Daniela Kengyel

Embodiment of Honeybee Behaviour in Artificial Agents: Swarm Robots and Swarm Level Optimisation

PhD thesis

Graz University of Technology

Institute for Softwaretechnology

Supervisor: Prof. Dr. Franz Wotawa

Graz, November 2015

Statutory Declaration

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

Graz, ______________________ ______________________

Date Signature

Eidesstattliche Erklärung[1]

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Graz, am ______________________ ______________________

Datum Unterschrift

[1]Beschluss der Curricula-Kommission für Bachelor-, Master- und Diplomstudien vom 10.11.2008; Genehmigung des Senates am 1.12.2008

Contents

List of Figures

List of Tables

Abstract

Honeybees are social insects which exhibit a wide range of collective behaviour. This leads to the emergence of abilities that single individuals wouldn't be capable of. For example, groups of young honeybees are able to collectively find a spot with optimal temperature while most single bees fail at this task. Such a behaviour is a good example for swarm intelligence in general and thus can be further developed for the usage in swarm robotics. Results retrieved from observations of swarms of bees are used to recreate the behaviour in simulations (computer models) and in real robots. The algorithm derived from the bees' behaviour is called "BEECLUST".

In the first part of this thesis, an experimental setup with temperature gradients (similar to the setup that was used for the observation of honeybees) is created to ensure physically realistic conditions. To emulate the bees' antennae for temperature measurement a swarm robot called "ePuck" is extended with temperature sensors. The BEECLUST algorithm is investigated in a set of collective choice experiments with different conditions in order to study

the strengths and limitations of the algorithm.

In the second part of the thesis, we investigate an idea called "Swarm Level Optimisation". Here we use different movement patterns of the agents (that were also observed in young honeybees) to optimise the aggregation process of the BEECLUST algorithm with respect to the experimental setup. Two different models are proposed: a rule-based behaviour model and an ODE-model that is used to find an optimal swarm composition for a specific setup with the help of evolutionary computation. Evolution reveals nontrivial distributions of movement patterns which indicate a complex underlying system.

Kurzfassung

Soziale Insekten, wie zum Beispiel die Honigbiene, zeigen viele verschiedene kollektive Verhaltenweisen, welche zur Entstehung von Fähigkeiten führen, die das einzelne Individuum nicht hat. Zum Beispiel kann ein Schwarm von jungen Honigbienen einen Wärmespot mit seiner optimaler Temperatur im Kollektiv auffinden, während eine einzelne Biene meistens nicht dazu fähig ist. Solch ein Verhalten ist ein gutes Beispiel für Schwarmintelligenz im Allgemeinen und kann daher gut für die Verwendung in der Schwarmrobotik weiter entwickelt werden. Ergebnisse von Beobachtungen von Bienenschwärmen werden verwendet um das Verhalten sowohl in Simulationen (Computermodellen) als auch in echten Robotern nachzustellen. Dieser Algorithmus wird "BEECLUST" genannt.

In ersten Teil dieser Arbeit wird ein Versuchssetup mit Temperaturgradienten (das gleiche Setup wird bei den Versuchen mit Bienen verwendet) konzipiert, um realistische physikalische Bedingungen sicherzustellen. Der Schwarmroboter "ePuck" wird um Temperatursensoren erweitert um die

Temperaturmessung der Biene mit den Antennen nachzuahmen. Weiters wird der BEECLUST-Algorithmus in mehreren Sets von Experimenten unter verschiedenen Bedingungen analysiert um die Stärken und Grenzen des Algorithmus zu untersuchen.

Im zweiten Teil der Dissertation wird die Idee der "Schwarm Level Optimierung" untersucht. Hier werden verschiedene Verhaltensmuster der Agenten (die ebenfalls bei jungen Honigbienen beobachtet wurden) benutzt, um den Prozess der Aggregation von Agenten, die mit dem BEECLUST-Algorithmus gesteuert sind, unter Berücksichtung des Versuchssetups zu optimieren. Hierbei werden zwei verschiedene Modelle vorgestellt: Ein Modell basierend auf Regeln und ein Differentialgleichungsmodell. Das Differentialgleichungsmodell wird mit Hilfe eines evolutionären Algorithmuses dazu verwendet, um die optimale Schwarmzusammensetzung für ein spezifisches Versuchssetup zu finden. Die Evolution zeigt dabei nichttriviale Verteilungen der Verhaltenstypen, was darauf hindeutet, dass dem Prozess ein komplexes System zugrundeliegt.

Acknowledgements

First, I am thankful for the help and support from my supervisor Prof. Franz Wotawa. He let me pursue my own ideas and always encouraged and motivated me even in tough times so that - in the end - I was able to finish my thesis. I would also like to thank Ass. Prof. Thomas Schmickl for leading the Artificial-Life-Laboratory at the Karl-Franzens-University and for all the good advice he gave to make the project and this thesis a success.

I want to to thank my collegues for all the fruitful discussions, especially Payam Zahadat, Thomas Kunzfeld, Sibylle Hahshold, Heiko Hamann and Gerald Radspieler who also helped me clear the mess and rebuild everything when the robot laboratory was destroyed due to a severe water damage. Thanks to Sabine Martinjak for proof reading.

I also want to thank my whole family, in particular my parents Andrea and Wolfgang Kengyel and my sister Nadja Kummer with family for their support and for always believing in me. I'm especially thankful for all the support and encouragement of my partner Gerhard Scheikl during writing this thesis.

1. Background & Motivation

Parts of this chapter are published in [24, 25, 26] or submitted for publication in [27].

1.1. Swarmintelligence

Ants, bees and termites belong to the group of eusocial insects. Such eusocial insects are a fascinating and inspiring group which live in well organized colonies. They are characterized by three properties: cooperative brood care, living together of all generations in one colony and distinction in fertile and infertile individuals.

Although a colony follows a certain goal, every individual acts unrestricted and without following orders of a central unit that makes decisions. The behaviour that originates from such a self-organized system is called "swarm intelligent“. Millonas [36] listed five principles a swarm intelligent system should have:

- Proximity principle
- Quality principle
- Principle of diverse response
- Principle of stability
- Principle of adaptability

The proximity principle indicates, that the group shows a direct behavioural response to environmental stimuli which leads to a collective space and time computation. The quality principle indicates, that the collective should be able to react on quality factors, like the quality of a food source. The principle of diverse response refers to the distribution of ressources in combination with many different modes to be safe if any sudden changes happen in the environment. The principle of stability demands, that the group should not switch its behaviour because of every small fluctuation in the environment. The last principle is the principle of adaptability and is related to the principle of stability. It demands, that the group should be able to switch its behaviour as a reaction to significant environmental changes.
Thus, a swarm system has to have a good balance between the fourth and fifth principle.

These properties also apply to colonies of social insects [10]. Craig Reynolds was a pioneer in the field of swarm intelligence. He simulated a swarm of birds where every bird acted independently but the swarm was still able to react on the environment [42]. He achieved this, by following just three basic behaviours: avoidance of collisions with other birds, adaptation of the speed to other birds and the attempt to stay close to the swarm.

In the field of swarm robotics, social insects are a good and often used source of inspiration. Swarm robots are very small and do not have a lot of computational power. Thus, the fact that swarms only need a few simple rules to create a complex behaviour can be taken advantage of here. In the work at hand, the behaviour of young honeybees is used as a source of inspiration. This behaviour is described in the following section.

1.2. Swarm Behaviour of Young Honeybees

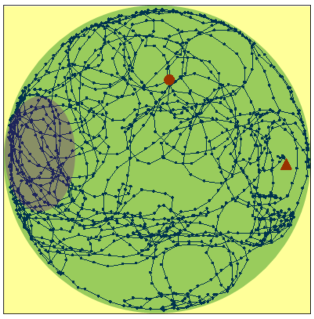
Colonies of honeybees are precisely regulated. For example, the temperature in the brood nest is kept constant at approximately 36°C because otherwise disorder in the development or even the loss of brood may happen [50, 25]. 36°C is also the preferred temperature of young honeybees [17, 20]. In a hive there is no light and thus, the bees have to face the challenge to organize and orient themselves only in a temperature gradient that is inhomogeneous and dynamic [25].

In previous experiments it was found that a single young honeybee (*Apis mellifera*) is mostly not able to locate the area with its preferred temperature, whereas a group of young honeybees is able to find the right spot collectively [31, 47, 54, 26]. From this behaviour the BEECLUST algorithm is derived and is described in the next section (section 1.3).

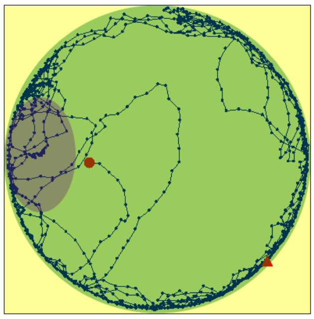
In the previous mentioned experiments with honeybees [54] younger than 24 hours in an arena with a temperature gradient, there were also four different kinds of movement behaviours observed (as mentioned in [45]. We classified these kinds of behaviour into the following classes: *Random Walker*, *Wall Follower*, *Goal Finder* and *Immobile Bee*. Figure 1.1 shows example trajectories of these four classes.

When these different movements of bees were discovered, several questions arose: What effect do the different movements have on other agents or on the swarm? Why are there different behavioural types although just a random-walking-behaviour is enough to reach the goal? What is the role of each behavioural type and how could it be used to predict or modulate the resulting aggregation-behaviour of the swarm?

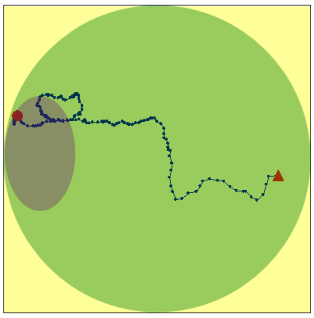
To investigate these questions we introduce four behavioural types to the BEECLUST-algorithm [47], that is a state-of-the-art algorithm for robot swarms and is derived from the aggregation behaviour of young honeybees. We assume that individual behavioural patterns contribute to different



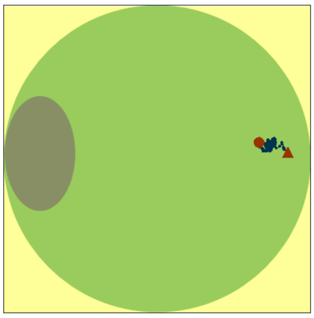
(a) Random-Walker: Trajectory of a bee that walks around randomly in the arena.



(b) Wall-Follower: The behaviour of a young honeybee that we describe as a Wall-Follower.



(c) Goal-Finder: This bee is able to find the area with its preferred temperature. We call a bee with this behaviour "Goal-Finder".



(d) Immobile Bee: This bee moves in the arena rarely.

Figure 1.1.: Behaviour of young honeybees: typical trajectories of the different types.

attributes of global swarm behaviour and can be used to optimise the swarm decision making process.

1.3. BEECLUST Algorithm

The BEECLUST algorithm is a state-of-the-art algorithm for robot swarms and is derived from the aggregation behaviour of young honeybees. Those bees are able to find a spot with their preferred temperature collectively, although this spot cannot be found by most of the single honeybees[54]. Agents controlled by this algorithm are able to find a certain point of interest in an area without using explicit communication, memory, ego-positioning and permanent measurements of the environment [31, 47, 26]. Thus, it can be used for swarm robots with very limited actuation and sensing capabilities. This algorithm is very simple but the swarm is still able to find a certain predefined point of interes (e.g. hottest or brightest spot in the arena). It is implemented as follows:

1. Each agent moves around in the arena according to its behavioural type until there is a collision with another agent or an obstacle.
    a) If the agent collides with an obstacle it makes a random turn and moves again according to its behavioural type.
    b) If the agent meets another agent it stops, measures the tem-

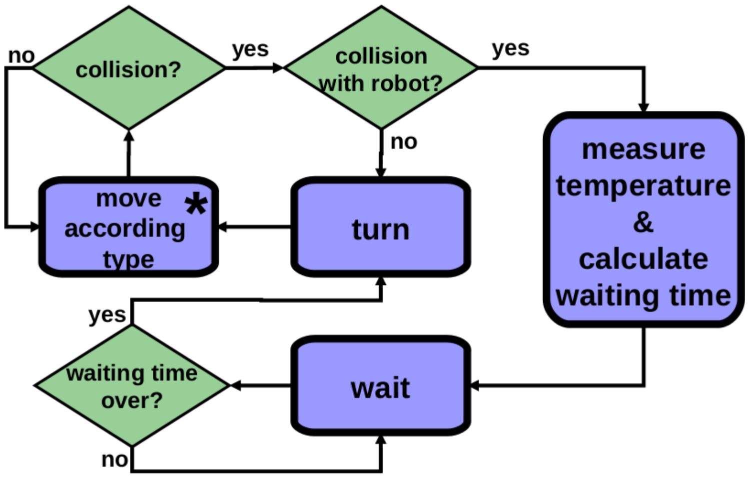


Figure 1.2.: BEECLUST algorithm derived from the behaviour of young honeybees.

perature once and calculates a waiting-time depending on the measured temperature.

2. If the calculated waiting-time is over, the agent turns randomly and moves around again according to its behavioural type.

The origincal BEECLUST algorithm as reported in [47] is based only on *Random Walkers*. and no other behavioural types are used.

A state-machine of the algorithm is shown in figure 1.2.

A formal description of the BEECLUST algorithm is shown in figure 5.14.

```
1.) Each agent moves straight until it
    perceives an obstacle O within
    sensor range.
2.) If O is a wall the agent turns
    away and continues with step 1.
3.) If O is another agent, the agent
    measures the local potential field value.
    The higher the scalar field value the
    longer the agent stays still.
    After this waiting period, the
    agent turns away from the other
    agent and continues with step 1.
```

Figure 1.3.: The BEECLUST algorithm [24].

1.4. Contribution

This thesis deals with fundamental mechanisms of swarm behaviour of young honeybees. As mentioned in the last chapter, the BEECLUST algorithm was already tested in swarm robots "Jasmin 2" with a light gradient. The BEECLUST algorithm was derived from experiments with young honeybees under a temperature gradient. Therefore, in this thesis a similar setup was created for robot experiments with a temperature gradient to bring the experiments closer to the situation honeybees are faced with. In contrast to robot experiments with light, this brings along the following challenges:

- warming up period of optima
- thermal diffusion
- turbulences in the airflow
- warming up of the environment
- time delays in measurement of temperature

These points increase the difficulty of the experiments and had to be considered while designing the arena, the extensions of the robot and also for designing the experiments.

Three different experiments were designed: ”impaired sensors“, ”dynamic environment“ and ”social seed“. In the experiment called ”impaired sensors”, some robots have broken temperature sensors and cannot measure the temperature. Therefore, they have always a waiting time of $t = 0s$ and cannot be part of an aggregation. It is analysed whether robots with impaired sensors do harm the swarms' decision making process or if they can still help other swarm members to form an aggregation. This experiment is done in simulation as well as real robot experiments. The second scenario is called ”dynamic environment“. Here, the optima are changing over time and it is tested, if the swarm is able to react on environmental changes even in a noisy temperature gradient. This experiment is done with real robots. The last experimental setting is called ”social seed”. This experiment shows, if the decision making can be influenced by a second kind of gradient and if the system reacts to a social stimulus.

In the second part of this work, different behaviour types (that were found in experiments with young honeybees before) are introduced to the BEECLUST algorithm in two different ways. In the first model the four types are modelled individually with a classical approach. An analysis is shown how the performance changes when combining each type with the *Random Walker*. The second model is an ODE-model where the four behaviour types could be generated by only varying two parameters. Here an evolutionary algorithm is used to determine a swarm composition of the four behaviour types so that the swarm performs better in a specific experimental setting than with only *Random Walkers* (as the original BEECLUST algorithm is defined with only *Random Walkers*).

The main scientific questions are:

- Is the efficiency of the BEECLUST algorithm affected by impaired agents that are not able to measure temperature?
- How stable is the algorithm against agents with malfunctioning sensors?
- How are the social and the environmental gradient's strengths balanced?
- What is the influence of the distribution of the behaviour types on aggregation speed and robustness?
- What is the benefit of introducing different behaviour types to the BEECLUST algorithm although only a random-walking behaviour is enough to reach the goal?

- Does a more complex algorithm have a better performance in certain environments than a simple one?

1.5. Outline

In chapter 1 an overview of swarm intelligence and why it is interesting for the field of robotics is given (section 1.1). In section 1.2 an introduction of the behaviour of young honeybees is given and in section 1.3 the BEECLUST algorithm that was derived from that behaviour is described.

Chapter 2 discusses related research.

Chapter 3 describes the hardware and also the experimental setup. In section 3.1 the swarm robot *e-Puck* and its modifications and extensions (section 3.1.1 & 3.1.2) are described. Section 3.2 describes the experimental setup that is used for most of the experiments. Potential changes to the setup are described in the respective chapter.

In chapter 4 the BEECLUST algorithm is analysed under three different conditions: robots with impaired sensors 4.1, dynamic environment 4.2 and experiments with a social seed 4.3.

Chapter 5 deals with the introduction of different behaviour types to the

BEECLUST algorithm. Two models are proposed: a model that implements the behaviour types in a common way 5.1 and an ODE-model that creates the different behaviour types by changing only two parameters 5.2.

In Chapter 6 a conclusion and also open issues and an outlook to future work is given.

2. Related Research

An algorithm for optimisation of non-linear continuous functions - the "Particle Swarm Optimization" algorithm - was developed by James Kennedy and Russell Eberhart [30]. The algorithm is based on a simple concept, so that it needs as little ressources in matters of memory and processing power as possible. The result is an algorithm that only needs a minimum of programming effort, a specification of the problem and only a few parameters for solving the problem.

There is a huge amount of papers dealing with several versions of the "Particle Swarm Optimization" algorithm. In [39] it is reported, that around 700 papers can be classified as applications of this algorithm. It is used in diverse research fields like: antenna design, biomedical, communication networks, clustering and classification, combinatorial optimisation (eg. travelling sales man problem), network, image and video, finance and economics but also in modelling and robotics.

An adaptive particle swarm optimisation (APSO) is suggested by Zhi-Hui Zhan [56]. It provides better global search efficiency by only introducing two new parameters and thus it does not increase complexity a lot in its implementation. The improvements were made by using evolutionary algorithms.

In [38] a simplified version of the Particle Swarm Optimisation is proposed by Pedersen et al.. An overlaid meta-optimiser is used to simultaneously tune parameters for multiple optimisation problems. With this method the authors could show, that the proposed simplified version of the PSO algorithm performs similar or - in some cases - slightly better in Artificial Neural Network problems.

Doctor [14] applies the PSO to search tasks for mobile robots in single and multiple target search scenarios. He shows, that the "Particle Swarm Optimization" algorithm is realiable for such search scenarios, but the optimal values for the parameters may be varying for different applications. Thus the parameters have to be chosen with regards to the task that should be solved.

Swarmintelligence is also used in technical tasks. In 1992, Marco Dorigo invented the "Ant Colony Optimization" algorithm [15]. When ants are searching for food they leave pheromone trails. If the distance to the food source is shorter, the trails gets more intense and thus, more and more ants choose this route. Marco Dorigo used this behaviour as an inspiration to solve the problem of searching the shortest path to a specified place.

Another bio-inspired algorithm called "AntNet" is used as a routing algorithm for networks and was developed by Gianni Di Caro and Marco Dorigo [11]. The basis of the algorithm is the behaviour "Stigmergy". This behaviour describes an indirect communication between agents via manipulation of the environment (here it is pheromone trails). The algorithm was compared to other routing algorithms like OSPF, SPF, BF, Daemon and SPF_1F and was amongst the best regarding data throughput, delay of packets and ressource management. In addition, the "AntNet" algorithm is robust.

The BEECLUST algorithm only has few requirements and has therefore been used more often in the last years. In [47] the authors study the BEECLUST algorithm in four different environments. They show, that a swarm of robots controlled by the BEECLUST algorithm is able to choose the brightest light source from several distinct light sources.

A study how the BEECLUST algorithm was derived from the behaviour of young honeybees is shown in [31]. The authors show a detailed analysis of the bee behaviour and how this behaviour was transferred to a micro-robotic swarm.

Two modifications of the BEECLUST algorithm were proposed by Farshad Arvin [3]: dynamic velocity and comparative waiting time. In case of dynamic velocity, he divides the arena into three zones based on the intensity of the light. In the area with high intensity of light the robots have low velocity and in the dark zone of the arena robots move faster. This behaviour leads to more robot-to-robot collisions in the bright area. For this implementation it is necessary that robots measure the light intensity continuously. The second modification - comparative waiting time - is dependent on the density of clustered robots. The calculated waiting time from the basic BEECLUST model is prolonged based on the number of neighbouring robots. Results show, that with these two modifications the swarm could aggregate at the brightest spot faster.

There are also analyses of the BEECLUST algorithm with macroscopic models. In [46] the authors analyse the swarm behaviour with two different approaches of macroscopic models: a Stock & Flow model and a spatially resolved model based on diffusion processes. Both models allow exhaustive parameter sweeps within some seconds and predict quite well the dynamics of robot aggregation at the target sites.

In [21] James Hereford analysis the effectiveness of the BEECLUST algorithm. A birth and death Markov chain is used to show that the agents aggregate at the optima and thus, is a promising algorithm for swarm search applications, where only very simple robots with limited sensors and computing capacity can be used.

A lot of research in behavioural patterns was done. Michelsen et al. [35] used a mechanical model to investigate how a honeybee perceives communication dances of other individuals. It was observed how many bees are flying to a specific bait because of the models' movement. In a second step the authors observed the behaviour of bees during simulated dances in which different components provide conflicting information about the baits. Results showed, that the wagging run is the part with most information for the bees whereas it seems that the figure-of-eight dance path does not hold any information. Although information seems to be coded in sound and wagging redundantly, both must be present in the dance.

Sumpter and Broomhead [53] studied the movement of individual honeybees in a thermoregulating cluster in a hive with the help of a multiagent-model. Disc- and ring-like cluster shapes were observed and in environments with lower ambient temperatures the shape of clusters were not always stable.

A study about flight patterns in honeybees is published in [41]. Bees are trained to an artificial feeder. To study the search patterns, the feeder was removed and the flight patterns of bees were recorded. Results showed that bees perform a scale-free (Lévy-flight) flight pattern. The authors show, that even if the Lévy-flight is imprecise, this search strategy is still optimal.

Bartumeus et al. [7] show an analysis of different random-walk models: a correlated random-walk and Lévy walks. The efficiency of both models is compared and results show, that the optimization mainly depends on the optimal temporal exectuion of reorientation events even when directional persistence is not high. The authors propose a combined model of random-walk and Lévy walk: the Lévy-modulated correlated random walk.

3. Hardware & Experimental Setup

This chapter is published in [24, 28]. It describes the robot *e-Puck*, modifications and extensions of the *e-Puck* and the experimental setup for the robot experiments and in simulation.

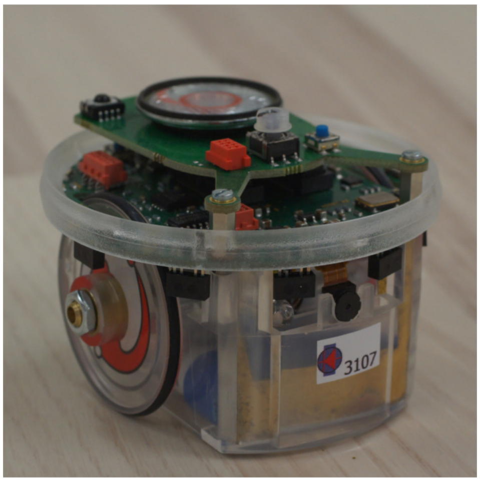
3.1. e-Puck Robot & Modifications

For our robot experiments, we used the mobile robot *e-Puck* which was developed at the École Polytechnique Fédérale de Lausanne (EPFL) for educational purposes at universities [37]. It has a 16 bit Microchip dsPIC microprocessor running at 64 MHz, which results in a peak processing power of 16 MIPS. Programming is done in the C programming language with a custom GCC compiler that is provided by Microchip. The e-puck features a broad array of different sensors, however, only the infrared (IR) proximity sensors are used here. In mobile robotics, IR proximity sensors are mostly used for simple proximity measurement. In this work the fact that the IR proximity sensors consist of an active and a passive part - namely an IR-emitting diode and a receiving phototransistor - is used to implement a robot-to-robot detection. The body of the robot has a set of red and green leds embedded to determine the status of the robot. Possible values are: "standby", "moving", "waiting" and "error".

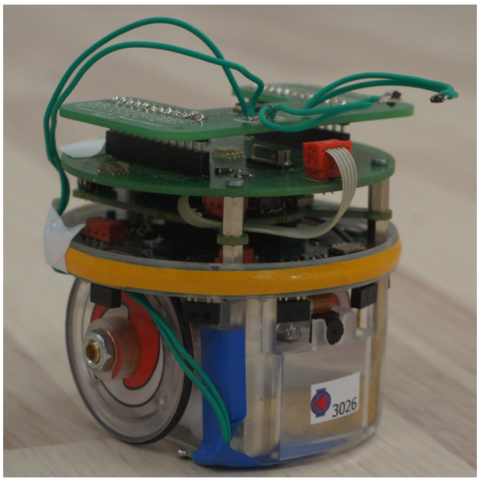
A bluetooth transceiver provides the possibility to communicate with an external computer and - in combination with the bootloader - a way to upload new programs to the flash memory.

3.1.1. Extension-Board

In order to provide a way of mounting additional sensors onto the *e-Puck*, an extension board has been designed. The extension board uses the UART serial interface of the *e-Puck* for interaction of the extension board with the robot itself. An Atmel Atmega8 microprocessor is used to allow easy interaction with various sensors. It runs at 16 MHz and offers 32 KB of flash memory and 2 Kb of SRAM, which is sufficient for this application. Directly on the board there are three color-sensors and an extension-header.



(a) E-puck without extension board



(b) E-puck with extension board and two temperature sensors imitating bees' antennae and a ground temperature sensor which is used in our experiments.

Figure 3.1.: Swarmrobot "e-Puck" used for our robot experiments.

3.1.2. Temperature Measurement Board

As the *e-puck* has no temperature sensors, we designed an extension board for the temperature measurement which uses the extension-header of the extension-board as an interface. On this header, a separate board for temperature measurement is mounted. Measurement is performed by a MAXIM MAX6636 integrated circuit, which supports up to 6 external temperature sensors and provides access to the data via a system management bus (SM-Bus). Temperature measurement is done using the temperature depended characteristics of diodes. In this case, PNP transistors with joined base and collector are used.

Two transistors are mounted in a way to imitate antennas, a third one is slightly above the ground. For our experiments we use only the ground sensor since the sensors measuring air-temperature have proven to be unreliable and extremely slow. Also only the ground temperature can be direclty observed by an IR camera.

Figure 3.1 shows an e-puck robot with its extension board and temperature sensor.

3.2. Standard Experiment

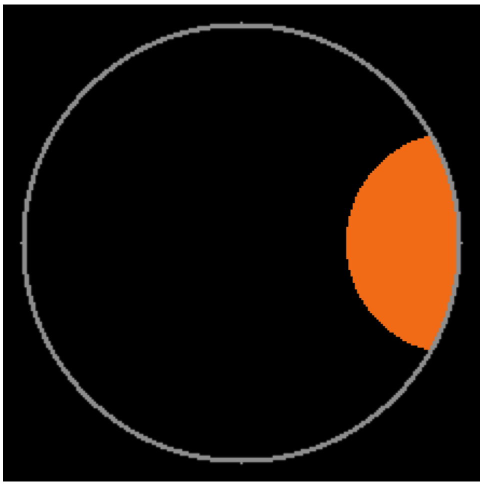
In the following, a standard-experiment is described that is used in most studies as a reference-experiment.

In the standard-experiment, the agents (meaning simulated agents and robots equally) move in a circular arena with either one goal area or two goal areas (see Fig. 3.2). Following our inspiration from the honeybee experiments we define the potential field P as a temperature field here. The potential field P is chosen in a way that there is a global optimum at the right-hand side and an optional local optimum at the left-hand side of the arena.

In the experimental settings, these optima are located at the wall, see Fig. 3.2. There are two standard experiments:

1. One global optimum at the wall
2. One global and one local optimum at the wall, called the choice-experiment

In the first experiment (Fig. 3.2(a)), there is an orange area around the global optimum on the right side. This area is called *global goal area* and has temperatures of 30°C to 36°C ranging from the boundary between the orange and black area to the wall.



(a) One global goal area at the right hand side of the arena.

(b) A local goal area at the left-hand side and a global goal area at the right-hand side.

Figure 3.2.: Experimental setup of the arena: the global goal area is located at the right-hand side of the arena and contains temperatures between 36°C and 30°C. The local goal area at the left-hand side of the arena contains temperatures between 32°C and 30°C. Each of the goal areas covers 11% of the arena.

In the second experimental setup (Fig. 3.2(b)) we create a binary choice-experiment that is often used to investigate swarm-intelligent behaviours and algorithms ([49, 52, 47, 51, 22]). Additionally to the *global goal area* on the right side, there is an orange area around the local optimum on the left side. This area is called *local goal area* and has a maximal temperature of 32°C at the outer side and 30°C at the boundary between the orange and black area. Each of the goal areas covers 11% of the total arena.The black area inside of the arena is called *Pessimum* (area that is neither the *Global Goal* nor the *Local Goal* inside the arena). The heat sources of the goals create a temperature gradient of 30°C to 20°C from the border of the goals to the middle of the arena.

These two experimental settings are used in most of the experiments. Any changes of these two settings are described in the respective chapter.

4. Analysis of the BeeClust-Algorithm [24, 28]

This chapter is based on the work published in [28, 29, 26, 34].

4.1. Impaired Sensors

During experiments with swarm robots, we raised the question if robots that are not fully functional do harm the performance of the swarm consisting of fully functional robots or if they can still help other swarm members. This is especially relevant if the presence of robots in the arena has other purposes in addition to finding the *global optimum*.

In the experiment here, the performance of a swarm of intact agents aiming to find the *global optimum* is investigated in presence of additional agents with malfunctioning temperature sensors. All the agents are controlled by BEECLUST algorithm. The impaired agents cannot measure the temperature and therefore never stop, but they can still create collisions and trigger other agents to measure the local temperature.

4.1.1. Simulation

Results: Impaired Sensors

In simulation we address the question, how the swarm is affected if sensors of agents break without being noticed by the observer.

Figure 4.1 shows the simulation results of experiments with impaired sensor agents. Here, the proportion of impaired agents to functional agents is varied. In the experiment with 10 functional agents, the median amount of agents in the *global goal area* is 7. Replacing functional agents by agents with impaired sensors one by one decreases the median amount of agents in the *global goal area* significantly. But statistics also shows, that the swarm can discriminate between the *global goal area* and the *local goal area* even when the majority of the swarm are agents with impaired sensors. Significances are tested with the Wilcoxon-Test (*global optimum* against *local optimum*) and the U-Test (global optimum of one experiment against global optimum of another experiment) with $p < 0.05$.

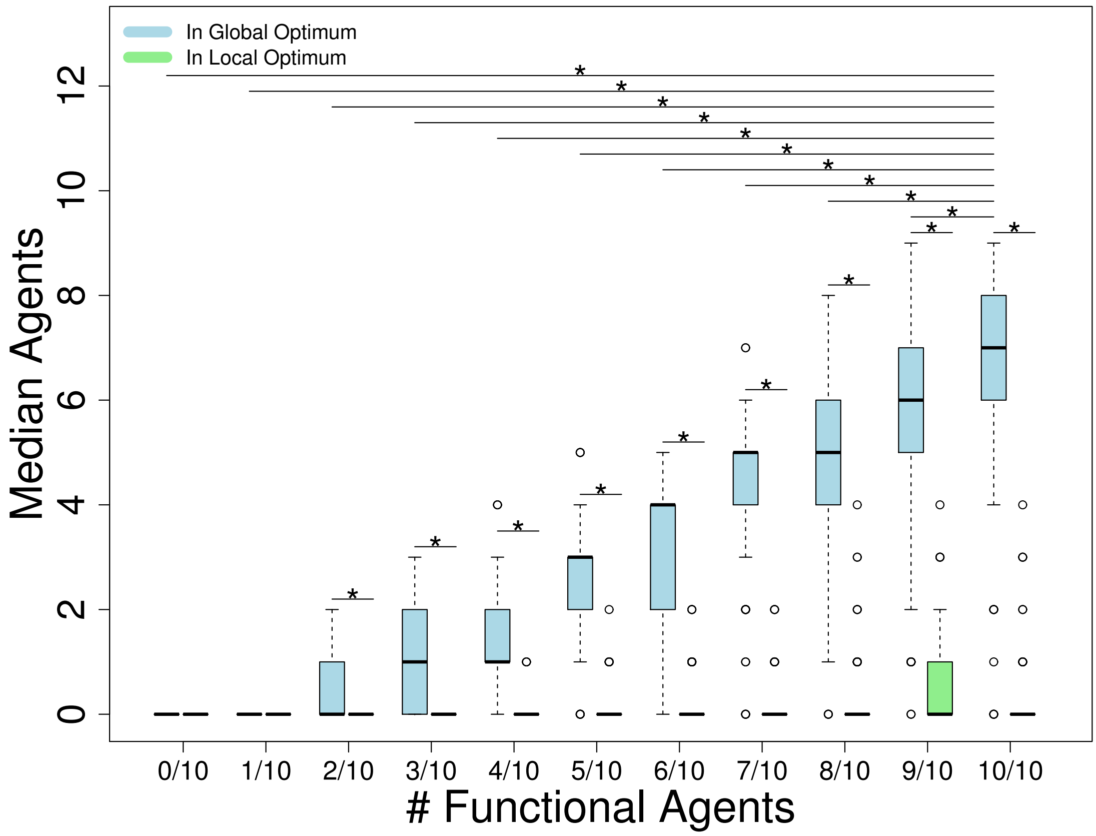


Figure 4.1.: Simulation results of experiments with different proportion of functional agents to impaired agents: Significances are tested with the Wilcoxon-Test (*global optimum* against *local optimum*) and the U-Test (global optimum of one experiment against global optimum of another experiment) with $p < 0.05$.

Discussion: Impaired Sensors

In this scenario, we study how agents that cannot measure temperature (thus, have always a waiting time of $t = 0s$) affect the performance of the swarm. Our results show, that the median amount of agents in the *global goal area* decreases when the amount of agents with impaired sensors increases. However, a decision can be made by the swarm even when only 2 functional agents (and 8 agents with impaired sensors) are present. We think, that the decision making process is still possible, because agents with impaired sensors trigger measurements of the temperature of the fully functional agents. If the fully functional agent stops in the *global goal area*, it operates as a "social seed" (see Section 4.3) and thus, attracts other agents until the waiting time is over.

4.1.2. Robot experiments

In this scenario, we study how the performance of a swarm of fully functional robots changes, if robots with impaired sensors are added to the swarm. Note, that here fully functional agents are not replaced by impaired agents but are added to the swarm. This addresses the question, if impaired agents can still help other swarm member by triggering measurements of the environmental gradient.

The intact swarm in this experiment consists of 3 fully functional agents. To measure the effect of impaired agents, we added 7 agents that are not able to measure the temperature. In the control experiment, the 3 fully functional agents were used. We used only 3 agents here, because the aggregation with 3 agents is stable enough to measure a decrease of performance if the impaired agents harm the decision making process. On the other hand, to measure if the performance can be increased with the help of impaired agents, the aggregation of 3 agents is instable enough so that there is potential to improve the aggregation.

Results: Impaired sensors

In figure 4.2 the results of robot experiments with impaired sensors are shown for both the control experiment and the impaired sensor experiment. For the control experiment, the median number of robots in the *global optimum area* is 1 and in the *local optimum area* is 0 (here the swarm consists of 3 fully functional robots). In the experiment with impaired sensors where the swarm consists of 3 fully functional and 7 impaired robots that cannot measure the temperature (thus having a waiting time of $t = 0s$), the median number of robots in the *global optimum area* is 1 and in the *local optimum area* is 0. Note that in both experiments (control and impaired) only the fully functional robots are counted. The results of the two experiments are not significantly different (tested with U-Test and $p < 0.05$).

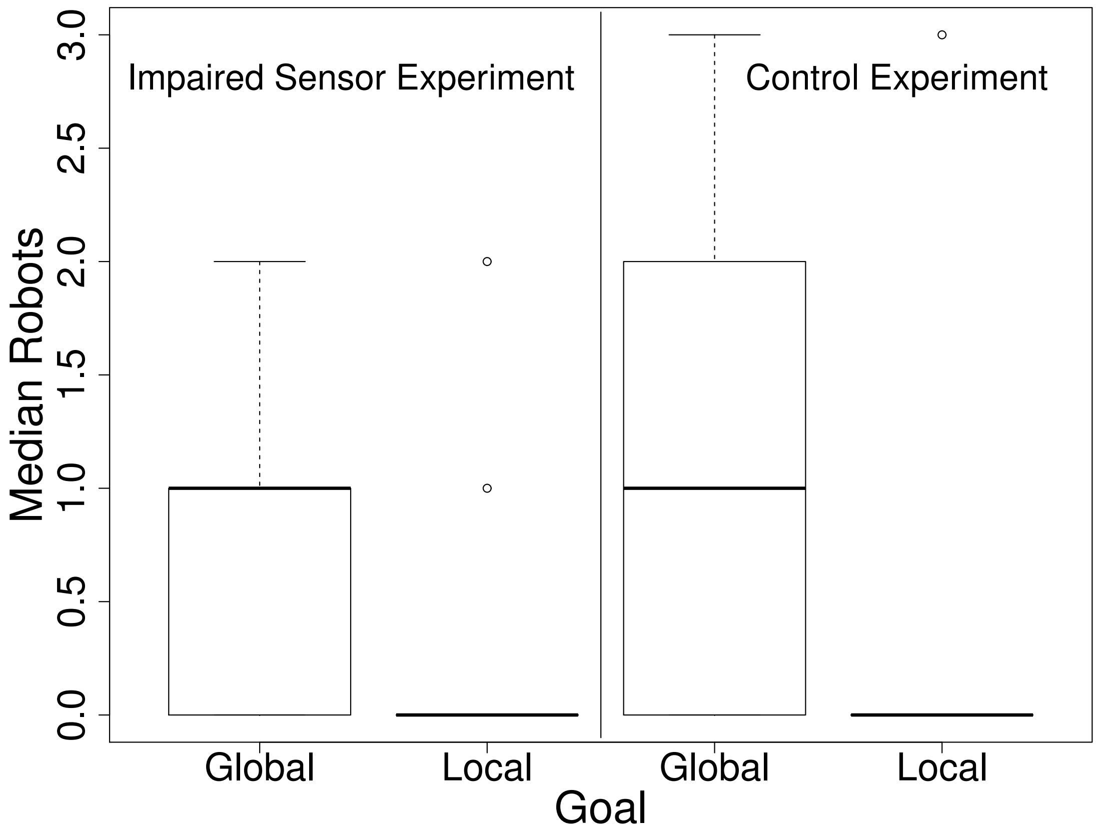


Figure 4.2.: Median number of robots in the two areas. Left: Impaired sensor experiment with 3 fully functional and 7 impaired robots. Right: Control experiment with 3 fully functional robots. There are no significant differences. Significances are tested with the Wilcoxon-Test (*global optimum* against *local optimum*) and the U-Test (optimum of one experiment against the same optimum of another experiment) with $p < 0.05$.

Discussion: Impaired sensors

Our results of robot experiments in a temperature gradient show that even adding a big number of impaired robots (more than 100%) does not harm the swarms decision making process significantly, although an aggregation of only 3 robots is unstable even without any disturbances. The aggregation stays unstable also in presence of the impaired robots. Thus adding impaired robots does not have any significant effect neither increasing nor decreasing the performance of the main swarm controlled by the BEECLUST algorithm.

4.1.3. Conclusion: Impaired sensors

In simulation we could show, that replacing fully functional agents by impaired agents does affect the performance of the swarm but the decision making is still possible even when the majority of the swarm are agents with impaired sensors.

Experiments with robots showed, that adding robots with impaired sensors to a swarm of fully functional robots does not improve the performance of the swarm but they also do not decrease the performance significantly.

Thus, we conclude that the BEECLUST algorithm is robust against agents with impaired sensors. The performance of the swarm decreases but the swarm is still able to discriminate between the *global goal area* and the *global goal area*.

4.2. Dynamic Environment

In [47] it was shown in a light gradient, that agents controlled by the BEECLUST algorithm are not only able to find the global optimum out of several local optima, but they are also able to react on environmental changes like changing the global and local optima. Here we want to show that it is also possible under more sophisticated environmental conditions like in a temperature gradient.

The experiment is carried out in two steps. The setup in the first step is identical to what is described in Sec. **??** with a *global* and *local optima* located respectively at the right and left side of the arena. The first step consists of a single observation phase and lasts for 15 minutes. At the beginning of the first step, robots are released in the middle of the arena. After 15 minutes, the second step starts with switching off the heat source of the *global optimum*. The second step consists of two phases which last for 5 and 10 minutes respectively. In the first 5 minutes the *global optimum* cools down to the ambient temperature (28°C). Then in the next 10 minutes the formerly called *global optimum* keeps a median temperature of 28°C and the formerly called *local optimum* becomes the new global optimum. Without changing the heat source of the former local optimum, its temperature decreases to 30°C due to the absence of the heat source of the former global optimum. The duration of a complete experiment is 30 minutes. See figure 4.3 for the changes in the median temperature during the three phases.

4.2.1. Results: Dynamic environment

The dynamic environment experiment was created to demonstrate that agents controlled by the BEECLUST algorithm are flexible in their decision-making in a dynamic environment even where the environment is very inert and hard to control (like the temperature-gradient). Figure 4.3 shows the changes in temperature over time. In phase 1, both optima, *global optimum area* and *local optimum area* are present. After 15 minutes, the heat source of the *global optimum area* is switched off and a cooling phase of 5 minutes starts (phase 2). Then only one optimum is present (phase 3). In figure 4.4 the corresponding distribution of robots over time is shown. In phase 1 all robots start in the middle of the arena. After a short period, most of the robots are aggregated in the *global optimum*. In phase 2, the *global optimum* is shut down and therefore the cluster dissolves. After a few minutes, the robots start to cluster in the *local optimum* which is now the new global optimum in the arena. At the end of the experiment most robots are clustered in the formerly called *local optimum*.

Figure 4.5 shows the median number of robots in the *global optimum* and *local optimum* for the different phases. In phase 1 the median number of robots in the *global optimum* is 7 and in the *local optimum* is 0. The median number of robots in the cooling phase is 6.5 and 0 in the *global optimum* and in the *local optimum*, respectively. In the last 10 minutes of the experiment (phase 3) the median number of robots in the *global optimum* is 0 and in the *local optimum* is 6. Statistical significances are tested with the Wilcoxon-Test (*global optimum* against *local optimum*) and the U-Test (number of robots in the optimum area of one phase against the same optimum area of another phase) with $p < 0.05$. All boxplots in the figure are significantly different to the other boxplots, except *global optimum* of phase 1 with *global optimum* of phase 2, and also *local optimum* of phase 1 with *local optimum* of phase 2.

4.2.2. Discussion: Dynamic Environment

Results of experiment with a dynamic environment in a temperature gradient showed that robots controlled by the BEECLUST algorithm are able to react on changes in the environment reliably. In [47] this experiment was performed in a light gradient and was shown that the robots were able to choose the brightest source of light out of several light spots and that they also were able to react on environmental changes. We performed this experiment in a physically more complex environment. Compared to light, temperature differs in its physical characteristics for example in: warm-

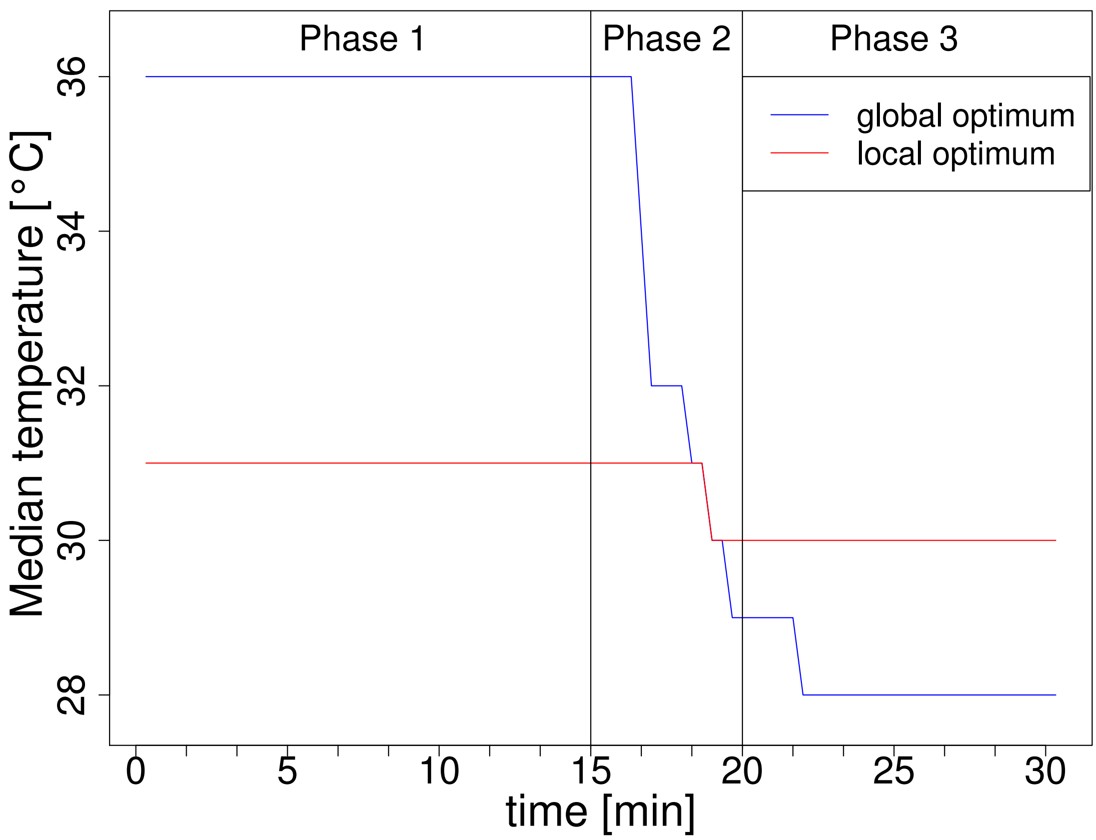


Figure 4.3.: Course of temperature in the different phases over time in the dynamic environment experiment.

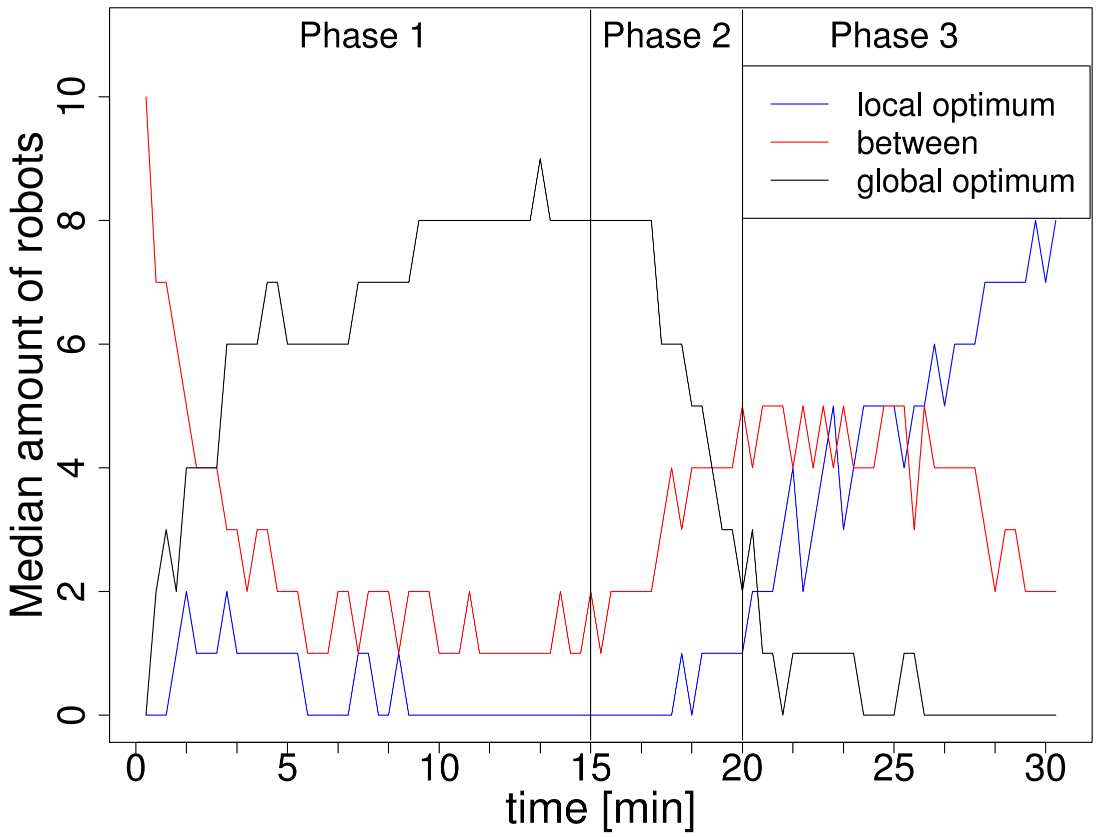


Figure 4.4.: Course of experiment in the dynamic environment: Median number of robots in the different areas and phases.

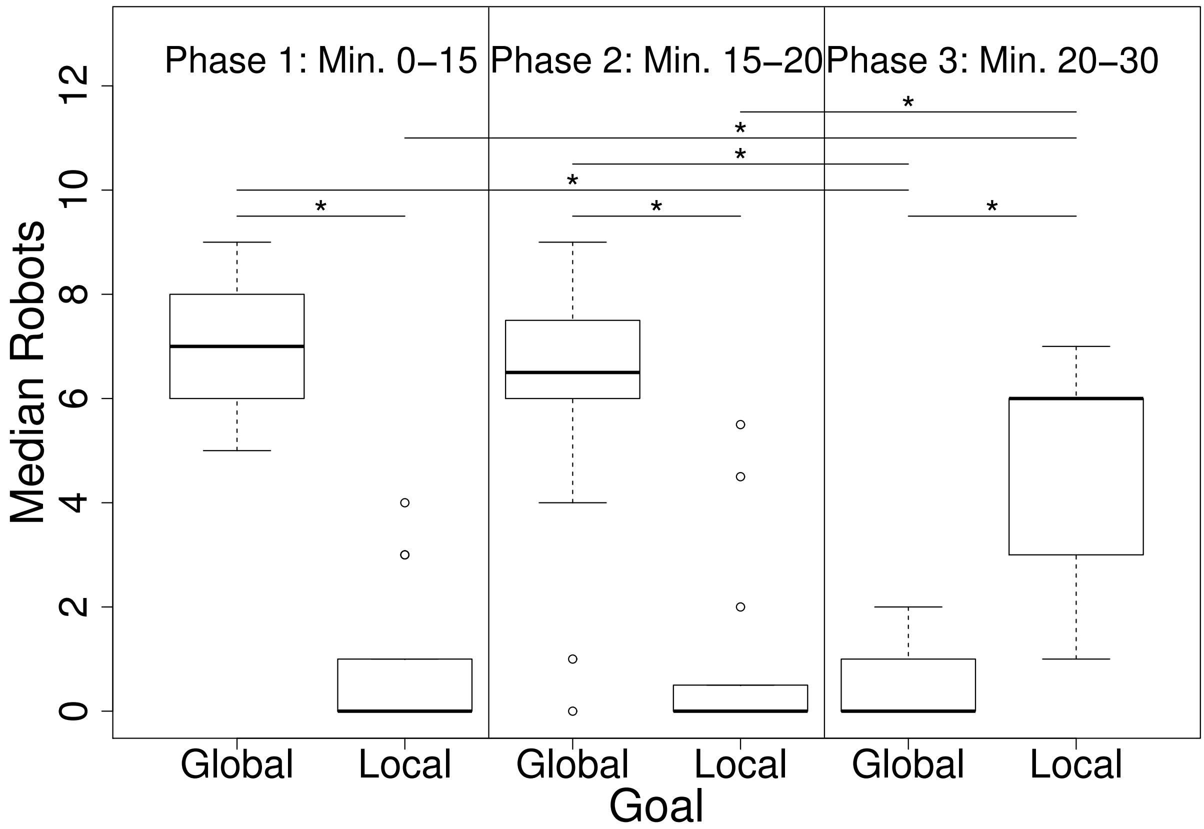


Figure 4.5.: Median number of robots in the respective area over time per phase. Results are significantly different, except in the case of *global optimum* of phase 1 with *global optimum* of phase 2, and also *local optimum* of phase 1 with *local optimum* of phase 2.

ing up period, cooling down, thermal diffusion, turbulences in the airflow. As the experiment lasted 30 minutes, there was also an air-conditioning necessary so that the temperature of the room (28°C) was stable. Another challenge of working with temperature is the time delay of the measurement. Because of this, after the measurement of temperature was triggered, we had to measure it again a few seconds later and correct the first measurement. Although these physical conditions make experiments more complex, it is shown that robots controlled by the BEECLUST algorithm are still able to react on environmental changes. After switching off the heat lamp of the *global optimum area* the aggregation started to dissolve in the *global optimum area* 2 minutes later and after another 3 minutes robots start to form a cluster in the *local optimum area* (which is now the new global optimum).

4.2.3. Conclusion: Dynamic Environment

We conclude that robots controlled by the BEECLUST algorithm are still able to react on environmental changes although the physical conditions in a temperature gradient are more difficult (e.g. inert and unstable) than in a light gradient.

4.3. Social Seed

The following section deals with another feature of the BEECLUST algorithm: “Can the decision-making be influenced by an additional kind of gradient?” An easy way to create a second kind of gradient is to place immobilized agents into the arena and thus creating something we define as “social gradient”. In the BEECLUST algorithm there is only a minimal social component modelled which is the discrimination of an obstacle and another agent. However, we assume that the system reacts on the social stimulus without changing the original algorithm and therefore the minimal social component plays an important role.

[1] showed that a social component can improve the success of algorithms based on genetic evolution or individual learning. An experimental setup was created artificially to analyze the social component in the BEECLUST-algorithm: Although the optima are typically not known a priori, immobilized agents were placed into the suboptimum allowing us to understand the effects of the social component of the algorithm. By investigating this behaviour our goal is to create new hypotheses for the swarm research of young honeybees and other social species. Please note, that these experiments aim for an improved understanding of the BEECLUST algorithm and not improvement of the efficiency.

It has been previously shown in [1] that the success of algorithms based on genetic evolution or individual learning can improve by adding a social component. Here we used a social seed as the social component in the arena. The social seed is a robot that is immobile but its social effect is the same as other robots meaning that other robots can perceive its presence and react to it as a robot. In a previous work [26] we investigated the effect of social seeds in simulation.

4.3.1. Simulation

In the experiments described in [25, 31, 47], the BEECLUST algorithm was derived and tested with a single kind of gradient: a light- or temperature-gradient. The agents, regardless of the type of agents, were always able to determine the global optimum. In [47] the algorithm is also tested with two gradients of different intensity and also in dynamically changing environments, showing that the algorithm is flexible enough to react on these dynamic changes of the environment.
The cooperation of two swarms with different waiting time curves is investigated: It is shown that the two swarms benefit from the cooperation with each other if there is only a small swarm [8].

In this chapter we present the simulation results of experiments in which we investigate how a second, different kind of gradient influences the decision-making of the bio-inspired swarm-algorithm. This chapter deals with the following questions:

H1: Is the clustering behaviour of the BEECLUST algorithm sensitive to a social gradient?

H2: Does a swarm have to trade off between two different gradients?

H3: How many social agents are necessary to influence the aggregation behaviour?

To test the hypotheses we designed four different experiments (figure 4.6):

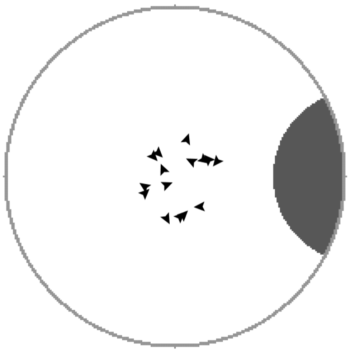
1) This experiment is used as a reference-experiment. Here we test the BEECLUST algorithm for the given arena with the global optimum on the right side and without the suboptimum on the left side (figure 4.6(a)).
2) In this experiment we additionally provide the suboptimum on the left side of the arena (figure 4.6(b)).
3) To test how a social stimulus affects the behaviour of aggregation in experiment 3) we place immobilized agents in the suboptimum

with 32°C and a dummy-agent in the global optimum with 36°C to avoid side-effects (e.g. jamming-effects) (figure 4.6(c) and 4.6(d)). This experiment is conducted with different numbers of social agents to demonstrate how the system reacts to different sizes of a social seed:

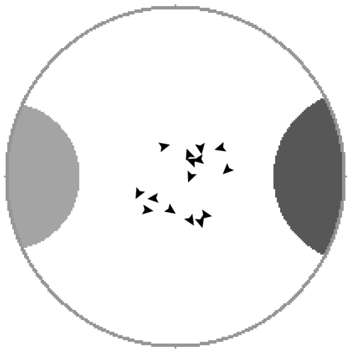
(a) with 1 agent acting as a social gradient (as depicted in figure 4.6(c)).
(b) with 2 agents
(c) with 3 agents and
(d) with 4 agents (as depicted in figure 4.6(d)).

Each experiment was repeated 100 times. At the beginning of each experiment the agents are placed randomly inside a central area which has a diameter of 10 agent-lengths and is located in the middle of the arena (figure 3.2(b)). In all six experiments 15 agents perform the BEECLUST algorithm with identical parameter settings. The agents move around in the arena with a speed of two agent-lengths per second. Agents which generate the social stimuli are immobile and do not perform the BEECLUST algorithm. To ensure that placing agents into the suboptimum has no side-effects (e.g. regarding jamming-effects due to overcrowding of an optimum) we also place dummy-agents into the global optimum which are perceived as obstacles and not as an agent.

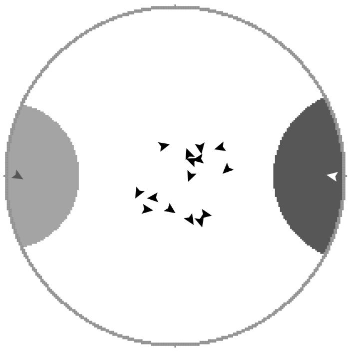
Changes to the original BEECLUST Algorithm as published in [47] The BEECLUST algorithm is not changed in its sequence. We didn't have to



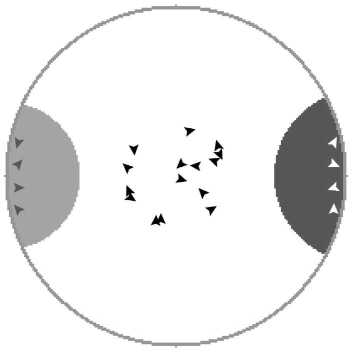
(a) Arena with global optimum



(b) Arena with local and global optimum



(c) Arena with one social agent placed in the suboptimum on the left side (dark-gray triangle) and a dummy-agent in the global optimum (light-gray triangle) which is perceived as an obstacle by the other agents.



(d) Arena with four social agents placed in the suboptimum on the left side (dark-gray triangles) and four dummy-agents in the global optimum (light-gray triangles) which are perceived as obstacles by the other agents.

Figure 4.6.: Different experimental settings. The dark-gray area on the right side of the arena indicates the global optimum with a temperature of 30°C - 36°C. The light-gray area on the left side is the local optimum with 30°C - 32°C. The black triangles indicate agents that are controlled by the BEECLUST algorithm.

adapt the algorithm so that it responds to the social seed.

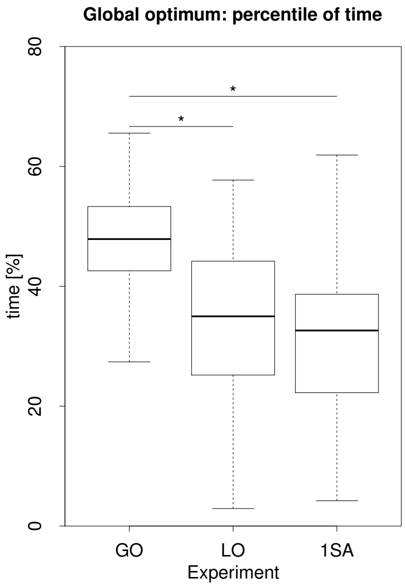
Results: Social Seed

On the x-axes of figure 4.7(a) and 4.9(a) the abbreviation "LO" is referring to experiment (2)) with only a global and a local optimum. "1SA", "2SA", "3SA" and "4SA" are referring to experiments with a social gradient with 1, 2, 3 or 4 social agents, respectively.

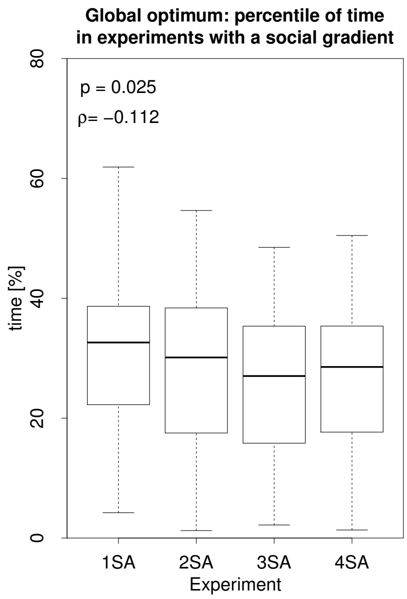
The time the agents spent in the global optimum is shown in figure 4.7(a). In experiment 1) ("GO") with just one optimum of 36°C the median time the agents spent in the global optimum is 47.88%. The median time in the experiment (2)) with a local optimum ("LO") is 34.98% and with 1 social agent ("1SA") the median time is 32.62%. The results of experiment 1) is significantly different to the results of experiment 2) and (a). The significances were tested with a level of p=0.05 (Wilcoxon-Mann-Whitney-U test).

Figure 4.7(b) shows the time the agents spent in the global optimum of the experiments with a different amount of social agents. The median times of experiments with 1, 2, 3 and 4 social agents are 32.62%, 30.12% , 27.03% and 28.54%, respectively. Here, the significances were tested with Spearman-statistics and showed no significant correlation between the amount of social agents and the time the agents spent in the global optimum ($p = 0.025$ and $\rho = -0.112$).

Figure 4.9(a) shows the percentage of time the agents spent in the local

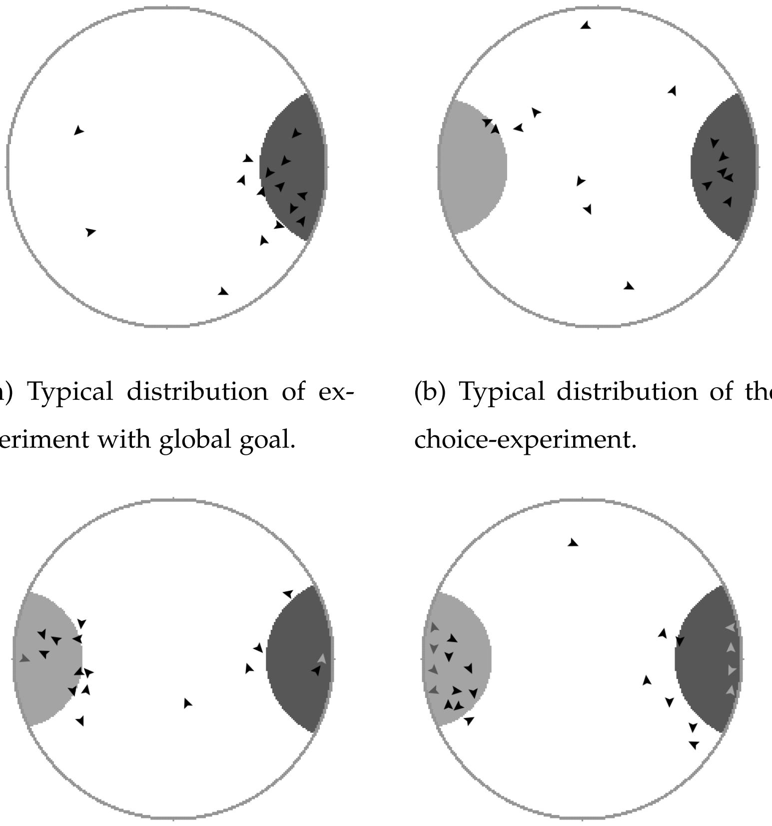


(a) Percentile of time the agents spent in the global optimum. The plot shows the median with 1st and 3rd quartile. n=100. The bars with asterisks indicate significances at a significance-level of p=0.05 and were tested with the Wilcoxon-Mann-Whitney-U test (nominal scaled).



(b) Percentile of time the agents spent in the global optimum in experiments with an increasing amount of social agents. The statistics is made with Spearman-statistics (ordinal scaled).

Figure 4.7.: Time the agents spent in the *global goal area* in the different experiments.



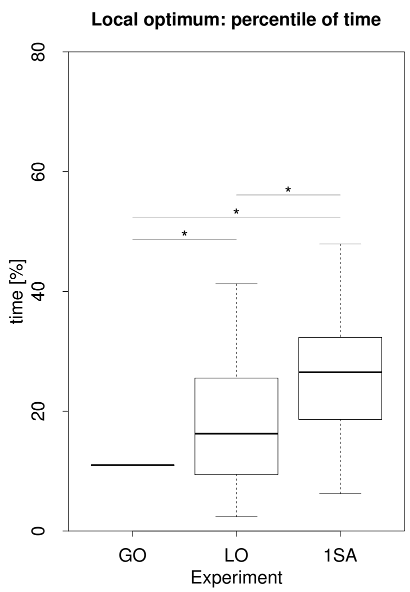
(a) Typical distribution of experiment with global goal.

(b) Typical distribution of the choice-experiment.

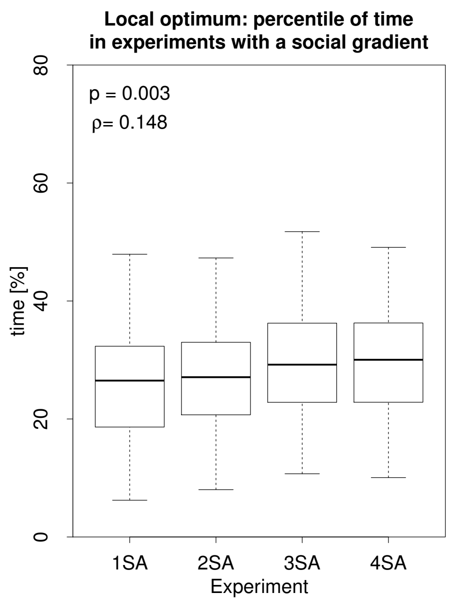
(c) Typical distribution of experiment with one agent indicating the social gradient placed in the suboptimum (left side, dark-gray triangle) and one dummy-agent in the global optimum (light-gray triangle) which are perceived as obstacles by the other agents.

(d) Typical distribution of experiment with four agents indicating the social gradient placed in the suboptimum (left side, dark-gray triangles) and four dummy-agents in the global optimum (light-gray triangles) which are perceived as obstacles by the other agents.

Figure 4.8.: Examples of results of different experimental settings. The dark-gray area on the right side of the arena indicates the global optimum with a temperature of 30°C - 36°C. The light-gray area on the left side is the local optimum with 30°C - 32°C. The black triangles indicate agents that are controlled by the BEECLUST algorithm.



(a) Percentile of time the agents spent in the local optimum. The plot shows the median with 1st and 3rd quartile. n=100. The bars with asterisks indicate significances at a significance-level of p=0.05 and were tested with the Wilcoxon-Mann-Whitney-U test (nominal scaled).



(b) Percentile of time the agents spent in the global optimum in experiments with an increasing amount of social agents. The statistics is made with Spearman-statistics (ordinal scaled).

Figure 4.9.: Time the agents spent in the *local goal area* in the different experiments.

optimum. The time for experiment 1) ("GO") is calculated with a uniform distribution model, due to the fact that in this experimental setting no local optimum is available. As the defined area of the local optimum covers 11%, the agents would spend 11% of the time in this area. In experiment (2)) the median time spent in the local optimum of 32°C was 16.25%. The median time for experiment ((a)) with one social agent is 26.50%. The results of the experiments here are all significantly different to each other. The significances were tested with a level of p=0.05 (Wilcoxon-Mann-Whitney-U test). In figure 4.9(b) the time the agents spent in the local optimum are shown for the experiments with an increasing amount of social agents. The median times for 1, 2, 3 and 4 social agents are 26.50%, 27.08%, 29.21% and 30.04%, respectively. The significances were tested with Spearman-statistics and showed no significant correlation between the amount of social agents and the time the agents spent in the local optimum ($p = 0.003$ and $\rho = 0.148$).

Discussion: Social Seed

The main feature of the BEECLUST algorithm is to find the global optimum within a complex environment, as shown in [47] experiments with light-gradients in a dynamic environment were conducted. In the following, we will discuss the three questions mentioned above:

Is the clustering behaviour of the BEECLUST algorithm sensitive to a social gradient?

The BEECLUST algorithm as tested in [47] is able to locate the global optimum in static and dynamic environments robust. In the simulation experiments we showed that this stable decision-making can be influenced by adding another gradient - a social gradient. By just using one additional agent - functioning as a source of a social gradient - we were able to increase the percentage of time in the local optimum from 16.25% to 26.50% (compare figures 4.8(b) and 4.8(c).

How many social agents are necessary to influence the aggregation behaviour?

Adding just one single social agent had a huge effect. We were able to bound agents for more than 10% of the time to the suboptimum. To reach the threshold were the agents spend more time in the suboptimum than in the global optimum, three social agents were necessary (see figure 4.7(b) and 4.9(b)). This leads us to the next question:

Does a swarm have to trade off between two different gradients?

A swarm of agents which is controlled by the BEECLUST algorithm always decides for the global optimum even if a second suboptimal gradient of the same type is present. Thus a discrimination of the local and the global optimum is possible. If there is a second gradient of another type, the decision-making of a swarm is not that clear anymore. In a weak gradient, agents which were undecided start to decide for the social gradient, but also agents from the global optimum reconsider their decision. If the social gradient gets stronger, no more agents are bound from the pessimum but some additional agents from the global optimum change their minds (see figure 4.8).

The percentage of time that agents spent in the pessimum is significantly lower in the experiments with social agents ((a), (b), (c)) compared to the experiment without a social stimulus (2)). Increasing the amount of social agents had no significant effects.

If we compare the results of experiment ((a)) with experiment (2)) it appears that fewer agents stay in the pessimum or global optimum and more agents stay in the local optimum. We can conclude that agents get bound not only from the global optimum but also from the pessimum (figure 4.7(a) and 4.9(a)). This effect can also be observed in the results of experiment ((b)). Three social agents is the minimum number of agents that are needed so that more agents place themselves in the local optimum than in the global optimum (figure 4.7(b) and 4.9(b)). Adding another social agent - in total 4 social agents - leads to no significant changes in the percentage of time the agents spent in the optima.

Figure 4.10.: The figure shows the setup of the social seed. Left: *local optimum area* with a social seed (immobilised robot). Right: *global optimum area* with a dummy robot to ensure similar circumstances of the two optima areas.

4.3.2. Robot Experiments: Social Seed

Here we are interested in investigating if the effect of a social seed also exists in robots controlled by the BEECLUST algorithm and how strong the effect is compared to the results from the simulation experiment. Therefore, we created a setup similar to the simulation setup: The main swarm consists of 10 robots. In the *local optimum* we placed an immobile robot as a social seed which can be recognised by other robots (see figure 4.10). To keep the conditions fair for the two optima (e.g. in terms of available space), we placed a dummy robot in the *global optimum*. The dummy robot is perceived as an obstacle by other robots.

Results Robot Experiments: Social Seed

In figure 4.11 the median time that each agent spends in the *global optimum area* and in the *local optimum area* are depicted for both control experiment (no social seed) and the experiment with social seed. In the control experiment, the robots spend 67% of the time in the *global optimum area* and 4.6% in the *local optimum area* (left side of figure 4.11). In the experiment with social seed (right side of figure 4.11), the median time each agent spends in the *global optimum area* is 26% and in the *local optimum area* is 45.8%. Statistical significances are tested with the Wilcoxon-Test (global optimum tested against local optimum) and the U-Test (global optimum of one experiment against global optimum of another experiment) with $p < 0.05$.

Discussion Robot Experiments: Social Seed

The experiment with a social seed was originally designed to analyse the social component of the BEECLUST algorithm. The simulation results [26] showed that the decision-making can be influenced by a social stimuli. In the simulated experiment, the minimum of three social agents were needed to be placed in the *local optimum area* in order to make the main swarm to spend more time in the *local optimum area* than in the *global optimum area*. In real robot experiments, the effect of social seed is even stronger. Here, a single robot as a social seed is enough to influence the decision making

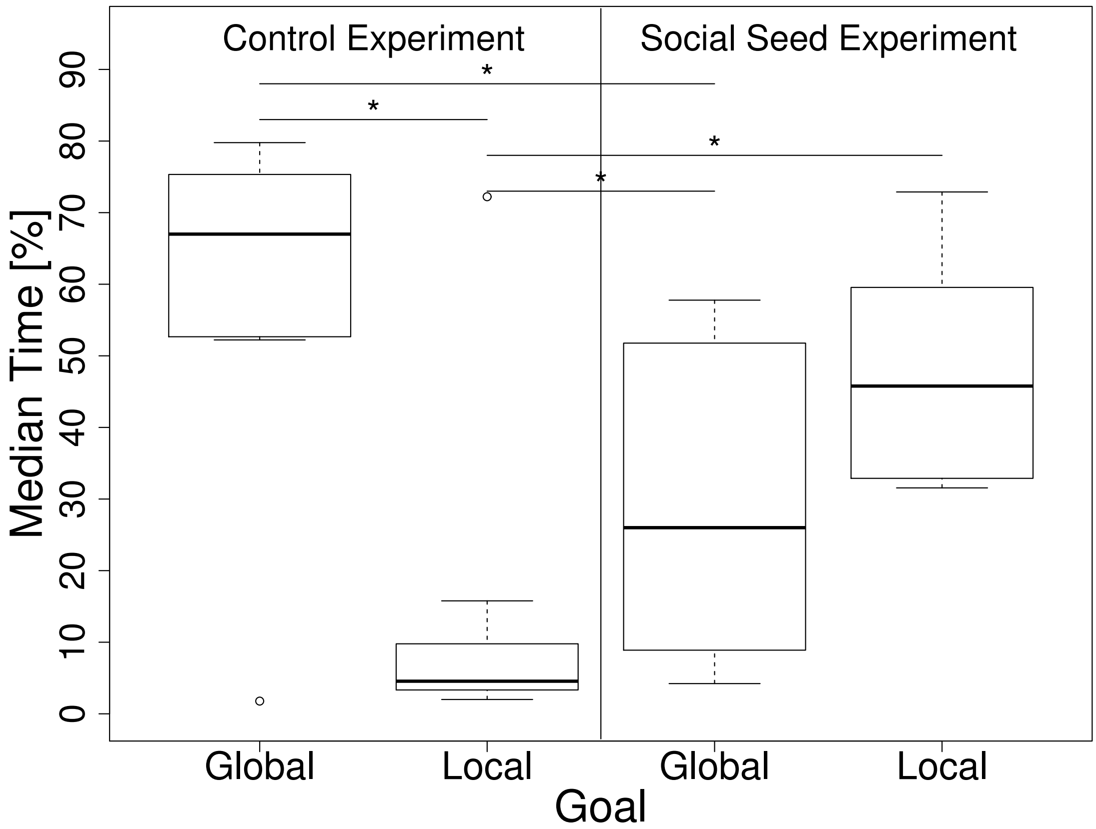


Figure 4.11.: Median time the robots spent in the respective optimum area. Left: Control-experiment with 10 robots. Right: Experiment with an immobilized agent placed in the *local optimum area*. Asterisks indicate significant differences with $p < 0.05$.

process of the swarm significantly.

Conclusion Robot Experiments: Social Seed

When placing a social seed in the *local optimum*, the effect is even stronger than in the simulation results presented in [26], although there is only a minimal social component included in the BEECLUST algorithm: namely recognition of another robot. As the BEECLUST algorithm is derived from the behaviour of young honeybees, experimenting with a temperature gradient gets us closer to the situation that young honeybees are faced with. Because of the strong effect of the social stimuli we think, that in honeybees this social stimuli can also be very strong.

4.3.3. Conclusion & Future Work

Simulation results showed that the BEECLUST algorithm can be influenced by using a social gradient induced by immobile agents placed in the *local optimum area* of the arena. In robot experiments the effect is even stronger, although there is only a minimal social component included in the BEECLUST algorithm: namely recoginition of another robot. As the BEECLUST algorithm is derived from the behaviour of young honeybees, experimenting with a temperature gradient gets us closer to the situation that young honeybees are faced with.

As the social gradient had an unexpected big effect, we also want to introduce the social gradient to experiments with real honeybees. We think, that in honeybees this social stimuli can be very strong and we expect that the decision-making of young honeybees can also be influenced by offering a second, different type of gradient. These results can then be used for further investigations of the swarm-intelligent behaviour of honeybees by creating bio-hybrid systems consisting of real honeybees and artificial autonomous robots [43].

4.4. Cooperation of Two Different Swarms

In the work at hand we investigate how honey bee's age polyethism influences this system. Age polyethism means that in a honeybee colony individuals of the same age perform the same task, and that a given task is often associated with a given age. Examples for such tasks are, collecting nectar in the environment, brood care and the cleaning of the honeycombs. The location of these tasks are not always spatially separated, but can be located near each other, or even within the same area, e.g., broodcare and wax manipulation. It was shown by [9] that agents, controlled by BEECLUST, that have identical sensors, but differ regarding their temperature optimum, are able to cooperate well in a complex environment. The question we raise here is, how good can two different groups of agents, that have different sensors (and therefore different tasks) cooperate, if the tasks are located near each other. Based on the results of [9], who described the negative influence of jamming effects on groups of agents operating in the same area, and the positive affects of cooperation subgroups we formulated the following hypothesis: Two non-identical agent swarms controlled by BEECLUST are able to cooperate for a given ratio of the agents in the two groups.

4.4.1. Experimental Setup

Here we deviate from the standard setup that is described in section 3.2. To answer the question mentioned above, we created an area of 16x16 patches. We implemented two different task-areas with a distance of 5 patches and two different agent-swarms A and B, acting parallel in the same environment. Both swarms had the same properties and comply with the rules of the BEECLUST algorithm. The two different task-areas T_A and T_B were implemented as gradients in the environment, scaling from a value of 1 in the maximum to 0 in the environment. The size of T_A was the quater of the size of T_B to simulate a highly specialised task near an area of a more general task.

As mentioned above the length of the waiting time of a single agent is determined by the local value of the gradient. The sigmoidal waiting-curve was identical for both agent-swarms A and B. The maximum waiting time for both swarms was 1000 timesteps.

We observed and analysed the percentage of agents of A aggregated at T_A. The tested population size, including A and B, ranged from 3 to 20 individuals. The ratio of A to B was varied from 0.2 to 1, rounding was always done towards the next bigger number of A. Each experiment ran for 3600 timesteps and was repeated 100 times.

4.4.2. Results and Discussion

It showed, that, by increasing the total amount of agents, the average number of A increased linearly in the target area T_A (see figure 4.12). Surprisingly it further showed, that, in contrast to our hypothesis (mentioned above), the relative amount of A in the target zone was highly stable against changes of the ratio of A to B (see figure 4.13). This means, that even a single agent can operate within a group of agents with another task (or even another sensory system) without any loss of efficiency. Due to this we can falsify our hypothesis, that the cooperation of two BEECLUST controlled groups of agents is depending on the ratio of these two groups.

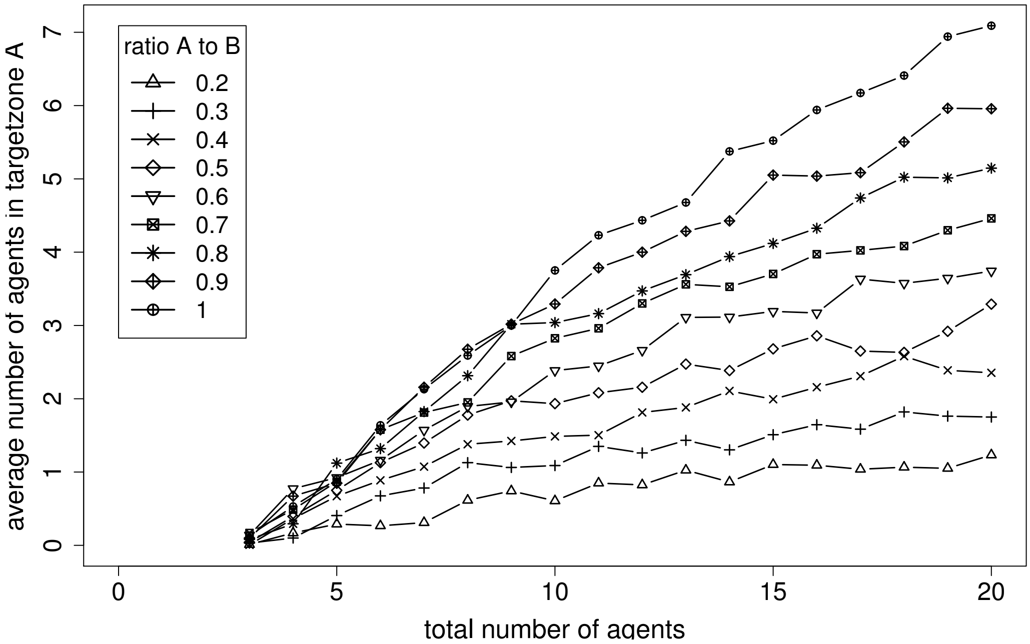


Figure 4.12.: Absolute number of A agents, aggregated in T_A; $n = 100$.

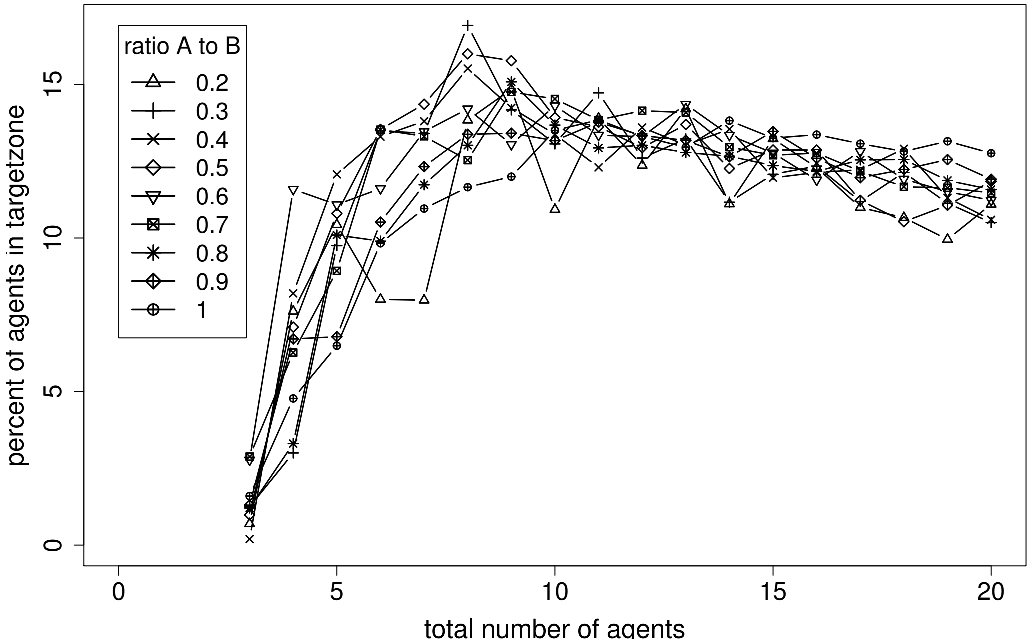


Figure 4.13.: Average percentage of A agents, aggregated in T_A; $n = 100$.

5. Swarm Level Optimisation

This chapter is part of [27] which is submitted for publication (rule-based model) and [24] (ODE-model).

5.1. Rule-based Behaviour Model

This chapter deals with the four behaviour types that were found in experiments with young honeybees. When these different movements of bees were discovered, several questions arose: What effect do the different movements have on other agents or on the swarm? Why are there different behavioural types although just a random-walking-behaviour is enough to reach the goal? What is the role of each behavioural type and how could it be used to predict or modulate the resulting aggregation-behaviour of the swarm?

To investigate these questions we introduce four behavioural types to the BEECLUST-algorithm [47], that is a state-of-the-art algorithm for robot swarms and is derived from the aggregation behaviour of young honeybees. Those bees are able to find a spot with their preferred temperature collectively, although this spot cannot be found by most of the single honeybees [54]. Agents controlled by this algorithm are able to find a certain point of interest in an area without using explicit communication, memory, ego-positioning and permanent measurements of the environment [31, 47, 26]. Thus, it can be used for swarm robots with very limited actuation and sensing capabilities. This algorithm is very simple but the swarm is still able to find a certain predefined point of interest (e.g. hottest or brightest spot in the arena). There exist various analysis of the BEECLUST algorithm [2, 3, 5, 22, 21, 31, 47, 26]. There has been a lot of other work in the domain of swarm algorithms already published, e.g. [12, 33, 40].

We assume that individual behavioural patterns contribute to different attributes of global swarm behaviour and can be used to optimise the swarm's decision making process. We want to optimise a swarm system by composing the swarm from a selection of members with specific behavioural traits. We call this concept "Swarm Level Optimisation": like type setters in former days, who were picking letters from boxes to compose a journal page, a swarm engineer could "engineer" a swarm by picking specific agents and arranging them to a specific goal-tailored swarm system.

We aim for a swarm system that can be optimised for a specific experimental setting by choosing the accurate swarm composition. Here we present the first experiments and results of a whole set of swarm composition experiments. In the end we want to have a set of swarm compositions to pick from according to how the environment looks like. For example, there could be an optimal swarm composition for experiments with one goal, another optimal composition for two or more goals, another one if the goals are located in the middle of the arena or at the walls and so on.

To achieve this goal, we show here how different types of motion behaviour can change the overall swarm-behaviour. Therefore we formulate the following hypotheses:

H1 The Goal-Finder is able to locate itself at different goals in the given arena, but is not able to discriminate between a Local and a Global Goal.

H2 Introducing the Wall-Following-Behaviour to a swarm of Random-Walkers raises the success of aggregation for the given setup.

H3 Immobile-Agents have an attractive effect on other swarm members just as Social Agents [26] have.

H4 Immobile-Agents have an effect on the success of aggregation depending on their position.

5.1.1. Material & Method

Implementation of Behavioural Types

We implemented 4 behavioural types according to the trajectories created from the movement of young honeybees shown in figure 1.1:

Wall-Follower: Once a Wall-Follower reaches a wall, it follows the wall at a certain distance. To achieve this the sensor-input has to be between two thresholds. If the sensor-input exceeds the thresholds, the agent makes a small turn either left or right depending on which side the threshold is exceeded. As it is not known how two Wall-Followers behave if they meet each other, we implemented three different possibilities:

1. The agents move randomly until they find the wall again (figure 5.1(a)). We use the acronym "WFrw" for this implementation.
2. If the agents meet at a wall, they turn 90° to the inside of the arena (figure 5.1(b)). We use the acronym "WF90" for this implementation.
3. The third possibility is, that the agents both turn 180° and follow the wall in the other direction (figure 5.1(c)). We use the acronym "WF180" for this implementation.

Goal-Finder: The Goal-Finder is implemented like a greedy uphill walker. It compares the temperature on its front left side with the temperature on its front right side and moves in the direction of the higher temperature. If both temperatures are equal, the agents moves forward (figure 5.1(d) shows an example behaviour).

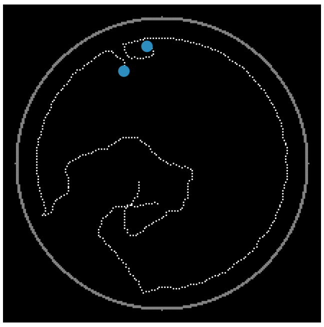
Random-Walker: In each step the Random-Walker is moving, it simultaneously makes a randomly generated turn between -35° and +35°. This leads to a trajectory as shown in figure 5.1(e).

Immobile Agent: In figure 5.1(f) it can be seen that this behavioural type does not move very far from its origin and is thus called "Immobile Agent". This trajectory is created by agents with a high turning-angle between -180° and +180°and with a slow speed (a quarter of the speed of the other behavioural types).

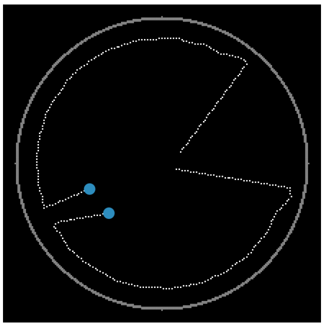
Apart from the motion pattern the agents execute the normal BEECLUST algorithm which is described in section 1.3.

Setup of the Experimental Arena

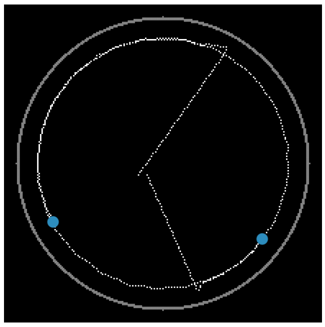
To test our hypotheses we used the classical binary choice setup which is described in Section 3.2.



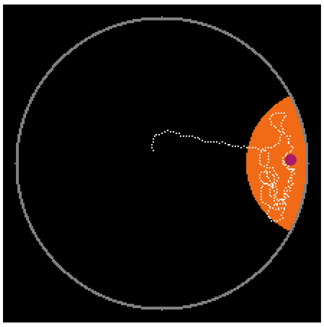
(a) Typical trajectory of a Wall-Follower with a Random-Walk in the middle of the arena.



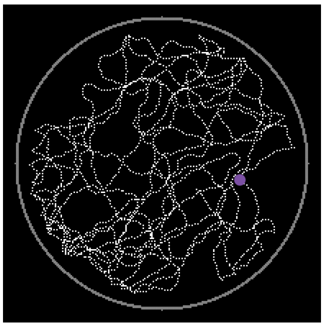
(b) Typical trajectory of a Wall-Follower with a 90° turn after meeting another agent.



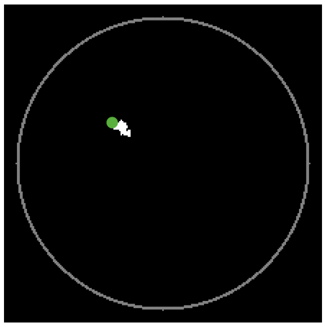
(c) Typical trajectory of a Wall-Follower with a 180° turn after meeting another agent.



(d) Typical trajectory of a Goal-Finder. The orange area indicates the source of the temperature gradient.



(e) Typical trajectory of a Random-Walker.



(f) Typical trajectory of an Immobile Agent.

Figure 5.1.: Typical trajectories of the different implemented behavioural types in simulation.

Experiments

Experiment 1: In order to test Hypothesis 1 we used 12 agents of the type "Goal-Finder" performing the BEECLUST algorithm and released them in the middle of the arena.

Experiment 2: To investigate the influence of the Wall-Following behaviour to a swarm of Random-Walkers (Hypothesis 2), 3 Wall-Followers and 9 Random-Walkers (in total 12 agents) were released in the middle of the arena. The agents moved around randomly. Whenever they reach a wall, they change their behaviour to a Wall-Following-Behaviour until they lose contact with the wall again due to a close encounter with another agent (see next section for a more detailed description).

Experiment 3: To test Hypotheses 3 and 4 the influence of the Immobile Agent has to be analyzed. In the first experiment (Hypothesis 3) the goal was to compare the effects of Immobile-Agents with the effects of Social Agents [26] (Social Agents are fully immobilized agents (do not turn or move at all), which create a "social gradient" through the minimal social component modelled in the BEECLUST algorithm). To achieve this, we created an experimental setup similar to what was used in [26]. Three Immobile-Agents were placed in the "Local Goal" on the left side of the arena and in the Global Goal we placed three dummy-agents which are perceived as obstacles. In the middle of the arena 9 Random-Walkers are released, so that we have 12 agents in total.

Experiment 4: A series of experiments was made to test Hypothesis 4.

> First 12 Random-Walkers were used in an arena with a Global and a Local Goal and no Immobile-Agents are placed in the arena. Then three Immobile-Agents were placed in the Global Goal, in the Local Goal and in the middle of the arena consecutively. In all three cases 9 Random-Walkers were released in the middle of the arena. The same series of experiments was done with Wall-Followers (WF180) and Immobile-Agents instead of Random-Walkers and Immobile-Agents.

We also made an exhaustive analysis with (in total) 15 agents. We start with 15 agents of one type and then replace the agents one by one with another type. The results of these experiments provide a swarm-designer with a guideline how the composition of the swarm with different types shall be engineered.

All experiments were repeated with "number of repetitions $n = 100$". The simulated time is 30 minutes per experiment and repetition.

5.1.2. Results

The success of aggregation was measured by counting the median amount of agents over time in the Global, in the Local Goal and in the Pessimum. On the y-axis the figures show the median amount of agents and the x-axis show the different areas for every tested behavioural type. All significances were tested with the Wilcoxon-Test (Global Goal tested against Local Goal) and the U-Test (Global Goal of one experiment against Global Goal of another experiment) with $p < 0.05$.

Figure 5.2 shows the median aggregation count for a group of Goal-Finders, Random-Walkers and different implementation-methods of Wall-Followers. The median amount of Goal-Finders was in the Global and in the Local Goal 6 agents and in the Pessimum 0 agents. In the experiment with 12 Random-Walkers, the median amount of agents in the Global Goal was 4 agents, 1 agent in the Local Goal and 6 agents in the Pessimum. For the experiment with 9 Random-Walkers and 3 WFrw the median amount of agents in the Global Goal is 6, in the Local Goal 1 and in the Pessimum 5. The median amount of agents for 3 WF90 and 9 Random-Walkers is in the Global Goal 5, in the Local Goal 1 and in the Pessimum 6. In the experiment with 3 WF180 and 9 Random-Walker the median amount of agents in the Global Goal is 5, in the Local Goal 1 and in the Pessimum 6. For significances and the exact values see table 5.1.

The results for comparing the effects of Immobile-Agents with the effects of Social Agents in [26] are shown in figure 5.3. On the x-axis the different areas of the arena in which the agents are counted are shown. The y-axis shows the median and quartiles of the time the agents spent in the respective area. In both cases, 3 Social Agents or Immobile-Agents were placed in the Local Goal. In the experiment with Social Agents, the median time that agents

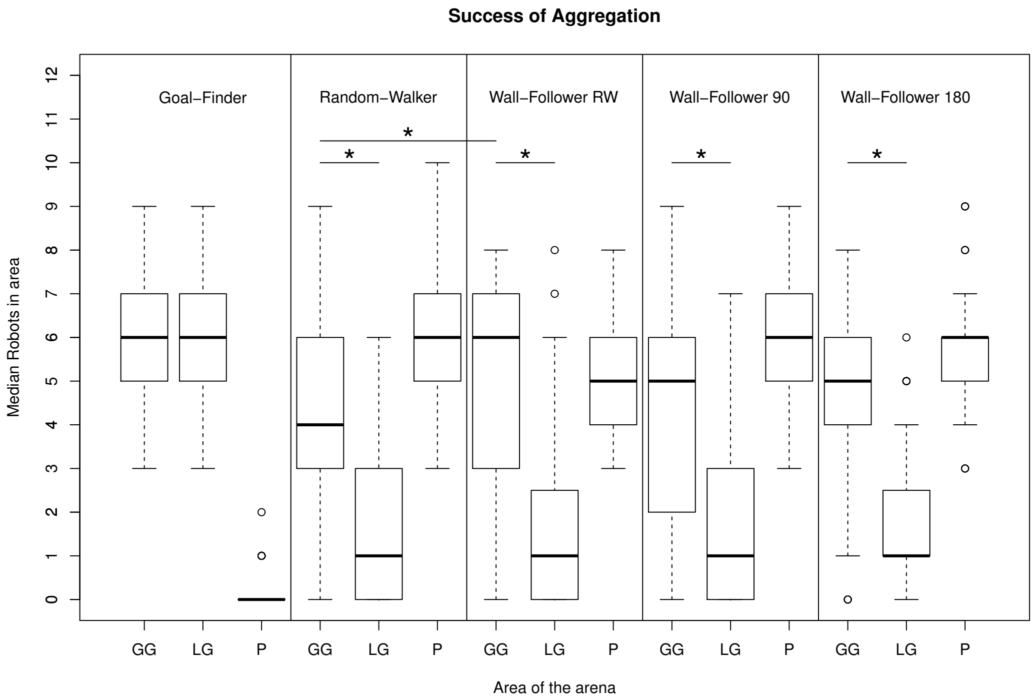


Figure 5.2.: Success of aggregation for different behavioural types. "GG", "LG" and "P" refers to the "Global Goal", "Local Goal" and "Pessimum", respectively. X-axis shows the different areas for every tested behavioural type. The y-axis shows the median amount of agents in the respective area. Significances are tested with the Wilcoxon-Test (Global Goal tested against Local Goal) and the U-Test (Global Goal of one experiment against Global Goal of another experiment) with $p < 0.05$.

Experiments	Significance	p-value
Goal-Finder GG - LG	n.s.	0.743
Random-Walker GG - LG	s.	< 0.01
Wall-Follower RW GG - LG	s.	< 0.01
Wall-Follower 90 GG - LG	s.	< 0.01
Wall-Follower 180 GG - LG	s.	< 0.01
RW GG - WFrw GG	s.	< 0.01
RW GG - WF90 GG	n.s.	0.5595
RW GG - WF180 GG	n.s.	< 0.01

Table 5.1.: Significance-values for the different behavioural types tested in figure 5.2. WF = Wall-Follower, RW = Random-Walker, GF = Goal-Finder.

Experiments	Significance	p-value
SA GG - IA GG	s.	< 0.01
Social Agents GG - LG	s.	0.02142
SA LG - IA LG	s.	< 0.01
Immobile-Agents GG - LG	s.	< 0.01

Table 5.2.: Significance-values of the comparison of Social Agents (SA) and Immobile-Agents (IA) in figure 5.3.

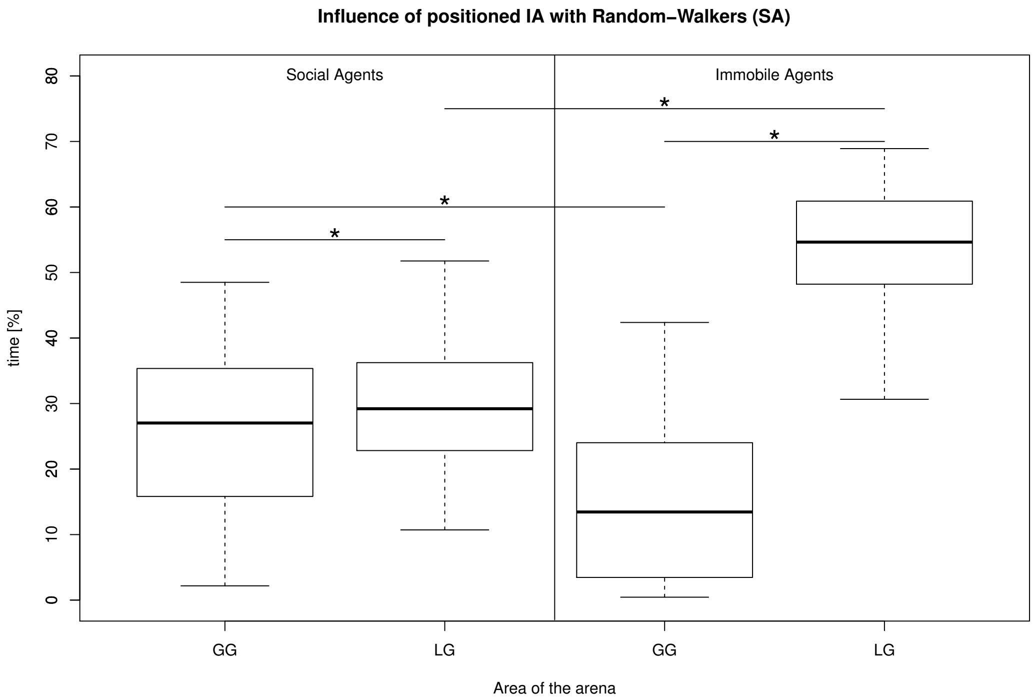


Figure 5.3.: The figure compares the influence of Social Agents [26] with the influence of Immobile-Agents to a swarm performing the BEECLUST algorithm. X-axis shows the different areas of the arena: "GG" and "LG" refers to the "Global Goal" and "Local Goal", respectively. Y-axis shows the median amount of time the agents spent in the respective area. Significances are tested with the Wilcoxon-Test (Global Goal tested against Local Goal) and the U-Test (Global Goal of one experiment against Global Goal of another experiment) with $p < 0.05$.

Experiments	Significance	p-value
Random-Walker GG - LG	s.	< 0.01
IA in GG: GG - LG	s.	< 0.01
IA in LG: GG - LG	s.	< 0.01
IA in P: GG - LG	s.	< 0.01
RW GG - IA in GG: GG	s.	< 0.01
RW GG - IA in LG: GG	s.	< 0.01
RW GG - IA in P: gg	s.	< 0.01

Table 5.3.: Significance-values of the influence of Immobile-Agents (IA) on a swarm of Random-Walkers (RW) in figure 5.4.

spent in the Global Goal was 27% and in the Local Goal 29.2%. If we use Immobile-Agents instead of Social Agents, the median time the agents spent in the respective area was 13.45% in the Global Goal and 54.64% in the Local Goal. All tested significances are significant ($p < 0.05$). For significances and the exact p-values see table 5.2.

The results of the experiments with positioned Immobile-Agents in different areas are shown in figure 5.4. The x-axis shows the position of Immobile-Agents and different areas of the arena, whereas the y-axis shows the median of agents in the corresponding area. The results are grouped into three boxplots, where a group of 3 boxplots represents an experiment. The first three boxplots show the median aggregation count without any Immobile-Agents and are used as a reference to the other experiments.

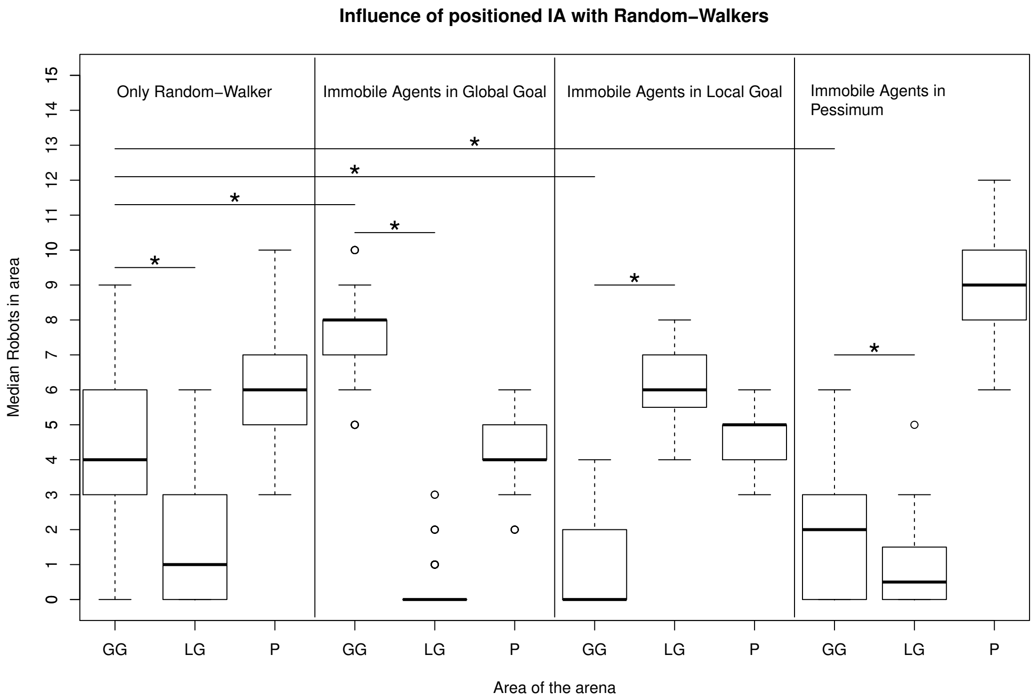


Figure 5.4.: Influence of Immobile-Agents positioned in different areas of the arena on a swarm of Random-Walkers. X-axis shows the different areas of the arena: "GG", "LG" and "P" refers to the "Global Goal", "Local Goal" and "Pessimum", respectively. The y-axis shows the median amount of agents in the respective area. Significances are tested with the Wilcoxon-Test (Global Goal tested against Local Goal) and the U-Test (Global Goal of one experiment against Global Goal of another experiment) with $p < 0.05$.

Experiments	Significance	p-value
Wall-Follower GG - LG	s.	< 0.01
IA in GG: GG - LG	s.	< 0.01
IA in LG: GG - LG	s.	< 0.01
IA in P: GG - LG	s.	< 0.01
WF GG - IA in GG: GG	s.	< 0.01
WF GG - IA in LG: GG	s.	< 0.01
WF GG - IA in P: GG	s.	< 0.01

Table 5.4.: Significance-values of the influence of Immobile-Agents (IA) on a swarm of Wall-Followers (WF) in figure 5.5.

Here the median amount of agents is 4, 1, and 6 for the Global Goal, Local Goal and Pessimum, respectively. The second group of boxplots shows the results of the experiments where the Immobile-Agents were placed in the Global Goal. The median amount of agents for the Global Goal was 8, for the Local Goal 0 and the Pessimum 4. If the Immobile-Agents were placed in the Local Goal (group three), the median amount of agents is 0, 6 and 5 for the Global Goal, Local Goal and Pessimum, respectively. The last group shows the results of the experiments where the Immobile-Agents were placed in the middle of the arena. Here the median amount of agents was 2, 0.5 and 9 for the Global Goal, Local Goal and Pessimum, respectively. All tested significances are significant ($p < 0.05$). For significances and the exact p-values see table 5.3.

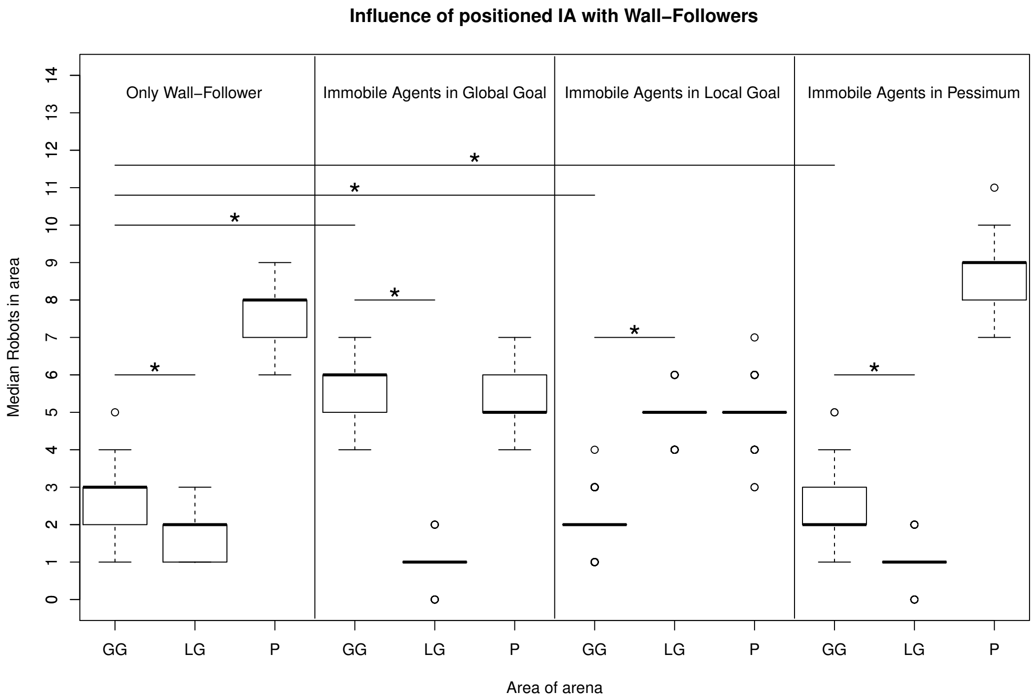


Figure 5.5.: Influence of Immobile-Agents positioned in different areas of the arena on a swarm of Wall-Followers (WF180). X-axis shows the different areas of the arena: "GG", "LG" and "P" refers to the "Global Goal", "Local Goal" and "Pessimum", respectively. The y-axis shows the median amount of agents in the respective area. Significances are tested with the Wilcoxon-Test (Global Goal tested against Local Goal) and the U-Test (Global Goal of one experiment against Global Goal of another experiment) with $p < 0.05$.

Figure 5.5 shows the results of how Wall-Followers (WF180) are influenced by Immobile-Agents. The x-axis shows the position of Immobile-Agents and different areas of the arena, whereas the y-axis shows the median amount of agents in the corresponding area.The first three boxplots represent the results of 12 Wall-Followers (WF180) without Immobile-Agents and are used as a reference. The median amount of agents are 3, 2 and 8 for the Global Goal, Local Goal and Pessimum, respectively. The second group of boxplots show the data when the Immobile-Agents are placed in the Global Goal at the beginning of the experiment. For the Global Goal the median amount of agents is 6, for the Local Goal 1 and for the Pessimum 5. If the Immobile-Agents are placed in the middle of the arena at the beginning of the experiment, the median amount of agents in the Global Goal, Local Goal and Pessimum are 2, 1 and 9 respectively. The last group of three boxplots shows the median amount of agents if the Immobile-Agents start in the Local Goal. Here the median amount of agents are 2, 5 and 5 for the Global Goal, Local Goal and Pessimum respectively. All tested significances are significant ($p < 0.05$). For significances and the exact p-values see table 5.4.

Figure 5.6 - 5.11 depict results of an exhaustive analysis. For a better representation of the swarms' decision making process, here the fitness function is represented as

$$F = G - L \tag{5.1}$$

where G is the number of agents within the global goal area and L is the

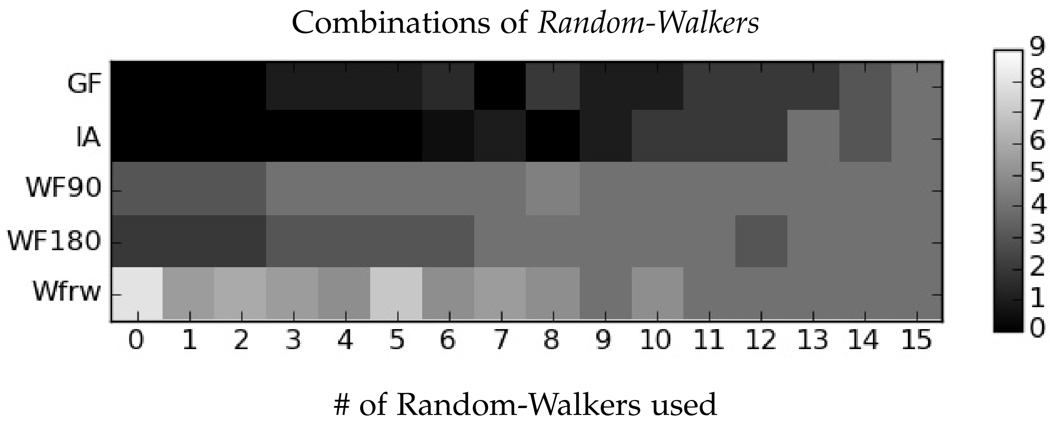


Figure 5.6.: Results of exhaustive analysis. The agents of a swarm of 15 Random-Walkers were replaced one by one with another behavioural type. We always used 15 agents in total. The x-axis shows the amount of Random-Walker. Black = low fitness, white = high fitness.

number of agents within the local goal area. Note that some results are plotted twice in reversed order to have a better overview (for example line 1 of figure 5.6 and line 5 of figure 5.7).

In figure 5.6 the x-axis shows the amount of Random-Walker of the swarm, whereas the total amount of agents used is always 15 agents. Black means low fitness and white higher fitness of the swarm. Thus, for the binary choice-experiment the swarm accomplishes a high fitness if many WFrw are used (left lower corner). If many Goal-Finders or Immobile-Agents are used, the swarm accomplishes a low fitness (left upper corner).

The next figure shows combinations of *Goal-Finders* with the other types.

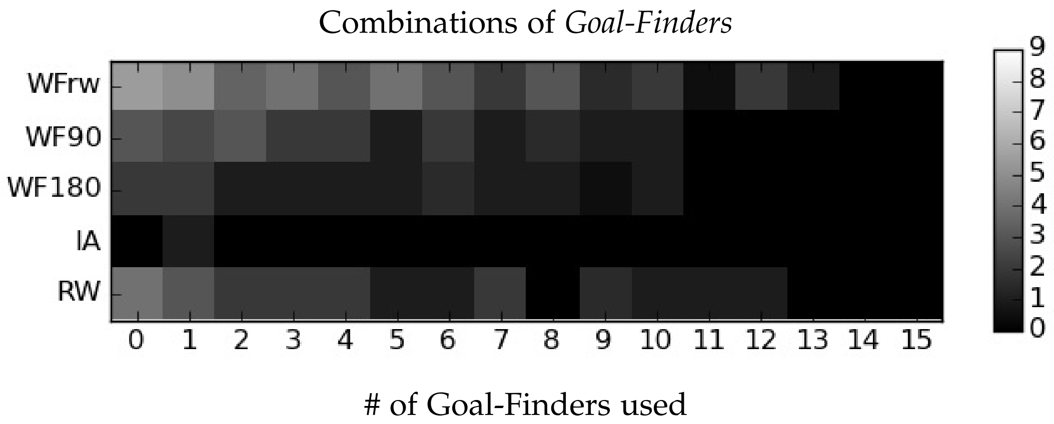


Figure 5.7.: Results of exhaustive analysis with *Goal-Finders*. The agents of a swarm of 15 *Goal-Finders* were replaced one by one with another behavioural type. We always used 15 agents in total. The x-axis shows the amount of *Goal-Finders*. Black = low fitness, white = high fitness.

Here we can see, that using a high amount of *Goal-Finders* decreases the performance of the swarm for the binary choice setup in all combinations.

The next three figures (5.8 - 5.10) show combinations with *Wall-Followers*. Here we show only three different combinations with the *Wall-Follower*, because WFrw, WF90 and WF180 are only variations of the wall-following behaviour and not three totally different behaviour types. It can be seen, that with an increasing amount of *Wall-Followers* the variation of WFrw is the only type that improves the performance of the swarm. An increase of the amount of WF90 or WF180 leads in both cases to a decrease of the performance when combined with the *Random-Walker* (figure 5.9 and 5.10).

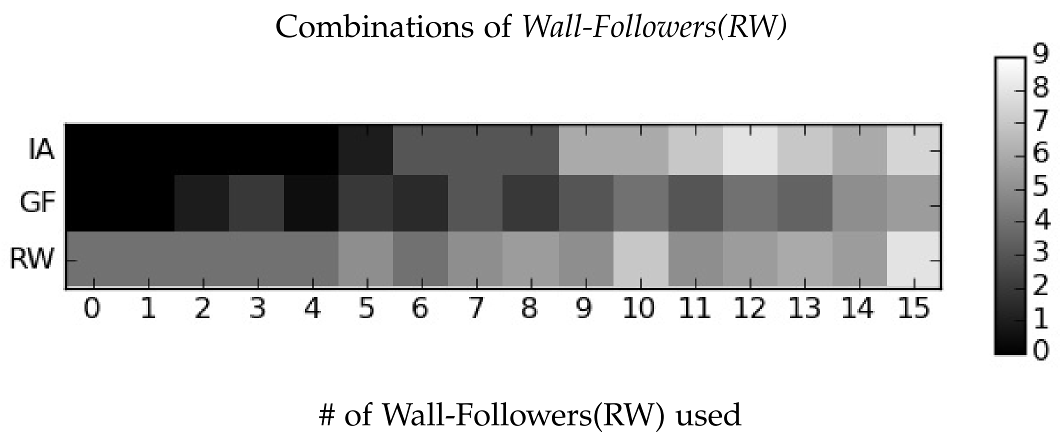


Figure 5.8.: Results of exhaustive analysis with *Wall-Followers(RW)*. The agents of a swarm of 15 WFrw were replaced one by one with another behavioural type. We always used 15 agents in total. The x-axis shows the amount of WFrw. Black = low fitness, white = high fitness.

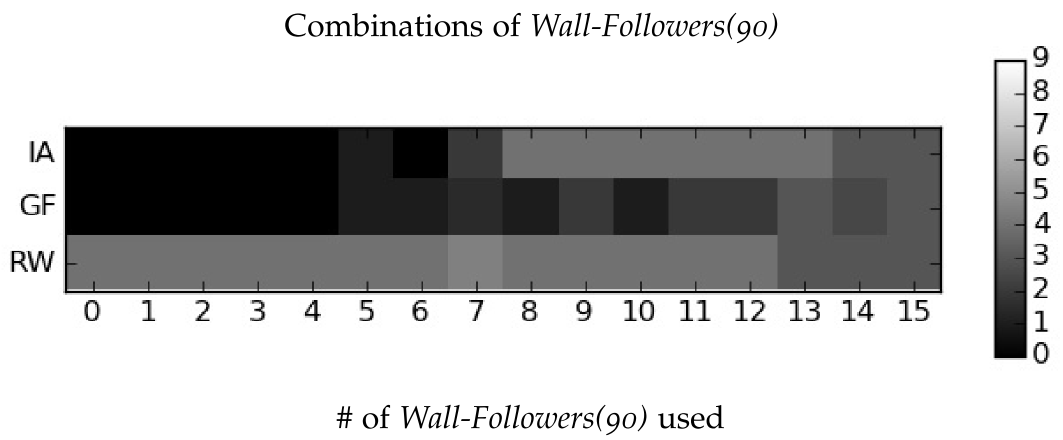


Figure 5.9.: Results of exhaustive analysis. The agents of a swarm of 15 *Wall-Followers(90)* were replaced one by one with another behavioural type. We always used 15 agents in total. The x-axis shows the amount of *Wall-Followers(90)*. Black = low fitness, white = high fitness.

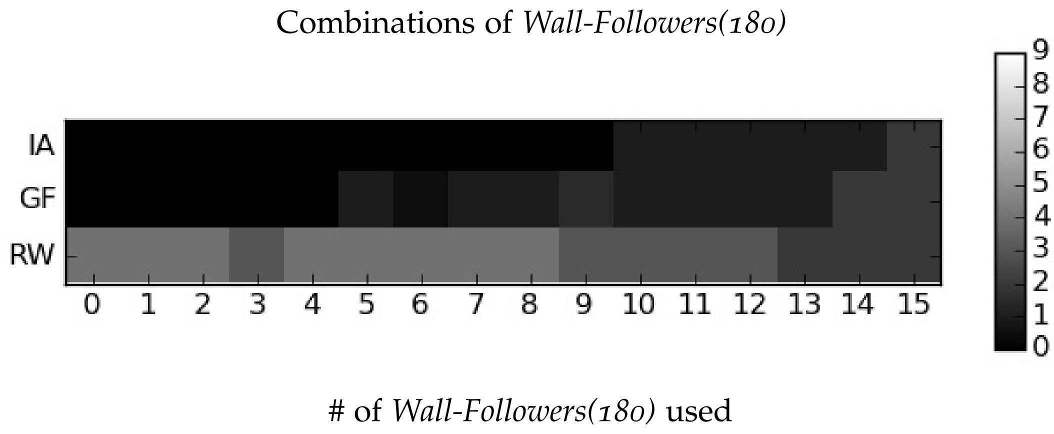


Figure 5.10.: Results of exhaustive analysis. The agents of a swarm of 15 *Wall-Followers(180)* were replaced one by one with another behavioural type. We always used 15 agents in total. The x-axis shows the amount of *Wall-Followers(180)*. Black = low fitness, white = high fitness.

Figure 5.11 shows combinations of *Immobile Agents* with all other types.

Timelines of 5 exemplary compositions of behavioural types are shown in figure 5.1.2 and figure 5.1.2 ($n = 100$). Figure 5.1.2 shows 3 combinations of Random Walker and Goal Finder: The fitness of a swarm with 15 Goal Finders and 0 Random Walkers (solid line) increases in the first 5 minutes sligthly, but then decreases again and stays at 0 until the experiment ends. At a ratio of 7 Random Walkers and 8 Goal Finders (dashed line), the fitness of the swarm starts to increase during the experiment, although the fitness is still very low in the end. When removing all Goal Finders (0 Goal Finder and 15 Random Walker, dotted line), the fitness increases from the beginning of the experiment until 25 minutes are over. Then the fitness

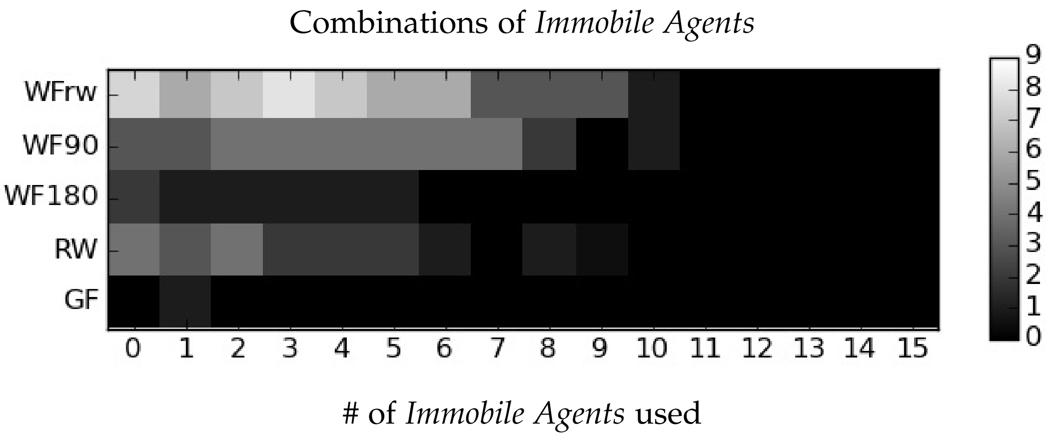


Figure 5.11.: Results of exhaustive analysis. The agents of a swarm of 15 *Immobile Agents* were replaced one by one with another behavioural type. We always used 15 agents in total. The x-axis shows the amount of *Immobile Agents*. Black = low fitness, white = high fitness.

stays the same for the last 5 minutes of the experiment. In figure 5.1.2 the timeline of combinations with Random Walkers and WFrw are depicted. The dotted line shows the fitness of 15 Random Walkers (same as in figure 5.1.2). The dashed line shows a combination of 7 Random Walkers and 8 WFrw. Here, the fitness rises faster compared to the homogeneous swarm with 15 Random Walkers. 15 agents of WFrw are used to create the solid line. In the first 15 minutes, the fitness of this swarm (solid line) and the one with 7 RW and 8 WFrw (dashed line) is almost the same. After 15 minutes, the fitness of the swarm with 15 WFrw increases more and has a slightly better fitness at the end of the experiment.

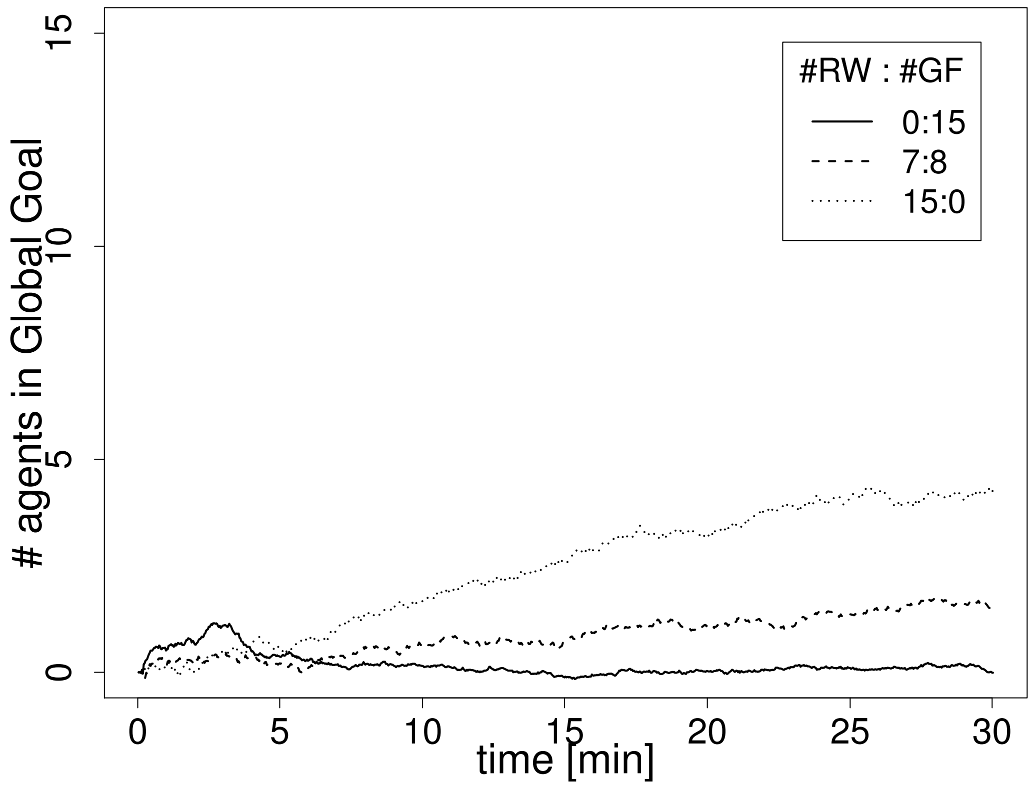


Figure 5.12.: Timeline of combinations of Random Walker and Goal Finder. Swarm with 15 Random Walkers (dotted line) achieves the best fitness compared to a swarm with partially Random Walkers and Goal Finders (dashed line) or only Goal Finders (solid line). Simulated time is 30 minutes (x-axis). The y-axis shows the mean amount of agents in the Global Goal (repetitions $n = 100$).

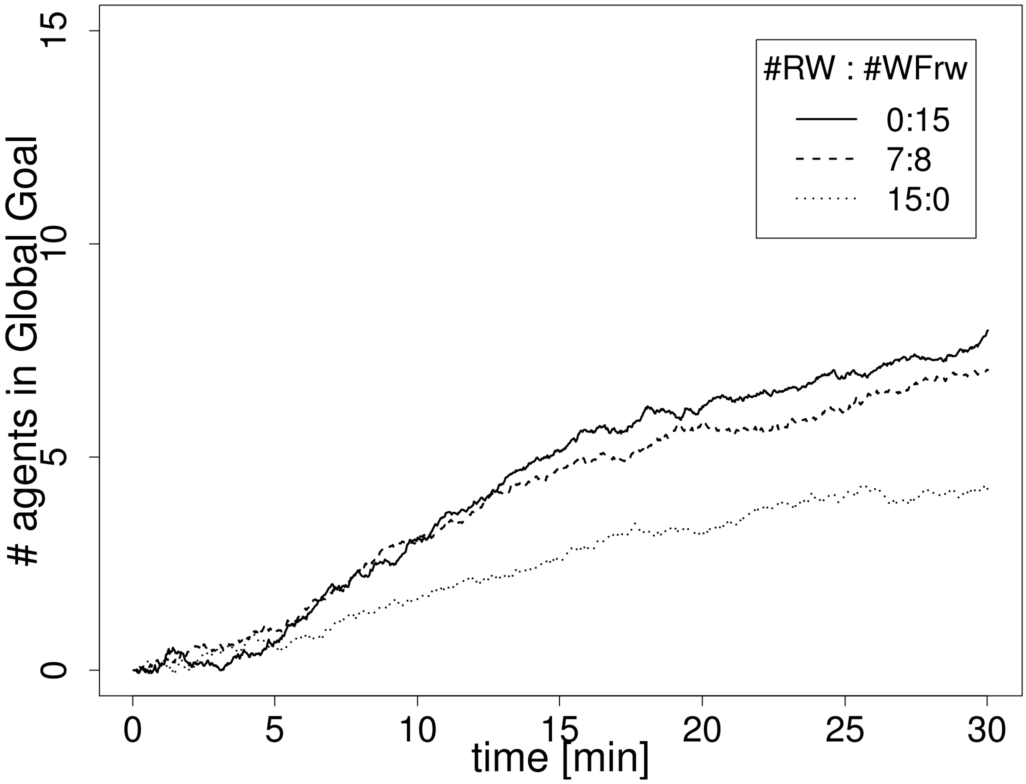


Figure 5.13.: Timeline of combinations of Random Walker and Wall Followers (WFrw). 15 Random Walker as reference (dotted line). Replacing Random Walker with Wall Follower (WFrw, dashed and solid line) increases fitness of the swarm. Simulated time is 30 minutes (x-axis). The y-axis shows the mean amount of agents in the Global Goal (repetitions $n = 100$).

5.1.3. Discussion

In the following section, we want to discuss the influence of the four motion patterns to the original BEECLUST algorithm which is only defined by Random-Walkers [31].

H1: *The Goal-Finder is able to locate itself at different goals in the given arena, but is not able to discriminate between a Local and a Global Goal.*

The results of our experiments showed that different behavioural types lead to different aggregation-behaviour. As it can be seen in figure 5.2 (group 2) half of the Random-Walker-agents are located in either the Global Goal or the Local Goal although each of these two areas cover only 11% of the whole arena. Due to the still remaining low amount of agents in the Local Goal it is possible for the swarm to distinguish between the Global and the Local Goal. However, a clear majority decision is made. If we look at the results of the Goal-Finders, half of the agents are located in the Global Goal and the other half in the Local Goal. This can be explained by the implementation method used: as a Goal-Finder compares the temperature on the left side with the temperature on the right side, those agents always move to the side where it is warmer (locally). The coldest area is in the middle of the arena. If the agents' starting position is slightly left or right of the coldest area, the starting position is the crucial factor whether the agents are moving to the left side (Local Goal) or to the right side (Global Goal). Once an agent has located itself in an optimum, it is stuck there because whatever direction the agent tries to go the temperature gets colder. The results of the Goal-Finders can be interpreted as they are able to mark the different goals although they are not able to make a swarm-decision for the Global Goal.

The difference between a Goal-Finder and a leading- or informed agent is, that the Goal-Finder uses no ego-positioning and does not know where the goal is located. Therefore a Goal-Finder is an uninformed agent. If a Goal-Finder reaches a Local Goal it is trapped there and does not know if there is a better goal somewhere in the search area. In contrast, a leading agent (or informed agent) knows where the best goal is, where it is located relatively to the goal and thus, where it has to go to reach the goal.

H2: *Introducing Wall-Following-Behaviour to a swarm of Random-Walkers raises the success of aggregation for the given setup.*

To investigate this hypothesis, we combined the Wall-Following-Behaviour with the Random-Walking-Behaviour. If the agents have contact with the wall, they follow the wall. But if the agents lose the wall (eg. if they meet another agent and have to avoid a collision) the behaviour switches to a Random-Walker until the agent has contact with the wall again. The results in figure 5.2 (last three boxplots) and figure 5.6 show, that the introduction of the Wall-Following-Behaviour leads to a more distinct decision-making of the swarm regarding the Global Goal. The amount of agents in the Global Goal is nearly doubled by the combination of these two behavioural types compared to experiments where only Random-Walkers were used. Combining the Wall-Following-Behaviour with a 90° or 180° turn if they lose the wall, does not lead to a significant increase regarding the amount of agents in the Global Goal.

However, the introduction of the Wall-Following behaviour could also lead to a deterioration of the median aggregation count. If - for example - the goals are in the center of the arena and the swarm system consists mostly of Wall-Followers, the agents will spend most of the time at the wall and will have problems to detect the goals in the middle of the arena. Therefore, the use and also the amount of Wall-Followers in a swarm system has to be wisely chosen considering the goal of the experiment and the swarm-systems' intention.

H3: *Immobile-Agents have similar effects on the swarm as Social Agents.*

In [26] it was shown that the swarm-behaviour of agents controlled by the BEECLUST algorithm is affected by artificially placed agents functioning as a social seed. Figure 5.3 shows the results of experiments with Social Agents (reprint from [26]) compared to the results of our experiments with the behavioural type "Immobile Agent". In those experiments with Social Agents it was possible to influence the swarm decision making in a way, that the agents spent significantly more time in the Local Goal instead of the Global Goal. If we use Immobile-Agents, the swarm decision gets clearer and it can be seen that the Immobile-Agents have the same but also a bigger effect than the Social Agents. Due to the very limited movement but high turning-angle, the Immobile-Agents are better recognised by other agents than the Social Agents with no movement at all.

H4: *Immobile-Agents can have positive and negative effects on the success of aggregation depending on their position.*

To show the different possible effects a series of experiments is made with Immobile-Agents and Random-Walkers (figure 5.4) and with Immobile-Agents and Wall-Followers (figure 5.5). In the experiments with Random-Walkers, the area where the Immobile-Agents were placed in the beginning was always the region with the highest amount of agents at the end of the experiment. Because it is the swarm's intention to locate the Global Goal, Immobile-Agents have - as expected - a positive influence on Random-Walkers if they start in the Global Goal, but have a negative influence on the swarm decision if they are placed in the Local Goal or Pessimum.

If we look at the same experiments with Wall-Followers (figure 5.5), it is not that clear anymore. Using only Wall-Followers most of the agents are located in the Pessimum, but a significant decision between the Local and the Global Goal can be still made. Placing Immobile-Agents in the Global Goal leads to a clear decision making. Here, most of the agents locate themselves in the Global Goal whereas very few agents are located in the Local Goal. Thus, placing Immobile-Agents in the Global Goal has in fact - as expected - a positive effect on the median aggregation count compared to the results with only Wall-Followers. As can be seen in figure 5.5 the swarm always decides (significantly) for the area where the Immobile-Agents are located. Thus, the Immobile-Agents attract other agents to the region where they are located.

As Immobile-Agents do not move but attract other agents, this behavioural type can be used to mark different locations that are possible candidates for Global Goals. The swarm's task is then to decide which of the marked goals is the best option.

The exhaustive analysis (shown in figures 5.6 - 5.11) depicts that for the binary choice-experiment the introduction of Goal-Finders and Immobile-Agents to a swarm of Random-Walkers do deteriorate the decision-making of the swarm. Introducing WF90 and WF180 to the swarm do not change the decision-making-process of the swarm significantly, although the exhaustive analysis shows a tendency to decrease the fitness. An improvement and also the best results for the binary choice-experiment can be achieved by only using WFrw.

5.1.4. Conclusion

We conclude, that introducing different individual, thus local acting, behaviour types to the swarm of agents controlled by the BEECLUST algorithm does influence the swarm decision and therefore supports the idea of Swarm Level Optimisation. Depending on the behavioural types that we use, the influence can be positive or negative for the given setup. If one composes a swarm of different behavioural types, one should always have the given setup and the goal of the swarm in mind. Goal-Finders - for example - are able to locate goals but are not very useful if a decision about the quality of the goals shall be made by the swarm. However they can be used to “mark” a goal. Wall-Followers should be used to increase the time the agents spend at the wall. If we know that most of the goals are located next to the wall, the decision making process will be influenced positively. Immobile-Agents have a strong attracting effect. If the swarm shall decide about the quality of marked goals, this behavioural type can be useful. However this type should only be used very carefully as the attracting effect is very strong.

If it is not definitely clear which type has a positive effect for the setup, it is safe to use only Random-Walkers because they will cover the whole search space.

5.2. ODE-model

In the following investigations we focus on an extreme case by not allowing any task switching. Agents start with a predetermined behavior and keep it for the whole experiment. The motivation is to simplify the swarm system and to investigate the potential capabilities of such a static non-task-switching system. We hypothesize that swarms with predetermined and fixed behavioral heterogeneity can outperform homogeneous swarms for certain sets of predetermined behaviors. This idea is inspired by the behavior of juvenile honeybees that were found to show several behavioral roles in an aggregation behavior while not switching between them during the whole experiment [**?**, 27]. Such a complex swarm system with heterogeneous behaviors is an interesting research object in itself but also as an inspiration for how to design swarm robotic systems. We focus on an aggregation task in which the swarm has to find a single target area or to choose between two target areas. In the latter case, the behavior can also be interpreted as a collective-decision making process [**?**, 19, 16]. This setting is subject to many studies on an algorithm for homogeneous swarms called BEECLUST [44, 9, 23, 18, 6, 4, 25, 26, 47, 31].

The BEECLUST algorithm (see Fig. 5.14) is actually inspired by the above mentioned behavior of young honeybees. Agents controlled by the BEECLUST algorithm move around randomly (step 1 and step 2 create trajectories of straight lines interrupted by rotations due to collision avoidance), whenever they meet another swarm member (step 3) they stop, measure the local potential field value (e.g., temperature, light, gas concentration), wait for a time proportional to that value, and continue to move randomly afterwards. As a result, the robots form clusters, which is followed by a competition of growing and 'dissolving' robot clusters until one big cluster remains with robots leaving and returning occasionally. The BEECLUST algorithm simplifies the situation found in bees by reducing the different behavior types to only one: random walk. BEECLUST implements a homogeneous approach. The following work can be viewed as an extension of the BEECLUST algorithm to the domain of heterogeneous behavior. In contrast to the study reported in [27], here we investigate behavior compositions with arbitrary numbers that are optimized by evolutionary algorithms, we rely on a mathematical model to represent the individual behavior types, and we investigate different environments.

In this paper, we investigate the above mentioned hypothesis whether a swarm that is heterogeneous in its behavior can outperform a homogeneous swarm under the condition that there are only predetermined basic behaviors and agents are not allowed to switch between them. The motivation is our finding in the behavior of juvenile honeybees that take behavioral roles and never switch them during the run of the experiment [**?**, 27]. Aggregation at appropriate spots within the bee hive is essential for survival of honeybees and hence we follow that the observed heterogeneous swarm behavior is a well adapted product of natural evolution. In this study, we investigate whether we can reproduce that behavior in simulated agents

```
1.) Each agent moves straight until it
    perceives an obstacle O within
    sensor range.
2.) If O is a wall the agent turns
    away and continues with step 1.
3.) If O is another agent, the agent
    measures the local potential field value.
    The higher the scalar field value the
    longer the agent stays still.
    After this waiting period, the
    agent turns away from the other
    agent and continues with step 1.
```

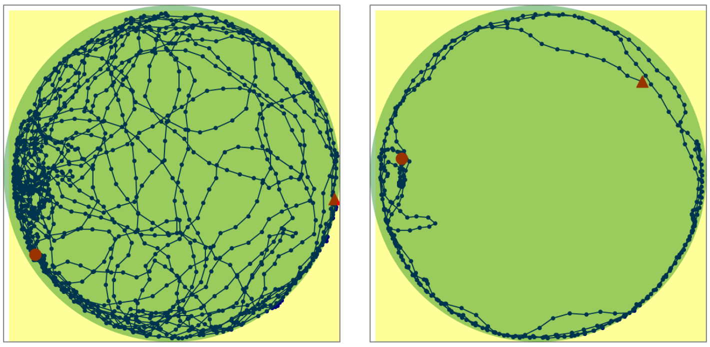
Figure 5.14.: The BEECLUST algorithm [44].

and test the hypothesis whether heterogeneity outperforms homogeneity in the investigated setting. The results of this study might help to make the right design decisions for systems of swarm robots, such as considering a heterogeneous approach in the first place and then choosing appropriate compositions of predetermined behaviors.

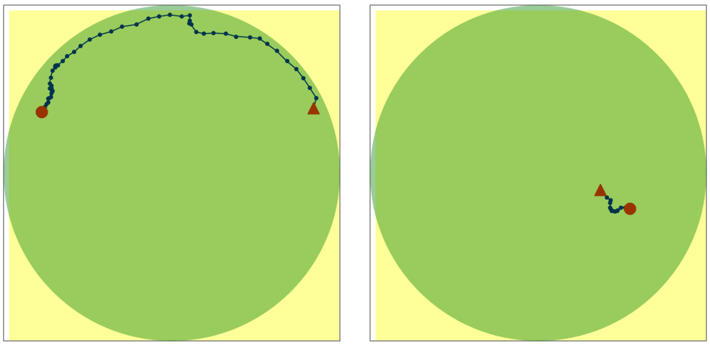
In the following, we limit our case study to a selection of four predetermined behavior types inspired by the biological system of juvenile honeybees. Our study might be considered as an example of biomimicry research due to this choice. However, we also motivate this choice by the opinion that these naturally evolved behaviors might be well adapted to the investigated task of aggregation. The definition of the four behavior types found in juvenile honeybees and a novel model to describe them are our next steps.

5.2.1. Four behavior types in juvenile honeybees

Honeybees (*Apis mellifera*) of age younger than 24 hours show four types of behaviors when allowed to move in a bounded temperature field [**?**]. The experiments were done in a circular arena surrounded by walls that cannot be climbed by the bees. Heat lamps create a distinct temperature field and it is known that juvenile honeybees have a preference for areas of 36°C [47, 31]. Each of the four behavior types consists of up to two actions: moving and stopping. Except for one type (immobile) all behavior types are combinations of both actions. Switching between the two actions is not considered task switching. The types differ in their movement pattern; there are: *random walker* (no bias found, neither due to walls nor due to temperature), *wall follower* (bias towards walls), *goal finder* (bias towards warmer areas), and *immobile agent* (no or slow movement only). See Fig. 5.15 for typical trajectories assigned to their respective behavior type based on tracking data of young honeybees. Note, that the young honeybees never switch between the different behavior types during an experiment.



(a) Tracked bee trajectory of type *random walker*. (b) Tracked bee trajectory of type *wall follower*.



(c) Tracked bee trajectory of type *goal finder*. (d) Tracked bee trajectory of type *immobile agent*.

Figure 5.15.: Typical tracked trajectories of young honeybees (same length of experiment), assigned to the four behavior types, start of trajectory at triangle, end at circle, 36°C target area at the left hand side of the arena.

5.2.2. Mathematical model of the behavior types

The behaviors of our agents are directly inspired by the behaviors observed in young honeybees. These behaviors are logically separated in two components: individual behavior aspects differ according to the four identified types and the collective behavior aspects that are identical across all types except for the immobile agents that do not show a reaction to social interactions because they only stay stopped always.

Individual behavior

We give a general, unified model here that is parametrized to describe all four behavior types. These behavior types are instantiated through different sets of parameters (see Section 5.2.2). An agent has a position $\mathbf{x} = (x_0, x_1)^\intercal$ (arena limits are $\sqrt{x_0^2 + x_1^2} < 1$), a heading $\phi \in [0, 2\pi)$, and a nominal velocity $v \in [0, 5]$ which is downscaled by discretization to $v/100$ per time step. An agent can measure an environmental feature, which is temperature in the case of young honeybees but it could also be light, ground color, gas concentration, etc. The environmental feature is modeled by a potential field $P(\mathbf{r})$, $\mathbf{r} \in \mathbb{R}^2$. An agent's turning behavior depends on the environmental feature and/or random effects. The parameter $\alpha \in [0, 1]$ is a weighting factor that determines how intensively an agent follows the gradient of the potential field. A 100% greedy agent following the gradient is defined by $\alpha = 1$. An agent that moves randomly is defined by $\alpha = 0$. Any intermediate value of α defines a corresponding agent that follows the gradient to some extent but is also subject to noise. We define the change of an agent's heading (for simplicity without units) by

$$\begin{aligned}\frac{d\phi(t)}{dt} =& \alpha \min\left(\operatorname{atan}\left(\frac{\partial P(\mathbf{x}(t))}{\partial x_0}, \frac{\partial P(\mathbf{x}(t))}{\partial x_1}\right), \phi_{\max}\right)\\ &+ (1-\alpha)\xi(\sigma, t),\end{aligned} \tag{5.2}$$

for a stochastic process ξ based on Gaussian noise with zero mean, standard deviation σ, and maximal turning angle $\phi_{\max} = 7/18\pi$ ($\phi_{\max} = 70°$). An agent's velocity (for simplicity without units) is defined by

$$\frac{d\mathbf{x}}{dt} = \begin{pmatrix} \cos\phi(t) \\ \sin\phi(t) \end{pmatrix} v(t)m(t), \tag{5.3}$$

for its current nominal speed $v(t)$ and $m(t) \in \{0, 1\}$ giving the agent's current state: $m = 0$ for *stopped*, $m = 1$ for *moving*. Note that the nominal speed v is irrelevant in state stopped ($m = 0$). The transitions between *stopped* and *moving* are modeled as probabilistic state machine with probability to move again P_{move} and probability to stop P_{stop}. Finally, the change of an agent's nominal speed v over time is modeled by a simple Markov chain. The interval of possible velocities $[0, 5]$ is discretized as a set of 51 velocities. For each of these discrete velocities v we have a probability of increasing the velocity $P_{\text{incr}}(v)$ by one step (i.e., $v' = v + 1/50$) and a symmetrical probability of decreasing the speed $1 - P_{\text{incr}}(v)$ (i.e., we force a change). These probabilities P_{incr} define the velocity distribution that results from our model.

Social behavior

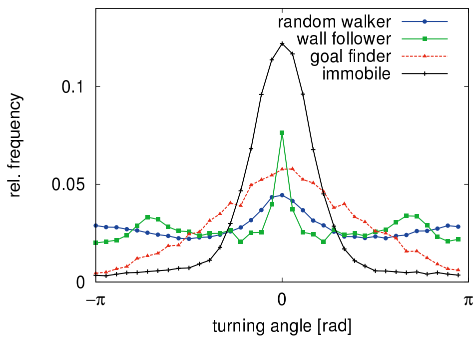
The agents' social behavior, that is the interactions between agents, are homogeneous across all behavior types. They follow the definition of the behavioral model of young honeybees [**?**] and the definition of the BEECLUST algorithm [44]. Once two agents perceive each other, they stop their motion, measure the local value of the potential field P, and wait for a certain period. This waiting time w is modulated proportionally to the measured potential field value P. It is defined by the function

$$w(P) = \frac{t_{\max} P^2}{\theta + P^2}, \tag{5.4}$$

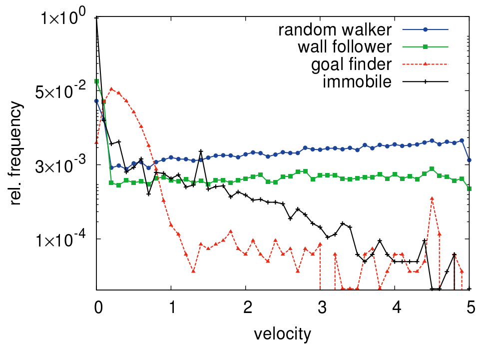
for parameters $t_{\max} = 132$ time steps and $\theta = 1.4 \times 10^4$. The parameters $t_{\max}$ and θ are chosen to generate an appropriate relation between the frequency of robot-robot encounters, the maximum of the potential field P, and the resulting interval of occurring waiting times. During the waiting time all features of the individual behavior are turned off (i.e., velocity v and agent state m are not relevant). Once the waiting time has elapsed the agents do a u-turn of $[-0.25\pi, 0.25\pi]$ and start to move again following their individual behavior type.

	RW	WF	GF	IA
σ [radian]	0.090	0.0004	0.57	0.885
α	0.016	0	0.99	0.481
P_{stop}	0	0	0.007	0.163
P_{move}	0	0	0.024	0.002

Table 5.5.: Typical parameters for the four behavior types of our model: *random walker* (RW), *wall follower* (WF), *goal finder* (GF), and *immobile agent* (IA).



(a) Histogram of turning angles for all 4 behavior types.



(b) Histogram of velocities for all 4 behavior types (note logarithmic scale on vertical axis).

Figure 5.16.: Histograms of turning angles and velocities for all four behavior types based on the mathematical model and parameters as given in table 5.5 (averaged over 200 repetitions of simulations, 1.5×10^4 time steps each).

Evolution of parameters for behavior types

Data acquired from experiments with single, young honeybees[1] are used to find appropriate parameters for our mathematical model. These bee-derived data were manually classified to the four behavior types. A parameter set (σ, α, P_{stop}, P_{move}, P_{incr}) for each behavior type is evolved using a simple genetic algorithm. The population size is 100, we evolve for 100 generations, the mutation rate is 0.25, we select based on proportionate selection, and 30 repetitions per evaluation are done. In each evaluation the agent operates in an arena with only one goal area to avoid side-effects of the symmetrical setting investigated in the swarm experiments. The agent is initially positioned far from that goal area with random orientation and random speed. The agent's behavior is defined by the considered parameter set and it is simulated for 1.5×10^4 time steps. During the simulation all turns and changes of velocity are stored in a histogram of turning angles and a histogram of velocities. The fitness function is a weighted sum of two features: First, it rewards similarities in the histograms of the simulated agent to the histograms acquired from the bee data. Second, type-specific qualities, that are not directly represented by the histograms of turning angles and velocities, are rewarded. In the case of the *goal finder*, turns towards the goal (i.e., maximum in the potential field P) are rewarded. The gradient of the potential field $(\frac{\partial P(\mathbf{x})}{\partial x_1}, \frac{\partial P(\mathbf{x})}{\partial x_2})^{\mathsf{T}}$ defines the optimal direction for each position $\mathbf{x}$. For each time step, the difference between the agent's direction and the optimal direction is cal-

[1] unpublished, publication in preparation

culated. The sum of these differences is part of the fitness function and hence imposes a minimization problem. In the case of the *wall follower*, time spent close to the wall is rewarded. This is done by defining three areas: a ring-shaped area R_{wall} directly at the wall $\mathbf{x} \in A_{\text{wall}} : \sqrt{x_0^2 + x_1^2} > 0.47$, a circular area far from the wall $\mathbf{x} \in A_{\text{center}} : \sqrt{x_0^2 + x_1^2} < 0.4$, and a second ring in between $\mathbf{x} \in A_{\text{neutral}} : 0.47 < \sqrt{x_0^2 + x_1^2} < 0.4$. In each time step, the agent is rewarded by a score of $+1$ when positioned on A_{wall}, it receives a penalty of -1 when positioned on A_{center}, and it is treated neutral (± 0) when positioned on A_{neutral}. This score needs to be maximized to evolve a wall following agent. In the case of the *immobile agent*, staying stopped is rewarded which is implemented by minimizing the agent's average speed. In the case of the *random walker*, no type-specific quality is defined.

The results of these evolutionary runs are shown in table 5.5 (except for the 50 values of P_{incr}). The resulting histograms of turning angles and velocities (due to $P_{\text{incr}}(v)$) for these four behavior types as defined by our model and the parameters given in table 5.5 are shown in Fig. 5.16. These results do not allow for a simple interpretation but a few features can be discussed here. The lowest peak for turning angle 0 is found for the *random walker* which indicates that the turning angle distribution is close to a uniform distribution. The *random walker* is also one of the fastest. The next peak for angle 0 is that of the *goal finder* but it also has low values for extreme turning angles. Hence, the *goal finder* approaches the goal area in a rather straight trajectory. In addition, the *goal finder* moves slowly. The *wall follower* has a distribution of turning angles that is close to a uniform distribution similarly to the *random walker*. However, the maximal turning angle σ is small, which leads to the behavior of a *wall follower*. Additionally there are two more peaks for big turning angles which are the required corrections when following the curved wall around the circular arena. The *wall follower* moves rather fast. In the case of the *immobile agent* the turning angle is of limited relevance, instead its low average velocity is of more importance.

5.2.3. Setup of experiments

In the following experiments the two standard setups are used (see Section 3.2).

5.2.4. Evolution of behavior type compositions

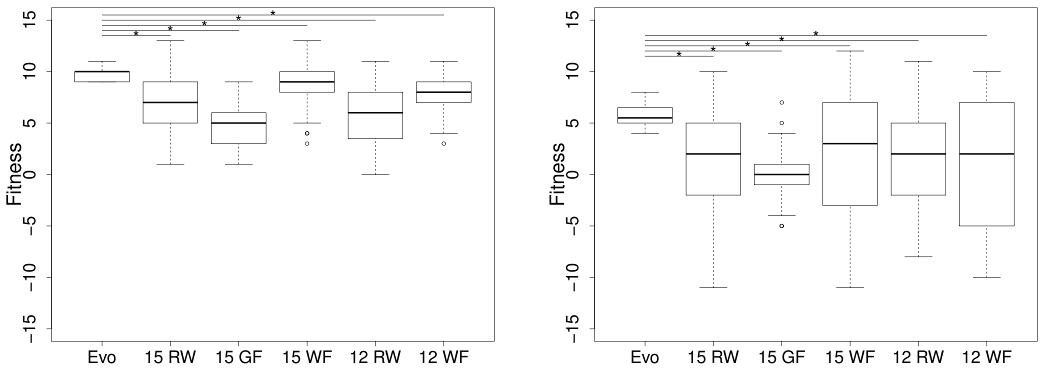
A variation of evolutionary algorithms, called wolf-pack-inspired evolutionary algorithm [55], is used to evolve the composition of behavior types in the swarm. The algorithm maintains overlapping generations and considers a fixed maximum population size. Proportional selection (fitness-based) is used to select individuals (i.e., compositions of behavior types) for mutation that fill empty places in the population. In every generation, one of the individuals, that have not been evaluated yet, is evaluated (alternatively the least evaluated individual if all the individuals have been evaluated already). The algorithm maintains the hierarchy in the population and keeps its diversity by removing older individuals with an equal or lower fitness than a newly evaluated individual (with a probability factor). The fitness function is defined by

$$F = G - L \tag{5.5}$$

where G is the number of agents within the global goal area and L is the number of agents within the local goal area (if there is one).

5.2.5. Results

For both experimental settings (one global goal and choice-experiment), we investigate the potential of heterogeneous swarms. As described in the above section, we use evolutionary algorithms to adapt the swarm's behavior-type composition to the environment. The experiments are based on a fixed swarm size $N = 15$. The results are based on $n = 18$ independent runs of the evolutionary algorithm and the population of compositions was initialized to a random uniformly distributed setting of behavior types. The evolved approach is compared to the fitness of several homogeneous swarm settings (Fig. 5.17) that were evaluated in $n = 100$ independent simulation runs (no evolution because composition is predetermined). In the first three homogeneous swarm settings we use a swarm size of $N = 15$. For the last two settings we used a swarm size of $N = 12$ to test for a potential density



(a) Setting with one goal area at the wall.

(b) Setting with a *global goal area* and a *local goal area* (choice experiment).

Figure 5.17.: Comparison of the best fitness between one evolved heterogeneous setting and several homogeneous swarm settings for one goal area at the wall (left) and the choice experiment on the right (*global goal area* and *local goal area*); heterogeneous swarm (labeled 'Evo'), homogeneous swarms with only *random walkers* (RW), *goal finders* (GF), or *wall followers* (WF). In both settings, the heterogeneous swarm is significantly better than all homogeneous swarms (based on Wilcoxon rank sum test, $p < 0.05$). Other significances are not shown.

dependency.

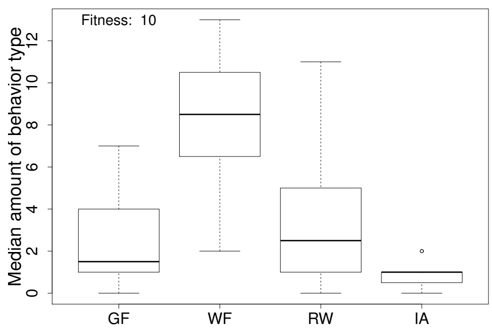
First we focus on the experiment with only one goal area (Fig. 5.17(a)). The median fitness for 15 *random walkers* is 7, for 15 *goal finders* it is 5, for 15 *wall followers* it is 9, for 12 *random walkers* it is 6, and for 12 *wall followers* it is 8. For the heterogeneous swarm optimized by evolution the median fitness is 10 ($n = 18$). The evolved behavior-type composition is found to be significantly better than the homogeneous swarms (based on Wilcoxon rank sum test, $p < 0.05$).

Figure 5.17(b) shows the results of the choice experiment (*global goal area* and *local goal area*). Here we compare the evolved heterogeneous behavior-type composition (first box plot, labeled 'Evo') to homogeneous behavior-type compositions.

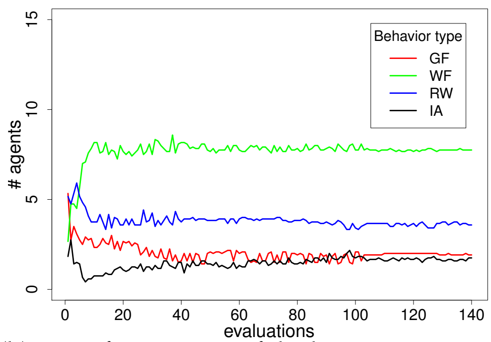
The median fitness for 15 *random walkers* is 2, for 15 *goal finders* it is 0, for 15 *wall followers* it is 3, for 12 *random walkers* it is 2, and for 12 *wall followers* it is 2. For the heterogeneous swarm optimized by evolution the median fitness is 5.5 ($n = 18$). The evolved behavior-type composition is found to be significantly better than the homogeneous swarms (based on Wilcoxon rank sum test, $p < 0.05$). Hence, our heterogeneous approach is the most effective variant of all tested configurations. The results for 12 *random walkers* and 12 *wall followers* indicate no dependency on density. The motivation of this test is based on results we report below and the consideration that *immobile agents* might potentially be used to virtually decrease the agent density.

Next, we evolve behavior-type compositions for different environments (one or two goal areas) and different initializations of the composition populations. We start with the setting that has only one goal area (see Fig. 3.2(a)). The evolutionary approach is as described above, that is, the initial population of compositions is sampled from a random uniform distribution. For our analysis, we take the best composition of the last population from each evolutionary run. The box plots shown in Fig. 5.18(a) give a summary of these best compositions. The number of occurrences for each behavior type is given for the $n = 18$ best evolved compositions. The median number of *goal finders* is 1.5, the median of *wall followers* is 8.5, the median of *random walkers* is 2.5, and the median of *immobile agents* is 1. It is counterintuitive that *goal finders* are relatively infrequent while the high number of *wall followers* might seem reasonable because the goal area is located at the wall. In Fig. 5.18(b) we give an overview of the type frequencies of the current best compositions over the number of evaluations averaged over all evolutionary runs. We started with compositions that are in average uniformly distributed. During the first 10 evaluations the number of *immobile agents* is decreased while the number of *wall followers* is increased quickly. The number of *random walkers* increases initially but then decreases again. The number of *goal finders* is decreased over a long period during the first 40 evaluations. After about 100 evaluations a saturation effect is observed.

Next we investigate the choice experiment (*local goal area* on the left side and a *global goal area* on the right side of the arena). The box plots of Fig. 5.19(a)



(a) Results of the evolution with one goal area. The plot shows the median amount of behavioral types that are used to compose a heterogeneous swarm with the highest fitness.



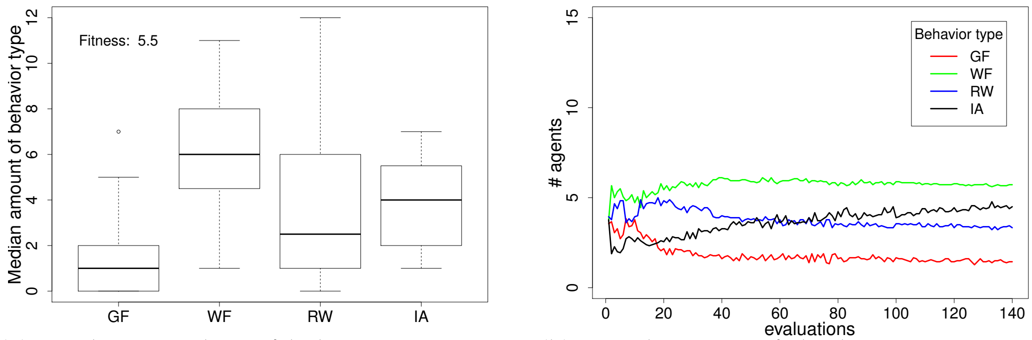
(b) Type frequencies of the best composition over evaluations

Figure 5.18.: Results of evolved swarm compositions with one goal area and all four behavioral types: *random walker* (RW), *goal finder* (GF), *wall follower* (WF), and *immobile agent* (IA).

give the number of agents for each behavior type as they occurred in the best compositions of $n = 20$ independent evolutionary runs. The median number of *goal finders* is 1, the median of *wall followers* is 6, the median of *random walkers* is 2.5, and the median of *immobile agents* is 4. As expected the number of *goal finders* is smaller in comparison to the setting with only one goal (cf. Fig. 5.18(a)) because *goal finders* merely follow the gradient and the swarm separates between the two goal areas. The number of *wall followers* is decreased, the number of *random walkers* is increased in its variance, and the number of *immobile agents* is increased in comparison to the one-goal setting. Especially the increase of *immobile agents* is counterintuitive because they are of no direct use to maximize the fitness function. In Fig. 5.19(b) we give an overview of the type frequencies of the current best compositions over the number of evaluations averaged over all evolutionary runs. Starting from approximately uniformly distributed compositions the number of *immobile agents* first decreases and is then increased slowly over about 120 evaluations at the cost of random walkers. After about 130 evaluations a saturation effect is observed.

5.2.6. Discussion

Concerning the results for the one-goal setting (Fig. 5.17(a) and 5.18) one would expect that the best fitness in this setup is achieved by making exclusive use of *goal finders* only. From our experience with the simulation



(a) Median number of behavior types as they occur in the best swarm composition.

(b) Development of the best composition that is evaluated at each time step.

Figure 5.19.: Results of the evolution for the choice experiment (*local goal area* and *global goal area*) and all four behavioral types: *random walker* (RW), *goal finder* (GF), *wall follower* (WF), *immobile agent*) (IA).

we can tell that too many *goal finders* actually block each other in areas before the goal area which results in clusters outside of the goal area. Instead, a limited number of *goal finders* turns out to be useful because such deadlock situations are then avoided. They serve as seeds within the goal area and help agents of other types to form clusters inside the goal area more easily, which is an example of how the different behavior types create opportunities of cooperation between agents. Most of the agents of the evolved heterogeneous swarms are *wall followers*. With only one goal area present, the *wall followers* always end up in the goal area and form a cluster. In comparison, the number of *random walkers* is low. Their approach to the goal area is slower because they might form clusters within the center of the arena. Eventually, they join the cluster in the goal area and join the *wall followers*. Therefore, in this setup a high amount of *wall followers* is the better choice. In an extended study, that is in preparation, we have also done experiments with goals not positioned at the walls. The number of *wall followers* decreases for that setting as expected but the qualitative result of our study is not influenced by the positions of the goal areas.

Concerning the results for the choice experiment (Fig. 5.17(b) and 5.19) the small number of *goal finders* is explained by the fact that they are not able to distinguish between a global and a local goal area because they merely follow the local gradient. Hence, they are not able to increase fitness ($F = G - L$). This is also indicated by the zero median for homogeneous *goal finder* swarms (Fig. 5.17(b)). Still, *goal finders* might be useful in a heterogeneous swarm to mark the goal areas and to serve as social seeds that attract others. Compared to the results of the experiment with only one goal area, the median amount of *immobile agents* is higher. Intuitively it seems inappropriate to use any *immobile agent* because they never enter the goal area when placed outside of it initially. However, they are part of many evolved swarm compositions although the optimization algorithm is effective [55] and we also do not enforce that all four behavior types have to be included in the solution. Thus, additional experiments are required to investigate the role of *immobile agents* and to find a sound explanation of why *immobile agents* are useful for the swarm in both our model and also in the natural swarms of honeybees. We can only speculate that *immobile agents* might have the functionality of a barrier and might slow down or even block agents that switch between goals. That way *immobile agents* might prevent other agents from visiting the *local goal area* and hence might stabilize the whole decision-making process. However, this requires more investigations and will be done in future work.

5.2.7. Conclusion

In this paper we have investigated swarms of agents that are heterogeneous in their behaviors. The idea is to simplify the swarm system by predefining static roles for certain swarm fractions. Even for the investigated extreme case without task switching, the heterogeneous swarm outperforms homogeneous swarms in the investigated aggregation scenario for the selected, predetermined behavior types. For now, all our results are based on one set of predetermined behaviors and one kind of collective task. However, the selection of behaviors was not arbitrary but inspired by results from biological experiments with juvenile bees. Still, the generalization of this work is left for future work.

The evolved compositions of behavior types indicate a rather complex underlying system that creates nontrivial distributions of behaviors which might even be perceived as counterintuitive. While the behavior types themselves were simple and predefined here, it is of course an option to determine the behavior types also by evolutionary computation or other methods of machine learning. However, for applications of swarm robotics, such as nanorobotics [32], it is attractive to make use of simple predetermined behaviors.

The effectivity of the evolved behavior compositions is certainly interesting, raises questions, and allows for different interpretations. While the four behavior types all score low in homogeneous swarms, they allow for a much more efficient aggregation behavior once combined. Obviously cooperation among different types is crucial and teamwork of a diverse team is essential. A tempting interpretation is that the results might be compared to findings in natural swarms that rely on certain degrees of leadership [13]. Only leadership is difficult to define here. The goal-oriented and greedy behavior of the *goal finder* is not helpful for the swarm per se. It requires a *random walker* and a *wall follower* to make use of the social seed within the goal area created by a *goal finder*. Hence, we observe a sophisticated interplay of agents with different approaches and capabilities that outperform their homogeneous counterparts as a heterogeneous swarm.

Also note that the use of simulations is potentially the only means to investigate the concept of predetermined behavioral roles in the natural complex system of young honeybees. Following the common standards of experiment design in biology it is not an option to use the same subjects (bees) in several replications of the experiment. In our case here an initial experiment would be necessary to label the bee with its behavioral role and in a second experiment we could create the desired swarm composition of behavior types. However, the bee might be influenced by the initial experiment and show a different behavior. Hence, simulations are a useful tool to investigate this complex system of interacting honeybees.

The results of this study support a core idea of swarm robotics that the interplay of several simple behaviors generates complex behaviors due to multiple interactions. This case study's main result is that heterogeneous swarms based on predetermined behaviors without task switching can perform well. Our approach is not limited to the study of the BEECLUST algorithm. Also other collective behaviors can be explored, such as heterogeneity in the stimulus-response functions of bees in their waggle-dance behavior [48]. In future work, we plan to do a complete sensitivity analysis of the many paramters in our model. In addition, we plan to work our way towards a generalization of our approach, for example, allowing different sets of predetermined or even learned behaviors. Although this study was guided by the biological inspiration of young honeybees' behavior, our future research will focus more on engineering applications of heterogeneous swarms in (evolutionary) swarm robotics.

6. Conclusion & Future Work

In this work the BEECLUST algorithm derived from the swarm behaviour of young honeybees is analysed and further developed in simulation as well as in real robot experiments. The first part of this work deals with the analysis of the BEECLUST algorithm in three different scenarios: robots with impaired sensors, dynamic environment and the social seed experiment. All three scenarios were analysed in real robot experiments, the social seed scenario was tested in simulation additionally. Before the experiments could be performed, temperature sensors had to be developed for the robots. Then the original BEECLUST algorithm was implemented. Because temperature has different and more sophisticated physical characteristics, the original BEECLUST did not work and the implementation had to be modified. Usually, the temperature is measured after a robot-to-robot encounter and then the waiting time is calculated as a function of the just measured temperature. The temperature sensors did not measure the correct temperature because they have to heat themselves up. Because of this, we measure the temperature right after a robot-to-robot encounter, calculate the waiting-time and set the state of the robot to ”wait“. After three seconds, the temperature is measured again and the waiting-time is corrected. No other adjustments were necessary.

Our results show, that the decision making of agents controlled by the BEECLUST algorithm is stable in various conditions but also dynamic enough to react to environmental changes. Experiments with agents that are not able to measure temperature show, that such agents do not harm the efficiency when adding them to a swarm of fully functional agents. Another point of view is, what happens when a swarm of fully functional agents is deployed and temperature sensors break during an experiment without being noticed by the observer. If only one agent gets impaired, the performance does not decrease significantly. If more and more agents break, the swarm is not able to form an aggregation anymore and thus, the swarm cannot make a decision. However, there will be no false decision made by the swarm even when the majority of the swarm is not able to measure the temperature. We therefore conclude, that the swarm is - to a certain extent - stable against malfunctioning members.

Experiments with a dynamic environment show, that even in a noisy and unstable environment the swarms' decision making process is reliable and the swarm is able to react to environmental changes.

Adding a second kind of gradient - the social seed - to an environmental gradient raises the question how the swarms' decision making reacts to such a gradient. Our results in simulation and in real robot experiments show, that the social component in the BEECLUST algorithm is very strong. One single social agent placed in the *local goal area* is enough to influence the decision making process significantly. However, if the environmental gradient is too weak where the social agent is placed, the aggregation will not be formed around the social agent but will be formed in the *global goal area* on the environmental gradient. We conclude that a social seed can be used to influence the swarms' decision making if need be.

The second part deals with Swarm Level Optimisation, where we introduce four different behaviour types to the BEECLUST algorithm. From the results we conclude that the aggregation process can be optimised by composing different swarm settings with respect to what the environment looks like. If chosen wisely, such a heterogeneous swarm performs better than a homogeneous swarm with only *Random Walkers* (like it was defined in the original BEECLUST). Evolution showed, that a more complex setting regarding the composition of the swarm can have a better performance, even when the results are counterintiutive because of the inclusion of *Immobile Agents*.

However, there are several questions left open. In the case of agents with impaired sensors, it is still unresolved how the aggregation process is affected if the sensors always measure the maximum temperature (thus having always the maximum waiting time) or if the sensors transmit random values. Also, more analysis need to be done regarding the *Immobile Agents*. It is still not clear, why *Immobile Agents* are part of the swarm that the evolutionary algorithm has found. To further improve the Swarm Level Optimisation with heterogeneous swarm settings, a lot of different things can be done: So far, the different behaviour types were assigned statically. Changing those roles during an experiment could further increase the performance of the swarm. For example, if a *Random Walker* managed to find the *global goal area* the agent could change its role to an *Immobile Agent* (this could also be a useful scenario for *Immobile Agents*). As the agent itself can only guess if it is aggregating in the *global goal area*, this could be done with the help of swarm size estimation or learning algorithms (online unsupervised learning).

Appendix

Appendix A.

List of Publications

A.1. Published

1. Daniela Kengyel, Heiko Hamann, Payam Zahadat, Gerald Radspieler, Franz Wotawa, and Thomas Schmickl. Potential of heterogeneity in collective behaviors: A case study on heterogeneous swarms. *In Principles and Practice of Multi-Agent Systems*, PRIMA 15, page 1, 2015. Nominated for best paper award.
   Heiko Hamann and I developed the idea of this paper. Heiko Hamann implemented the evolutionary algorithm of the four behaviour types. Gerald Radspieler provided the data of bee experiments. Payam Zahadat wrote the basic implementation of the evolutionary algorithm

and helped me with integrating it into the simulation. I wrote the simulation, performed and evaluated the experiments. Heiko Hamann wrote the introduction, about the mathematical model and the individual behaviour. I wrote about the behaviour of young honeybees, the simulation, experiments and their results and the discussion and conclusion. Franz Wotawa and Thomas Schmickl helped in proof-reading the paper.

2. Daniela Kengyel, Gerald Radspieler, Franz Wotawa, and Thomas Schmickl. Emulation of collective honeybee behaviour by a swarm of simple robots. *Abstract at IUSSI 2012*, 2012.
   I wrote the abstract and had a talk at the conference to present the project "Rebodiment".

3. Daniela Kengyel, Ronald Thenius, Karl Crailsheim, and Thomas Schmickl. Influence of a social gradient on a swarm of agents controlled by the BEECLUST algorithm. *Advances in Artificial Life, Proceedings of the 12th European Conference on the Synthesis and Simulation of Living Systems, ECAL13*, 12:1041–1048, 2013.
   I developed the idea, implemented the simulation, made experiments and the evaluation and wrote the paper. Ronald Thenius adviced me during the evaluation. Thomas Schmickl helped in proof-reading the paper.

4. Daniela Kengyel, Payam Zahadat, Thomas Kunzfeld, and Thomas

Schmickl. Collective decision making in a swarm of robots: How robust the beeclust algorithm performs in various conditions. *In Bio-inspired Information and Communications Technologies, BICT 2015*, page 1, 2015.
I had the idea about the paper and the experiments. Thomas Kunzfeld wrote about the hardware and helped in performing the robot experiments. I made the evaluation of the experiments. I wrote the paper with the help of Payam Zahadat.

5. Tobias Meister, Ronald Thenius, Daniela Kengyel, and Thomas Schmickl. Cooperation of two different swarms controlled by beeclust algorithm. In *Advances in Artificial Life, ECAL*, volume 12, pages 1124–1125, 2013.
Ronald Thenius had the idea, Tobias Meister wrote the simulation and the paper. I helped with advices during evaluation of the simulation and writing the paper. Ronald Thenius proof-readed the paper.

A.2. Unpublished

1. Daniela Kengyel, Franz Wotawa, and Thomas Schmickl. Towards swarm level optimisation: The role of different movement patterns in swarm systems. *Artificial Intelligence*, 2016. submitted.
   Thomas Schmickl had the idea about "Swarm Level Optimisation". I implemented the model, made the analysis and wrote the paper. Franz Wotawa proof-readed the paper and gave helpful advices.

2. Daniela Kengyel, Payam Zahadat, Thomas Kunzfeld, and Thomas Schmickl. Collective decision making in a swarm of robots: Investigation of the beeclust algorithm in various conditions. In *Complex Systems*, 2016. Submitted.
   This paper is an extended version of a paper mentioned above. I had the idea about the paper and the experiments. Thomas Kunzfeld wrote about the hardware and helped in performing the robot experiments. I made the evaluation of the experiments. I wrote the paper with the help of Payam Zahadat.

Bibliography

[1] Alberto Acerbi, Davide Marocco, and Stefano Nolfi. Social facilitation on the development of foraging behaviors in a population of autonomous robots. In Fernando Almeida e Costa, LuisMateus Rocha, Ernesto Costa, Inman Harvey, and António Coutinho, editors, *Advances in Artificial Life*, volume 4648 of *Lecture Notes in Computer Science*, pages 625–634. Springer Berlin Heidelberg, 2007.

[2] Farshad Arvin, Shyamala Doraisamy, Khairulmizam Samsudin, Faisul Arif Ahmad, and Abdul Rahman Ramli. Implementation of a cue-based aggregation with a swarm robotic system. In *Knowledge Technology*, pages 113–122. Springer, 2012.

[3] Farshad Arvin, Khairulmizam Samsudin, Abdul Rahman Ramli, and Masoud Bekravi. Imitation of honeybee aggregation with collective behavior of swarm robots. *International Journal of Computational Intelligence Systems*, 4(4):739–748, 2011.

[4] Farshad Arvin, Ali Emre Turgut, Farhad Bazyari, Kutluk Bilge Arikan, Nicola Bellotto, and Shigang Yue. Cue-based aggregation with a mobile robot swarm: a novel fuzzy-based method. *Adaptive Behavior*, 22(3):189–206, 2014.

[5] Farshad Arvin, Ali Emre Turgut, and Shigang Yue. Fuzzy-based aggregation with a mobile robot swarm. In *Swarm intelligence*, pages 346–347. Springer, 2012.

[6] Farshad Arvin, Ali Emre Turgut, and Shigang Yue. Fuzzy-based aggregation with a mobile robot swarm. In *Swarm Intelligence (ANTS'12)*, volume 7461 of *Lecture Notes in Computer Science*, pages 346–347, Berlin, 2012. Springer.

[7] Frederic Bartumeus, M. G. E. da Luz, G. M. Viswanathan, and J. Catalan. Animal search strategies: a quantitative random-walk analysis. *Ecology*, 86(11):3078–3087, 2005.

[8] Michael Bodi, Ronald Thenius, Martina Szopek, Thomas Schmickl, and Karl Crailsheim. Interaction of robot swarms using the honeybee-inspired control algorithm beeclust. *Mathematical and Computer Modelling of Dynamical Systems*, 18(1):87–100, 2012.

[9] Michael Bodi, Ronald Thenius, Martina Szopek, Thomas Schmickl, and Karl Crailsheim. Interaction of robot swarms using the honeybee-

inspired control algorithm beeclust. *Mathematical and Computer Modelling of Dynamical Systems*, 18(1):87–100, 2012.

[10] Scott Camazine, Jean-Louis Deneubourg, Nigel R. Franks, James Sneyd, Guy Theraulaz, and Eric Bonabeau. *Self-Organizing Biological Systems*. Princeton Univ. Press, 2001.

[11] Gianni Di Caro and Marco Dorigo. *AntNet: A Mobile Agents Approach to Adaptive Routing*. IRIDIA - University Libre de Bruxelles, 1997.

[12] Jimming Cheng, Winston Cheng, and Radhika Nagpal. Robust and self-repairing formation control for swarms of mobile agents. In *AAAI*, volume 5, pages 59–64, 2005.

[13] Iain D. Couzin, Jens Krause, Nigel R. Franks, and Simon A. Levin. Effective leadership and decision-making in animal groups on the move. *Nature*, 433:513–516, February 2005.

[14] Sheetal Doctor, Ganesh K Venayagamoorthy, and Venu G Gudise. Optimal pso for collective robotic search applications. In *Evolutionary Computation, 2004. CEC2004. Congress on*, volume 2, pages 1390–1395. IEEE, 2004.

[15] Marco Dorigo. Optimization, learning and natural algorithms. *Ph. D. Thesis, Politecnico di Milano, Italy*, 1992.

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

[17] Przemyslaw Grodzicki and Michal Caputa. Social versus individual behaviour: a comparative approach to thermal behaviour of the honeybee (*Apis mellifera* l.) and the american cockroach (periplaneta americana l.). *Journal of Insect Physiology*, 51(3):315 – 322, 2005.

[18] Heiko Hamann. Towards swarm calculus: Urn models of collective decisions and universal properties of swarm performance. *Swarm Intelligence*, 7(2-3):145–172, 2013.

[19] Heiko Hamann, Bernd Meyer, Thomas Schmickl, and Karl Crailsheim. A model of symmetry breaking in collective decision-making. In S. Doncieux, B. Girard, A. Guillot, J. Hallam, J.A. Meyer, and J.B. Mouret, editors, *From Animals to Animats 11*, volume 6226 of *LNAI*, pages 639–648. Springer-Verlag, 2010.

[20] H. Heran. Untersuchungen über den Termperatursinn der Honigbiene (*Apis mellifica*) unter besonderer Berücksichtigung der Wahrnehmung strahlender Wärme. *Zeitschrift für vergleichende Physiologie*, 34:179–206, 1952.

[21] James Hereford. Analysis of a new swarm search algorithm based on trophallaxis. In *Evolutionary Computation (CEC), 2010 IEEE Congress on*, pages 1–8. IEEE, 2010.

[22] James Hereford. Analysis of beeclust swarm algorithm. In *Swarm Intelligence (SIS), 2011 IEEE Symposium on*, pages 1–7. IEEE, 2011.

[23] James M. Hereford. Analysis of BEECLUST swarm algorithm. In *Proc. of the IEEE Symposium on Swarm Intelligence (SIS 2011)*, pages 192–198. IEEE, 2011.

[24] Daniela Kengyel, Heiko Hamann, Payam Zahadat, Gerald Radspieler, Franz Wotawa, and Thomas Schmickl. Potential of heterogeneity in collective behaviors: A case study on heterogeneous swarms. In *Principles and Practice of Multi-Agent Systems*, PRIMA 15, page 1, 2015.

[25] Daniela Kengyel, Thomas Schmickl, Heiko Hamann, Ronald Thenius, and Karl Crailsheim. Embodiment of honeybee's thermotaxis in a mobile robot swarm. In *10th European Conference on Artificial Life (ECAL'09)*, volume 5777/5778 of *LNCS*. Springer-Verlag, 2011.

[26] Daniela Kengyel, Ronald Thenius, Karl Crailsheim, and Thomas Schmickl. Influence of a social gradient on a swarm of agents controlled by the beeclust algorithm. *Advances in Artificial Life, Proceedings of the*

12th European Conference on the Synthesis and Simulation of Living Systems, ECAL13, 12:1041–1048, 2013.

[27] Daniela Kengyel, Franz Wotawa, and Thomas Schmickl. Towards swarm level optimisation: The role of different movement patterns in swarm systems. *Artificial Intelligence*, 2016. Submitted.

[28] Daniela Kengyel, Payam Zahadat, Thomas Kunzfeld, and Thomas Schmickl. Collective decision making in a swarm of robots: How robust the beeclust algorithm performs in various conditions. In *Bio-inspired Information and Communications Technologies*, BICT 2015, 2015.

[29] Daniela Kengyel, Payam Zahadat, Thomas Kunzfeld, and Thomas Schmickl. Collective decision making in a swarm of robots: Investigation of the beeclust algorithm in various conditions. In *Complex Systems*, 2016. Submitted.

[30] James Kennedy and Russel C. Eberhart. *Swarm Intelligence*. Morgan Kaufmann, 2001.

[31] Serge Kernbach, Ronald Thenius, Olga Kornienko, and Thomas Schmickl. Re-embodiment of honeybee aggregation behavior in an artificial micro-robotic swarm. *Adaptive Behavior*, 17:237–259, 2009.

[32] S.C. Lenaghan, Yongzhong Wang, Ning Xi, T. Fukuda, T. Tarn, W.R.

Hamel, and Mingjun Zhang. Grand challenges in bioengineered nanorobotics for cancer therapy. *IEEE Transactions on Biomedical Engineering*, 60(3):667–673, 2013.

[33] James Dwight McLurkin. *Stupid robot tricks: A behavior-based distributed algorithm library for programming swarms of robots*. PhD thesis, Massachusetts Institute of Technology, 2004.

[34] Tobias Meister, Ronald Thenius, Daniela Kengyel, and Thomas Schmickl. Cooperation of two different swarms controlled by beeclust algorithm. In *Advances in Artificial Life, ECAL*, volume 12, pages 1124–1125, 2013.

[35] Axel Michelsen, Bent Andersen, Jesper Storm, Wolfgang Kirchner, and Martin Lindauer. How honeybees perceive communication dances, studied by means of a mechanical model. *Behavioral Ecology and Sociobiology*, 30(3-4):143–150, 1992.

[36] Mark M. Millonas. Swarms, phase transitions, and collective intelligence. In Christopher G. Langton, editor, *Artificial Life III*, pages 417–445, Reading, MA, 1994. Addison-Wesley.

[37] Francesco Mondada, Michael Bonani, Xavier Raemy, James Pugh, Christopher Cianci, Adam Klaptocz, Stephane Magnenat, Jean-Christophe Zufferey, Dario Floreano, and Alcherio Martinoli. The

e-puck, a robot designed for education in engineering. In *Proceedings of the 9th conference on autonomous robot systems and competitions*, volume 1, pages 59–65. IPCB: Instituto Politécnico de Castelo Branco, 2009.

[38] Magnus Erik Hvass Pedersen and Andrew J Chipperfield. Simplifying particle swarm optimization. *Applied Soft Computing*, 10(2):618–628, 2010.

[39] Riccardo Poli. Analysis of the publications on the applications of particle swarm optimisation. *Journal of Artificial Evolution and Applications*, 2008:3, 2008.

[40] Jim Pugh and Alcherio Martinoli. Inspiring and modeling multi-robot search with particle swarm optimization. In *Swarm Intelligence Symposium, 2007. SIS 2007. IEEE*, pages 332–339. IEEE, 2007.

[41] Andrew Reynolds, Alan Smith, Don Reynolds, Norman Carreck, and Juliet Osborne. Honeybees perform optimal scale-free searching flights when attempting to locate a food source. *Journal of Experimental Biology*, 210(21):3763–3770, 2007.

[42] Craig W. Reynolds. Flocks, herds, and schools. *Computer Graphics*, 21(4):25–34, 1987.

[43] Thomas Schmickl, Stjepan Bogdan, Luis Correia, Serge Kernbach,

Francesco Mondada, Michael Bodi, Alexey Gribovskiy, Sibylle Hahshold, Damjan Miklić, Martina Szopek, Ronald Thenius, and José Halloy. The path to assisi: Mixing social animals with robots to a bio-hybrid society. In *Living Machines - The Internation Conference on Biomimetic & Biohybrid Systems*, page in press. IEEE, 2013.

[44] Thomas Schmickl and Heiko Hamann. BEECLUST: A swarm algorithm derived from honeybees. In Yang Xiao, editor, *Bio-inspired Computing and Communication Networks*. CRC Press, March 2011.

[45] Thomas Schmickl and Heiko Hamann. Beeclust: A swarm algorithm derived from honeybees. derivation of the algorithm, analysis by mathematical models and implementation on a robot swarm. *Bio-inspired Computing and Communication Networks. CRC Press (March 2011)*, pages 95–137, 2011.

[46] Thomas Schmickl, Heiko Hamann, Heinz Wörn, and Karl Crailsheim. Two different approaches to a macroscopic model of a bio-inspired robotic swarm. *Robotics and Autonomous Systems*, 57(9):913–921, September 2009.

[47] Thomas Schmickl, Ronald Thenius, Christoph Möslinger, Gerald Radspieler, Serge Kernbach, and Karl Crailsheim. Get in touch: Cooperative decision making based on robot-to-robot collisions. *Autonomous Agents and Multi-Agent Systems*, 18(1):133–155, February 2008.

[48] Thomas D Seeley. Honey bee foragers as sensory units of their colonies. *Behavioral Ecology and Sociobiology*, 34(1):51–62, 1994.

[49] Thomas D Seeley and P Kirk Visscher. Assessing the benefits of cooperation in honeybee foraging: search costs, forage quality, and competitive ability. *Behavioral Ecology and Sociobiology*, 22(4):229–237, 1988.

[50] Anton Stabentheiner, Sigurd Schmaranzer, Herbert Heran, and Renate Ressl. Verändertes Thermopräferendum von Jungbienen durch Intoxikation mit Roxion-S (Dimethoat). *Mitteilungen der Deutschen Gesellschaft für allgemeine und angewandte Entomologie*, 6:514–520, 1988.

[51] Katie L Stammler and Lynda D Corkum. Assessment of fish size on shelter choice and intraspecific interactions by round gobies neogobius melanostomus. *Environmental Biology of Fishes*, 73(2):117–123, 2005.

[52] David JT Sumpter and Madeleine Beekman. From nonlinearity to optimality: pheromone trail foraging by ants. *Animal behaviour*, 66(2):273–280, 2003.

[53] D.J.T. Sumpter and D.S. Broomhead. Shape and dynamics of thermoregulating honey bee clusters. *Journal of Theoretical Biology*, 204(1):1–14, 2000.

[54] Martina Szopek, Thomas Schmickl, Ronald Thenius, Gerald Radspieler,

and Karl Crailsheim. Dynamics of collective decision making of honeybees in complex temperature fields. *PloS One*, 8(10):e76250, 2013.

[55] Payam Zahadat and Thomas Schmickl. Wolfpack-inspired evolutionary algorithm and a reaction-diffusion-based controller are used for pattern formation. In *Proceedings of the 2014 Conference on Genetic and Evolutionary Computation*, GECCO '14, pages 241–248, New York, NY, USA, 2014. ACM.

[56] Zhi-Hui Zhan, Jun Zhang, Yun Li, and Henry Shu-Hung Chung. Adaptive particle swarm optimization. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 39(6):1362–1381, 2009.