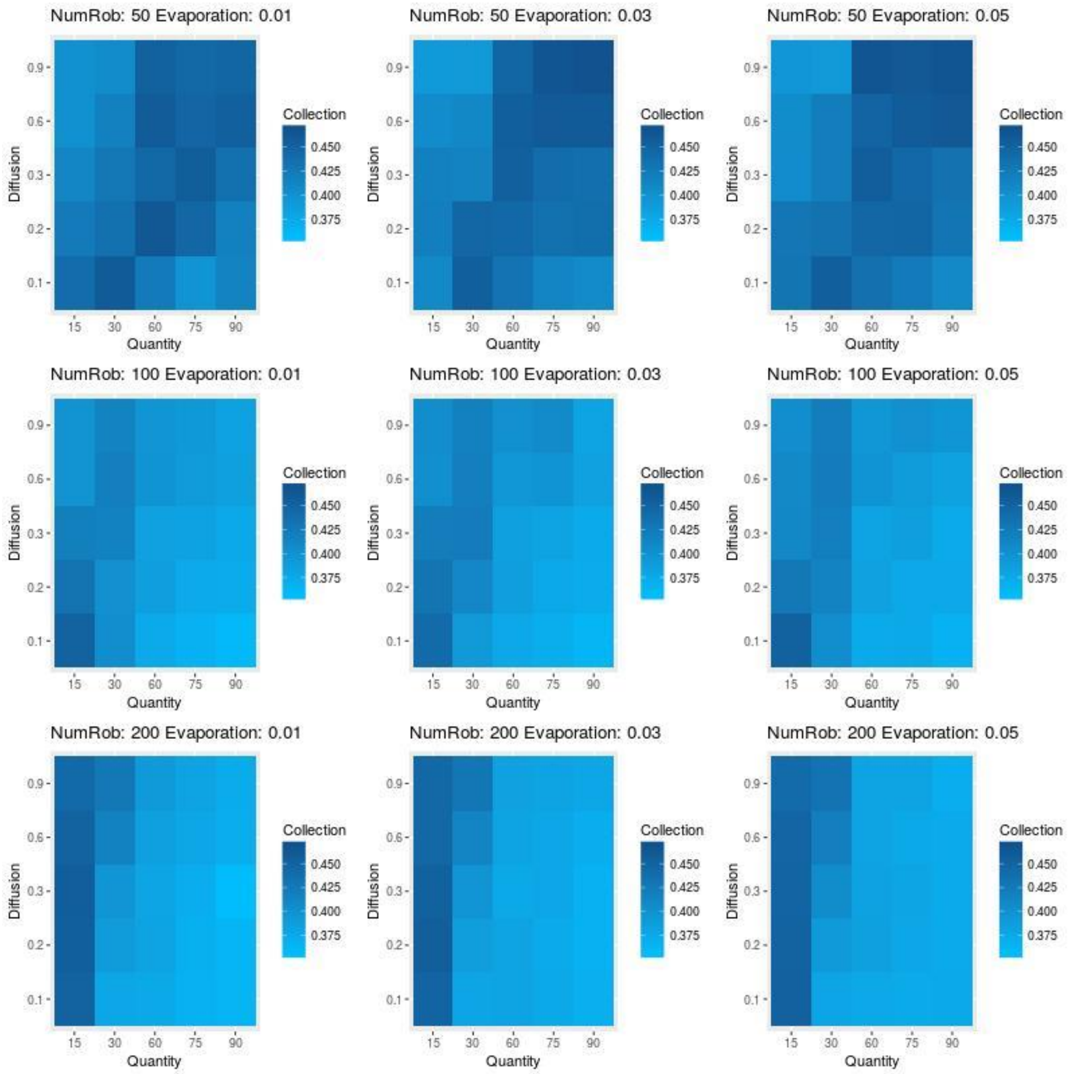


Figure 29. Ratio difference collection 4 food sources

It can be analysed that the best quality performance is varying with the number of robots, the diffusion and the quantity. Both plots, comparing the number of robots variation are creating a movement from the top right corner to the left down corner, independently of the evaporation.

This reveals that an environment with low number of robots requires pheromone parameters high in terms of diffusion (0.9) and in terms of quantity (90). When the number of robots is getting increased, the diffusion should be reduced to 0.1 and the quantity also, to a numbers of 15.

This behaviour happens due to the fact that a high number of robots spread quickly discovering all the food sources. Thus, a low quantity and low diffusion in the pheromone will be able to just keep the main trails. These main trails will be the ones produced with high qualities due to the constant floor painting.

On one hand, low number of robots requires high quantity of pheromone to keep the pheromone trail because there are fewer robots in this path to discover the food and print the trail. The use of low number of robots is already making a distinction of qualities due to the fact that all the robots cannot deal with all the food sources and the robots will prioritize easily the highest quality. In this case, it is needed to use high quantity and high diffusion in order to keep the pheromone trail. A low quantity and low diffusion is making the robots to collect food by chance, without discerning between qualities because the pheromone trail will not be able to be achieved. Therefore, they are not accomplishing a swarm behaviour based on the quality.

On the other hand, it is clear that within environments with the same food source the ratio of collection is quite similar. However, when the two food source environment is compared with the four food source environment, it can be observed a huge different ratio between them. For the environment of two food sources, the percentage of food collected for the quality 10 over the total collection is between 60% and 80%. It is quite good results with a deviation of 20%. That is why, to enable a high ratio, the choice of the pheromone parameters should be done accurately. Nonetheless, in the four sources environment the percentage collected from the food source of quality 10 is always less than 50%. That means that more than the half of the times, the food collected is coming from a poorer food source.

The chances of finding three food sources over the chances of finding the food source with quality 10 are higher. Due to this fact, during the exploration the food source that has a quality of 10 has lower chances to be discovered, while in the first case the chances between the food source with quality 10 and the food source with quality 5 are the same. In addition, the second scenario uses a quality of 8, which can create a trail of pheromone quite easy printing 80% of the pheromone each time. These two facts are producing this ratio reduction.

5.3.3. Choice and analysis of the best parameters

The information about the best performance depending on the weighted collection and the best quality performance has been found. To continue, the analysis should be focused in these two cases taking an evaporation of 0.03, the middle point of evaporation, due to the fact that the results are not varying between evaporation coefficients. The parameters chosen for each goal are summarized in the following table:

Study	Evaporation	Diffusion	Quantity
Weighted collection	0.03	0.1	75
Best quality collection	0.03	0.1	15

Table 8. Choice for the weighted collection and for the best quality collection

It is analysed how the collection of food and the distribution of the robots are changing using two food sources, one with quality 10 and the other one with qualities from 1 to 10.

What is more, it is added another constraint, the distance. In the next case, the food source of quality 10 is located further than the food source with varying quality.

This last one is located at 0.33 m from the nest while the other one is located at 0.66 m. As proven in the Distance experiment section, as further as the food source is from the nest, the collection is decreasing.

This experiment will enable to find a trade-off between quality and distance of the food source. The environment described above is the following one:

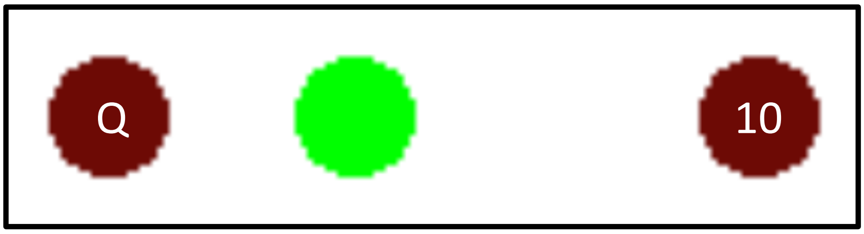


Figure 30. Quality and distance balance

The Q is representing the quality of the food source. This quality will have the following values for each swarm size (50, 100 and 200 robots) and for each quantity of pheromone (75 and 15 for each goal), to maximise the throughput and to choose the best quality food source, respectively.

This experiment will analyse where the threshold of collection is reached. Namely, in which point the food source that is close but with worse quality is collecting more food than the further good quality food source.

A plot of the weighted collection is shown in the next figure:

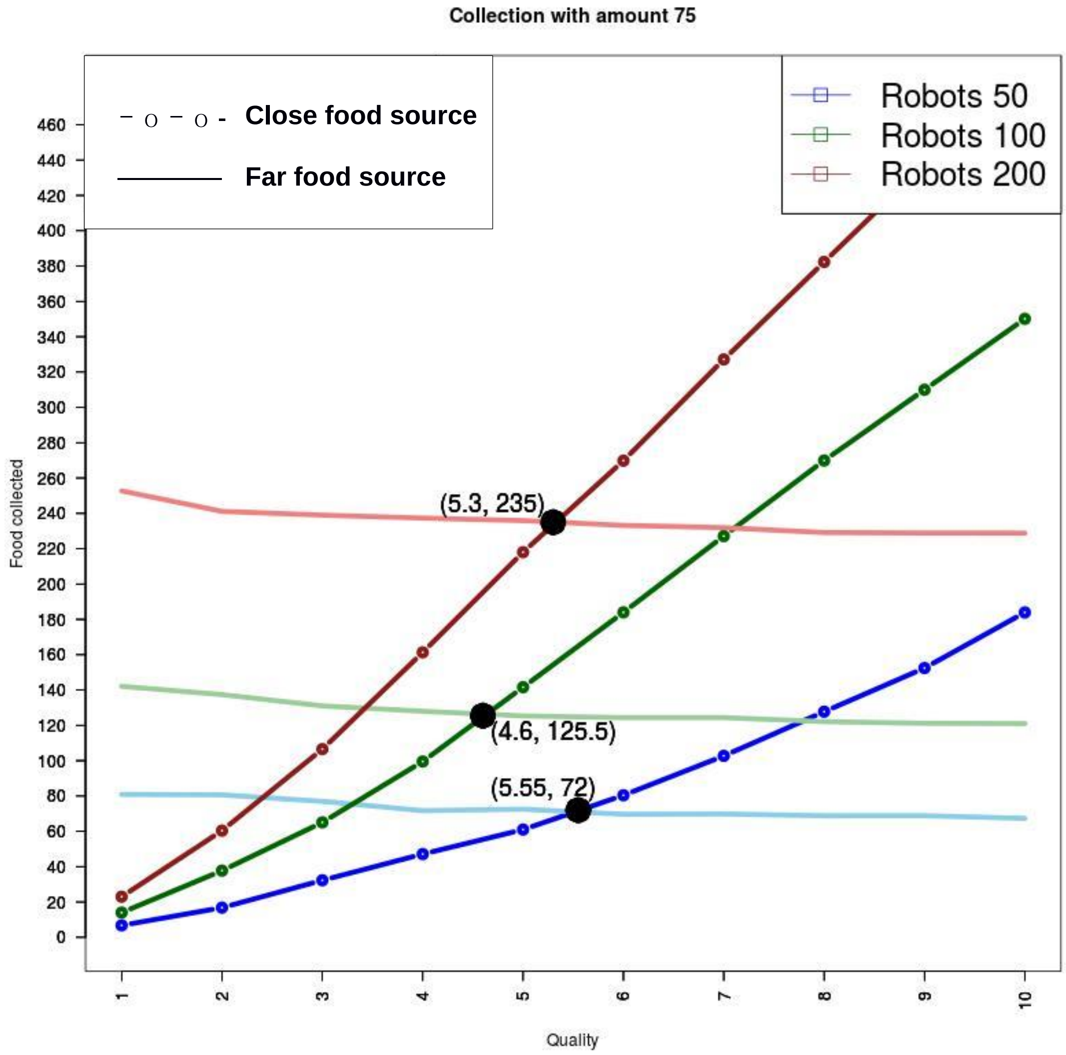


Figure 31. Collection maximizing the throughput

With dark colours, the collection of the food of the varying quality food source and with light colours the collection provided from the collection of the food source with quality of 10 is found.

It is marked in the plot the point where the close food source starts performing better. It is shown that, for close food sources with qualities less than 5, normally, the collection is higher than for the far food source of quality 10. Once it overcomes this quality threshold, the collection of the closest food source becomes higher than the furthest one.

To continue, the plot for the best quality priority choice, namely, the ones which pheromone coefficients that has better performance for quality 10, is plotted below:

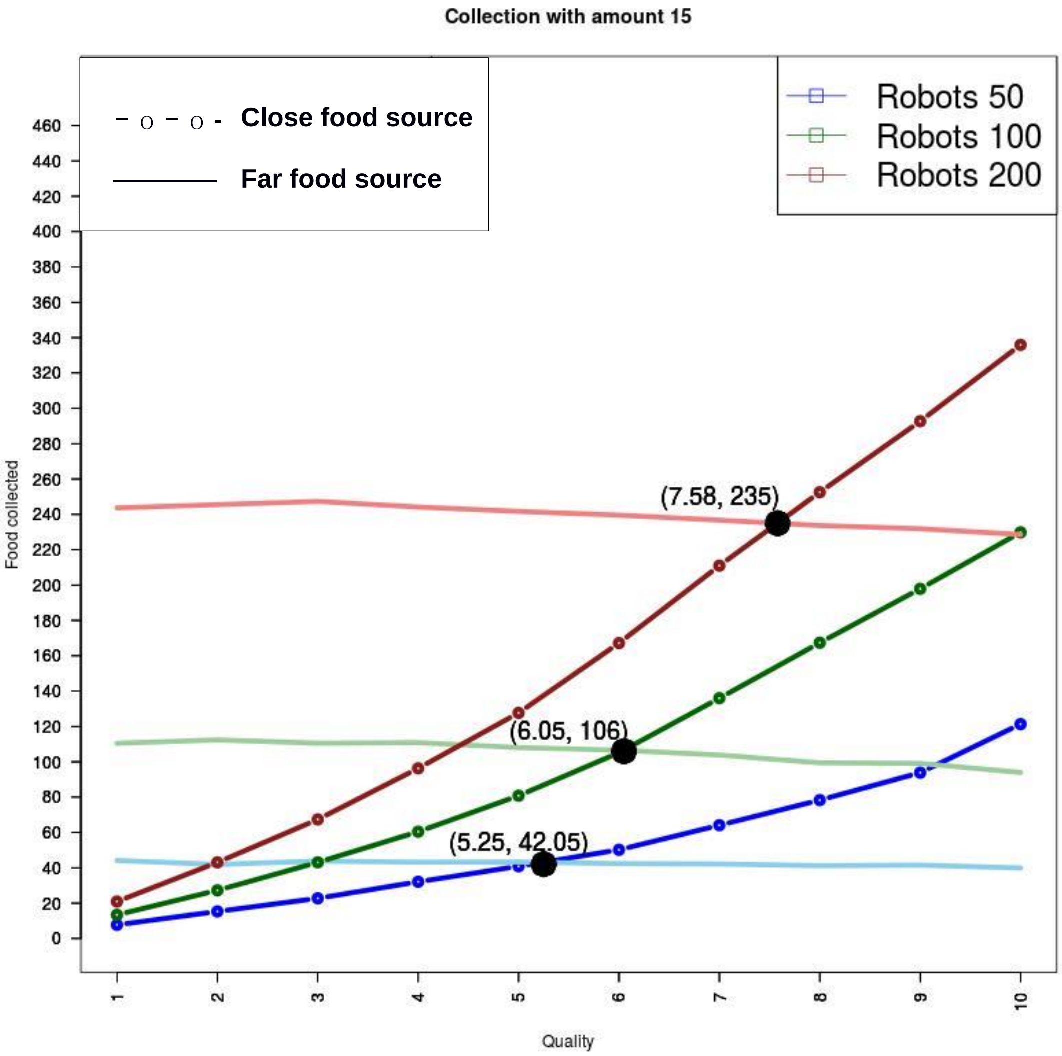


Figure 32. Collection maximising the best food source

In this plot, it is observed clearly that the threshold of the food collected now is much higher due to the fact that the parameters chosen for the environment are likely to prioritize the higher collections. In this case, the threshold is located when the qualities are higher than 5 and in 200 robots this quality threshold is higher than 7. This means that the collection from the food source that is closest only will be bigger when the qualities are 8, 9 or

10. It can be summarized that the fact of prioritizing the high qualities food sources has been achieved successfully.

On the other hand, the distribution of the robots for these two cases should be analysed in order to understand how the robots are working through this new environment. For both amounts of pheromone the same graphs have been performed:

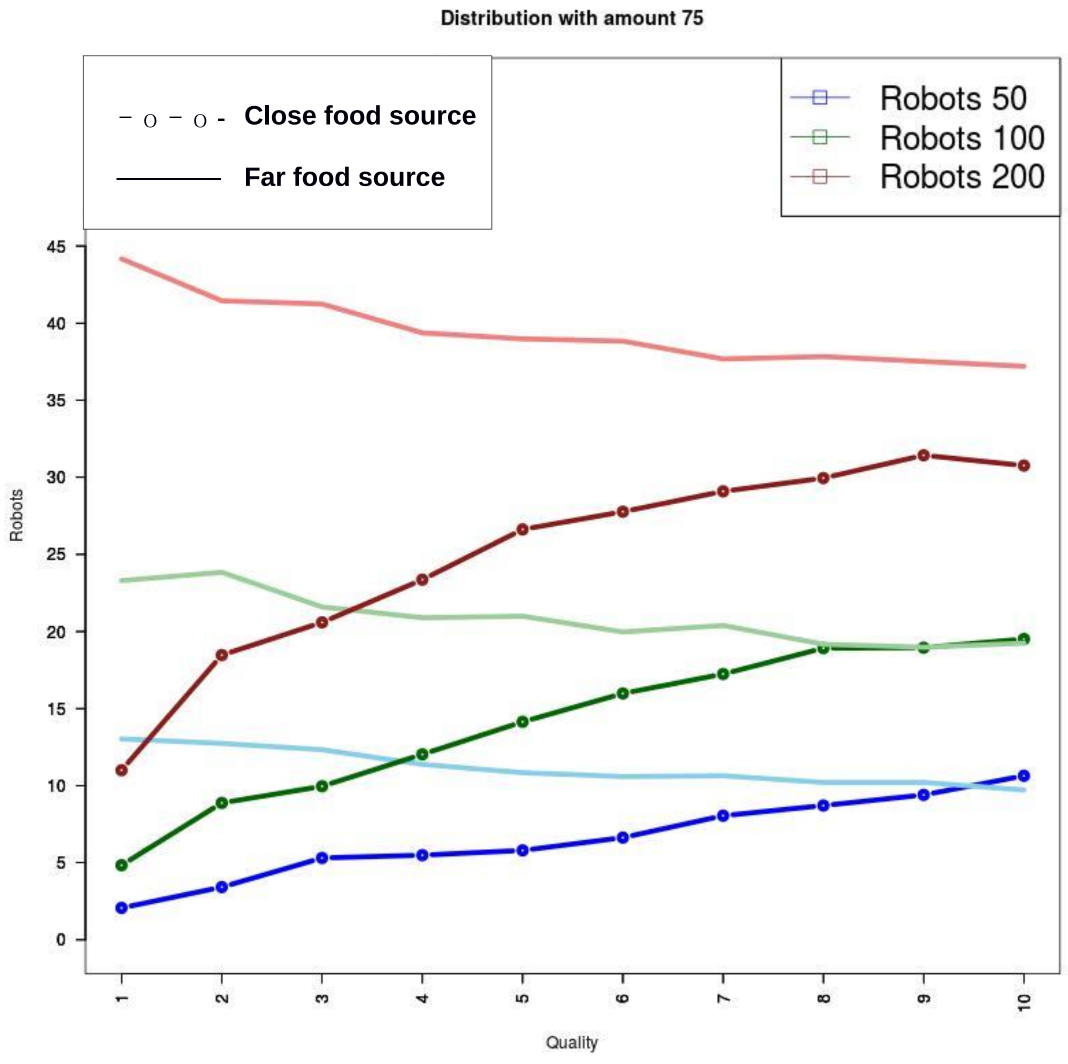


Figure 33. Distribution maximising the throughput

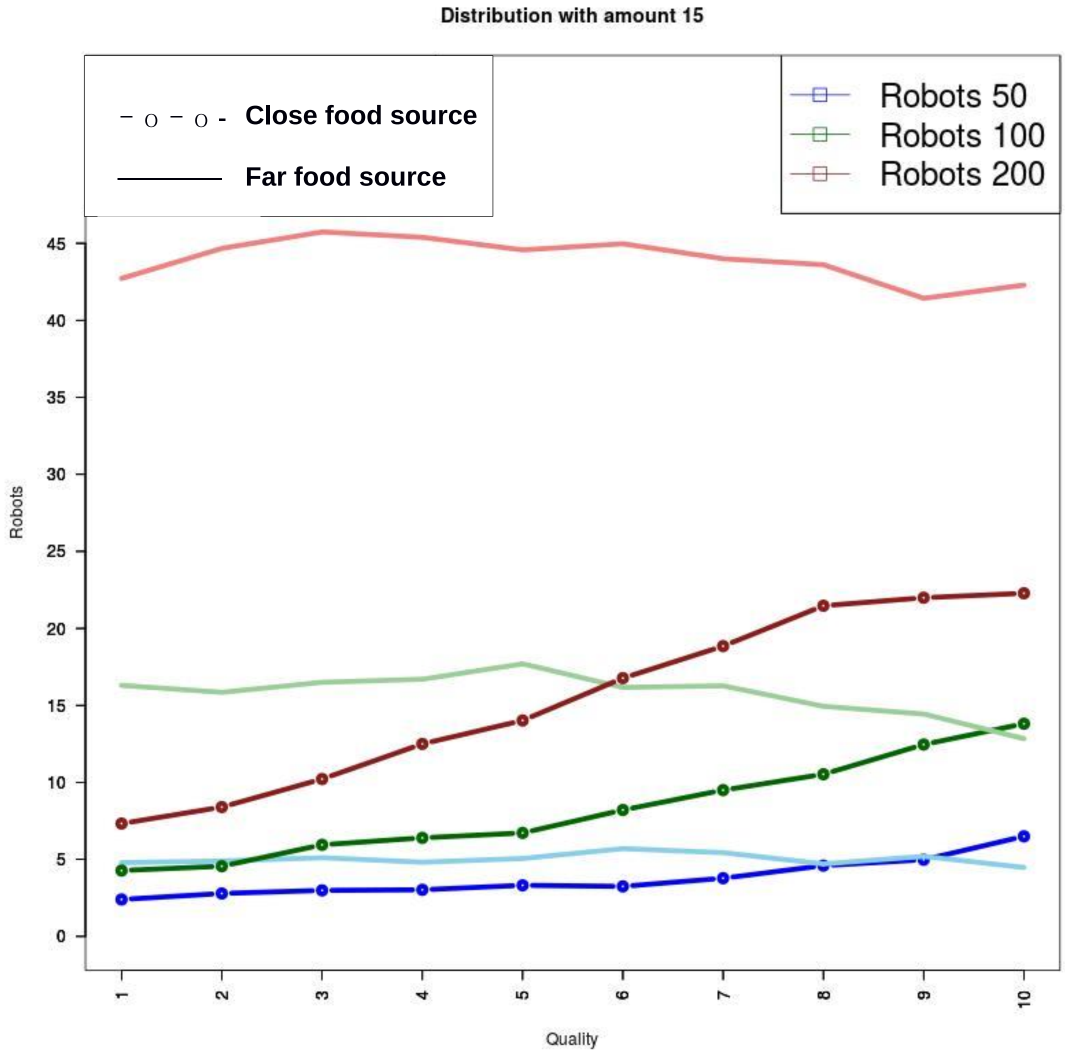


Figure 34. Distribution maximising the best option

In almost all the environment, the number of robots located in the pheromone trail between the nest and the quality of 10 is the highest one. This corroborates the analysis performed in the Distance experiment section, which was found that the furthest food source requires a higher number of robots. What is more, it is observed a higher number of these ones for an environment with 200 robots. This is because they have the chance to spread more and found the best quality and create a long path, handling a large number of robots without being crashed or lost.

5.3.4. Methods comparison

It is important the fact to consider the robustness of the current experiments. In order to prove this robustness other simple methods to analyse the environment have been used. These ones are just using other strategies to perform the foraging, changing the pheromone printing criteria presented in this paper. One of the alternative methods is the non-pheromone printing, namely, the collection of the food found by chance not for the swarm behaviour. To perform this experiment the food sources were fixed with a quality of 0. Thus, 0% of the time the robots are not printing pheromone. These results will show how many robots can find the food source just walking using a random move.

The other alternative method is the state-of-art pheromone. This method consists on printing pheromone all the time, without taking into account the quality sources. Despite this fact, once the collection is saved, the food collection will be weighted depending on the current quality. This method is going to show the ideally performance that it is expected with our method. That is why, when a quality is assigned is expected to have a total collection according to the previous equation described (

Equation 3. Weighted food collection).

Combining the three methods, a summarized table is done in order to understand each method in terms of quality relation and data classification, namely, how this food collected is weighted to distinct between qualities and which results are expected to be found for each method.

The table obtained is the following one:

	Quality used	Weighted collection	Expected
No pheromone	0	100%	Weak performance
State of art pheromone	10	Quality / Max. Quality	Linear performance
Our method	Actual quality	Quality / Max. Quality	Almost linear performance

Table 9. Method comparison

A plot comprising the three methods has been performed in order to check the performance of each method. This plot contains also, three different swarm sizes, 50, 100 and 200 robots respectively. In addition, the environment is performed in the same previous environment; with a close food source with vary quality and a far food source with quality 10. The method comparison obtained is the following one:

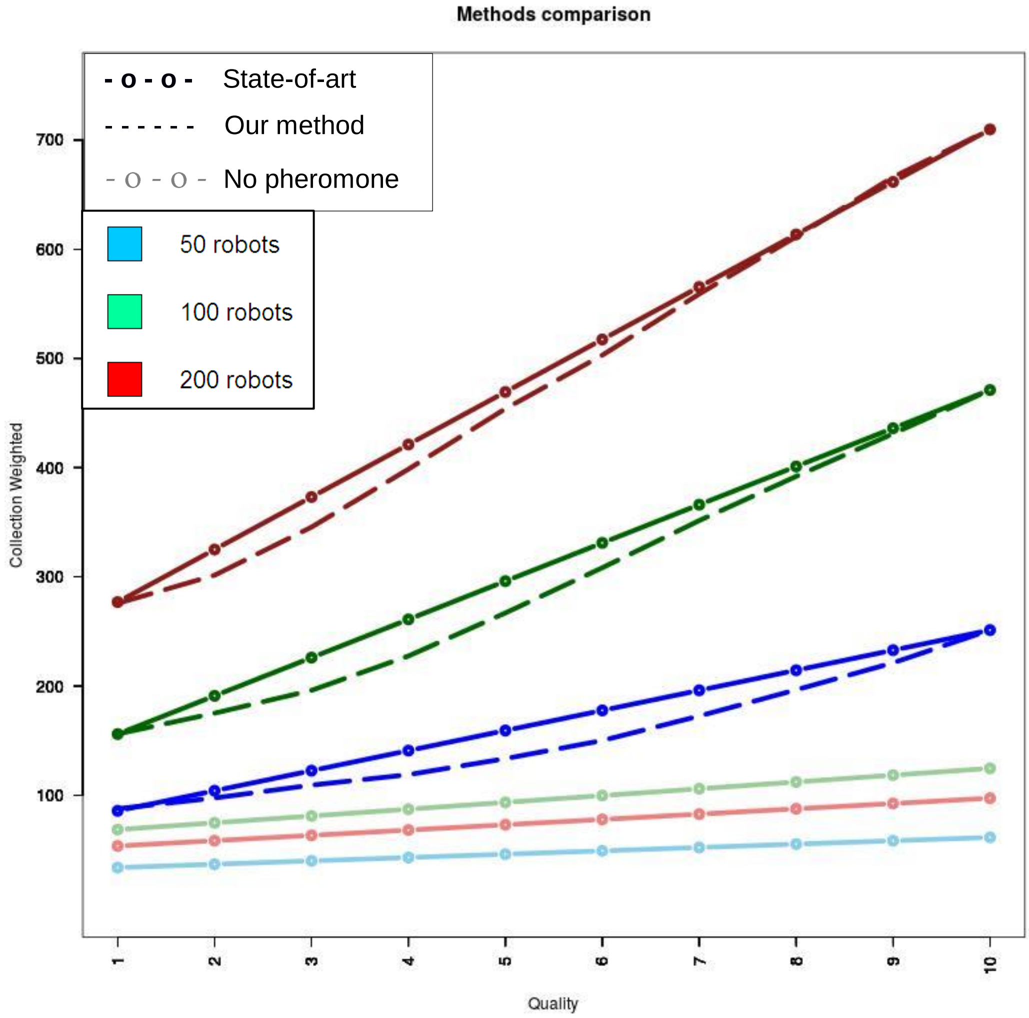


Figure 35. Methods comparison

The plot above shows the performance of the weighted collection (the total collection taking into account the quality of each food source) in the vertical axis compared with the quality of the closest food source in the horizontal axis. It is shown the different scenarios with different colours, the environment with 50 robots in blue, the one with 100 robots in green and the one with 200 robots in red.

It is clearly observed how the method of not using pheromone, the one with light colours, is the worst one. This was expected due to the fact that the robots are not able to follow a pheromone path to get to the food. Therefore, the food is found by chance.

What is more, the state of art pheromone is the one with better performance due to the fact that the collection of the low quality food source is done as a quality of 10 and the weighting is done afterwards. The linearity expected is achieved, which reveals that the collection is proportional with the quality of the food source. However, this case is not taking into account that the food source with highest collection should be prioritize in order to be as similar as possible to the real ants foraging. Namely, the pheromone should be the indicative to discern between trails of higher or lower quality food source.

The method created in this project is implemented to perform similar as the state-of-art pheromone. That is why; the printing of the pheromone has been done to reflect the current quality found. Thus, printing only pheromone in X% of the times according to the quality, a linear behaviour can be almost obtained. The lower values obtained in the method presented, in comparison with the state-of-art method, are because of the difficulty to build a complete path. In addition, the robots can crash within them and get shifted to one side when they are printing pheromone, allowing a worst path printed. Furthermore, the robots in the method presented can get lost easily in comparison with the other method.

It is observed that, with high qualities, the differences mentioned above are getting reduced because of the thicker pheromone trail. When the quality is 1, most of the collections are made by chance because of the difficulty of following such a split pheromone trail. This behaviour is observed in both methods.

It can be concluded that the method used in this work is suitable for the ants colony model.

5.3.5. Swarm size analysis

The previous simulations have been done using a limited number of robots (50, 100 and 200 robots). It is proven that the collection increases due to the increase of the number of robots. However, this increase is not proportional to the swarm size, as shown in Table 4.

Some researches such as (Couzin, Krause, James, Ruxton, & Franks, 2002) investigated this behaviour in the swarm insects and it was discovered that for each group of individuals exists an optimal group size.

It is true that increasing the group size, the benefits increase. However, the benefits while adding individuals creates a decelerating function and the costs of increasing the group creates an accelerating function leading to this optimal point, as shown in the next figure:

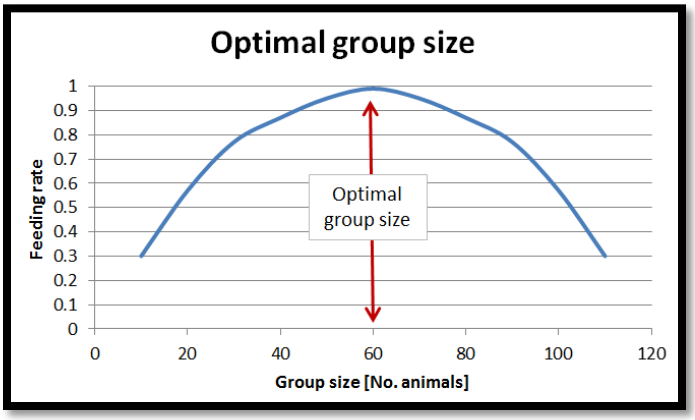


Figure 36. Optimal group size

There is one limit in each group where the costs overcome the benefits, as shown above; this optimal group size is 60 individuals in this example. It has to bear in mind that this number varies for each environment and for each type of individuals.

Due to this discovers, a wide number of different swarm sizes have been tested in order to find the best number of robots to achieve the highest collection. The parameters selected are the ones that maximize the throughput and the ones that prioritize the best food source.

First of all, the distribution of robots in the pheromone trails have been analysed for the case of prioritizing the best food source, namely, for the case with a pheromone amount of 15. This ratio distribution is calculated as:

$$Ratio\ of\ distribution = \frac{\frac{R\_Q10}{R\_Trail}}{R\_AllEnvironment}$$

Equation 4. Ratio of robots distribution

Where:

- R_Q10 = Number of robots in the pheromone trail of the quality 10 food source and the nest at the end of the experiment.

- R_Trail = Total number of robots in a pheromone trail at the end of the experiment.

- R_AllEnvironment = Total number of robots in the environment.

In the experiments, it is taken into account the number of robots in the environment in order to normalize the collection allowing a comparison within the different swarm sizes.

The first experiment is comparing how many robots are capable to be located in the pheromone trail of quality 10, in comparison with the total robots at other trails. The results obtained are the following ones:

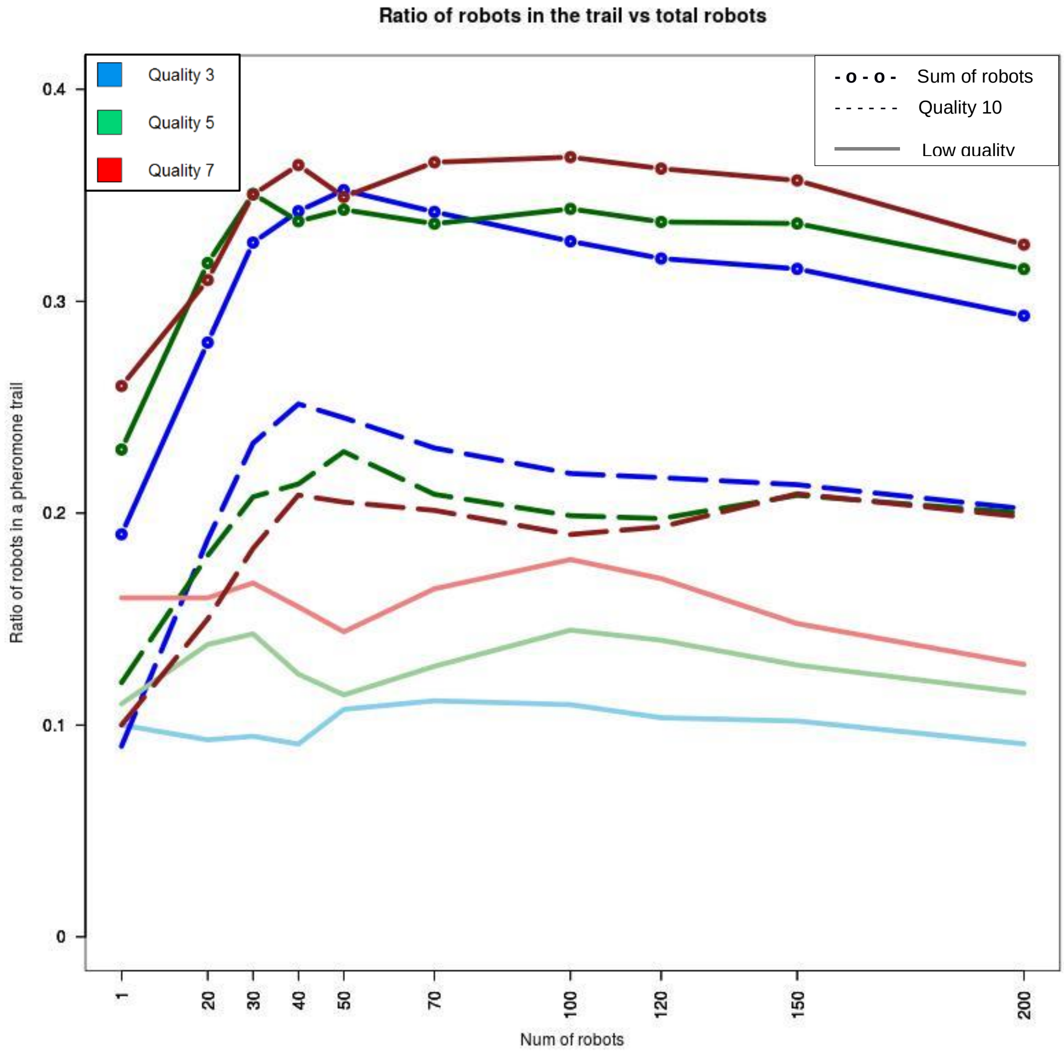


Figure 37. Ratio of robots distribution

It can be observed how the swarm size of the close food source is lower (light colour) than the robots in the far source (dashed line) corroborating the deductions of the Distance experiment section.

For a low quality food source, the number of robots in the close food source trail is the lowest one (blue line). However, this fact allows a higher number of robots in the farthest path that is the one with richest food, allowing a rich collection of food. It can be evaluated that the best number of robots for this case is between 40 and 50 robots which allows the richest collection and the most amount of robots working. Nonetheless, the ratio of robots in a path is less than 0.5, namely, half of the robots are foraging for food performing a random walk. This fact happens because of the Arena size which enables the robots to spread to the total environment.

It is observed that the minimum number of robots to achieve a good foraging is 30 robots. Once this point is reached, the number of robots in the pheromone trail remains quite constant.

It is interesting to remark the point that the far food source has a higher dependence with the number of robots. This is because two reasons. Firstly, the quality of this source is the maximum one, so, it is desirable for the robots. A low number of robots focus in the best food source because with the low food source is not possible to maintain a path capable to be followed. The other main reason is the distance; a far food source has a higher robot capacity.

Furthermore, the close food source has an optimal point around 100 robots. This value is achieved when the robots are capable to create a perfect pheromone trail from the nest to the low-quality food source and when this size is not big enough to create an overcrowding behaviour.

On the other hand, the collection of food depending on the number of robots is another fact that should be analysed in detail. In this case, two plots are created; the first one to maximise the throughput (amount of 75) and the second one to prioritize the best option selection (amount of 15). The weighted collection maximizing the throughput per robot is shown below:

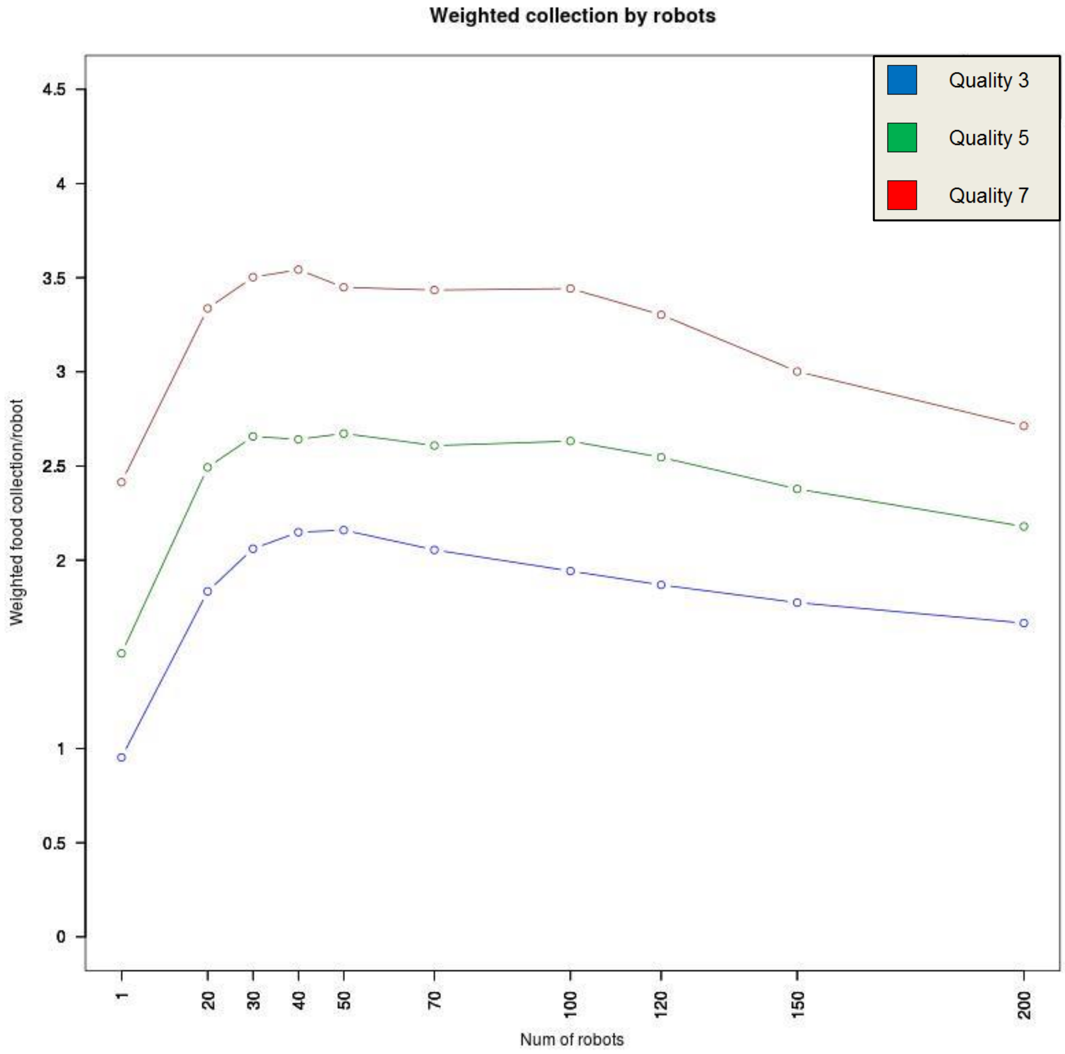


Figure 38. Weighted collection by swarm size

The plot above is showing a similar behaviour as the robots distribution, the food is weighted and added together so, as shown before, the collections is being quite stable due to the fact that, at the optimal size of the far food source the collections are done for the quality of 10 food source (when there are around 40 robots) and in 100 robots is the closest food source which is incrementing this collection. It can be evaluated how after this optimal point the collections are getting reduced.

It is also evaluated the fact that a number of robots lower than 20-30 is given a really bad performance because the robots are not able to follow the pheromone trail due to its quick evaporation.

The ratio of the collection with a quality of 10 in comparison with the total collection divided by the total number of robots in the environment in a two food source experiment with short food source of quality 2 and far food source of quality 10 is shown below:

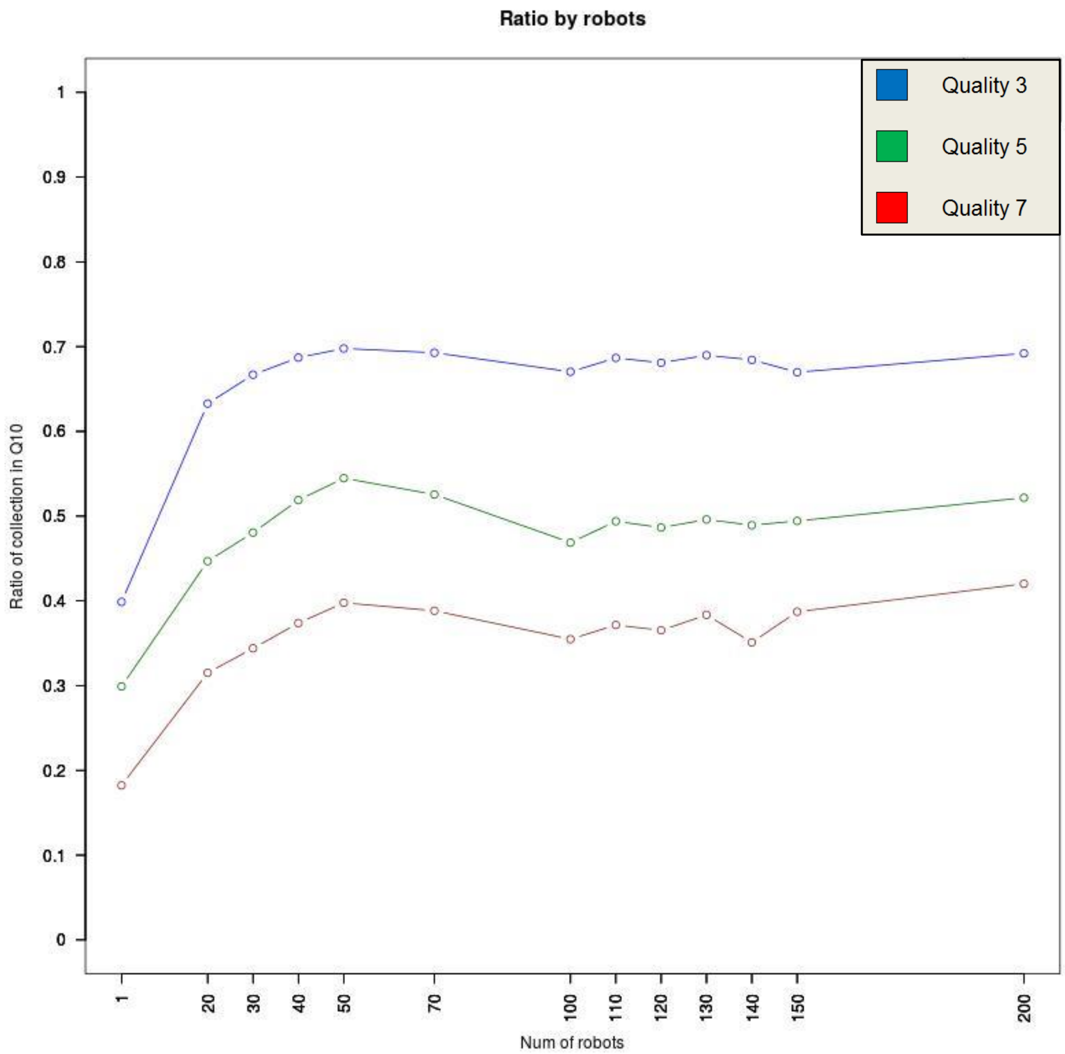


Figure 39. Ratio of collection in Q10 by swarm size

It is observed that reducing the quality of the close food source, the second optimal point has been almost reduce in the total collection, achieving a stable line, once the first optimal point is overcome.

To sum up, it can be concluded that there is not a certain number of robots that can be applied for all the experiments and all the goals in order to have the most efficient system. It has to bear in mind that the environment is always determining whether or not a high number of robots should be applied. Despite this fact, it is true that each individual food source, depending on the environment, has an optimal swarm size in order to maximise the throughput and the number of robots collecting food.

5.3.6. Execution time

The execution time used in all the experiments is around 20 minutes; to be precise 1000 ticks that are equal to 16.67 minutes.

This value has been chosen after checking the distribution of the robots along this time. Despite this fact, a deeply analysis should be done to evaluate how this collection increases over the time.

Using the same scenario as before, the collection has been recorded every 50 milliseconds for each of the food sources. The results using 100 robots, a far food source of quality 10 and a close food source of a varying quality are the following ones:

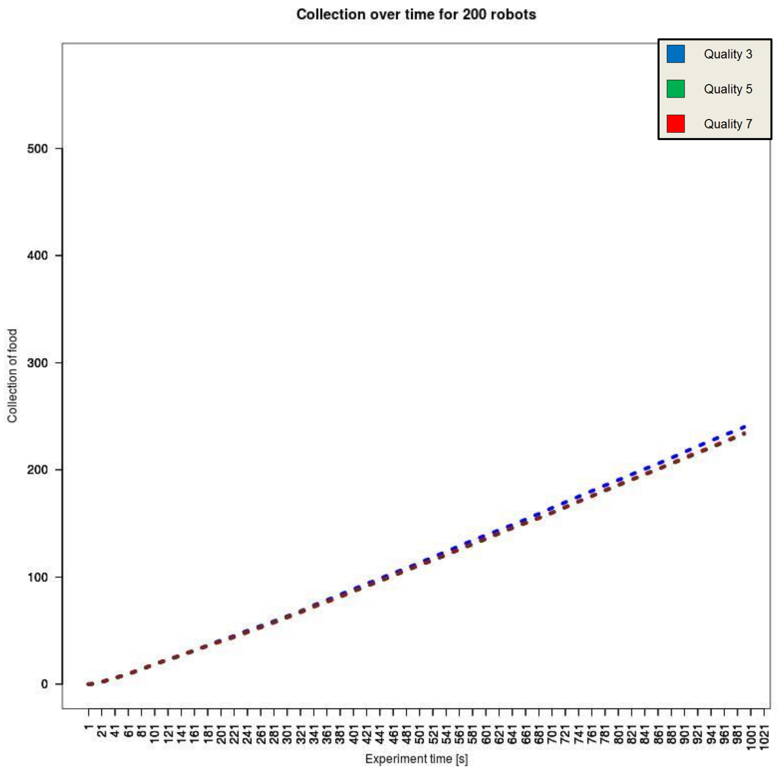


Figure 40. Collection vs time in Q10

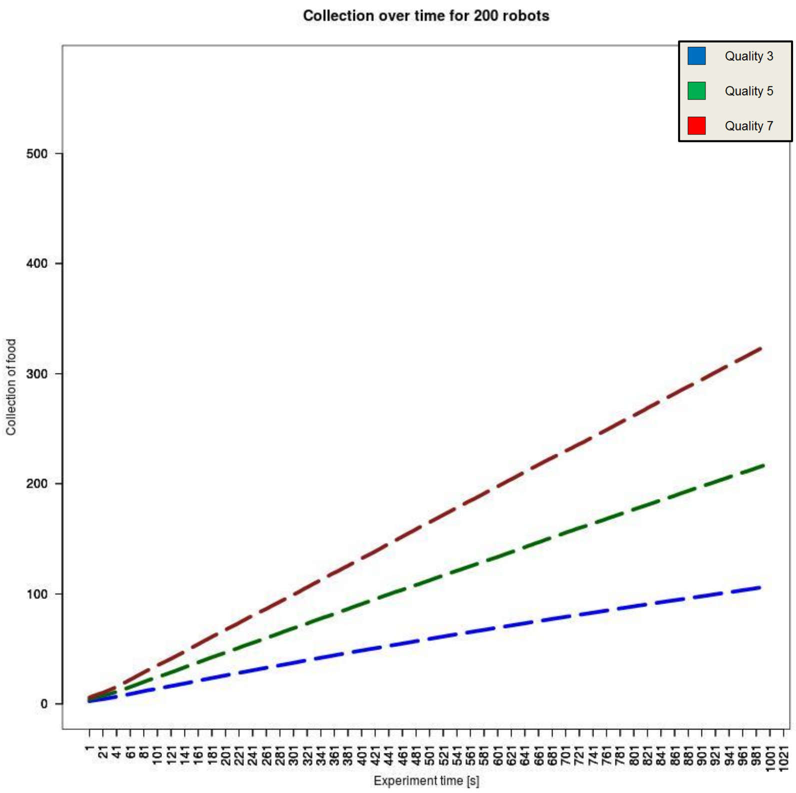


Figure 41. Collection vs time in the close food source

Analysing the plots, some assumptions are extracted. It can be observed how the collections increase over the time. In Figure 40, the collections over the time made by the far food source of a quality of 10 are showing that there exists a period from 0 to 30 seconds where the collection stills around 0. On the contrary, in Figure 41 this time is slightly lower. This phenomenon is because the robots find easier to find the closest food source when they are foraging.

It can be observed how the collections of the far food source do not altered for the different parameters in the quality of the close food source in almost all the cases. It is true that the blue line, the quality of 3 is showing slightly higher collections for the quality of 10 but between the quality 7 and 5 this variations are not perceived.

Another interesting fact observed in Figure 41 is the comparison between the different qualities. It can be observed how the difference between collections for each quality is getting higher over the time. This assumption reveals that a long execution time is allowing a better quality discern.

6. Critique

Once the project is completed, it can be affirmed that the achievements purpose in the different task has been achieved. However, some challenge had been overcome making the performance of the project longer and more difficult.

There were some unexpected constrains found during the project. These ones are mainly focused in the real implementation part due to the fact that the laboratory is really complex and the setup of the system is not an easy task. Some of the main unexpected difficulties are the fact that the robots are difficult to be programed, can lose its identification, can work in a wrong manner if the battery is not charged enough, its position is not always track for the cameras allowing a non-printing of the pheromone in some areas and the robots has to be constantly recalibrated. What is more, the author had to deal with the Kilobots memory that is really limited so, the program should be adapted in order to fit in each Kilobot.

Definitely, after analysing the point mentioned above related about the Kilobots constrains, it can be confirmed that the most challenging part is the adaption of the simulation to be performed in the real world with a huge amount of Kilobots. There are many constraints that should be checked for each of the robots including the battery level (a robot with a low battery level is not turning as fast as a full battery one and also, if this is so low, can cause some strange walking behaviour), their calibration should be done robot by robot in order to fix the left and right motor to the same level and the orientation of the robot in the moment of the ID assign can be not performed when the LED is not visible for the camera and the assign of the IDs should be done again until all the robots are found.

On the other hand, during the analysis of the experiments some difficulties have been appeared. Find the correlation of the environment parameters have been a hard task. It has been very controversial the discussion of which parameters are more suitable to be compared, which range of values the parameters should have and which the better graphical expression of the data is. A lot of parameters have been varied to obtain the best performance such as the number of robots, the diffusion, the evaporation, the quantity, the food sources, the quality and the distance.

There is an important constraint found during the execution of the experiments. This is the time that an experiment last. When a high number of experiments are performing, the view is deactivated to decrease the experiment time. However, for high number of experiments as the last one which implies 100 experiments for each parameter variation, which means, 32,400 experiments, just deactivate the visualization is not enough. The experiment in a computer lasts six days. Due to this fact, a computer cluster has been used. This cluster is called Iceberg and it is capable to run the same experiment in less than six hours. The problem related with the cluster is that requires an authorised account to get into and the ARGoS platform has to be installed in the cluster, including the related libraries. Furthermore, the program has been adapted to fit in this ARGoS due to the compilation is done different than in the ARGoS installed in the computer.

This project has been an important tool to increase the author's technical skills while expanding the authors' knowledge in the swarm robotics fields.

Many platforms have been used to achieve the aims of the project as R, ARGoS, Bash, HTML, QT and C++.

In addition, this project brought collaboration between the Sheffield Hallam University and the University of Sheffield in order to unify both knowledge and obtain such a successful project that creates an advance in the swarm robots fields. Due to the results obtained during this research, an article may be published in the ANTS conference (Birattari & Trianni, 2018). This conference is a Swarm Intelligence conference held in Rome from 29th to 31th of October in 2018.

There is no costing of the project because the laboratory, the computer, the Kilobots and the ARK system were provided forming the Robotics department and no extra purchases were needed. What is more, the programs used are open-source so; it has been obtained for free.

Comparing the real timings with the ones proposed at the start, there is a delay in the testing and programing section due to the fact that it was not taking into account the hard work of transforming the simulation program to the real execution and the performance of the huge amount of experiments in the laboratory.

Despite this fact, the report writing task was done in less time than expected. Thus, the schedule of the project was fitted in the desired timing. However, the article is not included in this schedule because it was not planned until this last month. The Gant chart is shown in the next figure:

Task Name	Start Date	End Date
Programming	**18/07/17**	**15/12/17**
Familiarize with ARGOS	18/07/17	27/07/17
Program basics	30/07/17	08/09/17
Program pheromon trails	10/09/17	29/09/17
Program sensitive decisions	02/10/17	03/11/17
Program improvements	06/11/17	15/12/17
Literature review	**17/07/17**	**15/12/17**
Background literature	17/07/17	08/09/17
Related literature	11/09/17	15/12/17
Testing	**04/09/17**	**02/03/18**
Program tests	20/11/17	22/12/17
Improvements tests	25/12/17	02/02/18
Report writing	**04/09/17**	**02/03/18**
Writing	04/09/17	29/12/17
Reviewing	01/01/18	02/03/18

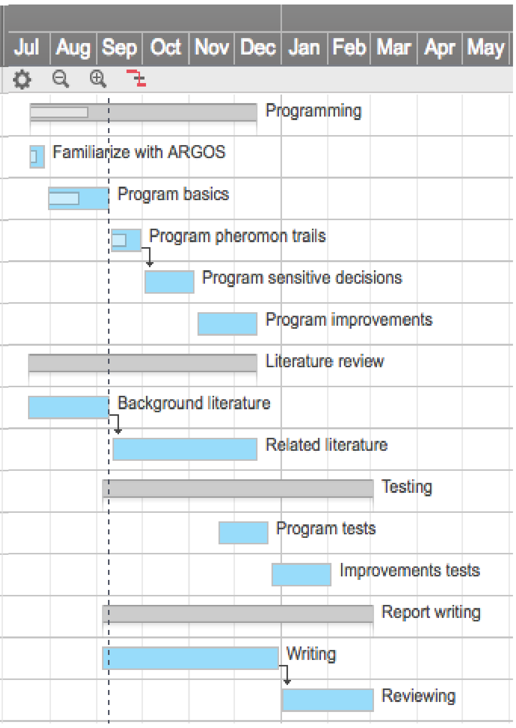


Figure 42. Gant chart

During the realization of the thesis, the project management skills of the author's improved sharply including the academic writing because of this report, the communication skills due to the constant meetings and the presentation of this work in both universities and the presentation skills because of the requirements in the report, presentation and article.

However, some limitations of time have been found due to the fact of combining classes and the project. Furthermore, there were some constraints in the mobility because of the constant working in both universities and the limitations of opening hours of the robotics laboratory of the University of Sheffield.

7. Conclusions and further work

After analysing the points mentioned above, it can be concluded that using a swarm robotic strategy the stigmergy behaviour of the ants can be implemented with a simple and low cost technology, the Kilobots.

It has been proven that there is not a unique and efficient environment that is the best one for all different experiments. Each situation should be evaluated individually in order to achieve the best performance, taking into account the coefficients of the pheromone, the environment and the number of workers able to forage in the experiment, agreeing with the discussion provide in (Garnier, Tâcheb, Combe, Grimal, & Theraulaz, 2007).

Once the project is finished, it can be declared that the goals proposed in the Introduction section has been achieved in an accurate, reliable and robust manner. Each of the goals has been achieved firstly, in the simulation field and previously, in the real experiments. However, more experiments using real robots are being conducted.

The fact that the closest food source is the one that needs less robots and achieves a better collection has been proven in the Distance experiment where, running hundreds of experiments and using three food sources with different locations but same qualities, the closest food path has been reveal in all the tests to have the best performance.

On the other hand, the quality of the food source has been evaluated deeply, in the first instance, just comparing similar distances in an environment composed for four food sources, as shown in the Quality experiment, and secondly, deeply analysing each quality.

In addition, a trade-off between the distance and the quality has been performed afterwards using different swarm sizes and pheromone coefficients. Evaluating this experiment, it is proven that there exists a real balance between these two factors. This experiment helps to understand how the distance and the quality values are distributing the swarm robots through the environment in order to achieve the best collection taking into account two different goals.

The first one is the fact of maximising the throughput weighting the food and achieving the highest collection for each environment choosing the best environment coefficients previously analysed. The second one is the fact that the best option can be prioritized using a certain amount of pheromone allowing the robots to forage further from the nest and explore better food sources.

One of the key points of the project that has been analysed is the importance of how each parameter choice in the experiment can affect the distribution and collection and also, in which situations these ones should be incremented or decremented. It has been analysed how the robots can really create a higher collection when the swarm size gets bigger, but it has been proved that there is a point where the environment becomes overcrowded, decreasing the performance.

In addition, the increase of the swarm size in the animals' world has been thoroughly analysed by some researchers as (Davies, Krebs, & West, 2012). This proves benefits in foraging terms when the size is increased but also has negative consequences that cannot be contemplate in this work as the disease or the food distribution in the nest when the swarm group overcome a maximum limit.

On the other hand, the amount of pheromone that helps the robots to achieve a better following of the pheromone trail in order to get to the food source is obtained. However, this is creating a negative effect when the amount is increased too much due to the fact that it is not allowing the robots to discern between a high quality food source and a low one.

The evaporation coefficient is a tool to overcome the problem explained before by increasing its value. However, these values should be chosen accurately in order to avoid a non-construction of a pheromone trail. What is more, the diffusion is a remarkable pheromone coefficient that allows the pheromone to spread through the environment to increase the path trail allowing more robots to follow a pheromone trail. It has been discovered that this value can be maximized to a certain point. If this limit is overcome, the spread can generate a slow move of the robots to the food source through the thick path created. What is more, a high number of diffusion is producing a faster reduction of the pheromone in the initial point due to its spread to other points.

This work is making an improvement in the scientific world and concretely, in the swarm fields due to the capability of comprising a wide variation of the parameters in the environment, creating a huge amount of deductions such as the ones previously discussed in papers (Deneubourg, et al., 1991) and (Goss, Aron, Deneubourg, & Pasteels, 2009), such as the fact of an implementation of an autonomous robots behaviour with just local knowledge achieving a collective behaviour supported by a certain swarm size.

Furthermore, the ants' colony model, previously defined in (Dorigo, Birattari, & Stutzle, Ant Colony Optimization, 2006), has been implemented as close as possible to the ants behaviour.

Despite all the achievements so far, some further work could still be done. The analysis of the food sources can be widely explored, for instance, increasing the food sources and checking the behaviour with too much sources or using one food source with a lot of robots to check the performance in an overcrowded scenario. What is more, the pheromone coefficients are based on the well-known equation of (Nakamura & Kurumatani, 1997) and can be improved by adding levels of complexity.

Another piece of further work that could be evaluated is the analysis of this technology for other robots such as the E-puck robots, the Bot robots, among others, in order to prove the reliability of the system designed, as done in the (Nouyan, Campo, & Dorigo, 2007) paper.

On the other hand, the real implementation could be performed with more experiments and higher numbers of robots in order to contrast with the huge amount of simulations made using ARGoS. However, the ARK system explained in (Reina, Cope, Nikolaidis, Marshall, & Sabo, 2017) of the University of Sheffield is currently in development and will be improved to enable faster and more reliable simulations in the near future.

Overall, the study of the swarm foraging based on the ants' colony behaviour has been extensively analysed, achieving coherent conclusions using a consistent system. This thesis enables a contrast of the different environments and particularly in the innovation of the trade-off comparison between the distance and the food quality resources in the environment.

References

Arvin, F., Yue, S., & Xiong, C. (2015). Colias-Φ: An Autonomous Micro Robot for Artificial Pheromone Communication. *International Journal of Mechanical Engineering and Robotics Research*.

Birattari, M., & Trianni, V. (2018). *ANTS 2018*. Retrieved from Swarm-intelligence.eu: http://www.swarm-intelligence.eu/ants2018/

Bonani, M., Raemy, X., Pugh, J., Mondana, F., Cianci, C., Klaptocz, A., . . . Martinoli, A. (2009). The e-puck, a robot designed for education in engineering. . *Proceedings of the 9th Conference on Autonomous Robot Systems and Competitions, vol. 1*.

Boogert, N. J., Fawcett, T. W., & Lefebvre, L. (2011). Mate choice for cognitive traits: a review of the evidence in nonhuman vertebrates. *Behavioral Ecology*.

Brambilla, M., Ferrante, E., Birattari, M., & Dorigo, M. (2013). Swarm robotics: a review from the sarm engineering perspective. *Swarm Intell*.

Bunn, A., & Korpela, M. (2014). Crossdating in dplR.

Caprari, G., Balmer, P., Piguet, R., & Siegwart, R. (1998). The autonomous micro robot "Alice": a platform for scientific and commercial applications. *Micromechatronics and Human Science, 1998*.

Chan, F., & Kumartiwari, M. (2007). *Swarm Intelligence, Focus on Ant and Particle Swarm Optimization.*

Chen, C. P., & Zhang, C.-Y. (2014). Data-intensive applications, challenges, techniques and technologies: A survey on Big Data. *Elsevier*.

Ciupa, I., Leitner, A., Oriol, M., & Meyer, B. (2008). ARTOO: Adaptive Random Testing for Object-Oriented Software. *ICSE '08 Proceedings of the 30th international conference on Software engineering*.

Couzin, I., Krause, J., James, R., Ruxton, G., & Franks, N. (2002). Collective Memory and Spatial Sorting in Animal Groups. *Journal of Theoretical Biology, 218(1), pp.1-11.*

Davies, N. B., Krebs, J. R., & West, S. A. (2012). *An introduction to behavioural ecology.* Wiley-Blackwell.

Deneubourg, J. L., Pasteels, J. M., & Varhaeghe, J. C. (1982). Probabilistic Behaviour in Ants: A Strategy of Errors? *Journal of Theoretical biology*.

Deneubourg, J., Goss, S., Franks, N., Sendova-Franks, A., Detrain, C., & Chretien, L. (1991). The dynamics of collective sorting robot.

Deutsch, A., Brusch, L., Byrne, H., Vries, G. d., & Herzel, H. (2007). *Mathematical Modeling of Biological Systems, Volume I: Cellular Biophysics, Regulatory Networks, Development, Biomedicine, and Data Analysis.* Springer Science & Business Media.

Dorigo, Birattari, & Stutzle. (2006). Ant Colony Optimization. *Iridia*.

Dorigo, M., Birattari, M., & Stutzle, T. (2006). Ant Colony Optimization. *Iridia*.

Dorigo, M., Floreano, D., Gambardella, L. M., Mondada, F., Nolfi, S., Baaboura, T., . . . Decugniere, A. (2013). Swarmanoid: A Novel Concept for the Study of Heterogeneous Robotic Swarms. *IEEE Robotics & Automation Magazine (Volume: 20, Issue: 4, Dec. 2013)*.

Ekman, G. E. (1959). Weber's law and related functions. *Journal of Psychology*.

English, S., Gough, J., Johnson, A., Spanton, R., Sun, J., Crowder, R., & Zauner, K.-P. (2008). Strategies for Maintaining Large Robot Communities. *ALIFE XI.*

Fujisawa, R., Dobata, S., Sugawara, K., & Matsuno, F. (2014). Designing pheromone communication in swarm robotics: Group foraging behavior mediated by chemical substance. *Swarm Intelligence*.

Garnier, S., Combe, M., Jost, C., & Theraulaz, G. (2013). Do Ants Need to Estimate the Geometrical Properties of Trail Bifurcations to Find an Efficient Route? A Swarm Robotics Test Bed. *PLOS*.

Garnier, S., Guérécheau, A., Combe, M., Fourcassié, V., & Theraulaz, G. (2009). Path selection and foraging efficiency in Argentine ant transport networks. *Behav Ecol Sociobiol*.

Garnier, S., Tâcheb, F., Combe, M., Grimal, A., & Theraulaz, G. (2007). Alice in Pheromone Land: An Experimental Setup for the Study of Ant-like Robots. *Proceedings of the 2007 IEEE Swarm Intelligence Symposium (SIS 2007)*.

GitHub. (2018). *Build software better, together*. Retrieved from https://github.com/.

Goss, S., Aron, S., Deneubourg, J. L., & Pasteels, J. M. (2009). Self-organized Shortcuts in the Argentine Ant. *Springer-Verlag*.

Gutierrez, A., Campo, A., Dorigo, M., Donate, J., Monasterio-Huelin, F., & Magdalena, L. (2009). Open E-puck Range & Bearing miniaturized board for local communication in swarm robotics. *IEEE International Conference on Robotics and Automation* .

Heck, P. S., & Ghosh, S. (2000). A Study of Synthetic Creativity: Behavior Modeling and Simulation of an Ant Colony. *Arizona State University*.

Herianto, Sakakibara, T., & Kurabayashi, D. (2007). Artificial Pheromone System Using RFID for Navigation. *Science Direct*.

Hettenhausen, J., Lewis, A., & Mostaghim, S. (2010). Interactive multi-objective particle swarm optimization with heatmap-visualization-based user interface. *Engineering Optimization*.

Jackson, D. E., Holcombe, M., & Ratnieks, F. L. (2004). Trail geometry gives polarity to ant foraging networks. *Nature*.

Kazama, T., Sugawara, K., & Watanabe, T. (2006). Traffic-like Movement on a Trail of Interacting Robots with Virtual Pheromone.

Kernbach, S. (2013). *Handbook of Collective Robotics: Fundamentals and Challenges.* Boca Raton, Fla.: Pan Stanford.

Kernbach, S., Thenius, R., Kernbach, O., & Schmickl, T. (2009). Re-embodiment of Honeybee Aggregation Behavior in an Artificial Micro-Robotic System . *SAGE*.

Kim, D. H., Wang, H., & Shin, S. (2006). Decentralized Control of Autonomous Swarm Systems Using Artificial Potential Functions: Analytical Design Guidelines. *Journal of Intelligent and Robotic Systems*.

Knaden, M., & Wehner, R. (2005). Nest mark orientation in desert ants Cataglyphis: what does it do to the path integrator? *Animal Behaviour*.

Krieger, M. J., Billeter, J.-B., & Keller, L. (2000). Ant-like task allocation and recruitment in cooperative robots. *Nature*.

Kube, C. R., & Bonabeau, E. (2000). Cooperative transport by ants and robots.

Marinakis, Y., & Marinaki, M. (2009). A Hybrid Multi-Swarm Particle Swarm Optimization algorithm for the Probabilistic Traveling Salesman Problem. *Computers & Operations Research*.

Mayet, R., Roberz, J., Schmickl, T., & Crailsheim, K. (2010). Antbots: A Feasible Visual Emulation of Pheromone Trails for Swarm Robots. *Springer, Berlin, Heidelberg*.

Morgan, E. D. (2008). Trail pheromones of ants. *Physological Enthomology*.

Nakamura, M., & Kurumatani, K. (1997). Formation mechanism of pheromone pattern and control of foraging behavior of an ant colony. *Artificial Life V*.

Nouyan, S., Campo, A., & Dorigo, M. (2007). Application of Supervisory Control Theory.

Payton, D., Daily, M., Hoff, B., Howard, M., & Lee, C. (2001). Pheromone Robotics. *Autonomous Robots*.

Pearce-Duvet, J. M., Elemans, C. P., & Feener, D. H. (2011). Walking the line: search behavior and foraging success in ant species. *Behavioral Ecology*.

Pinciroli, C. (2017). *argos3-kilobot*. Retrieved from Github.

Pinciroli, C., Trianni, V., Grady, R. O., Pini, G., Brutschyt, A., Brambilla, M., . . . Dorigo, M. (2011). ARGoS: a Modular, Multi-Engine Simulator for Heterogeneous Swarm Robotics. *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Purnamadjaja, A. H., & Russell, R. A. (2007). Guiding robots' behaviors using pheromone communication.

Quigley, M., Gerkey, B., Conley, K., Faust, J., Foote, T., Leibs, J., . . . Ng, A. (2009). ROS: an open-source Robot Operating System. *Proc. Open-Source Software Workshop Int. Conf. Robotics and Automation*.

Reina, A., Cope, A. J., Nikolaidis, E., Marshall, J. A., & Sabo, C. (2017). ARK: Augmented Reality for Kilobots. *IEEE Robotics and Automation letters*.

Reina, A., Miletitch, R., Dorigo, M., & Trianni, V. (2015). A quantitative micro–macro link for collective decisions: the shortest path discovery/selection example. *Swarm Intell*.

Ross, K. G., & Matthews, R. W. (1991). *The social biology of wasps.* Cornell University Press.

Rossum, G. V., & Drake, F. (1995). *Python Tutorial.*

Rubenstein, M., Ahler, C., & Nagpal, R. (2012). Kilobot: A Low Cost Scalable Robot System for Collective behaviors. *Proceedings of 2012 IEEE International Conference*.

Sahin, E. (2015). Swarm Robotics: From Sources of Inspiration to Domains of Application. *Middle East Technical University*.

Sakiyama, T., & Gunji, Y.-P. (2016). Directional ambiguity in trail-laying algorithms. *Elsevier*.

Sharma, G., & Martin, J. (2009). MATLAB: A Language for Parallel Computing. *International Journal of Parallel Programming*.

Sievert, C., Parmer, C., Hocking, T., Chamberlain, S., & Ram, K. (n.d.). plotly: Create Interactive Web Graphics via 'plotly. js'. *2016*.

von Thienen, W., Metzler, D., Choe, D., & Witte, V. (2014). Pheromone communication in ants: a detiled analysis of concentration-dependent decisions in three species. *Behav. Ecol. Sociobiol.*

Wendt, S., & Czaczkes, T. J. (2017). Individual ant workers show self-control. *Royal Society*.

Williamson, D. F., Parker, R. A., & Kendrick, J. S. (1989). The Box Plot: A Simple Visual Method to Interpret Data. *Academia and Clinic*.

Wilson, E. O. (1965). Chemical communication in the social insects. *Science*.

Witte, V., Attygalle, A., & Meinwald, J. (2007). *Complex chemical communication in the crazy ant Paratrechina longicornis Latreille (Hymenoptera: Formicidae).* Chemoecology, 17, 57–62.

Appendices

A1. Table of equations, figures and table

A2. ARGoS file explanation


```xml
<?xml version="1.0" ?>
<argos-configuration>

  <!-- ************************* -->
  <!-- * General configuration * -->
  <!-- ************************* -->
  <framework>
    <system threads="4" />
    <experiment length="1000"
                ticks_per_second="10"
                random_seed="RANDOMSEED" />
  </framework>

  <!-- *************** -->
  <!-- * Controllers * -->
  <!-- *************** -->
  <controllers>
    <kilobot_controller id="kbc">
      <actuators>
        <differential_steering implementation="default" />
        <leds implementation="default" medium="leds" />
        <kilobot_communication implementation="default" />
      </actuators>
      <sensors>
        <kilobot_communication implementation="default" medium="kilocomm" show_rays="false" />
        <kilobot_light implementation="rot_z_only" />
      </sensors>
      <params behavior="build/examples/behaviors/experiment2" />
    </kilobot_controller>

  </controllers>

    <!-- ****************** -->
    <!-- * Loop functions * -->
    <!-- ****************** -->
    <loop_functions library="build/examples/loop_functions/libark_loop_functions2"
      label="ark_loop_functions2"
      datafilename="FILENAME"
      frequency="3000"
      cells_per_metre="150">

      <pheromone_params evaporation_rate="EVAP" diffusion_rate="DIFF" pheromone_amount ="AMOUNT"/>

      <option id="1" quality="15" position="0,0" radius="0.1" color="3">
      </option>

      <option id="2" quality="10" position="0.67,0.67" radius="0.1" color="9">
      </option>

      <option id="ENV1" quality="10" position="-0.67,-0.67" radius="0.1" color="9">
      </option>

      <option id="ENV2" quality="10" position="0.67,-0.67" radius="0.1" color="9">
      </option>

      <option id="ENV3" quality="10" position="-0.67,0.67" radius="0.1" color="9">
      </option>

    </loop_functions>



  <!-- *********************** -->
  <!-- * Arena configuration * -->
  <!-- *********************** -->
  <arena size="2.55, 2.55, 1" center="0,0,0.5">

    <light id="l0"
```

```
         position="0,0,0.95"
         orientation="0,0,0"
         color="yellow"
         intensity="1.0"
         medium="leds"/>

    <box id="wall_north" size="2.5,0.1,0.5" movable="false">
      <body position="0,1.25,0" orientation="0,0,0" />
    </box>
    <box id="wall_south" size="2.5,0.1,0.5" movable="false">
      <body position="0,-1.25,0" orientation="0,0,0" />
    </box>
    <box id="wall_east" size="0.1,2.5,0.5" movable="false">
      <body position="1.25,0,0" orientation="0,0,0" />
    </box>
    <box id="wall_west" size="0.1,2.5,0.5" movable="false">
      <body position="-1.25,0,0" orientation="0,0,0" />
    </box>


    <distribute>
      <position method="grid"
      center="0,0,0"
      distances="0.07,0.07,0"
      layout="LAYROBX,LAYROBY,1" />
      <orientation method="gaussian" mean="0,0,0" std_dev="360,0,0" />
      <entity quantity="ROBOTS" max_trials="1">
        <kilobot id="kb">
          <controller config="kbc" />>
        </kilobot>
      </entity>
    </distribute>

   <floor id="floor"
    source="loop_functions"
    pixels_per_meter="100" />


  </arena>

  <!-- ******************* -->
  <!-- * Physics engines * -->
  <!-- ******************* -->
  <physics_engines>
    <dynamics2d id="dyn2d" />
  </physics_engines>


  <!-- ********* -->
  <!-- * Media * -->
  <!-- ********* -->
  <media>
    <kilobot_communication id="kilocomm" />
    <led id="leds" />
  </media>


<!-- ****************** -->
<!-- * Visualization * -->
<!-- ****************** -->

<visualization>
    <qt-opengl>
      <camera>
        <placement idx="0" position="-0.660983,0,0.6875" look_at="0.0531593,0,-0.0125011"
lens_focal_length="15" />
        <placement idx="1" position="-0.0229259,-0.177184,0.0725521" look_at="0.0273839,0.812385,-0.0624333"
lens_focal_length="20" />
        <placement idx="2" position="0,0,0.8" look_at="0,0,0" lens_focal_length="20" />
      </camera>
    </qt-opengl>
</visualization-->
</argos-configuration>
```

The file presented above is just an example of the ARGoS files, the ones capable of executing the program while fixing all the general parameters of the environment and are calling the behaviour and the loop functions. In this file there are some parameters defined in capital letters to enable the bash file to change their values when multiple experiments are performed.

First of all, in general information the length of the experiment in terms of ticks is specified. The experiments have to have a certain length in order to obtain reliable results. In addition, the number of threads executed for every experiment is specified in order to make the execution faster and the ticks per second to make the experiment faster. The last parameter is the random seed. This is extremely important because is specifying which randomly chosen number is going to be used to perform the experiment. This random number will affect to all the random numbers of the experiment providing a repeatable behaviour for the same random number used. This random number is very powerful due to the fact that any experiment can be exactly reproduced just using the same random seed.

To continue, the controller section is the one that specifies which the behaviour to use providing its full path and name. One of its important functions is to define the sensors and actuators in the experiment including its communication. As observed in the code above, the LED sensor should be specified.

The next section is the loop function that calls the loop function using its path and defining some constants of this loop function. In our case, some of the parameters predefined are the evaporation, diffusion and amount of pheromone in order to be fixed in the bash code for each experiment performed. Furthermore, in the ARGoS file is also defined the sources. Every source has an id, a quality source specification (in the nest's case, this quality is fixed to 15), the position in terms of vertical and horizontal axis, the radius of the source circle and the colour. As shown in the ARGoS code, the ID, in some cases, is represented with capital letters in order to be change from the bash file when multiple environments are being performed. Thus, the number of food sources can be varied decreasing them, writing a 0 in the ID or increasing them, writing any other number.

The Arena Configuration consists in the distribution of the Arena, the floor. It is defined its size, its colour, orientation, intensity and its walls, stopping the robots to move far from the current environment. It is also determined the pixels per meter and also the robots position in the space. The kilobots can be allocated in a grid structure or in a random position over the current environment choosing the orientation required. In this case, a Gaussian orientation is chosen and the allocation in the nest is done due to a grid structure that will get modified depending on the number of robots. At the start, the robots will be moving in a random walk. Therefore, due to the fact that their orientation is random too, their movement will be completely different and randomized.

The physical engines and the media section are just used as the template to specify the “dy2dn” for the dynamics and the “kilocomm” for the communication. These names are reserved in ARGoS and are related to predefined dynamics and communications configurations.

The last bit of the code of the ARGoS file is the visualization. This is defining the placement of the cameras. Remembering that this file is used only for the simulation, it can be simulated that the cameras are defined at the top, in one side, on the floor… Its position lens focal and its orientation are defined to allow a better visualization while the experiment is performed. What is more, a maximum of nine cameras can be defined with different parameters and it is allowed to keep changing between one camera and another during the experiment.

A3. Behaviour file explanation

There are a wide range of functionalities in the behaviour file which are responsible of defining the robots to take their own decision, reading the virtual sensors received from the ARK system.

The first code remarkable is the set random motion function:

```
void set_random_motion(){
   if (countForward < 4){
      countForward++;
      set_motion(FORWARD);
   }else{
      countForward = 0;
      if ((rand() % 2) == 0){
         set_motion(LEFT);
      }else{
         set_motion(RIGHT);
      }
   }
}
```

This function is capable to execute a random movement for the kilobots. It is executed every 10 ticks when the swarm robots are foraging for food. It can be seen that the most probable motion to be executed by the robots is the forward, go straight, namely, both motors spinning with the same power. As shown, the robots has a movement of four times straight and in the fifth time a random decision is made choosing to go left or right. Then, the local counter is reset. Thus, after 10 ticks the robot is going to go straight again.

This behaviour is imposed to the robot instead of allowing a total freedom due to the fact that it has been proven by using this capability the robots have the opportunity to forage further incrementing the chances to find food quickly. Furthermore, the real ants have a similar behaviour going straight after a certain amount of time, changing its direction and going straight again.

Another interesting section of the code is the performance of how the robots print pheromone according to the quality found. It is essential to bear in mind that the print of the pheromone is performed by the ARK system while the decision to print pheromone is made for the robots. This printing is known for the ARK due to the LED colour located in each robot, allowing a pheromone printing when the LED is blue. The check probability function is created to enable the LED to be blue when the pheromone is needed to be printed:

```
void checkProbability(){
   //% of probability of quality
   float probability = foodQuality*100/MAX_PROB;

   //Random number between 1 and 100
   float randNum = rand() % 100 + 1;

   //If probability bigger, print ph
   if(probability > randNum){
      set_color(RGB(0, 0, 3));
   }else{
      //Else, no pheromone, led off
      set_color(RGB(0, 0, 0));
```

```
    }
}
```

The function above is composed of a probability variable, which corresponds to the food quality in percentage. In this thesis the maximum quality allowed is 10. Thus, a quality of 5 corresponds a pheromone printing of 50% of the times and a quality of 10, of 100%.

The next step is to calculate a random number from 1 to 100. This number is compared to the probability previously found and if the probability is bigger than the random number, the LED is set to blue, if not; the LED will be switched off.

The check probability function is implemented through the way from the food to the nest, because in this moment the robot is collecting the food and printing the pheromone trail. The check probability function is executed every 5 ticks to achieve a floor painting according to the LED.

An example of a five-time sequence of some pheromone printing is shown below:

- Quality food source found: 6
- Probability according to quality: 6/10 *100 = 60%
- Random number found: 23, 73, 92, 11 and 55
- Times to print pheromone: YES, NO, NO, YES and YES.

As shown, the printing times are set according to the quality of the food source. After iterating this sequence for a long time, the pheromone will be printed according to its quality.

The robots turning time should be calculated when they are following the pheromone or when they are returning to home. It is used a specific angle that can be received from ARK or calculated by the robots. In the first case, ARK is sending the angle to home. This angle is received in 4 bits that means that only 8 numbers can be distinguished. That is why; the number of bits to be sent is restricted to 10 bits only. These 8 numbers are assigned each one to a region. Each region has a range of 45 degrees. Thus, the angle is divided by 45 and rounded achieving one number between 0 and 7. The numbers achieved are de-codified multiplying by 45. The angle to follow the pheromone is calculated by the robots. This is also limited in regions with the same precision.

Once the angles are ready to be computed and executed in the robots, the times that the robot should be turning to one side or the other should be calculated. The robot does not have a GPS; therefore, the turn has to be calculated using the angle explain above and transforming this to ticks. This method is really unprecise; that is why, a constant recalculation of the angle to turn should be done. The execution to calculate the ticks in order to turn is really simple:

$$\boxed{\text{timesToTurn} = \text{ceil}(\text{angle} * 47/360)}$$

Equation 5. Turning time calculation

This is compound by the relation known between 47 ticks and 360 degrees. Therefore, multiplying this relation for the current angle and rounding it, the ticks turning will be found. The problem is that depending on the battery the turn will be faster or slower, also, depends on the floor, on the environment around and the most important thing, on its calibration.

This function is executed in a case and once is executed goes to another case to execute the next computation:

```
//Turn x times depending on the angle to home
          if(countTurn < timesToTurn){
             countTurn++;
          }else{
             timesToTurn = 0;
             caseCollectFood = 2;
          }
          break;
```

This code is turning the robots while incrementing a counter that counts the ticks. Once the counter reaches the value desired, the parameter will be reset and the function will go to another case to recalculate the motion of the robot according with the environment found.

An important functionality of the robots behaviour is how to process the messages that comes from the ARK. The messages are located in a 10 bits variable and extracted depending on the necessities. The most remarkable bit of the message is the parameter that receives the knowledge of the pheromone in the robot surroundings. If the four bits of this information are not equal to zero, it means that pheromone is around. This parameter is analysing each of the four sections. When one or more sections are greater than 0, the current section is evaluated. The section chosen between the ones that has pheromone is the furthest from the nest due to the fact that the ants need to go in the way round to home to find the food source. However, this affirmation is not always the correct decision; therefore, the recalculation of this angle will be done every time that a new message from ARK is received to the Kilobot.

At the start of the execution of the behaviour file, the system is set up and the main loop is executed. This main loop has a switch case comprising all the main functions that will execute other functions to achieve the global goal. The cases of the switch are:

- SEARCH_FOOD: This one is capable to implement the random move, previously explained, while keep checking every time step if the pheromone or a food source is found.

- GO_AT_HOME: This case is reserved for the robots that are bringing food from a food source to the nest. This is the one in charge to change the robot LED to print the pheromone, while keep recalculating the angle to get home.

- TURN_KILOBOT: This function is executed from the FOLLOW_PHEROMONE and the GO_AT_HOME cases. This as explained above, is capable to turn the robot according to a certain angle.

- FOLLOW_PH: This process is executed when the robots are foraging for food and they find a pheromone trail. To calculate the new orientation for the ants some calculations are needed to allow the better path decision; therefore, this function is in charge to make this decision and send to the TURN_KILOBOT the angle needed.

A4. Loop function file explanation

The loop function is in charge to simulate the virtual sensors having a global knowledge while limiting the robots one in order to have a similar and accurate behaviour to the ants' behaviour. What is more, the loop function file has to create all the communications, process functions to explore the environment, read all the information from the robots, among others functionalities.

This process is the most complex because it has to deal with each robot communication while keep looking the environment and also, keep updating the environment data, processing it and understanding how the environment should change. It is structured in a set of functions explained in the following pages.

First of all, an "init" function is created to initialize the variables of the system and set up the parameters according to the ARGoS file, such as get the name of the file to write the data, the number of the cells per metre of the pheromone matrix, the configuration of the parameters of the evaporation, the diffusion and the amount of pheromone that are extracted from the ARGoS file. What is more, the floor matrix is initialized depending on the size of the matrix. This is defined as a dynamic variable to optimise the memory to the maximum.

It is also included a reset function in order to reset all the variables in the case that is needed to restart the program and a destroy function to delete all the dynamic memory of the program such as the matrix of the pheromone in the floor

The next important function writes the data extracted from the experiments to a file. This process is really useful due to the fact that the information about the performance requires a deep analysis. The file created will contain the collections of the food for every food source and also the distribution at the end. The distribution stands for the number of robots that are in the trail pheromone path taking food to home or going to collect the food. This function is performed using an array that counts the collection. A matrix of these arrays is built containing this counting for every food source. The information is saved in the file just at the end of the experiment to not slow down the program during the performance. The file should be correctly defined and closed after the writing.

The evaporation and diffusion calculation function is executed after 10 times step. This function goes through all the matrix of the floor and recalculates the diffusion and the evaporation for every cell using the functions defined in the Relevant Theory and Analysis section. An auxiliary matrix is built to record the past value of each cell in order to perform the diffusion. Due to its slow performance, the calculation is only done every 10 times step.

Another two particular functions typical for the loop functions is the pre step and the post step function. The pre step function is executed every time step just before the main execution. This function is needed in order to actualize some variables as the message counting, actualized every two times steps including the communication settings with the robots. Also, it includes the calling of the evaporation matrix, as explained before.

The post step function is executed after the main function. This process is useful to actualize the information about the sensors of each robot such as the LED colour in order to be process in the next time step.

Some small functions to normalize and de-normalize the angles needed to send to the robot and the Kilobot orientation and position are needed in order to process the data. It is including also functions such as the ID token and the LED colour of the robot.

The most important function of the loop functions is processing the message to be sent to the Kilobot. This function uses the message and the Kilobot entity, comprising the information of the current Kilobot. This function is compound for lots of small functions to enable to create a reduce message to be sent to the current robot as virtual sensors.

The first step is to calculate the position and the orientation using the functions above and adapting them to the Euler angles from the radiant system to the degrees system.

After that, the ARGoS file should be analysed storing where the position of food and of every food source are located. What is more, the quality and the radium of the food sources are saved in the list foods parameters. After that, an adaption should be done. This adaption consists of a relation between the floor matrix and the real arena sizes. As explained, the floor matrix have a fixed number cells per meter defined in the ARGoS file. The positions extracted from the ARGoS file are referred to the real system. Therefore, they should be adapted to this matrix to know where the environment parameters are located. The food list class explained is defined by a structure called Food Class as shown below:

```
struct FoodClass{
        int quality;
        float FOOD_X;
        float FOOD_Y;
    };
```

Once the relation of the matrix with the real arena is found, it can be analysed if the current robot have found the food. To perform this action a loop analysing each food source should be done. For each instance of the loop, the positions of the robot are compared with the position of the current food source taking into count its size. If this comparison is positive, a variable will be saved indicating that the current robot has reached. In addition, the quality of the food source found will be also saved. A similar comparison is done with the pheromone matrix and the robot position, to check if the robot is in a pheromone trail and also, with the home position in comparison with the robot position.

To check if there is pheromone in the robot surroundings a huge loop should be realized. This check is executed in all the cases except when the LED of the robot is blue, which means that the robot is taking the food to the nest and does not need to know if there is pheromone in its surroundings. This loop goes through all the floor matrix and check if around 3 cm from the robot there is pheromone. In addition, the loop is filtering these values to just take the ones that are in front of the robot. A comparison with the normalize angle from the robot to the pheromone should be done. If this comparison is less than 90 degrees, the robots have its pheromone in the front. The last step of this process is to save the variable that indicates the finding of pheromone.

There are four sections in order to send only 4 bits indicating a one or a zero for each of the zones corresponding with a 1 if the pheromone is found and with a 0 if the pheromone is not found.

Once the current environment is defined in relation with the robot, the distance to home is calculated in order to teach the robot always its path to return to the nest when this is needed. To do this performance a tangent relation of the y coordinates with the x coordinates is performed and transformed to degrees.

After that, the orientation is added due to the fact that the angle needed for the robot is taking into account its point of view. Then, the angle is de-normalized to be understandable for the robot.

The end of this process is to construct the message to send to the current Kilobot. First of all the pheromone zones Is allocated in the first four bits creating a loop through all the areas and allocating a one or a zero. The message is shifted four positions to allow space to the other variables. Later, a bit indicating if the robot is at home or not is allocated in the LSB. Then, the parameter that indicates the food reached by the robot is analysed. When the robot reaches the food source, the variable is saved in the 2LSB of the message. The quality is saved with the food source using 4 bits in the position of the pheromone zones due to the fact that this are not needed anymore. To continue, the angle to home divided by 45, allowing just four bits is saved and the flag to define that the message is ready to be sent changes to true.

The last function that needs an explanation in the loop function is the one that prints the floor in the virtual environment. This is performed by ARK which has all the control and the knowledge of the current system.

First of all, the position in the plane analysed is taken and an iteration of all the sources of the ARGoS, including the home source and the food sources, is done. For each of the sources is checked the current position of the plane with the current position of the source analysed in a range within the radius of the source analysed. If this coincides, the colour is taken from the ARGoS file, that is located as the fifth parameter of the source and it is saved in the variable of colour. The iterator gives priority to high sources that means that if two sources are located in the same location, the one with higher ID will be the one printed in the floor. That is why; if the nest is always ID = 1 is located in the boundaries of a food source, with IDs bigger than 1, the one printed in the top will be the food source. However, the robot stills detecting both source but on the screen the only viewable is the food source.

To continue, the printing of the pheromone in the floor is done according to the floor matrix. This one adapts the current space to the floor matrix sizes and print the pheromone taking into account the current plane position and the quantity of pheromone in the matrix. For high quantities of pheromone, a dark colour is selected and for low quantities, a light colour is chosen to distinguish in the execution the diffusion, the evaporation and the amount performance in every time step.

Once the colour in the particular plane is defined the function is returning this colour to the system that which internals procedures will enable the printing in the screen which will allow the facilities for the user to understand the current behaviour of the robots.

What is more, predefined functions, such as the get option function (to take the options from ARGoS), the add option function (to include all the options that ARGoS has) and the into colour function (to transform a number to a particular colour) are added to make the program easier to be performed.

A5. BASH file explanation

The bash files are used to execute a certain number of experiments executing ARGoS but changing some of the parameters that are tagged in capital letters.

A lot of bash files have been created to analyse different situations of the swarm robots. All of them use a similar structure.

First of all, a template file of ARGoS with its full path is saved and the needed variables are created. A loop comprises the number of environments needed. It is created inside a case function that changes a wide range of values depending of the current environment in the loop. The structure of these loops is similar as the following one:

```
while [ $EnvironCount -le 2 ]
do
    case $EnvironCount in
    1) auxEnv=( 3 0 0 );;
    2) auxEnv=( 3 4 5 );;
    *) auxEnv=( 0 0 0 );;
    esac

    NumRobCount=0
    while [ $NumRobCount -le 2 ]
    do
        EvapCount=0
        while [ $EvapCount -le 2 ]
        do
            DiffCount=0
            while [ $DiffCount -le 2 ]
            do
                AmountCount=0
                while [ $AmountCount -le 2 ]
                do
                    #Call the function
                    execExp
                   ((AmountCount++))
               done
                #Increment the counter
               ((DiffCount++))
           done
            #Increment the counter
            ((EvapCount++))
        done
        #Increment the counter
        ((NumRobCount++))
    done
    #Increment the counter
    ((EnvironCount++))
done
```

Figure 43. Bash code

As it can be observed, in the image above, inside all the loops, a function is called to set each the experiments. This function is creating a file for each environment and a new loop is implemented. This loop executes hundred of experiments using the same environment. It is chosen a high number, normally 100 to create a robust and reliable data. Inside this loop, the parameters of the ARGoS file template are modified depending on the necessities of the experiment and later the ARGoS is executed to perform the current experiment. Once the experiment is finished, the counter of the loop is updated and the same performance is done again until both loops are completed.

A6. R file explanation

The R files process the big data stored in different files during the execution. R has a lot of capabilities such as reading and writing from files, sorting information and plotting multiple types of graphs. For each experiment, multiple R files are created to adapt them to the necessities of the experiment. Normally, it is used an R file to sort the data from the created files during the experiment and another R file to plot the data stored from the first one.

The R file to save the data is composed to multiple loops, one for each parameter modified in the R for each experiment. Inside this multiple loops, a filename for each condition is given as shown in this example:

paste("Experiment3/FourFoodSources/NumRob_", NumRob[k1], "Evap_", Evap[k2],"Diff_",Diff[k3],"Amount_", Amount[k4],"Food_4.txt", sep=" ")

For instance, this name is related to the folder of an environment of four food sources performed in experiment 3 that includes a current number of robots according to an array, a particular evaporation, diffusion and amount of pheromone and also, saved with Food_4 at the end to know that has four food sources.

After that, using the function “read.table”, the results of the whole file are saved in a variable. It is created a loop to go through all the experiments saved in the table and sort them depending on the food source, collection and distribution values. Once all of them are saved in a different array, it can be saved in a new file and in a table making a mean of the 100 experiments with the same parameters performed, as shown in the example before:

write (data, file = "Experiment3/data4.txt", ncolumns = if(is.character(data)) 1 else 6, append = TRUE, sep = "\t")

This function indicates first the data to save, later the name of the file, then how many columns and finally, how it is desirable to be split with, in this case, with a tab between each variable.

The data will have a similar shape as the following one:

data4.txt

NumRob	Evap	Diff	Amount	Collect	Distrib
50	0.005	0.5	30	43.455	5.6025
50	0.005	0.5	50	43.25	5.565
50	0.005	0.5	100	39.89	5.115
50	0.005	1	30	42.475	5.68
50	0.005	1	50	42.695	5.5325
50	0.005	1	100	39.99	5.185
50	0.005	1.5	30	7.4175	0
50	0.005	1.5	50	6.915	0
50	0.005	1.5	100	6.4925	0
50	0.01	0.5	30	43.5175	5.665
50	0.01	0.5	50	43.3375	5.7875
50	0.01	0.5	100	39.8175	5.0175
50	0.01	1	30	42.535	5.7025
50	0.01	1	50	42.7575	5.52
50	0.01	1	100	40.03	5.1
50	0.01	1.5	30	7.4175	0
50	0.01	1.5	50	6.91	0
50	0.01	1.5	100	6.485	0
50	0.03	0.5	30	43.45	5.8175

Figure 44. R data file

The other kind of file, for each experiment is following a different format due to the fact that some are preferable to print graphs, histograms and overlapping histograms like in the distance experiment, while others opt for the printing of box plots only like the quality experiments which prints 10 different box plot comparison between qualities. It is also used the printing of heat maps, using the ggplot function which allows to adapt the heat map according to the experiment results necessities, used in the last experiment to compare a huge amount of parameters. This last performance is the most difficult due to the fact that all the data should be sorted with different formats to achieve, as shown in Parameters experiment section, a completely figure composed for nine different heat maps which each one has two different parameters with three possible values each variable. It is important to choose a correct range of colours to be able to compare between closer values in a heat map. The heat map is performed changing the squares colours according to the value given for each situation, in our case, these values were the collection and the distribution.

What is more, to create a heat map, a data frame is required defining exactly which value ranges have the parameters that include the experiment and also adding afterwards the collection and distribution values. Later, the heat should be inspected using the data frame created. To finish, the ggplot function should be performed for each case including the data, axes, and colours and limits values.

Once all the plots are saved in each correct variable, a multi plot function can be executed in order to print in the same plot but not overlapped the nine heat maps of the collection. Another multi plot including the nine heat maps of the distribution comprising all the cases performed in the experiments is executed.

A7. 3D system designed

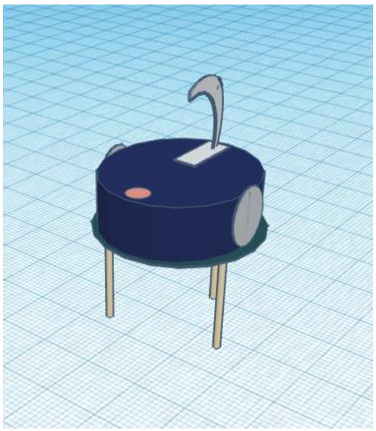


Figure 45. Kilobot 3D design

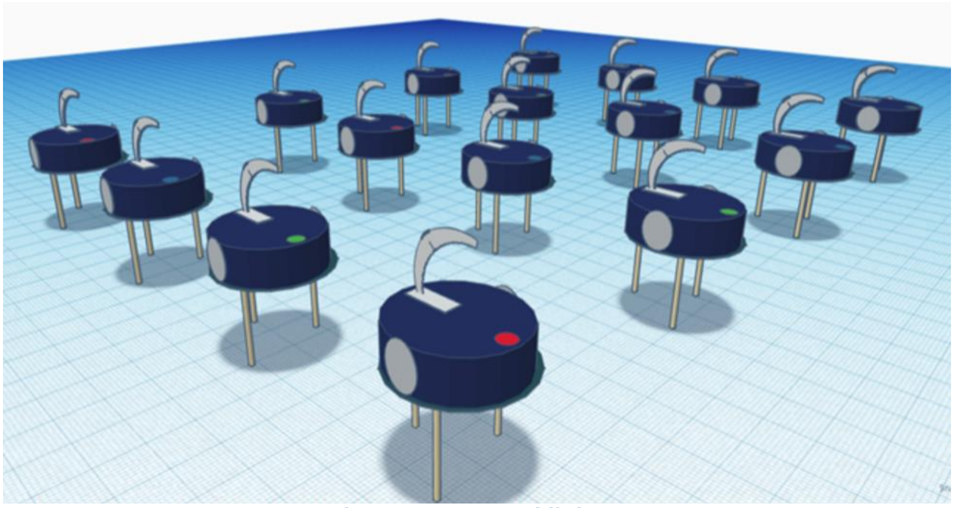


Figure 46. Swarm kilobots

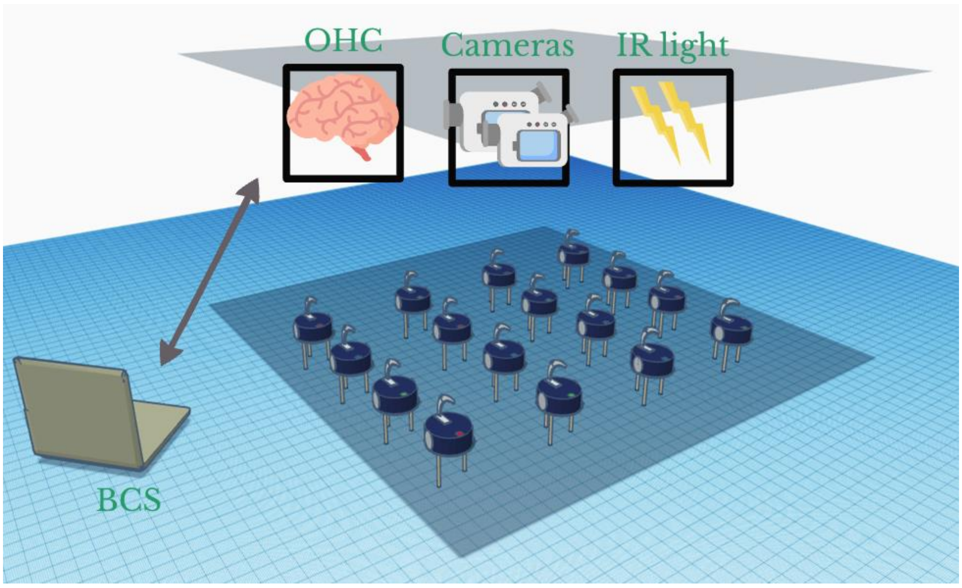


Figure 47. ARK system in 3D

A8. YouTube videos

During the realization of the project some videos have been performed to evaluate the performance of the experiments. These videos are available in YouTube and can be accessed using the following links:

- Fast foraging of kilobots

  https://youtu.be/11oY24RB-4Q

- Distance experiment of kilobots

  https://youtu.be/yu5tzv9HWmA

- Quality experiment of Kilobots

  https://youtu.be/0r3lUt_Z9qI

- Maximize throughput for Kilobots collection

  https://youtu.be/v8bfi2T0YgE

- Best option priority for Kilobots collection

  https://youtu.be/NK7hK3LKV2I

Sheffield Hallam University

A9. Experiment pictures

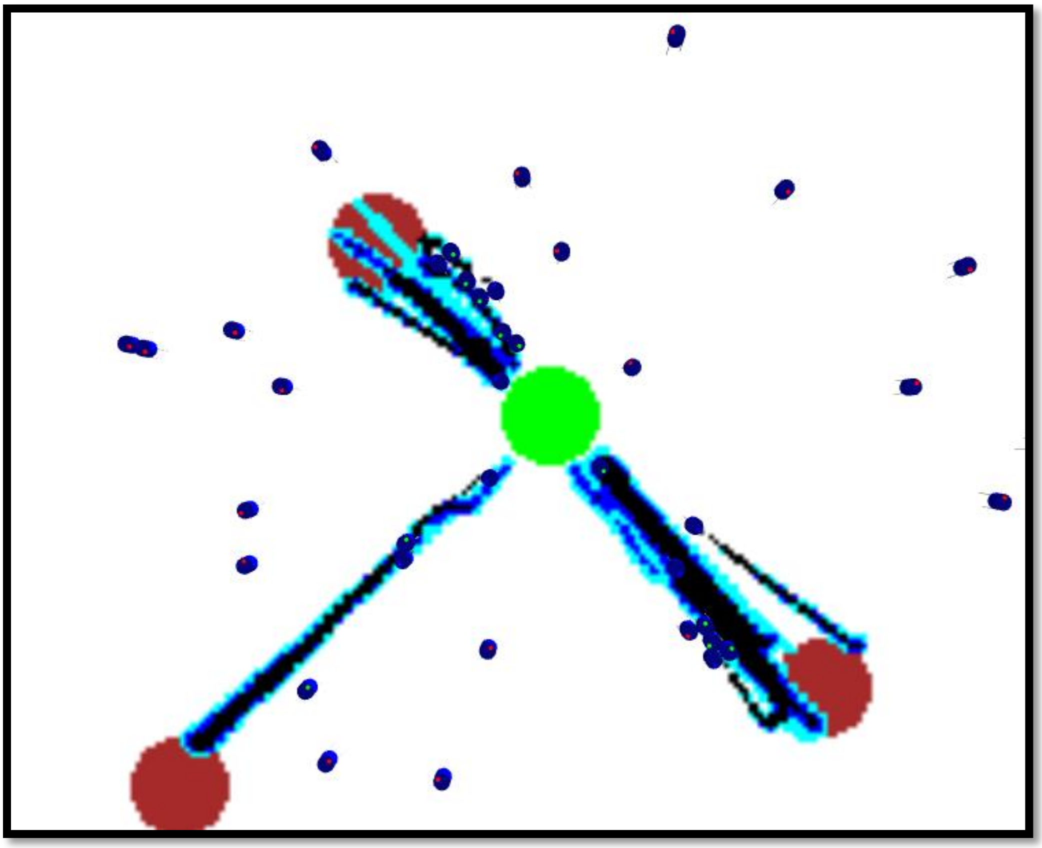


Figure 48. Short distance selection

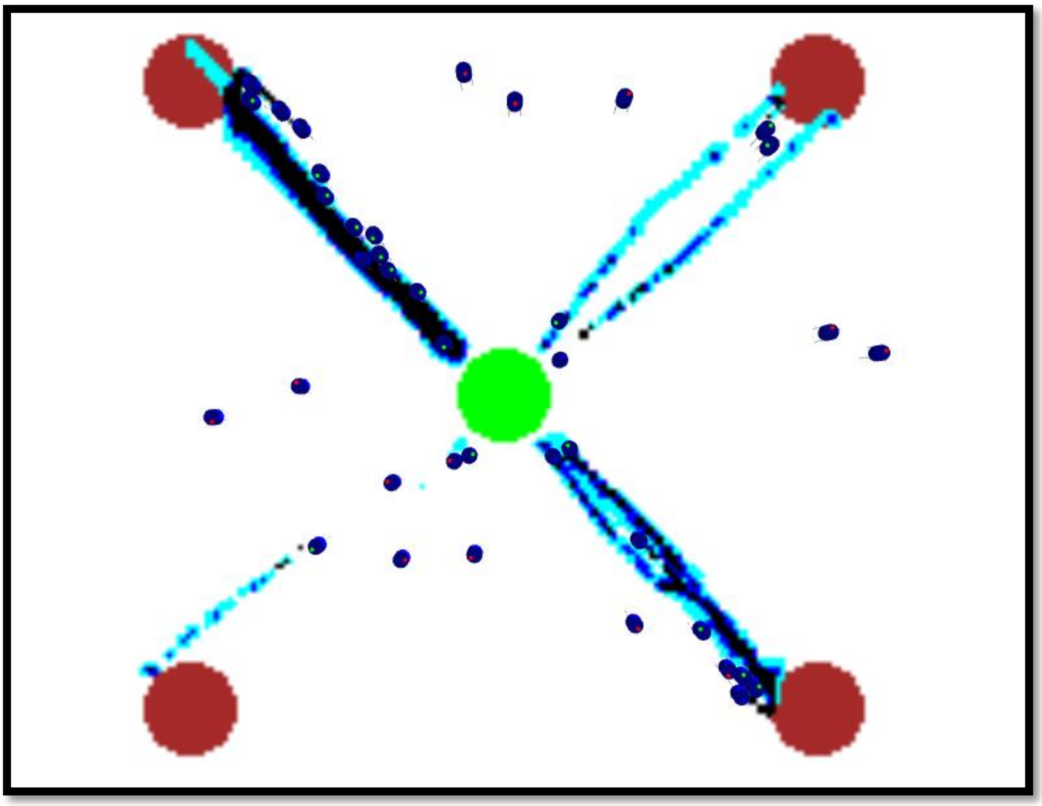


Figure 49. Best quality selection

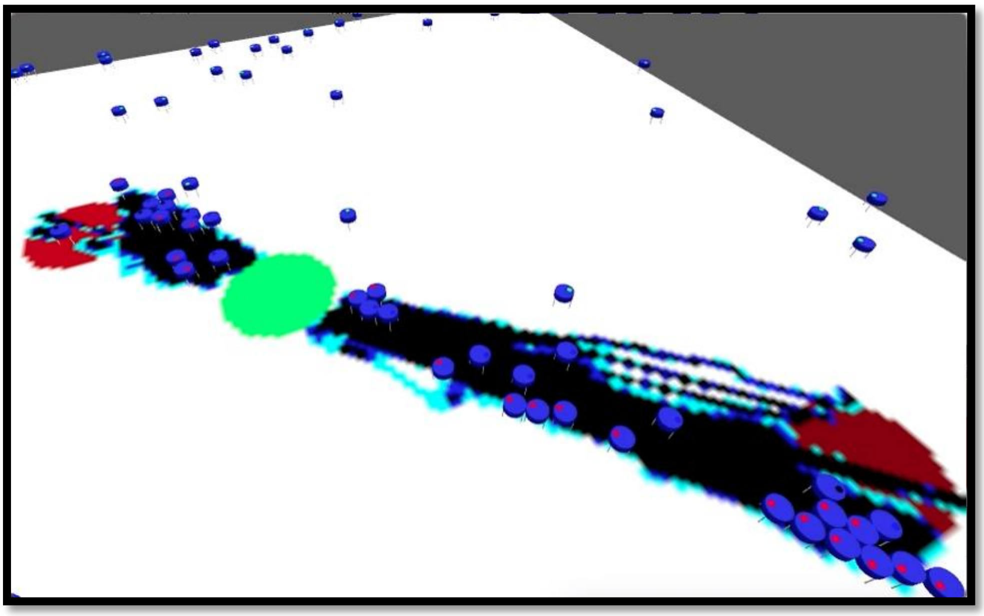


Figure 50. Maximise throughput

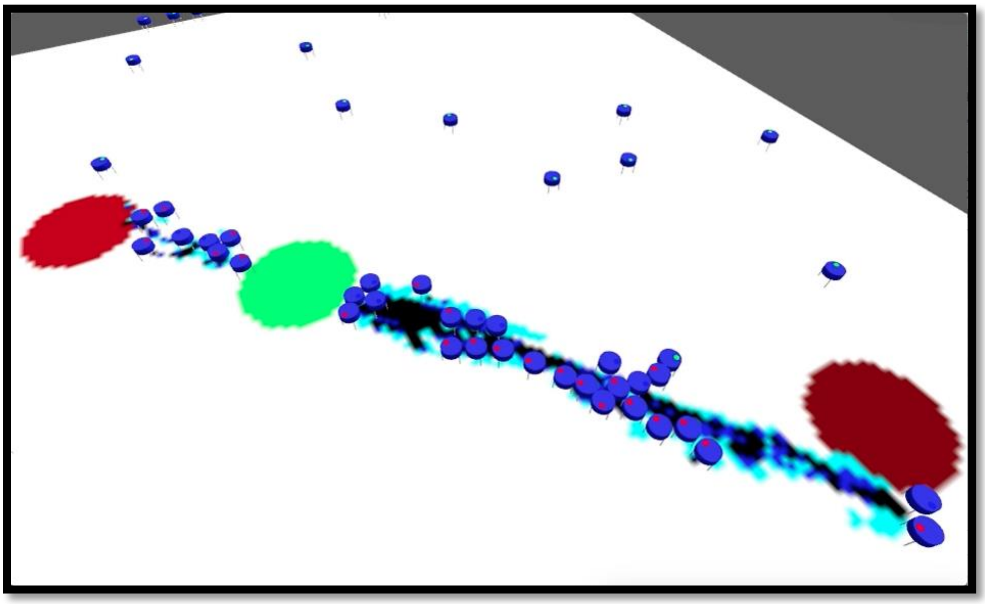


Figure 51. Choose the best option

A10. Lab pictures

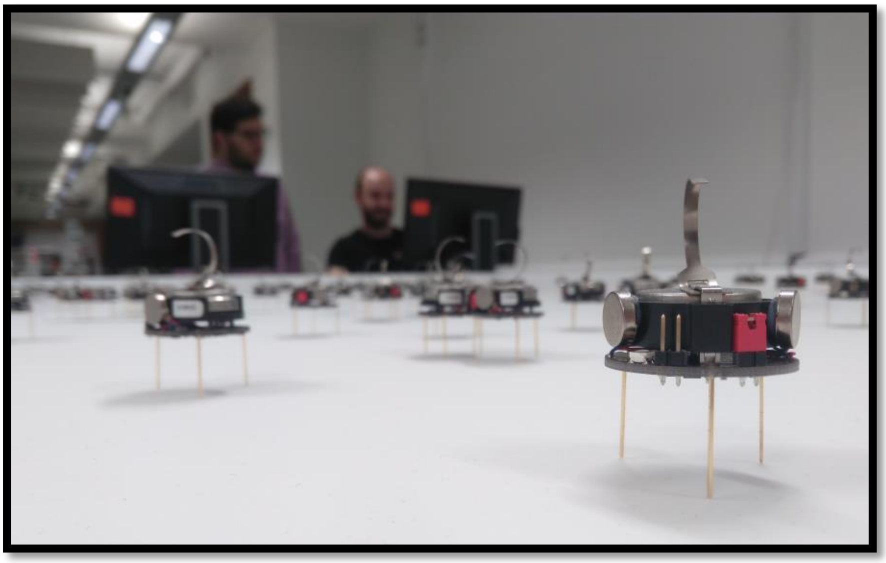


Figure 52. Real Kilobot

Figure 53. Aligned kilobots

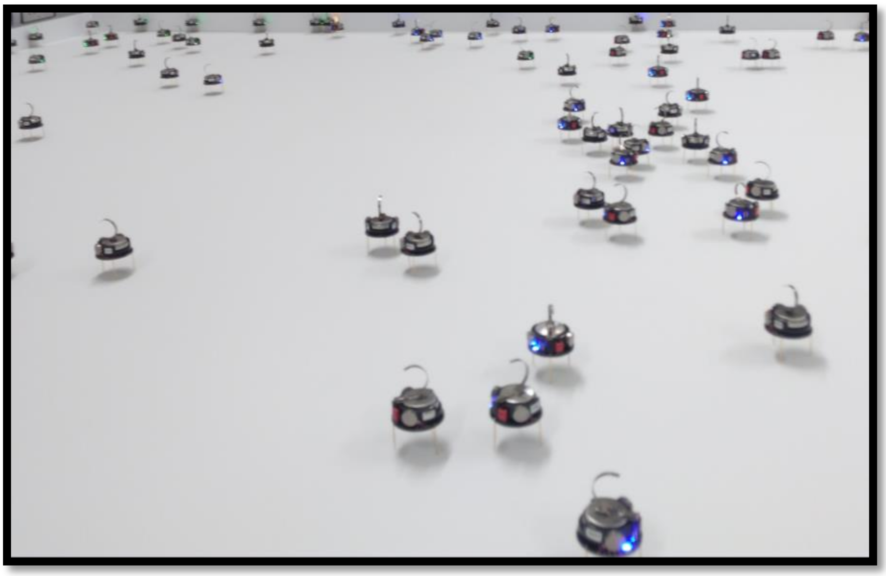


Figure 54. Kilobots performing an experiment