Self-Organised Task Differentiation in Homogeneous and Heterogeneous Groups of Autonomous Agents

Sven Magg

A thesis submitted in partial fulfilment of the
requirements of the University of Hertfordshire
for the degree of

Doctor of Philosophy

The programme of research was carried out in the School of Computer Science, Faculty of Engineering and Information Sciences, University of Hertfordshire.

May 2011

Acknowledgements

Hofstadter's Law: It always takes longer than you expect, even when you take into account Hofstadter's Law.

- Douglas Hofstadter, Gödel, Escher, Bach: An Eternal Golden Braid

I would mostly like to thank all people for having such patience, especially my parents and my supervisor René te Boekhorst. It took me a long time to finish this thesis and they for some reason never gave up hope. Thanks also go to my friends and colleagues whose discussions, suggestions and help greatly improved this thesis. I am also thankful for their constant joking about the time it took me to write. The bad conscience often kept me at my desk.

Sven Magg

Abstract

The field of swarm robotics has been growing fast over the last few years. Using a swarm of simple and cheap robots has advantages in various tasks. Apart from performance gains on tasks that allow for parallel execution, simple robots can also be smaller, enabling them to reach areas that can not be accessed by a larger, more complex robot. Their ability to cooperate means they can execute complex tasks while offering self-organised adaptation to changing environments and robustness due to redundancy.

In order to keep individual robots simple, a control algorithm has to keep expensive communication to a minimum and has to be able to act on little information to keep the amount of sensors down. The number of sensors and actuators can be reduced even more when necessary capabilities are spread out over different agents that then combine them by cooperating. Self-organised differentiation within these heterogeneous groups has to take the individual abilities of agents into account to improve group performance.

In this thesis it is shown that a homogeneous group of versatile agents can not be easily replaced by a heterogeneous group, by separating the abilities of the versatile agents into several specialists. It is shown that no composition of those specialists produces the same outcome as a homogeneous group on a clustering task. In the second part of this work, an adaptation mechanism for a group of foragers introduced by Labella et al. (2004) is analysed in more detail. It does not require communication and needs only the information on individual success or failure. The algorithm leads to self-organised regulation of group activity depending on object availability in the environment by adjusting resting times in a base. A possible variation of this algorithm is introduced which replaces the probabilistic mechanism with which agents determine to leave the base. It is demonstrated that a direct calculation of the resting times does not lead to differences in terms of differentiation and speed of adaptation.

After investigating effects of different parameters on the system, it is shown that there is no efficiency increase in static environments with constant object density when using a homogeneous group of agents. Efficiency gains can nevertheless be achieved in dynamic environments. The algorithm was also reported to lead to higher activity of agents which have higher performance. It is shown that this leads to efficiency gains in heterogeneous groups in static and dynamic environments.

Contents

Chapter 1

Introduction

With the recent advances in technology and the associated drop in hardware prices, autonomous robots have already started to appear in households. Since the beginning of robotics, they were envisioned to do work that is too dangerous or hard to do for humans, or sometimes just too boring and repetitive. With the growing complexity of the tasks those robots are built for, the robots inevitably also became more complex. Complexity means that robots have to have more processing power, sensors and actuators which increases maintenance, probability of failure and once again prices. There are also tasks that are inherently too difficult for one robot to solve and need robots to work together as a team.

These problems are known since the field of robotics started to grow and already early on, researchers looked at robot collectives. Using a group of simple robots which exhibit cooperative behaviour promised higher robustness and fault tolerance due to redundancy in the group. Also designing and replicating simple robots instead of one complex, powerful one was thought to be cheaper and quicker. Together with a higher performance on tasks that could be parallelised, all these features are desirable.

Researchers therefore started to look into control mechanism to control a group of agents. Approaches with a central controller that commands all the agents could solve the problem algorithmically since it had all the information and could compute an optimal solution. With communication being unreliable in real-world scenarios and a single point of failure which would lead to the failure of the whole group, central control had to be abandoned for many tasks. Decentralised approaches where agents communicate with each other in order to organise themselves became popular and proved to be more robust and fault-tolerant. Failure of single agents most likely reduce the performance of the group, but do not automatically lead to failure of the whole group.

In order to make good or even optimal decisions, robots have to have good information about the environment. This can be accomplished by improving the sensing abilities or through communication with other robots. The first again increases the complexity of the agents as well as maintenance and price. Communication is costly as well since robots need the processing power in order to parse messages and, when needed in between the whole

group, communication bandwidth increases exponentially.

A solution can be found in nature, where swarms of social insects are a living proof that organisation without central control is possible. Agents in a swarm act on local and often very limited sensor information. Communication between agents is often not needed or limited to agents in the vicinity. The French biologist Grassé (1959) identified a way in which termites "communicate" indirectly through the environment. When using stigmergy, clues in their local environment, which are constantly changed by other agents, guides the building and organisation process. This type of information transfer is cheap and does not need costly communication or sensors and can be accomplished by very simple agents. Implementing models derived from biological swarm systems has therefore been a main research focus in collective robotics, leading to the research fields of swarm robotics and swarm intelligence.

When tasks require a large set of abilities, implementing all of them in every agent of the group may not be necessary. Splitting the abilities between agents which become specialists for certain subtasks can lead to a group which is still capable of executing the whole task efficiently. In such a group, intelligent task allocation is necessary so that agents are used for the correct task they are built for. Allocating the right number of the right agents to a given task in a decentralised system is not trivial and significantly increases the complexity of the control algorithm.

So far most work on self-organised division of labour was done in homogeneous groups but heterogeneous groups are moving more and more into the focus. This thesis contributes to this research by investigating whether homogeneous groups can be exchanged with a heterogeneous group by splitting the its abilities between different agent types. This is done using a clustering task and it is shown that even in this simple setting, the behaviour of the homogeneous group can not be mimicked by a heterogeneous group of any fixed composition (i.e. size of the subgroups).

The second part of this work extends work on an algorithm for self-organised division of labour in a foraging scenario. The "Variable Delta" (VD) algorithm presented by Labella et al. (2004) was shown to lead to robust division of labour and enabled a homogeneous group to adapt to changing environments. Labella also demonstrated that the VD algorithm reinforced differences in performance, leading to a "selection of the best individual" in the task allocation process. The algorithm was chosen for investigation due to its minimal sensing and resource requirements and because it does not require communication between agents. In this thesis, the parameters governing the behaviour of agents and groups using the VD algorithm are more deeply explored and a non-probabilistic VD algorithm introduced. Also the efficiency gains by using this algorithm are investigated in more detail and reasons for those gains given.

The thesis is based on three hypotheses which give the main points of investigation:

Hypothesis 1 That it is possible to find a composition for a heterogeneous group of two types of agents in a clustering scenario so that this group creates the same pattern as a

homogeneous group of versatile agents, which have the combined abilities.

Hypothesis 2 The efficiency increase in a group when using the VD algorithm shown by Labella et al. (2004) can not be generalised and is dependent on environmental set-up and agent parameters

Hypothesis 3 The "selection of best individuals" by the VD algorithm is the main factor impacting efficiency and therefore the algorithm works best with heterogeneous groups

Although the inspiration for this work and the algorithms used are taken from swarm robotics, the hypotheses were explored and analysed on an abstract multi-agent based level. This allowed for a higher number of experiments in the given time due to the abstract nature of the simulations but limits the ability to generalise the results to robotic settings. Since algorithms were used that were already validated in real robot settings, their analysis and extensions introduced in this work on an abstract level may feed back to robotics where they might prove useful and inform future experiments.

1.1 Main Contributions

Following the three hypotheses, the main contributions of this thesis are:

- The rejection of the first hypothesis for the chosen setting by showing that a homogeneous group of versatile agents combining the abilities of two specialised agents (bulldozer and grabber) can not be replaced by a heterogeneous group of the two types (Chapter 3). No heterogeneous group was able to consistently produce the pattern of the homogeneous group in the conducted experiments.
- It is shown that there is no efficiency gain when using Labella's VD algorithm in an environment with constant object density, when using a fully homogeneous group. By adjusting the ratio of active to inactive agents, the VD algorithm even leads to a loss in efficiency when compared to a fully active, non-adaptive group (Chapter 4).
- It is confirmed that agents which have a higher performance are becoming active first when the VD algorithm is used, which leads to an efficiency increase in heterogeneous groups. This efficiency gain increases with heterogeneity and is due to agents with lower performance becoming less active first when success rates in the group drop (Chapter 4).

This means that the first hypothesis had to be rejected while the other two hypothesis were confirmed. In order to investigate the main hypotheses, the used experimental set-ups had to be explored which lead to a list of many smaller contributions. The detailed list of contributions is as follows:

- Three measurements were introduced in order to analyse the structure of the created patterns in the clustering experiments: Chamber count, pile count and structural complexity. All three can be calculated automatically.

- It was shown that the pattern created by a heterogeneous group can not be directly derived by combining the individual patterns created by a homogeneous group of each type and the composition of the group. In this instance, the pattern of the heterogeneous group is close to the pattern created by one agent type as soon as this type is present in the group (given a certain minimum threshold) and does not change for a range of group compositions.

- The behaviour along walls of both homogeneous and heterogeneous groups in the clustering task was investigated and the differences between the group behaviour shown. These are then used to explain the different outcome for homogeneous groups.

- In the foraging task, results from Labella et al. (2004) were successfully reproduced. The VD algorithm successfully adjusted the number of active foragers according to the density in the environment and reacted to changes in the environment.

- Parameters were put into two classes depending on their effects on the group. The success rate of agents is defined and it is shown that parameters affecting this success rate have an effect on the self-organised activity pattern of the group. Parameters affecting cycle times of individual foraging loops and the proportion an agent spends on given subtasks were shown to have no impact on group coordination.

- It is shown that replacing the VD algorithm, which uses a probability to determine when an agent leaves the base, with a direct calculation of idle times has no impact on the system's behaviour.

- The efficiency gains shown by Labella could not be reproduced and it is shown that in an environment with constant object density, a homogeneous group showed a decrease in performance when the VD algorithm was active.

- Results are presented which show that group efficiencies for dynamic environments can not easily predicted by using results from static environments for the same group. Groups are shown that have higher performance in a combined time interval when using the VD algorithm, although the performance in the individual intervals was lower compared to a control group.

- Labella's finding of the "selection of the best individuals" was reproduced in heterogeneous groups. Agents that have a higher individual performance become more active when the VD algorithm is used to determine idle times.

- It is shown that the "selection of best individuals" leads to efficiency gains in heterogeneous groups because agents with lower performance are becoming inactive first, thus raising the groups average performance. This is shown to be especially the case in environments with low object densities where using the VD algorithm can lead to higher gains.

Due to the nature of the experiments, which were all conducted in an abstract setting, the contributions are primarily in the field of agent-based systems. Future experiments in real robot settings are necessary to validate and generalise the findings to the field of robotics.

1.2 Organisation of Thesis and Publications

This thesis is split into five chapters. Chapter 2 starts with an overview of work related to this thesis. It gives a brief history of collective robotics before explaining the concepts of swarm intelligence and swarm robotics. Work in two application fields of swarm robotics which are especially relevant for this thesis is then highlighted: Collective clustering and collective foraging. After going through general work in these two fields, the focus is on division of labour and work on heterogeneous groups. In the second part of the second chapter, some terms are defined that are used throughout this thesis. Definitions for task, subtask and goal precede more complex concepts like differentiation, proficiency and specialisation. These definitions are within the broader context of heterogeneity and also the nature (or cause) of differences and the distinction between static and dynamic heterogeneity are discussed.

This distinction is the basis for the work presented in chapter 3. The experimental set-up and agent types are explained first before presenting results on groups with static heterogeneity. These groups consisted of two types of agents which were able to move boxes in either an enclosed or toroidal environment. The patterns of clusters created by groups of different composition were analysed and compared to investigate the influence of the different sized subgroup on the resulting pattern. The results were published in the conference paper

- Magg, Sven and te Boekhorst, René.: 2006, Interaction and emergent behaviour in heterogeneous groups of artificial agents, *in* Artmann, S. and P., Dittrich, *Explorations in the Complexity of Possible Life, GWAL 2006*, IOSPress, Dresden, Germany, pp.95–103

In the second part of chapter 3 results of homogeneous groups of versatile agents in the same environments are presented. The patterns created by such a group are compared to the patterns from heterogeneous groups. Due to significant differences in these patterns, the behaviour along the walls and the behaviour over time in both groups were investigated more deeply. These results were also published in:

- Magg, Sven and te Boekhorst, René.: 2006, Pattern Formation in Homogeneous and Heterogeneous Swarms: Differences Between Versatile and Specialized Agents, *in proceedings of* ALIFE '07. IEEE Symposium on Artificial Life, pp.311–316

In chapter 4, work on self-organised coordination in a foraging task is presented. The work is based on experiments from Labella et al. (2004) who used a self-organised system to regulate group activity. After describing the original control algorithm and the set-up used in this work, effects of two classes of parameters are investigated. The first class are parameters affecting the individual success rate of agents while the second class has effects on the length of individual cycles of the foraging loop. It is then shown that depending on agent parameters, the density range in the environment where the VD algorithm can have an effect on group activity, varies. After confirming that the system at hand shows the same behaviour as the original, a different control algorithm is introduced. In this new algorithm, agents leave the base according to a direct idle time calculation instead of a probability. It is shown that the new algorithm gives the same results as the VD rule used by Labella and can therefore be used interchangeably. Although the adaptation mechanism is the same as in the original work and shows the same behaviour, the gains in efficiency could not be reproduced and the results and discussion is shown in 4.4.4.

In the last part of the 4th chapter, the impact of heterogeneity on the adaptation mechanism is investigated. It is shown that differences between agents which lead to differences in success rates lead to division of labour within the groups. Also the finding of Labella that individuals that are "better" than other agents are becoming more active, is confirmed. The efficiency of adaptive heterogeneous groups are again compared to a non-adaptive baseline group and it is shown that heterogeneity leads to a gain. At the end of the chapter the behaviour of heterogeneous groups in a dynamic environment is discussed.

The thesis is concluded by chapter 5 which contains a summary of the contributions and a discussion of possible extensions and future work.

One more paper has been published which contains a possible extension of this work and was presented at a workshop to have a discussion on the feasibility of the approach:

- Magg, Sven and te Boekhorst, René.: 2008, Task allocation by dynamic specialisation, *in proceedings of the* German Workshop on Artificial Life, Leipzig, Germany, pp.91–100

Chapter 2

Related Work and Definitions

This chapter provides an overview over work related to this thesis and provides definitions for terms and concepts used in later chapters.

2.1 Related Work and Background

When research into collective robotics started about 20 years ago when prices of robots dropped. With groups of robots available it was possible to envision and build groups of robots to test models which could only be tested in simulation before. This was aided by the emergence of behaviour-based robotics which meant that robot controllers needed less resources, especially processing power. After giving a brief overview over behaviour-based and collective robotics in 2.1.1 the concepts of swarm intelligence and swarm robotics are explained in 2.1.2 and 2.1.3. Since by now the number of publications and projects on collective approaches go beyond the space available in this thesis, only tasks directly relevant to this work are presented. In 2.1.5 work on clustering and sorting is introduced which is most closely related to the experiments in chapter 3. Section 2.1.6 describes work using a foraging task which will also be used in chapter 4. Since work on division of labour and heterogeneity in robot collectives is of particular interest in this thesis, work that emphasises on these concepts is discussed in the last two subsections 2.1.7 and 2.1.8.

2.1.1 From Classic AI Robot Controllers to Swarm Robotics

In the early years of autonomous robotics most controllers followed a classical, symbolic artificial intelligence approach. The agent used its sensors and in many cases domain knowledge to construct an internal representation of the world. With this internal model and a set of actions, it then used a planning algorithm to decide on a sequence of actions which — when executed — would get the agent to a given goal state. After execution of one or more of these actions, the agent would compare its current state of the internal model to current sensor data to verify consistency between real world and internal representation.

With this last step, the agent could react to unpredictable changes in its environment by updating the sequence of actions if necessary. Presumably the most famous instantiation of those systems was "Shakey" (Nilsson 1984). To extend this symbolic AI approach to multi-robot systems which act in real-time within in a dynamic environment turned out to be extremely difficult or "entirely unrealistic" (Steels 1990). In order to achieve this, every robot would have had to have a logical inference machine which could hold and update a detailed symbolic representation of the world in real-time.

A solution to this problem was presented by Brooks (1986) who introduced a behaviour-based approach in form of the subsumption architecture. In this approach, complex "intelligent" behaviour is decomposed into layers of "simple" behaviour modules following an incremental, bottom-up design process. Brooks argued in (Brooks 1991) that in order to create artificial intelligence

- We must incrementally build up the capabilities of intelligent systems, having complete systems at each step of the way and thus automatically ensure that the pieces and their interfaces are valid.
- At each step we should build complete intelligent systems that we let loose in the real world with real sensing and real action. Anything less provides a candidate with which we can delude ourselves.

Brooks' approach marked the beginning of a new design paradigm for robot controllers. Instead of trying to mimic higher level — "human" — thought processes, roboticists now started by creating simple but robust low-level controllers and tried to increase complexity incrementally. Instead of complex, computationally intense algorithms to decide on actions, the robots now used a direct coupling of perception and action. This approach led to robots that were able to perform their (mostly simple) task in real-time even in highly dynamic environments, which presented a big step towards autonomous robots. Although simple tasks like obstacle avoidance or path following can easily be achieved by these simple agents, creating a behaviour-based controller for more sophisticated capabilities turned out to be by no means trivial.

It didn't take long until the first behaviour-based multi-robot systems were envisioned for tasks where a population of collaborating or even cooperating robots could replace a single, more complex robot. In (1992) Kube and Zhang conjecture that "by organizing multiple robots into collections of task-achieving populations [...] useful tasks can be accomplished with simple behaviour based control mechanisms". The tasks they refer to are collective tasks, either cooperative or just collaborative. The former require a certain amount of cooperation between agents to fulfil the task and therefore require at least two or more robots to successfully complete it. Tasks that can be performed by a single, complex robot fall into this class if the capabilities can be distributed over a group of simple robots that have to team up to achieve the same function. Collaborative, non-cooperative tasks can be successfully completed by a single agent given enough time (e.g. sorting) and using a multi-robot system can lead to performance gains due to parallel task execution.

This new research focus contrasted work on centralised multi-robot task allocation (e.g. (Noreils 1990) and (Caloud et al. 1990)). Different names were proposed for the new field, including "cellular robotic system" (Beni 1988), "collective robots" (Kube and Zhang 1992, 1993), "large scale micro robots" (Dudenhoeffer and Jones 2000), and "swarm intelligent systems" (Beni and Wang 1989).

2.1.2 Swarm Intelligence

Although Beni and Wang (1989) coined the term swarm intelligence for a robotic system, it quickly started to encompass a wider field. Bonabeau et al. (1999) extended the term to include all approaches that are inspired by the collective behaviour of animal or insect societies. These societies generally consist of a large number of simple agents that interact locally with the environment and neighbouring agents. These simple interactions often lead to the emergence of more complex behaviour or patterns. The term *swarm robotics* was later used to describe approaches where the agents are real-world autonomous entities, while the term *swarm intelligence* was used as a wider term to include all approaches that are inspired by biological societies. Examples for the latter are biologically inspired optimisation techniques like Ant Colony Optimization (Dorigo et al. 1991, Dorigo and Stützle 2004) and Particle Swarm Optimization (Eberhart and Kennedy 1995, Kennedy and Eberhart 1995).

Garnier et al. (2007) lists several concepts underlying swarm intelligence, focusing on stigmergy and self-organisation. Stigmergy was introduced by Grassé (1959) to explain nest construction in termites. Grassé showed that information from the local environment guides worker activity and leads to the coordination and regulation of the building process. This means that workers do not have to have a built-in representation of the structure but merely have to follow simple rules dependent on the structure of the local environment. This can be seen as a indirect communication between agents through the environment.

The concept of self-organisation plays an important role in many collective behaviours of social insects (Deneubourg et al. 1987). Camazine et al. (2001) defines self-organisation as

> "a process in which patterns at the global level of a system emerge solely from numerous interactions among the lower level components of the system"

Bonabeau et al. (1999) as well as Garnier et al. (2007) identify components that underlay such a decentralised self-organised system:

- **Positive feedback** which amplifies fluctuations of the system
- **Negative feedback** which acts as a regulatory mechanism and stabilises the system by counterbalancing effects of positive feedback
- **Balance between exploration and exploitation.** Since random fluctuations are

amplified, new solutions can be explored while already established solutions are stabilised.

- **Multiple interactions** — direct or indirect — to produce apparently deterministic outcomes

Beni (2004) discusses the properties of an "intelligent swarm" and finally defines it as:

> "a group of non-intelligent robots ("machines") capable of universal material computation"

2.1.3 Swarm Robotics

Since "swarm intelligence" became a widely used term and by now spans different research fields, the term "swarm robotics" emerged to describe swarm approaches involving robotic systems. As already mentioned, this term coexists with terms like "collective robotics" or "distributed robot system". All three are used for systems comprised of groups of autonomous agents with decentralised control. Beni (2004) wrote that the number of agents in the system is insufficient to identify a swarm system and lists scalability of the control mechanism as necessity. In addition to these two criteria, Şahin (2005) also required agents to be autonomous, relatively incapable or inefficient and have only local sensing and communication capabilities. According to Şahin a system can be labelled swarm robotic system when a group of such agents is homogeneous or is composed of only few homogeneous subgroups.

The most frequently mentioned motivations to use a swarm system are robustness, flexibility and scalability (e.g. (Brooks et al. 1990, Şahin 2005)).

Robustness. A swarm system can be more robust than a single robot system for three reasons: *Redundancy*, *individual simplicity* and *distributed control*. A swarm consists of many simple agents that should be less prone to failure than a more complex agent. In addition to the reduced failure rate, redundancy in the system enables the system to still carry out the task when one or more agents fail. Finally, distributed control enables parts of the system to still continue after other parts of the system failed or got disconnected (e.g. left communication range).

Flexibility. Distributed, self-organised control mechanisms give a swarm the ability to adapt to the local environment. Emerging patterns and coordination are dependent on the current local environment and the current state of the local sub-group.

Scalability. Agents act on local information about the environment and neighbouring agents. Emergent coordination should therefore be unaffected by changes in group size.

These desired properties of a swarm system follow from observations of social insects where the same properties have been observed.

2.1.4 Applications for collective/swarm robotics

In the last 20 years, collective approaches were proposed for a wide range of applications. Cao et al. (1997) gave three reasons why a collective approach may be of interest for a certain application:

1. tasks may be inherently too complex (or impossible) for a single robot to accomplish, or performance benefits can be gained from using multiple robots;

2. building and using several simple robots can be easier, cheaper, more flexible and more fault-tolerant than having a single powerful robot for each separate task; and

3. the constructive, synthetic approach inherent in cooperative mobile robotics can possibly yield insights into fundamental problems in the social sciences (organization theory, economics, cognitive psychology), and life sciences (theoretical biology, animal ethology).

The last point highlights one of two possible links to biological systems. Sharkey (2006) labels them bio-robotic modelling and biologically inspired approaches. The first describes research which models aspects of insect behaviour "to address a biological hypothesis or demonstrate understanding of a biological system" Webb (2001). The latter has "greater emphasis on borrowing ideas from biology in order to develop solutions for tasks and applications in the real world" (Sharkey 2006).

Cao et al. lists three canonical task domains which have driven research in collective robotics: Traffic control (including obstacle avoidance, path planning and resolving resource conflict), box-pushing/cooperative manipulation and foraging. A different classification was later proposed by Beni (2004) who identified again three main problem areas: Pattern formation (aggregation, distributed deployment, mapping of area, etc.), problems that focus on specific entities in an environment (collective searching, object retrieval, homing), and more complex group behaviour (cooperative tasks, flocking). Both authors do not claim that those categories are exhaustive but merely show areas that are in the main focus. Other authors introduced possible taxonomies for multi-agent robotics and task classes (Gerkey and Matarić 2004, Dudek et al. 1996) using different axes. For this thesis the main focus is on clustering and foraging tasks and a summary of relevant work can be found in the following subsections.

2.1.5 Clustering and Sorting

Collective clustering is the process where agents pick up objects in an environment and drop them in areas where other objects of the same type are present. This behaviour can

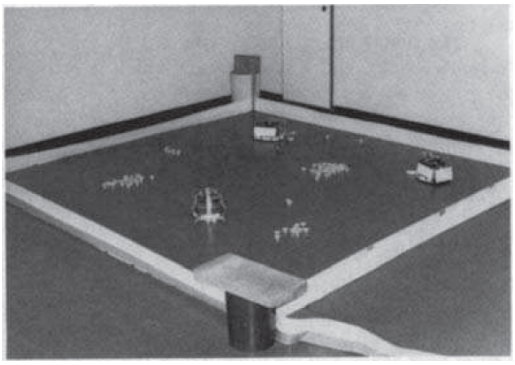


Figure 2.1: *Clusters are growing in a set-up with 3 robots (Beckers et al. 1994)*

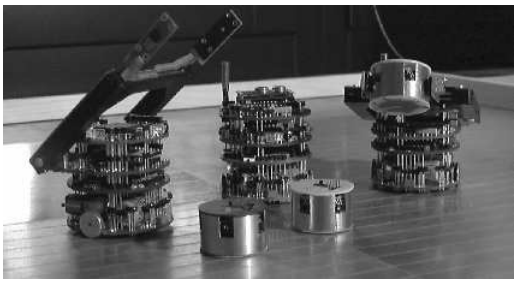


Figure 2.2: *Khepera robots and seeds as used by Martinoli et al. (1998)*

be found in biological systems, e.g. in wall building of ants or creation of comb patterns in honey bees (Camazine et al. 2001)

Deneubourg et al. (1991) proposed a model for clustering behaviour and compared it to corpse piling as observed in ants of the family *pheidole pallidula*. In this model, agents moved randomly through an environment and picked up and dropped objects that were randomly placed. The probability of picking up an object was inversely proportional to the density of objects in the immediate vicinity. When carrying an object, the probability of dropping it was directly proportional with object density. The system therefore used stigmergy and it is shown that the model exhibits many features of the biological system.

Collective clustering was also one of the first tasks used for a robot collective. Beckers et al. (1994) presented research where a homogeneous group of 1-5 robots clustered pucks in a rectangular environment. The robots were equipped with a C-shaped gripper to push pucks and travelled straight until an obstacle was detected in which case it was avoided by a random turn. A micro-switch behind the gripper was activated when the gripper contained three pucks or more at which point the robot reversed and turned away, leaving a small cluster of pucks behind. By following these simple behaviour rules, the robots formed clusters of pucks and Beckers et al. showed that the system always built one large cluster when given enough time.

The work was later repeated by Martinoli et al. (1998) who used 1-10 Khepera robots in a similar set-up. Robots were able to pick up one object, carry it and drop it next to another object when encountered. The group also created several clusters of objects, but in contrast to the work of Beckers et al. (1994), never created a single large cluster even when given enough time.

That robots moving through an area with objects can cluster those by "mistake" was shown by Maris and te Boekhorst (1996). Braitenberg type vehicles (Braitenberg 1986) were used in a rectangular area with randomly placed cubes. The control structure was such that agents avoided obstacles they sensed on one side by turning towards the opposite

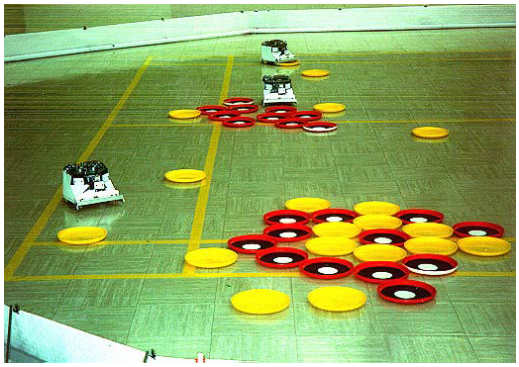


Figure 2.3: *Collective clustering of frisbees with different colour (Melhuish et al. 1998)*

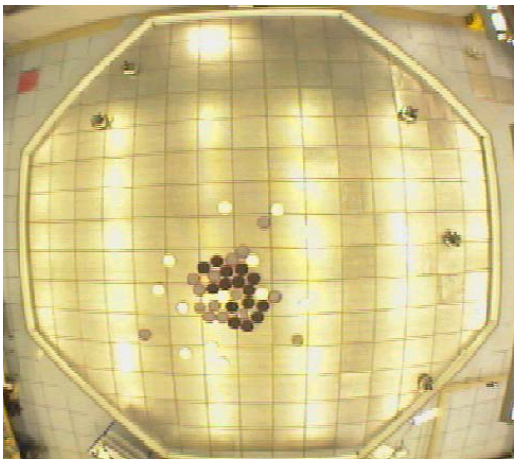


Figure 2.4: *Top view on arena with clusters after 4 hours (Wilson et al. 2004)*

direction. Due to the sensor configuration, robots could not detect cubes straight ahead and pushed those until another object made them turn. This simple behaviour lead to objects either being pushed to walls or to be clustered. This clustering was stable once the environment was "sufficiently structured [...] to manoeuvre, almost without hitting obstacles" (Maris and te Boekhorst 1996).

Using almost the same control algorithm as Beckers et al. (1994), Melhuish et al. extended collective clustering to collective sorting. Instead of pucks, their U-bots were able to pick up and carry frisbees of different colours (yellow and red-rimmed black). After reproducing the clustering results from Beckers et al. (1994), they added an additional pull-back rule: Before dropping a yellow frisbee, a robot reversed a certain distance and only then triggered the release. A red/black frisbee was dropped without pulling back. With this simple rule, clusters emerged with a centre of red/black frisbees surrounded by yellow ones. Experiments were also carried out with varying pull-back distances and the effect on the annular structures analysed. The work was later refined to deal with multiple coloured discs, also taking the colour of the frisbee into account that the robot collided with (Melhuish et al. 2001). It was shown that clustering accuracy dropped when the number of colours increased above 10 but worked well for lower numbers up to seven.

Wilson et al. (2004) introduced three more variants to the annular sorting algorithm:

- Object clustering using objects of different size
- Extended differential pull-back
- Combined leaky integrator

The results were measured using a performance measure that took into account separation, compactness, shape, and completeness of clusters. The annular sorting algorithm worked

well with objects of different sizes and after parameter optimisation, the leaky integrator algorithm was able to improve on the compactness loss of the differential pull-back.

All the mentioned approaches worked with distributed autonomous agents which had no means of direct communication but exploited stigmergy effects in order to achieve clustering.

2.1.6 Foraging

Foraging is a widely studied area and one of the main research areas lists by Cao et al. (1997). This task is based on foraging behaviour found in social insects where a number of agents have to find and retrieve objects scattered in an environment. It is particularly interesting for researchers since it can be part of a wider task, for example collective building, where objects have to be collected and then arranged in a specific order. A taxonomy of robot foraging systems which also includes performance criteria can be found in (Winfield 2009).

To my knowledge, one of the first implementations of foraging in a group of decentralised autonomous robots can be found in (Matarić 1992). She used a group of 20 robots, which used basic behaviour primitives in a subsumption architecture (Brooks 1986). In addition to performing basic foraging, robots were able to communicate success of searching to recruit nearby agents. When a robot found a puck, it broadcasted a signal. Other robots receiving this signal entered a "tracking" state to locate the source of the signal and search at that location. While tracking, robots also emitted a signal which prompted other agents to follow them (active recruitment).

At the same time, Arkin (1992) presented simulated multi-agent foraging without communication. The agents used a control structure based on motor schemas (Arkin 1987) and had to find and collect randomly placed "goals" in an environment and return them to a base. If two or more agents tried to retrieve the same goal, the speed of the retrieval was increased, leading to a performance increase due to cooperation. It was shown that agents successfully cooperated to retrieve the goals and that increasing the number of agents also increased the speed in which the task was completed. The work was then extended to include communication of behavioural states and it was shown that this improved cooperation between agents (Arkin et al. 1993). A more extensive investigation on the effects of communication on performance in multi-robot systems was presented by Balch and Arkin (1994). They extended the simulation used by Arkin to include also a "Consume" and "Grazing" task.

Consume was similar to Forage but agents did not retrieve the object. They had to stay at the location for a time proportional to the "mass" of the object before the object vanished. In the Graze task, agents had to visit ("cover") the whole environment and could sense already visited areas. Three types of communication were investigated: a) no communication between agents, b) state communication where agents show their current state, and c) goal communication where the position of a perceived goal is broadcasted. In the second case only a binary signal was transmitted (agent is searching or not) which corresponded to

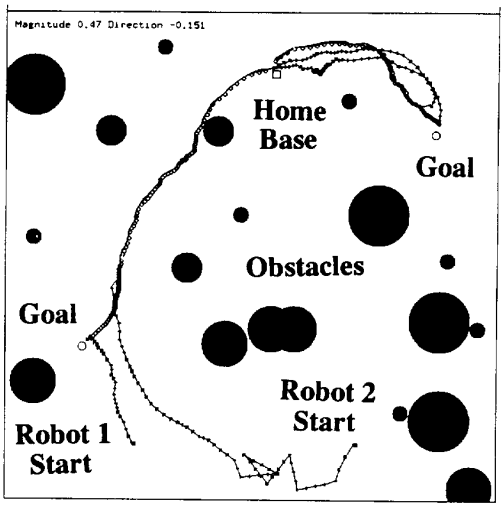


Figure 2.5: *Paths of two robots retrieving to goals (Arkin et al. 1993)*

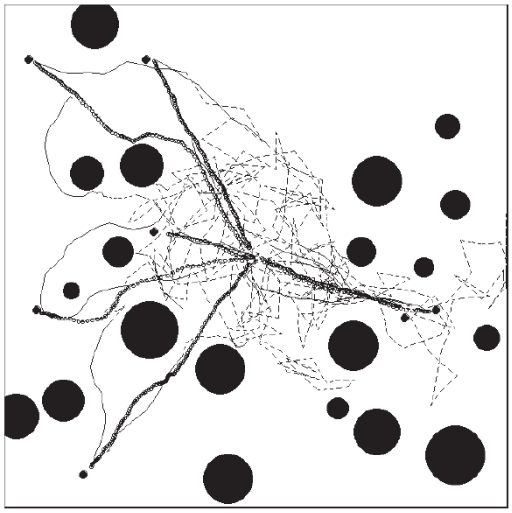


Figure 2.6: *Simulation of Forage with two robots and seven attractors (Balch and Arkin 1994)*

the behaviour communication in (Arkin et al. 1993). This type of information could also be perceived by observation without intent of sending it by the first agent whereas goal communication is active and intentional. All three communication types were tested on all three tasks in simulation and real robots and the results were:

- Communication improves performance significantly in tasks with little environmental communication (*Forage* and *Consume*)
- Communication is not essential in tasks which include implicit communication (*Graze*)
- Goal communication offers little benefit over state communication

In these experiments, it was also shown that task performance does not increase linearly with numbers of robots in the group and can even decline for larger group sizes. This is due to interference between robots and tackling this problem was also identified by Matarić (1992) as "the first goal in controlling a multi-robot system" . As soon as agents operate within, or have to pass through, a confined area, time spent on avoidance movements increases at the expense of performance.

2.1.7 Division of Labour

Division of labour means that different agents in a group execute different subsets of tasks and is one way to reduce interference within a group. For most systems there is an "optimal" number of agents to achieve maximum group performance (for a given performance measure).

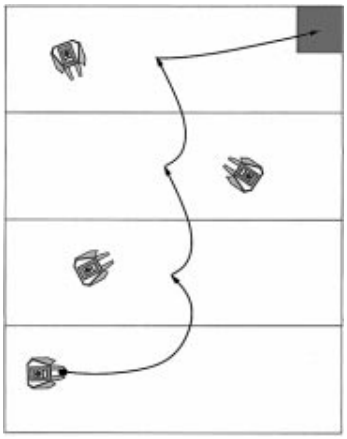


Figure 2.7: *A puck's path from a distant region to the "home" region (upper right corner) in (Schneider-Fontán and Matarić 1998)*

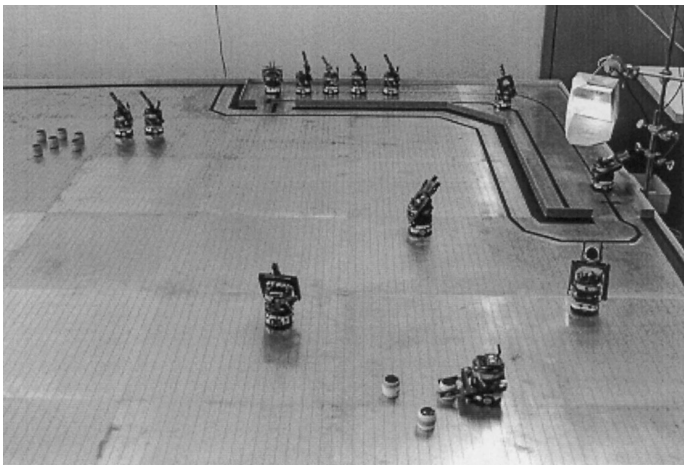


Figure 2.8: *Experimental set-up used by Krieger and Billeter (2000) with nest in top right corner.*

Therefore efficient division of labour, which keeps the number of agents near this optimum for each subtask, is a desired feature for a swarm system. The term "task allocation" is often used to describe the same process but the term "allocation" implies an allocator and therefore does not describe emergent processes accurately. Since interference quickly became a problem in collective robotics, many different mechanisms to reduce it have been proposed.

Schneider-Fontán and Matarić (1998) used a spatial approach to minimise interference in a foraging scenario. The environment was partitioned into different workspaces and each agent only moved within its designated area. Agents collected items within their work area and dropped them in the workspace closer to the base. They demonstrated that increasing group size had a negative effect on the performance of the group and it is difficult to judge whether there was improvement over a non-territorial approach for small group sizes. A similar approach is the idea of a "bucket brigade" in which robots transport items part of the way to the base and then hand them over to the next robot (Ostergaard et al. 2001, Goldberg and Matarić 2003, Lein and Vaughan 2008)

Instead of spatially dividing the agents, Parker (1998) presented an algorithm for intentional cooperation to achieve division of labour. ALLIANCE is a fully distributed and behaviour-based software architecture. Robots using this architecture communicate their current activities and make decisions incorporating the information gathered from other robots. Although reliable communication is not assumed nor required and ALLIANCE is designed to work even without, it was not intended for use in swarm applications.

Division of labour is well studied in social insects and several models have been proposed to describe it (see (Beshers and Fewell 2001) for an overview). In a threshold-based model (Theraulaz et al. 1998), each agent responds to an environmental cue when its internal threshold for this cue is lower. Different thresholds within a group lead to division of labour if the current environment only causes some agents to respond.

Krieger and Billeter (2000) used a threshold-based approach in a foraging scenario to modulate the number of agents engaged in two activities *foraging* and *inactive/waiting*. Successfully foraged items increased the nest's energy level, which was maintained centrally and communicated to all robots. Once the energy level dropped below a randomly chosen activation threshold of a robot, this robot became active. An activated robot was able to recruit other robots waiting in the nest if it still knew the location of food items. This simple set-up was enough to demonstrate self-organised division of labour in an almost decentralised group. The system was not fully decentralised since the nest maintained the global energy level.

Another threshold-based approach was presented by Agassounon and Martinoli (2002). They compared three different threshold algorithms: Fixed private threshold, variable private threshold, and public fixed threshold. The simulated environment was based on the aggregation task reported in (Holland and Melhuish 1999). As stimulus, a time T_{Search} was used which represented the time an agent spent searching and that was reset to zero when an object was found. When this time was above a threshold, the agent left the working area and became idle. In the fixed private and public threshold algorithm, every agent had the same fixed threshold, whereas in the variable threshold algorithm, agents estimated the stimulus and adjusted the threshold in a first phase and kept it fixed thereafter. In the public threshold algorithm, the stimulus was communicated to nearby agents and an average of the agent's own and all received stimuli used for threshold comparisons. The results showed that using a threshold-based algorithm lead to the same or better aggregation results when compared to a group with a fixed group size while using fewer agents. The differences between the threshold algorithms themselves were less apparent and they concluded that a public variable threshold algorithm might lead to better results regarding robustness to environmental changes. Also the inability to adapt the threshold to changes later in the experiment was pointed out as a drawback.

Labella et al. (2004) implemented a model proposed by Deneubourg et al. (1987) where an agent's probability to become an active forager was continually adapted according to this agent's foraging success. Labella used a circular environment with a central nest area in which agents had to find and retrieve objects. Each agent had an internal value representing the probability in each time step to leave the nest after a resting period. This internal value was increased or decreased by a fixed value according to success or failure on the last foraging attempt. This simple mechanism lead to successful division of labour in the group without communication or knowledge about other agents. Labella also showed that the efficiency of the adapting group was higher compared to a non-adapting group in terms of collected objects per time spent outside the nest. He also showed that the algorithm

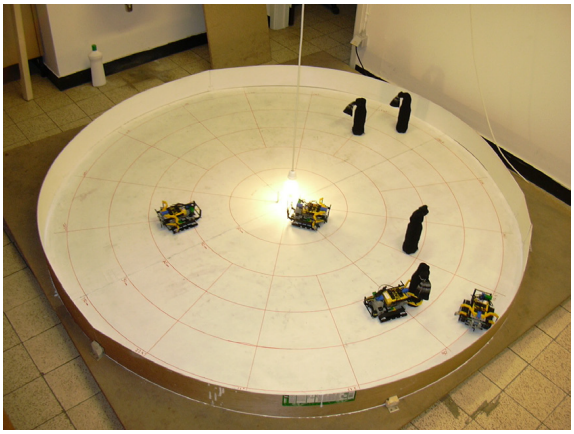


Figure 2.9: *Foraging in circular environment with nest in centre (Labella et al. 2004)*

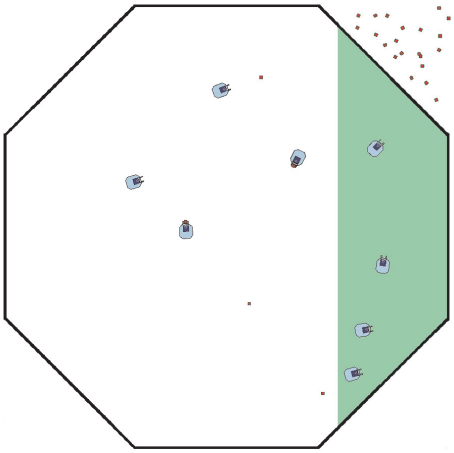


Figure 2.10: *Simulated environment with nest on right side as used in (Liu et al. 2007)*

lead to the "selection of best individuals", i.e. agents with a higher foraging performance (objects per time) had a higher probability to become foragers. This algorithm and the results will be explained in more detail later in this work.

Liu et al. (2007) have extended this approach to also take into account social and environmental cues. In addition to Labella's internal cue (success in object retrieval), they used environmental (number of collisions with other agents) and social cues (success of other agents) to adapt the time spent in the nest or searching. They used two different variables: T_{Search} and T_{Rest}, which are the time spent searching before returning and the time spent resting in the nest. After successful object retrieval, an agent reduced T_{Rest} and increased T_{Search} (internal) and also communicated its success to other agents in the group who would do the same adjustments (conversely for failure in retrieval). When an agent collided with another agent in the environment, it reduced T_{Search} and increased T_{Rest}. Experiments were performed using only internal cues, internal and environmental cues, and all cues together and compared to a baseline group using no cues to adapt. As a performance measure they used the net energy of the swarm which was increased by retrieved objects and decreased by agent activity. The results show that environmental and social cues increase the swarm's ability to adapt to changes in the environment. In terms of performance, using the additional cues was most profitable when the density of objects in the environment was low. In his PhD thesis (Liu 2008) he also presents a mathematical macroscopic probabilistic model for heterogeneous collective foraging which is validated using simulation.

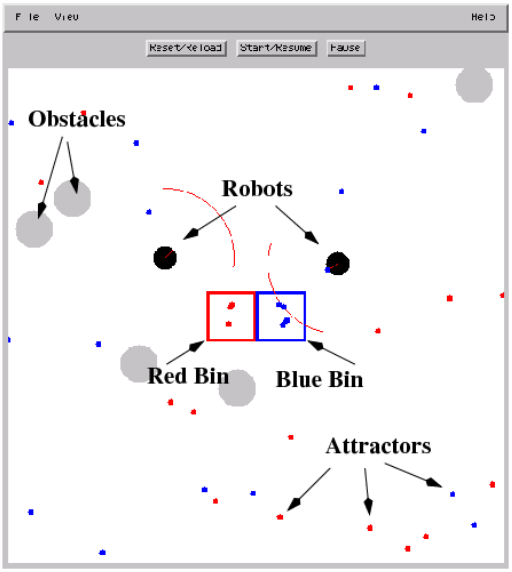


Figure 2.11: *Foraging and sorting of two objects (blue,red) into according areas in the centre (Balch 1999)*

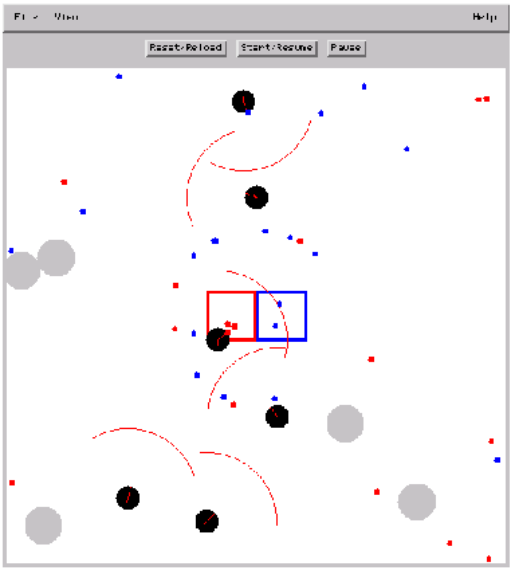


Figure 2.12: *Overworked sorter robot in heterogeneous territorial setting (ring of non-sorted objects around centre) (Balch 1999)*

2.1.8 Heterogeneous Foraging

Although many models and approaches mentioned above can be extended to heterogeneous groups, most of them were tested with homogeneous groups. The ALLIANCE framework (Parker 1994) was explicitly developed for heterogeneous groups and was used in different settings. Simmons et al. (2000) presented results on an autonomous assembly task by a heterogeneous group. They have used three robots (a roving eye, a crane and a manipulator) with ALLIANCE controllers to build a structure of beams. Howard et al. (2006) describes a heterogeneous team which can explore and map an environment and deploy sensor robots for intruder detection. The group consisting of mapper/leader and sensor robots used the ALLIANCE framework for controllers. Although these approaches were successful, the ALLIANCE framework does not scale very well and is not well suited for swarm approaches which require simple control structures.

Although the main body of swarm robot research was on homogeneous groups, there were projects that focused on heterogeneity. As already mentioned above, Labella et al. (2004) as well as Liu (2008) investigated the effects of the adaptation mechanisms on a heterogeneous group.

Balch (1999) presented a collective foraging/sorting approach with heterogeneous groups of simulated agents. Agents had to collect two different types of objects (red,blue) and deliver them to accordingly coloured areas in the centre of the environment. Three types of groups were used: Behaviourally homogeneous, specialise-by-colour and territorial. In

the homogeneous group, all agents indiscriminately collected objects and delivered them to the respective area, whereas the second group consisted only of agents specialised on one colour. The third group was split into foragers, which collected all objects and a sorter that sorted the objects by colour. The sorter stayed in a fixed area near the coloured bins, while the foragers collected objects and dropped them at the boundary of the sorter's area. All groups were tested with different group sizes from 2 - 9. The performance of the groups was measured in correctly delivered objects after a given time. Balch reported that the homogeneous group outperformed both other groups although the heterogeneous groups were designed to reduce interference. In the discussion, he already listed problems with the heterogeneous groups. There was only one sorter in the territorial group which was a bottleneck in larger groups leading to lower performance. In the specialise-by-colour group, the agents segregated themselves depending on the position of their corresponding delivery area. Agents that left the blue delivery area were more likely to stay on that side of the environment which got reinforced by collision avoidance when colliding with agents while trying to cross to the other side.

The related work presented in this section is by no means an exhaustive list but merely contains good representatives for the corresponding areas.

2.2 Terminology and Definitions

Many terms used in an interdisciplinary field like collective robotics are drawn from the different areas the field is composed of where the terms are often used with different connotations. It is therefore necessary to define how terms are used in this work and to point out that they are not used in any way that goes beyond the following definitions.

After a general definition of the terms group, task, goal and subtask, different aspects of the concept of heterogeneity are explained in 2.2.2. After describing the nature of heterogeneity, the terms differentiation, proficiency and specialisation are defined. Afterwards, ways of measuring heterogeneity or quantitatively distinguishing groups with different levels of heterogeneity are presented. Since two different types of heterogeneity are distinguished in this thesis, the difference between static and dynamic heterogeneity is defined at the end.

2.2.1 Task, Goal and Subtask

An **agent** is an autonomous entity which is able to execute a set of actions $\mathcal{A}_{agent} = \{action_1, action_2 \dots action_n\}$ in order to achieve given tasks. A **group** is a set $\mathcal{G} = \{agent_1, agent_2 \dots agent_n\}$ of agents and the capabilities of the group are the union of the capabilities of its agents: $\mathcal{A}_{\mathcal{G}} = \bigcup_{agent \in \mathcal{G}} \mathcal{A}_{agent}$.

According to Merriam-Webster a task is "a usually assigned piece of work often to be

finished within a certain time" [1], which is a rather general definition and can be used at different levels in a robotic system. In collective robotics for example, it can mean the overall task of the group (e.g. foraging), a task for a specific subgroup of agents (e.g. foraging one of many object types), or the current task of a single robot (find and retrieve one object). The difference between an action and a task is usually that a task was assigned and that task implies a more complex piece of work which can be performed by executing a series of actions.

In this work, a standard **task** is defined as a tuple

$$T_{x+1} = (G(\mathcal{T}_x, \mathcal{E}), Goal) \tag{2.1}$$

where $G(\mathcal{T}_x, \mathcal{E})$ is a directed graph, with a set of tasks $\mathcal{T}_x$ as vertices and a set $\mathcal{E}$ of ordered pairs of tasks $\in \mathcal{T}_x$. The second part is a **goal** against the performance of the task can be measured. By using this definition a task hierarchy is specified with x denoting the level of the task and $\mathcal{T}_x$ the set of tasks with level $\leq x$. Tasks $\in \mathcal{T}_0$ (i.e. atomic tasks) are defined as actions of agents together with the trivial goal "Action successfully executed". This is a simple definition of a task but is sufficient for the scope of this thesis.

A group has the ability to achieve $\mathcal{T}$ if $\mathcal{T}_0 \subseteq \mathcal{A}_{group}$, i.e. all actions in $\mathcal{T}_0$ are within the group's capabilities.

It is important to note that defining goals is not trivial, since goals in this interpretation can be defined in a wide variety of ways. The only requirement in this thesis is that performance of an agent executing the task can be measured against this goal, which encompasses binary goals (find object) but also goals like "Find as many objects as possible in the time given" (performance can then be defined as the number of objects found). It is not necessary that an agent can measure the performance itself, only that it can be measured by an observer.

Tasks themselves can be put together to compose more complex tasks. An example would be the task "foraging" which can be defined as task graph consisting of a repeated sequence of "searching", "retrieving", and "resting". In the case of nested tasks, tasks within the set would be called **subtasks** (containing sub-goals, etc).

2.2.2 Heterogeneity

The word heterogeneous originates from the Greek word *heterogenēs* which itself is composed of the words *heteros* (another, different) and *genos* (kind, clan) and is defined as "consisting of dissimilar or diverse ingredients or constituents" [2]. Following this definition and applied to robotics and multi-agent systems, a heterogeneous group contains at least one pair of agents which differ from one another.

[1] http://www.merriam-webster.com/dictionary/Task, accessed 17/05/2009

[2] Merriam-Webster Dictionary, http://www.merriam-webster.com/dictionary/heterogeneous, accessed 14/02/2010

This definition leads to two important aspects of heterogeneity: The nature of the differences between individual agents and the degree of heterogeneity of the group.

Nature of Heterogeneity.

Differences between agents can stem from either morphological or behavioural dissimilarities. In general these can easily be identified, e.g. when two agents have different actuators for different tasks (morphological heterogeneity). But the notion of "dissimilarities" is very general and also includes cases where it becomes more difficult to categorise agents of a group into subgroups. A simple example would be the colour of a robot since different coloured robots within a group would constitute a heterogeneous group due to visual dissimilarities. However, if this colouring has no influence on other robots nor the environment and hence no effect on the behaviour of the agents, the group should be defined as homogeneous (since it shows the same behaviour as a uni-coloured group). This shows clearly that it is not dissimilarities that a human observer would pick out but differences that have direct or indirect impact on the behaviour of the agents. These may or may not be directly perceivable by a human observer. In the field of collective robotics and especially swarm robotics there is often a feedback loop between agent and environment which enables small differences to be reinforced. The human observer, although being oblivious to the difference causing it, can then see the difference in behaviour after some time. When these differences are subtle (viewed from the agent's perspective), it can be difficult to categorise a group as heterogeneous or homogeneous before they are observed in action.

Once the differences are identified — either by design or observation — a metric can be defined to quantify the degree of difference between agents. The type of difference metric $d : \mathcal{A} \times \mathcal{A} \to \mathbb{R}$ for agents from an agent set $\mathcal{A}$, depends strongly on the type of differences that are of interest and the parameters of the system that can be measured or observed and is therefore context specific.

Specialisation and Differentiation

The terms "Specialisation" and "Differentiation" are sometimes used interchangeably in the literature on robotic systems; when agents differentiate by executing a task more often than other agents, they become "specialists" for that task. According to English dictionaries, the words "specialist" or "specialise" mean

- "very skilled at a particular subject" (specialist - Longman Dictionary of Contemporary English)
- "designed, trained, or fitted for one particular purpose or occupation" (specialised - Merriam-Webster Online Dictionary)
- "an organism specialised especially in food or habitat" (specialist - Merriam-Webster Online Dictionary)

- "concentrate on and become expert in a particular skill or area" (specialise - Compact Oxford English Dictionary)

All these definitions imply that a specialised agent is (a) better at a task than a non-specialist and (b) specialised on *one or a few* tasks only. Differentiation in turn is more general, denoting the act of becoming different and does not necessarily have any of the mentioned implications of specialisation. Since these terms are used with slightly different connotations in the literature the following paragraphs define how they are used within this work.

Differentiation. It is assumed that there exists a difference metric $d : \mathcal{A} \times \mathcal{A} \to \mathbb{R}$ on the set of agents $\mathcal{A}$ in the group which captures differences between agents that influence their behaviour, i.e. two agents when placed in the same environment behave differently. This metric can be defined by a simple categorisation of observed behaviours or as distance in a continuous behavioural space and is generally system dependent. With this metric a group $\mathcal{G}$ is defined as homogeneous when it satisfies $\forall a_1, a_2 \in \mathcal{G} : d(a_1, a_2) = 0$ and heterogeneous otherwise. Two agents a_1 and a_2 are differentiating when $\dot{d}(a_1, a_2) > 0$, i.e. when their difference is increasing over time. Therefore differentiation of a group at time t can be defined as cumulative difference of all pairs in G:

$$\text{Differentiation}_t(\mathcal{G}) = \sum_{i=1}^{|\mathcal{G}|-1} \sum_{j=i+1}^{|\mathcal{G}|} \frac{d_t(a_i, a_j)}{\binom{|\mathcal{G}|}{2}} \tag{2.2}$$

and a group is differentiating when

$$\frac{d}{dt}(\text{Differentiation}_t(\mathcal{G})) > 0 \tag{2.3}$$

Is has to be noted that this definition is independent of the reasons or the effects of differentiation and only describes the amount of differences within a group as measured by d. If differentiation leads to a different frequency of task execution among agents it is called **division of labour** or **task allocation**. Task allocation, following the definition of "to allocate", implies an external force which is responsible for the division of labour. Since there is no external force in a self-organised system, the term *differentiation* is preferred in this work to distinguish between externally and internally caused effects.

Proficiency. Every agent has capabilities which enable it to work towards achieving a task with a certain performance. Different agents in a group, all able to achieve a task, might have different ways of doing so with each way being somewhat efficient, leading to different performance values. Performance is again a context-specific measure and has to be defined for each task and purpose. Examples are time for achieving a task (e.g. time for finding and retrieving one object), number of times a task was executed within a given time

(e.g. number of objects retrieved in given time) or quality of result after achieving a task (e.g. detail achieved through a mapping task). Under the assumption that a performance measure $pf : \mathcal{G} \times \mathcal{T} \rightarrow [0, 1]$ exists for $agent \in \mathcal{G}$ and $task \in \mathcal{T}$ with $\mathcal{T}$ being the set of all available tasks, we can define the *proficiency* of an agent as

$$Proficiency(agent, \mathcal{T}) = \sum_{task \in \mathcal{T}} \frac{pf(agent, task)}{|\mathcal{T}|}. \tag{2.4}$$

Performance pf has to be normalised to $[0, 1]$ to allow for comparison between tasks (which may have different maxima). A value of zero means the agent cannot execute the task and one being the best performance possible in the current setting. The *proficiency* of an agent describes its overall ability to perform tasks from the given task set using the performance measure $pf(agent, task)$. A higher proficiency means an agent can execute tasks "better" than other agents with a lower score but does not necessarily mean an agent can perform all tasks $\in \mathcal{T}$. It also has to be emphasised that this does not necessarily mean that the agent is a specialist for any task $\in \mathcal{T}$ since it only fulfils the first condition for specialisation mentioned above. In a system where task performance affects agent behaviour it can generally be said different proficiency implies a difference, i.e.

$$Proficiency(a_1) \neq Proficiency(a_2) \Rightarrow d(a_1, a_2) > 0 \tag{2.5}$$

whereas the reverse does not necessarily hold.

Specialisation. In biology the term "specialisation" is used to describe functional differentiation of a group into distinct subgroups; agents within a subgroup are "specialists" for the task they execute. Therefore an agent specialises when executing a task more often than other agents in the group but without necessarily having a higher performance compared to other agents (if performance is defined on single tasks). Here a distinction has to be made between differentiation and specialisation to increase the emphasis on a higher performance of a specialists over a non-specialists. As already mentioned, the notion of proficiency is not enough to capture the second aspect of specialisation: being skilled in one or a few tasks only. To incorporate this aspect, the specialisation of an agent $a \in \mathcal{G}$ for a task $t \in \mathcal{T}$ can be defined as

$$Specialisation(a, t) = Proficiency(a, \{t\}) - Proficiency(a, \mathcal{T} \setminus \{t\}) \tag{2.6}$$

$$= pf(a, t) - \sum_{tk \in \mathcal{T}, tk \neq t} \frac{pf(a, tk)}{|\mathcal{T}| - 1} \tag{2.7}$$

This means that an agent is specialised if he has a high proficiency for this task in comparison to all other tasks. It should be noted that

$$Specialisation(a_1, t) = Specialisation(a_2, t) \nLeftrightarrow Proficiency(a_1, \{t\}) = Proficiency(a_2, \{t\}) \tag{2.8}$$

i.e. two agents a_1 and a_2 can have different proficiency for a task although being equally specialised and vice versa. Basing specialisation on proficiency entails that agents within a group, where differentiation leads to different frequencies of task execution, are not becoming specialists.

Degree of Heterogeneity

Often the difference between homogeneous and heterogeneous was seen as dichotomy — a group either consists of individuals with identical capabilities or not (see e.g. (Cao et al. 1997)). With increasing research interest in heterogeneous groups, this bipolar view limited quantitative comparison between groups of agents and researchers have started to use different metrics to be able to compare groups in terms of degree of heterogeneity.

Generally degree of heterogeneity has to be defined on a continuous scale where the minimum depicts a homogeneous group and the maximum a fully heterogeneous group with groups consisting of a number of distinct subsets in between.

Balch (2000) lists three aspects of a heterogeneous group a measure has to capture:

1. Number of distinct behavioural subsets in the group
2. Relative proportions of elements in each subset
3. Degree of difference between subsets (e.g. as distance in the behavioural or parameter space)

He then introduces *simple social entropy* based on Shannon's information entropy which captures the first two aspects and, in a second step, combines it with the hierarchical clustering algorithm C_u (Jardine and Sibson 1971) to define *hierarchical social entropy* which, according to Balch, fulfils all three requirements.

Condensing these three aspects down to a single number inevitably means to lose information. Depending on the context and the value needed to distinguish groups from one another in terms of heterogeneity, a simple measure will suffice or a combination of measures is needed. For example, Parker (1994) introduced the concept of task coverage to estimate the demand for cooperation. Task coverage is defined as the ability of an agent to achieve a given task and is high in homogeneous groups (all agents can achieve all tasks) and is lowest if the ability to achieve the given tasks is fully partitioned amongst agents in the group, i.e. no two agents can individually achieve the same task (under the assumption that the group covers all tasks. It can be even lower if a group can not achieve all tasks $\in T$. Since the focus of Parker's work was on cooperation within both homogeneous and heterogeneous groups, this measure, which captures points one and two of Balch's aspects, was sufficient.

Pre-set vs. Dynamic Heterogeneity

Another categorisation is the notion of static (or pre-set, predefined) and dynamic heterogeneity. This is comparable to the distinction between morphological and temporal polyethism in social insect division of labour (Beshers and Fewell 2001). Morphological differences (due to their fixed nature) almost always fall into the first category. Differences within this category are static and therefore do not change over the life time of an agent or the time of an experiment. They are present from the beginning and have a constant effect (for a given environmental context) on the behaviour of the agents. This can still mean that the behaviour of the group changes over time with changing environmental context, but the effect on the reaction of an agent in a given context stays constant. Examples are morphological differences (actuators, sensors, size,...) and differences in the control software (e.g. differing obstacle avoidance algorithms or parameter sets).

Dynamic heterogeneity means that differences appear and develop over time depending on the agent's individual history. For this, agents have to have memory to a certain degree, i.e. have to have the means to "remember" past actions and/or the state of the (local) environment or have a state that changes over time. This memory can range from complex world models to simple variables and has an impact on the control architecture, leading to different behaviour dependent on the history of this individual agent. Without static heterogeneity within the group, the behaviour of two agents will be equal when the representation of their memory is equal, which can lead to a temporally homogeneous group. Although dynamic heterogeneity in robotics is generally achieved by adapting internal variables which effect the control software, changes in morphology are also conceivable. Examples can be found in nature, for example in social insects like ants (e.g. Sendova-Franks and Franks (1993)) and honey bees (e.g. Beshers et al. (2001)). While state-of-the-art robots generally do not have the capability to change their morphology over time (unlike most living creatures, e.g. muscle growth) it is conceivable that parameters of sensors or actuators could be changed to mimic morphological adaptation in the natural world.

Chapter 3

Preset Heterogeneity and the Impact of Group Composition

3.1 Introduction

When designing agents for collective tasks, one approach is to design agents for each task and create a heterogeneous group with a ratio reflecting expected task availability to use the group to full capacity. Since different tasks often require different actuator and/or sensor configurations and each agent type is designed especially for one task, these single-task agents are generally less complex than agents capable of executing multiple tasks. This simplicity can result in higher reliability and lower production and maintenance costs but in turn requires knowledge about the environment to decide on optimal group composition to achieve high task performance. In comparison, a homogeneous group of versatile agents has the ability to adapt to changing or initially unknown task availability in the environment, which a heterogeneous group of single-task agents lacks. Although both types of groups are designed to have the competence to complete all tasks, the efficiency of an adapting group should be higher due to a more optimal distribution of agents over tasks. Therefore there is a trade-off between simplicity and adaptability which depends on the amount of change that is to be expected in the environment and the costs for combining tasks in single agent.

If both approaches are viable, the question is whether the two are interchangeable, i.e. whether a group composition for a static heterogeneous group can be found that leads to the same outcome as a homogeneous group of versatile agents for a given environment.

To investigate this question, a box pushing scenario was chosen in which agents can either carry or push boxes and emerging patterns in the environment can be used to categorise group behaviour. In a first step two agents were defined for each task (grabbing and pushing) and the behaviour of each, as well as the effects of group composition in a heterogeneous group investigated. In section 3.3 these findings are then compared to the results of a homogeneous group of versatile agents and. It was found that no heterogeneous

setting could produce the same result as the versatile group for the same starting conditions and differences that lead to this outcome were analysed.

3.2 Groups with Pre-set Heterogeneity

In order to examine the impact of group composition on the behaviour of the system, a box-pushing scenario was chosen. In this scenario agents push and carry boxes through the environment, creating dynamically changing patterns. In the first set of experiments the heterogeneity of the group was pre-set and static, i.e. individual agents do not change their behaviour contingent on internal variables and only react to the current structure of the local environment. The patterns created by heterogeneous groups with varying group compositions (ratio of different agent types) were categorised and used to compare the groups.

3.2.1 Experimental Setting

A 2-dimensional artificial world was simulated in NetLogo representing an area containing boxes and agents. Both a toroidal and an enclosed world were used to identify effects that are due to the interaction of agents with walls and distinguish them from agent-agent interaction effects that would also be prevalent in a toroidal setting. The space of the world was divided into discrete patches that could be occupied by either a box, a wall, or a single agent. Boxes and walls were represented by differently coloured patches and agents could detect their presence by querying the colour of the corresponding patch. "Pushing a box" was simulated by switching the colours of start and destination patch and no physical properties usually associated with boxes (e.g. friction, orientation of boxes, etc) was taken into account.

3.2.2 Agent Types

The world was inhabited by two kinds of agents: a bulldozer-type agent ("Dozer") and a grabber-type agent ("Grabber").

Dozers perform what is called "blind bulldozing". They always move straight forward until they encounter an obstacle. If the obstacle is a box and the patch behind the box is empty, the agent pushes the box into that patch and takes its place. If the patch behind is already occupied, the agent treats the box as obstacle and avoids it. Walls and other agents are always considered obstacles which are avoided by changing heading randomly (left or right) by a fixed angle of 45 degrees. Pseudocode of the Dozer can be seen in Algorithm 1.

Dozers push boxes until they run into another object and subsequently turn away, thus leaving the box behind. In a toroidal set-up, this may lead to endless loops where agents push boxes forever around the world. In order to avoid this, an additional rule was

Algorithm 1 Control loop of a "Dozer"-type agent in pseudo-code. The random number is chosen from [0,1] and max-push-distance is the maximum number of steps an agent can push a box.

```
loop {Control loop for Dozer agent}
  if Object one step ahead then
    if Object ahead = box then
      if random number < (push-distance / max-push-distance) then
        Turn 45° in random direction
        Reset push-distance
      else if No object two steps ahead then
        Move one step forward and push box
        Increment push-distance
      end if
    else
      Turn 45° in random direction
      Reset push-distance
    end if
  else
    Move one step forward
    Reset push-distance
  end if
  lay down pheromone
end loop
```

introduced. An agent loses the box it pushes with a probability proportional to the distance pushed, after which it turns 45 degrees in a random direction. At every step, Dozers also lay down an artificial "pheromone". A simple diffusion and evaporation model is used, i.e. the pheromone is represented by a float variable at every patch and diffusion (not influenced by objects) and evaporation are calculated every time step.

Grabbers "grab" and "carry" boxes (also when turning) instead of pushing them. Like Dozers, Grabbers move straight forward, until they encounter an object. In case the object is not a box, or if the Grabber already holds a box, it turns 45 degrees in a random direction. Otherwise, they grab it or treat it as obstacle depending on the concentration of pheromone at that location with the probability of grabbing the box increasing with local pheromone concentration. In turn, the probability of dropping the box increases with decreasing pheromone concentration and the distance the box was already carried. Grabbers possess a switch which, when on, enables them to continuously deposit pheromones with the same intensity as Dozers. This ability has to be pre-set for each experiment and can not be toggled by the agent itself. For pseudo-code of the control loop of a Grabber, see Algorithm 2.

Algorithm 2 Control loop of a "Grabber"-type agent in pseudo-code. The random number is chosen from [0,1] and max-carry-distance is the maximum number of steps an agent can carry a box. The amount of artificial pheromone laid down is either the same as for Dozers or 0 depending on experiment setting.

```
loop {Control loop for Grabber agent}
  if Carrying box then
    if Object one step ahead then
      Turn 45° in random direction
      Increment carry-distance
    else if random number > pheromone concentration then
      Drop box into patch ahead
      Reset carry-distance
    else if random number > (carry-distance / max-carry-distance) then
      Drop box into patch ahead
      Reset carry-distance
    else
      Move one step forward
      Increment carry-distance
    end if
  else if Object one step ahead then
    if Object ahead = box and random number < pheromone concentration then
      Move one step forward and carry box
    else
      Turn 45° in random direction
    end if
  else
    Move one step forward
  end if
  if Pheromones enabled then
    lay down pheromone
  end if
end loop
```

Both agents are acting on the same resource (boxes) and the algorithms were designed to get competing (i.e. essentially antagonistic) behaviours: Dozers push single boxes until they encounter another box, creating piles, whereas Grabbers were implemented to take boxes and drop them in areas with little agent activity. Since the activity of agents should be higher near walls due to avoidance movements, Grabbers should work against the Dozer's propensity to create piles. The algorithms were chosen in order to be minimal in terms of sensing and computing requirements and show antagonistic behaviour. It was assumed that this would result in two behaviours that are connected (i.e. the agents are not switching

between completely different tasks requiring different resources), which would allow for more interesting system dynamics and combination possibilities.

At the beginning of each experiment 65% of available patches were coloured as box and, in case of an experiment in an enclosed environment, the border patches were coloured as wall. The number of boxes was determined through tests to render the system unstable, thus more variable, for small differences to have a higher impact. A lower box count tends to result quickly in only one pile in the middle, while larger values often lead to one small chamber due to quick wall formation, blocking further travel. All agents started from the middle of the environment (the only case in which more than one agent can occupy the same patch) with a random orientation, spreading outwards after the beginning of the experiment. The central starting position was chosen over a random spatial distribution for comparability reasons. Random starting positions lead to a high number of agents trapped in small unconnected chambers, leading to varying numbers of active agents and thereby highly fluctuating group compositions within the starting phase of experiments. Tests showed no difference in later phases of experiments or in emerging patterns between the two starting conditions.

All experiments were run for 300,000 time steps and at each time step, all agents where consecutively activated. 300,000 time steps was found to be long enough for the system to settle in all experiments. For each parameter setting (see subsection 3.2.5) 10 runs were performed and the results averaged.

During an experiment different structural patterns emerged which, from the viewpoint of a human observer and depending on box density, can either be described as a number of piles or a number of "chambers" and "corridors". Examples of these structures can be seen in Figure 3.1.

3.2.3 Measurements

Three different measurements were used to characterise the emerging structures.

1. **Number of chambers.** This measure is calculated using a two-step algorithm. In a first step the distance from each empty patch to the nearest box is computed, which results in a 2-dimensional matrix of distance values. Local maxima within this matrix are identified in the second step and counted as number of chambers, i.e. a centre of a chamber is defined by a set of patches with larger or equal distance to the nearest box than all neighbouring patches (using a Moore neighbourhood). With increasing noise in the pattern (increasing number of free-standing boxes) the algorithm becomes unreliable as an ever-increasing number of chambers is detected. In such cases, the structure has to be reviewed manually.

2. **Structural Complexity.** Structural complexity is measured as the information dimension (Baker and Gollub 1990) by a multi-step algorithm. At each step, the pattern is covered by a grid and entropy is calculated by binning grid cells into three

classes: a grid cell either consists completely of white patches (boxes), black patches (empty) or contains both. Therefore, the entropy for a pattern and grid cell size x is:

$$E(x) = -\sum_{i=1}^{3} p_{x,i} \log_3 p_{x,i} \tag{3.1}$$

where $p_{x,i}$ is the probability that a quadrant with side length x belongs to class i. Structural complexity is measured as the slope of the regression of $E(x)$ for growing values of x starting with 2. The slope is negative, since entropy is decreasing with increasing grid cell size and is close to zero for patterns that consist of only a few piles/chambers. As for the chamber count, this measurement is also sensitive to noise due to the choice of bins, leading to lower values for noisy patterns (since there are fewer all-black cells with increasing cell size).

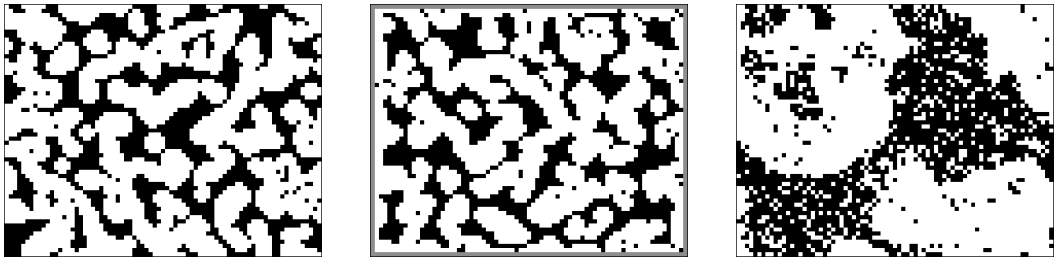
A correlation between structural complexity and chamber count can be expected, since the formation of chambers (connected areas with no boxes), combined with the corresponding increase in areas filled with boxes around a chamber, will inevitably lead to a higher value for structural complexity. After the initial phase, a decreasing number of chambers implies an increased size of the remaining chambers and therefore an increased value for structural complexity, leading to an inverse relationship between the two measurements.

3. **Number of piles** To complement the chamber-count measurement also piles of boxes are counted. A pile is defined as a connected conglomeration of boxes (using a von Neumann neighbourhood) with a cardinal number higher than 5. Piles are counted by a breadth-first search algorithm consecutively marking neighbouring box patches, incrementing the pile count, and starting again with an unmarked box patch. In comparison to the other measurements, counting piles is noise-resistant but doesn't capture the structure of the underlying pattern very well. For example, the pattern shown in Figure 3.1b consists of 8 piles, of which one pile contains about 80% of the boxes, implying it could be similar to Figure 3.1f which consists of 6 piles.

3.2.4 Behaviour of Homogeneous Dozer/Grabber Groups

Before experimenting with heterogeneous groups, results from homogeneous groups of each type were analysed to show differences between the agent classes. Both agent classes show different behaviour resulting in characteristic patterns.

The behaviour of Dozers can be described as "keep areas free of boxes". They push single boxes to the walls delineating chambers which leads to a structure always consisting of narrow corridors and small chambers (as can be seen in Figure 3.1a–(b)). This is due to the initial random distribution of boxes; the maximum distance a box can be pushed straight



(a) *Homogeneous Dozer group in toroidal environment*

(b) *Homogeneous Dozer group in enclosed environment*

(c) *Homogeneous Grabber group in toroidal environment*

(d) *Homogeneous Grabber group in enclosed environment 1*

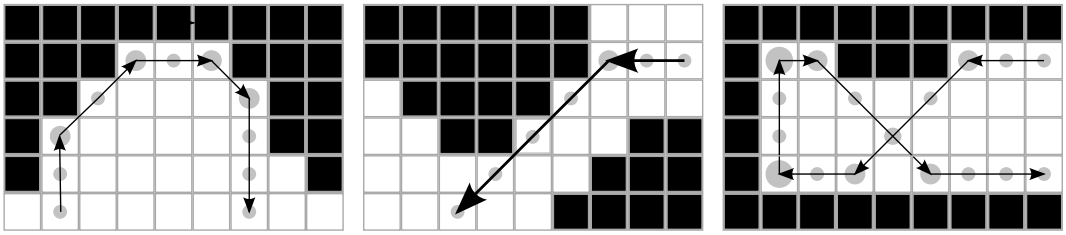
(e) *Homogeneous Grabber group in enclosed environment 2*

(f) *Heterogeneous group (90% Dozer) in enclosed environment*

Figure 3.1: *Examples of emerging structural patterns after 300,000 time steps. White squares depict boxes, black squares depict empty spaces and gray squares represent walls. Agents are not shown.*

without colliding with the next object is short. Once all movable boxes are pushed against each other no further changes can occur, since — due to the simplicity of the simulation and the lack of physical interaction between agents and boxes — boxes can not be removed from a pile by collisions. The final structure is static and its properties for groups with size > 10 independent of group size. For simplicity, this pattern is from now on referred to as "Dozer-pattern". Generally, the number of chambers in this characteristic structure is more than 30 and the value for structural complexity is around -0.25.

Due to mutual avoidance movements in a group, Dozers change directions more often in groups with a higher number of agents and therefore have an increased chance to encounter boxes that can be pushed away. In this way they open up the area for their fellow Dozers to travel further and therefore prevent single or small groups of Dozers from getting trapped. As a result, with increasing group size, also the chance to get the Dozer-pattern increases. If, in a group less of than 10 agents, all or most agents get trapped in fully closed chambers, the final structure often only partly resembled the Dozer-pattern with the rest being the initial random pattern. With groups of 10 or more agents there was no occurrence of this "unfinished" Dozer pattern in any experimental run. Since Dozers do not react to the



(a) *Concave wall* **(b)** *Convex wall* **(c)** *Corridor with dead end*

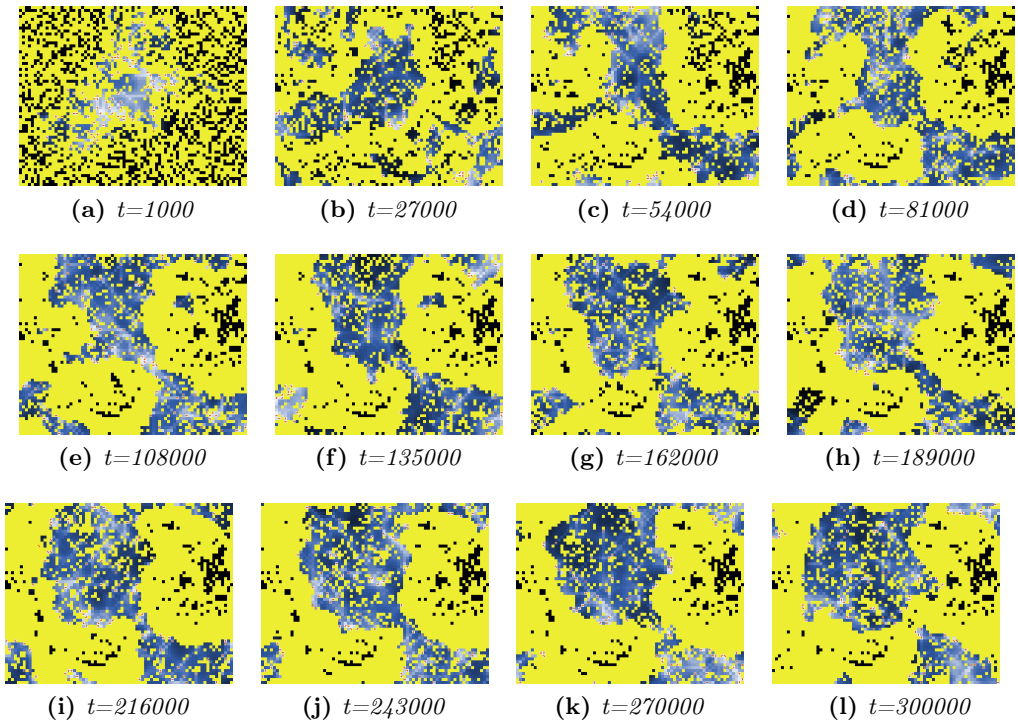
Figure 3.2: *Path of agents at convex and concave walls and along corridor dead ends due to 45° avoidance turns. Arrow ends depict turning points and grey circles outline differences in pheromone deposits due to movements. Since avoidance turns have random direction, shown paths are optimal (i.e. pheromone deposits minimal at turning points) and agents often turn back and forth at path nodes until possible move is found, leading to higher pheromone concentration. Agents may turn and travel the same path backwards, but, since the first possible move is taken after random turns, never deviate from shown path except when avoiding oncoming agents.*

pheromone they lay, concentrations of pheromone could be ignored.

In contrast to Dozers, Grabbers generally move boxes from areas with many agents to less densely populated areas. Because agents spend more time at obstacles and walls due to their avoidance algorithm and thus the concentration of the simulated pheromone in these areas is higher (see Figure 3.2), Grabbers tentatively move boxes from larger piles to the centre of chambers (due to wall following behaviour the concentration of pheromone away from walls is generally lower and the corresponding probability to drop the item higher. See Figure 3.2(a) and (c) for example). This activity can be seen as a counter force to the activity of Dozers which generally push free-standing boxes against walls.

A homogeneous group of grabbers is only active if they are able to drop pheromones since the grabbing behaviour is dependent on the presence of pheromones. Without pheromones, Grabbers only travel but never move a box. As can be seen in Figure 3.1c–(e), the pattern generated by such a group is noisy, but on a closer look generally consists of a few large chambers connected by corridors (or from a different point of view a few big piles). This pattern changes dynamically over time (see Figure 3.3) but keeps its basic structure (from now on referred to as "Grabber pattern"). If the noise, which strongly affects the measurements, is removed, the number of chambers is generally less than 10, the value for structural complexity is around -0.05 and the number of piles is usually less than 3.

Since Grabbers can always move boxes as long as there are agents laying down pheromone, they constantly change their environment. Due to this they are less likely to get trapped inside structures given that there are pheromone laying agents with them (or if they lay pheromones themselves). The only way to trap a pheromone laying Grabber is to enclose it



(a) *t=1000* **(b)** *t=27000* **(c)** *t=54000* **(d)** *t=81000*

(e) *t=108000* **(f)** *t=135000* **(g)** *t=162000* **(h)** *t=189000*

(i) *t=216000* **(j)** *t=243000* **(k)** *t=270000* **(l)** *t=300000*

Figure 3.3: *Patterns produced over time by a Grabber-only group in a toroidal environment. Grabbers are depositing pheromones (displayed as gradient black-blue-white), patches with box are coloured yellow.*

while it is holding a box since it would require a free patch to drop the box.

The reaction to higher pheromone concentrations can clearly be seen near corridor dead-ends, walls and concave chamber walls where the presence of Grabbers generally leads to an immediate widening of the space as reaction to high agent density.

It is known from other studies (e.g. by Maris and te Boekhorst (1996)) that agents moving through an environment filled with objects tend to push these objects together. In the case of Dozers this behaviour was expected to emerge since it is inherent in their implementation, but it turned out that also Grabber groups tend to amass boxes to piles despite their tendency to move boxes away from already aggregated piles on the individual level. This behaviour was clearly visible in toroidal environments (see Figure 3.1c), where the result generally was a single pile, but also in enclosed arenas where the end result was a few piles along walls (Figure 3.1d) or a main pile in the middle (Figure 3.1e). These two outcomes for enclosed environments were mutually exclusive since the formation of a big pile in the centre implied the destruction of all piles along the walls.

3.2.5 Results from Groups with Fixed Heterogeneity

To investigate the behaviour of mixed groups, composed of Dozers and Grabbers, and how the behaviour is affected by group composition, the following set of experiments was conducted; all experiments were run for 300,000 time steps and four parameters were changed:

- **Group composition.** This was the main parameter that was investigated. The composition is always given in percent of Grabbers in the group and was varied from 0% to 100%, usually in steps of 10%. If the pattern changed significantly within one step, one or more intermediate points were chosen to get better resolution.

- **Number of agents.** In a first set of experiments the total number of agents was varied from 1 to 100 while the group composition was kept constant (assuming that the ratio was possible with that agent count). In general, the result was that the number of agents only affected the speed (in number of time steps) in which the pattern, which was typical for the given ratio, emerged. Provided all other parameters being equal, a group of 50 agents produced a similar pattern to a group of 100 agents. The only notable difference was for small groups or subgroups (< 10) where the loss of agents due to entrapment had a high impact since the Dozer:Grabber ratio changed drastically when members of one subgroup got trapped. Here the resulting pattern was dependent on the number of agents trapped, their type, and also the time of entrapment. The non-trapped members of the group went on creating the pattern typical for a group with the new composition, if the pattern at the time of entrapment still allowed for it.

  An example would be a group with eight Dozers and two Grabbers where both Grabbers get trapped after half the experiment time, changing the group to a Dozer only group. Since it is unlikely that the Grabbers will be freed again by Dozer actions and the pattern was already influenced by the presence of Grabbers (and assuming the typical pattern for a 20% group wasn't reached yet), the resulting pattern would not resemble a Dozer pattern nor a pattern for a 20% group but some blended pattern. There was also no minimum number necessary for the typical pattern to emerge as long as the group ratio could be kept until the end of the experiment and enough experimental time was left (i.e. even one Dozer or Grabber could produce the Dozer or Grabber pattern if it didn't enclose itself in a small part of the environment.) Provided that the total group (and the size of the subgroups) was big enough for the effects of entrapment to have little effect on group composition, the resulting pattern was independent of group size. Therefore all subsequent experiments were run with a total agent count of 100 agents.

- **Toroidal vs. enclosed environment** All experiments were run in a toroidal world as well as an enclosed world. The density if boxes was the same for both environments.

- **Grabbers laying pheromones** While Dozers always deposit pheromones, Grabbers were laying pheromones only in half of the experiments. The main resulting difference was in the activity of Grabbers. When Grabbers deposit pheromones, the sum of pheromones in the environment stays constant since all agents continuously deposit a fixed amount which counters the effects of evaporation, leading to a stable pheromone level overall. In this case the average activity of a Grabber was independent of the group ratio. In the case that only Dozers deposit the pheromone, the average number of boxes moved by a Grabber decreases with increasing Grabber numbers, reaching zero with 100% group due to the lack of pheromone to trigger grabbing behaviour.

Dependent on these parameters, the results can be partitioned into three classes:

1. Grabbers not depositing pheromones
2. Grabbers deposit pheromones in a toroidal world
3. Grabbers deposit pheromones in an enclosed world

All differences in patterns that emerged over the set of experiments fall into one of these classes. As described above, a change in the total number of agents had the same effect in all experiments and all results shown are from experiments with 100 agents. All results are discussed by comparing pattern measurements from 0% to 100% Grabber rate, which were averaged over 10 runs each.

Grabbers not depositing pheromones Starting with a Dozer-pattern for 0% Grabbers, the results quickly change to a more Grabber-like pattern when the percentage of Grabbers increases (chambers < 10 and complexity value > -0.07). Compared to the pure Grabber-pattern described earlier, the pattern is noise free, i.e. contains no single standing boxes and still has a higher number of chambers. With further growth of the subgroup, their activity and therefore their influence on the resulting structure is decreasing, because less pheromones are dropped by the decreasing proportion of Dozers. In the extreme case of 100% Grabbers no box is moved and the initially random world is not changed. Starting from 40% the result slowly converges to that of a Dozer-only group. As can be seen in Figure 3.4 and 3.5 the presence or absence of walls has no impact.

Grabbers deposit pheromones in a toroidal world When Grabbers also drop pheromones they stay active up to 100% Grabber rate. When slowly increasing the percentage of Grabbers, the pattern changes quickly from the Dozer pattern to the standard mixed pattern (see Figure 3.7). Starting from 40%, the structure gets noisier and converges to the pattern of a Grabber-only group. The increase in free-standing boxes is due to the decreasing activity of Dozers that now pose a minority. With noise, the chamber counting algorithm becomes unreliable which results in the increasing chamber count in Figure 3.6.

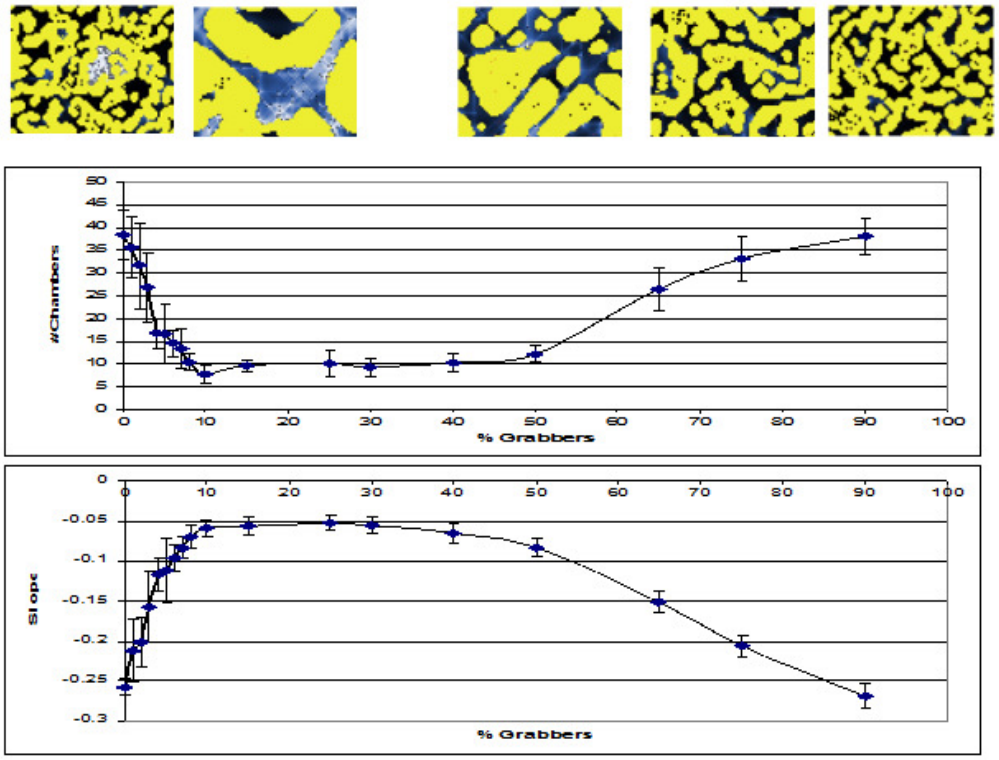


Figure 3.4: *Pattern examples (top, for 0, 10, 50, 75, and 90 % Grabbers), chamber count (middle) and complexity measure (bottom) for experiments with a heterogeneous group in a toroidal world and Grabbers not laying down pheromones. Error bars indicate the 97.5% confidence interval. 100% Grabbers is not plotted because they stay inactive and the result is the initial random box distribution.*

Other than the graphs may suggest, viewed sample patterns show no significant change in the basic structure. Changes in the graphs between 40% and 100% therefore have to be traced back to the increase in the number of free standing boxes.

Grabbers deposit pheromones in an enclosed world Other than in the case of Grabbers not dropping pheromones, the presence of walls in this setting has a big impact on the results of a group with more than 50% Grabbers. With increasing percentage of Grabbers, the chance to get one big pile in the middle of the world (as in Figure 3.1e) also increases. The explanation is that Grabbers are likely to move boxes away from walls because the pheromone level near walls is usually higher than average, which is due to agent's propensity of wall following (see subsection 3.2.4). Bumping against a wall and the consequent 45 degree turns make it likely that agents move along the wall until encountering another obstacle. The pile counts (Figure 3.8c) show, that the likelihood to build one pile

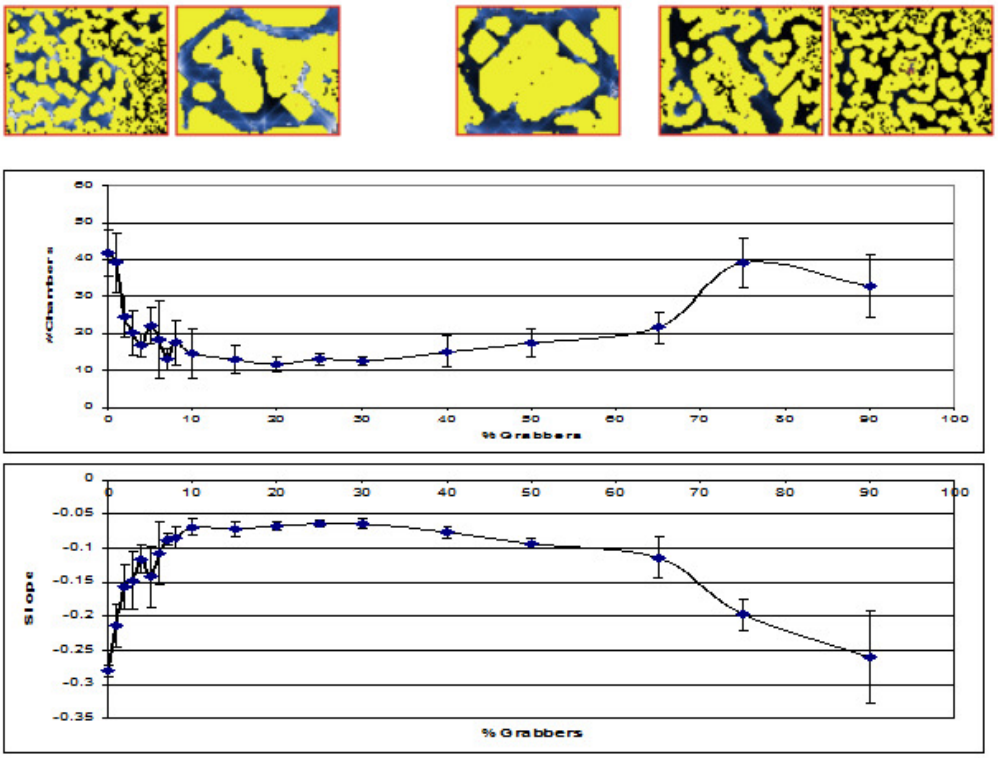


Figure 3.5: *Pattern examples (top, for 0, 10, 50, 75, and 90 % Grabbers), chamber count (middle) and complexity measure (bottom) for experiments with a heterogeneous group in a closed world and Grabbers not dropping pheromones. Error bars indicate the 97.5% confidence interval. 100% Grabbers is not plotted because they are inactive and the result is the initial random box distribution.*

increases when the percentage of Grabbers rises above 50% , but decreases again when it gets close to 100%. In the case of 90%, only single boxes can be temporally found at walls. The increasing chamber count (Figure 3.8a) and the dropping complexity values (Figure 3.8b) for more than 50% can be again partly explained by the increase in noise. However, the amount of free-standing boxes is less compared to the patterns of groups in the toroidal world which is due to the increased probability that areas along the wall get visited by Dozers travelling along them, thereby removing boxes faster, even when their number decreases.

3.2.6 Discussion

For this experimental setting, it became clear that it is difficult to derive the outcome for a heterogeneous group from the patterns produced by homogeneous groups of each agent type. Even after knowing the results for some groups compositions, extrapolating outcomes

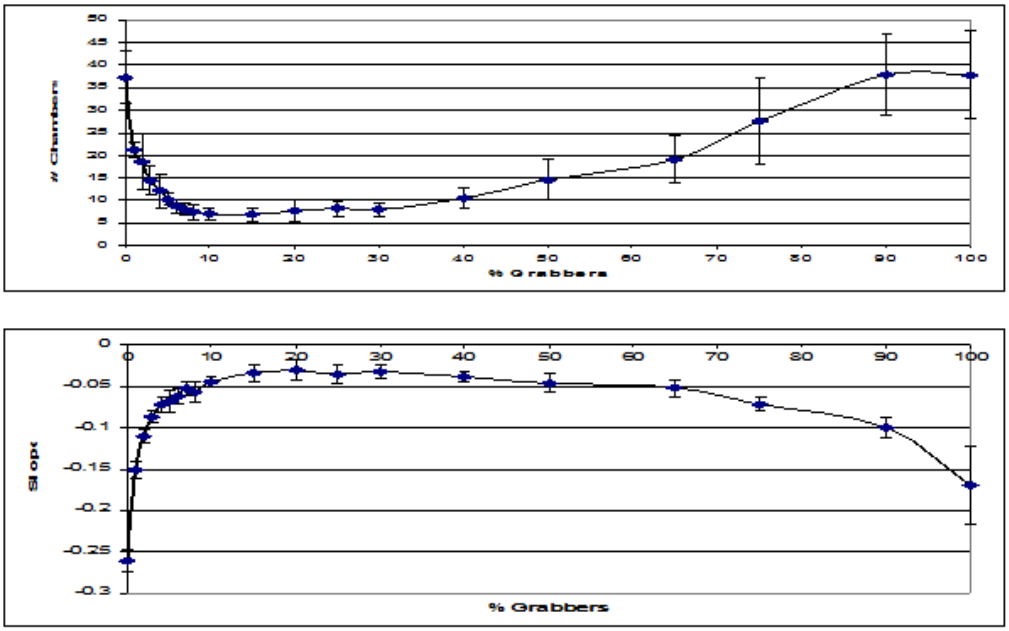
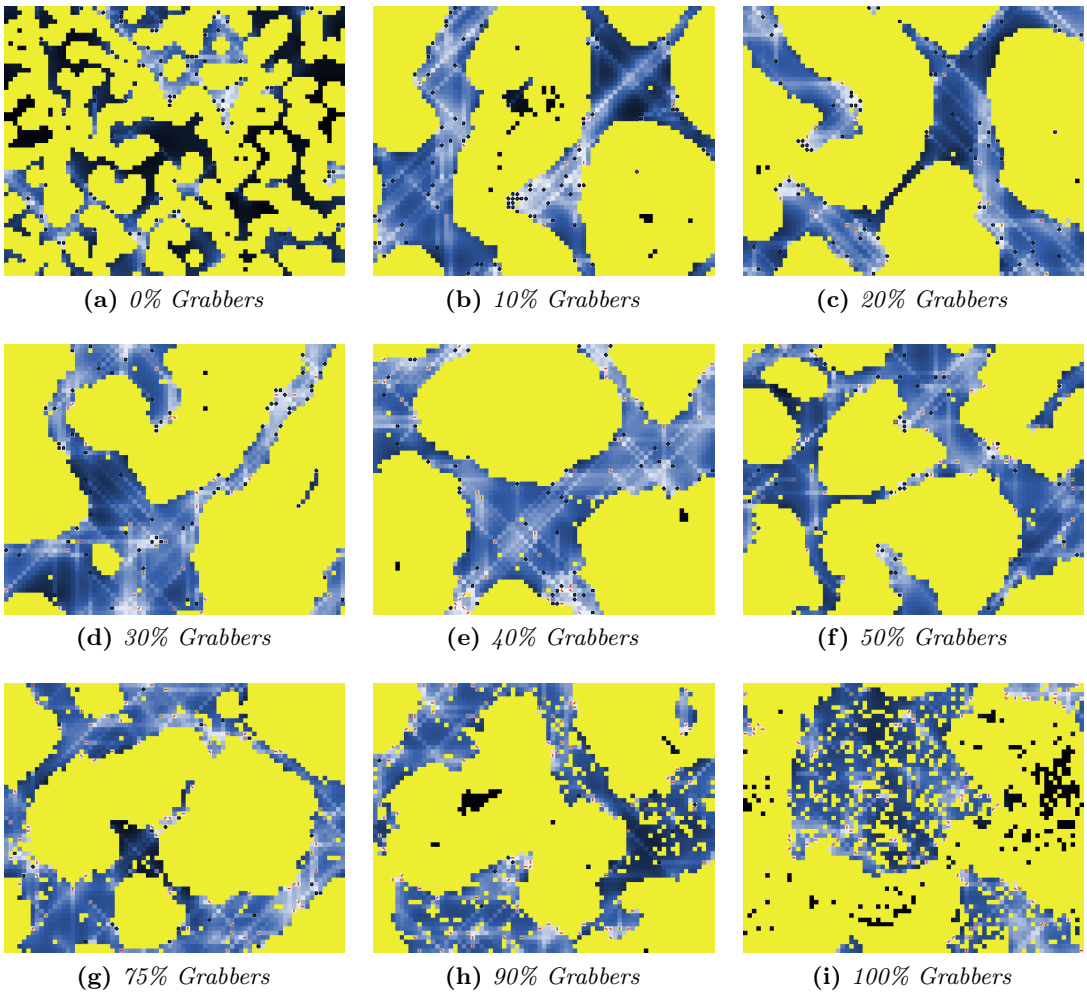


Figure 3.6: *Chamber count (top) and complexity measure (bottom) for experiments with a heterogeneous group in a toroidal world and Grabbers dropping pheromones. Error bars indicate the 97.5% confidence interval.*

for other groups proved difficult. Also, due to the amount and variety of local interactions even in this very simple setting, knowing the individual behaviour of agents does help little to predict behaviour at group level and aid pattern prediction.

Mixing Grabbers and Dozers and comparing the patterns to the ones emerging in the homogeneous experiments, it turned out that the basic structural pattern was more similar to that of a homogeneous Grabber group. Even a low percentage of Grabbers had a big impact on the resulting pattern and although the typical influences of Dozers could still be seen (little noise in the pattern), the overall structure did not resemble a typical Dozer pattern or a blended pattern with influences according to sub group size. Comparing homogeneously created patterns, one might say that the behaviour of Grabbers almost completely superseded Dozer behaviour in a mixed group. This result should not be over-interpreted since it is known from other experiments (e.g. by Maris and te Boekhorst (1996)) that agents moving through an environment with movable objects might push them together over time. This seems to be the case when agents can push an object without sensing it (or being unaffected by collisions) but are affected by a conglomeration of objects. Since Dozers cannot free objects from an already created pile, they can not form bigger piles although this should be possible with noise that makes boxes available again to be pushed elsewhere. The actions of Grabbers can be seen as this random noise for the Dozers, which, in a real world setting, could come from random collisions and subsequent destruction



(a) *0% Grabbers* **(b)** *10% Grabbers* **(c)** *20% Grabbers*

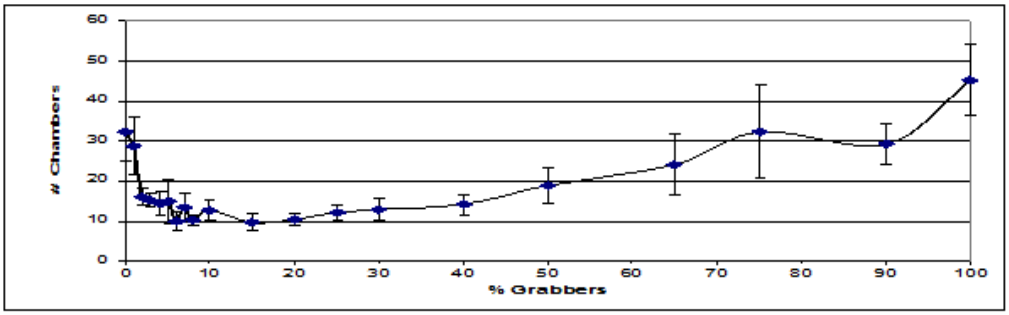
(d) *30% Grabbers* **(e)** *40% Grabbers* **(f)** *50% Grabbers*

(g) *75% Grabbers* **(h)** *90% Grabbers* **(i)** *100% Grabbers*

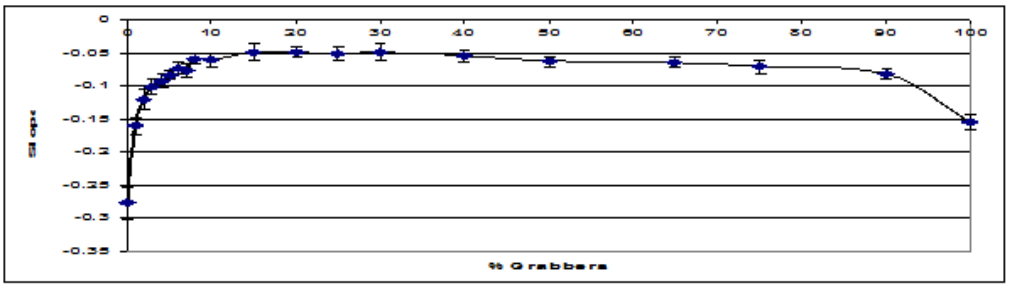
Figure 3.7: *Patterns produced by mixed group with the given percentage of Grabbers at the end of the experiment (t=300000). Grabbers were depositing pheromones (displayed as gradient black-blue-white) and patches with box are coloured yellow.*

of small piles leading to a comparable setting as in (Maris and te Boekhorst 1996). As soon as Grabbers are present which provide the ability to keep boxes pushable, the system naturally converges to a pattern with a low number of piles. With increasing percentages of Grabbers, the number of boxes that are removed from walls exceeds the number of boxes that are being pushed to walls, which results in an increasing number of free boxes in the environment.

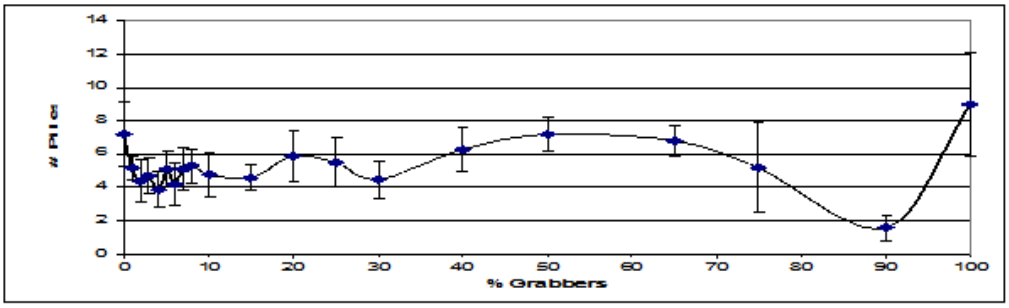
Looking at pile formation and box movements over time, it is noticeable that piles act as attractors for other boxes. Simply put, the bigger a pile, the higher the probability



(a) *Chamber Count*



(b) *Complexity Measure*



(c) *Pile Count*

Figure 3.8: *Chamber count, complexity measure and pile count for experiments with a heterogeneous group in an enclosed world and Grabbers dropping pheromones. Error bars indicate the 97.5% confidence interval.*

that agents push a box into it. A pile also "loses" boxes on a regular basis but since the presence of agents and the resulting avoidance movements are lower at a convex surface compared to a concave chamber wall, this loss is lower at piles than at straight or concave walls. This means that boxes are removed faster from concave surfaces than from convex surfaces, leading to pile growth over time. As can be seen in the toroidal experiments, a group containing Grabbers tends to push/carry all boxes into one big pile if given enough

time.

In enclosed environments the walls also act as fixed attractors which seem to be strong enough to counter the formation of a central pile (and cannot be removed like the smaller attractors in the toroidal case). In a group with a majority of Dozers, this attraction is strong enough to prevent a large pile to form in the middle of the environment. A hypothesis was that the system would always converge towards the central pile pattern given enough time and that the time of 300,000 time steps was too short to reach it. A quick analysis of the patterns emerging over time does not support this. The final patterns shown were often present after one third or half the time and then kept their basic properties. With numbers of Grabbers increasing from 50% to 90%, also the probability that a central pile was formed increased and was one for 90%. Above 90% the probability dropped again to a value of 0.5 for a homogeneous Grabber group.

We presumed at the time that the reason for this behaviour was the higher concentration of pheromone at walls. Agents are likely to follow walls due to 45 degree avoidance movements when bumping into a wall, therefore spending more time near walls, which results in a higher concentration of pheromones and finally a higher activity of grabbers in this area. Hence, the amount of boxes carried in the area along the border is above average, facilitating the process of moving boxes away from the wall.

We were able to prove this presumption in later experiments (see section 3.3) by measuring the average amount of pheromone in the fringe of the enclosed world compared to the overall concentration. Figure 3.10b shows that, for a group with 50% Grabbers, the average pheromone concentration in the fringe was not significantly different from the overall concentration, whereas the concentration in the 90% case was about 2.5 times higher. This is also reflected in the increased grabbing behaviour within the fringe region (see Figure 3.11b).

In all analysed cases the central pile was formed early in the experiment and stayed until the end; in not a single analysed run was it destroyed once being formed. In turn, if a group with a composition capable of building a central pile started out with larger piles along walls, this pattern was also stable. Large piles along walls, when arranged in a certain way, seem to prevent a pheromone build-up and the subsequent high grabbing activity, thus leading to a stable pattern. Although the agents were implemented in a way to exhibit competing behaviours, the central pile can be seen as cooperative effort since both behaviours are needed to build and maintain it. The freeing of boxes through Grabbers and the pushing of that box later in time by a Dozer can be seen as cooperative behaviour, depending of the viewpoint of the observer and the implied aim of the group.

The main question that was investigated in these experiments was the stability of patterns over different group compositions. The emerging pattern was always very stable between 10% and 40% in both, enclosed and toroidal settings. Towards the lower end ($<$ 10%) the pattern changed quickly with small changes in group composition. Entrapment of agents had a big impact in these runs, leading to high uncertainty for pattern prediction. Above 40% the pattern, although visually keeping its basic properties became more and

more noisy (many free-standing boxes) and changed gradually towards the Grabber pattern for the given environment.

3.3 Groups of Versatile Agents

Now that the behaviour and the resulting patterns for heterogeneous groups are known, the question becomes where a group of versatile agents, that can act as Dozer and Grabber, has to be placed. Combining both agents into one leads to an agent that can push boxes when no pheromone is around, but can also carry boxes from areas with high pheromone concentrations. Since this new type of agent can dynamically decide whether to push or grab boxes, the question is whether the emerging behaviour can be compared to the behaviour of any heterogeneous group with a given pre-set composition from the last experiments.

3.3.1 Grabdozers

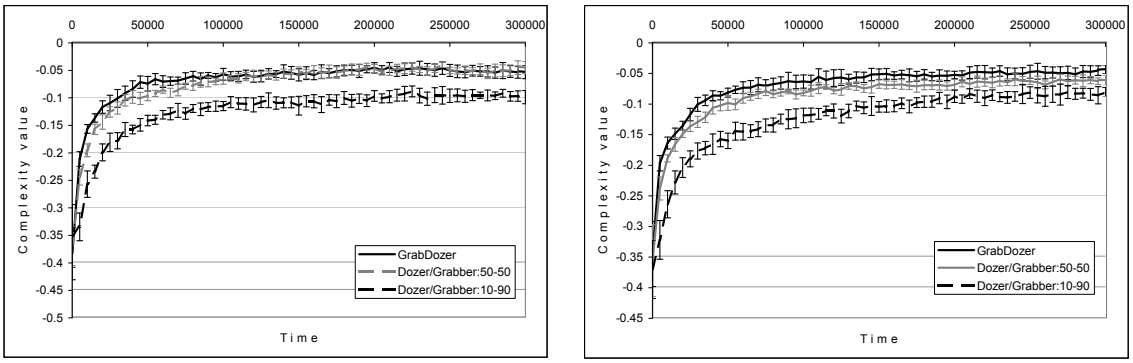
A GrabDozer combines both abilities, grabbing and pushing boxes, in one agent. When encountering a box, a GrabDozer can switch between "pushing" and "carrying" behavior, depending on the pheromone level at its location. The decision making process follows the same rules as used in a Grabber when encountering a box. If the pheromone level at that location exceeds a threshold, the GrabDozer grabs the box; if not, it behaves as a Dozer and pushes it one step forward. If no box is ahead, a GrabDozer will move straight on.

Although the implementations of Dozers and Grabbers are taken directly to implement a GrabDozer, there are some important differences. The decision on which behaviour to exhibit is made every time step leading to a blend of Dozer and Grabber behaviour. A GrabDozer can push a box from an area with low pheromone concentration to an area with high concentration where it subsequently grabs the box and carries it away again. A Dozer would have pushed it until it hit an obstacle where another Grabber could pick it up leading to a time gap between the two actions. Even though a pushing action can change to a grabbing action any time, a carrying phase can only be ended by dropping the box, i.e. a "Grabber phase", once entered, always has to be fully completed, while a "Dozer phase" can be interrupted any time following environmental cues at the current location.

3.3.2 Results of Homogeneous GrabDozer Groups

The crucial difference between the current simulation and those of the previous experiments is that now the agents adapt their behaviour — and hence the composition of their population in terms of Dozer and Grabber activities — in accordance to local changes in the structure of the environment. In the earlier experiments no such "dynamic task differentiation" was possible because the number of Dozers and Grabbers was fixed for a particular run.

Because the action of a GrabDozer depends on local pheromone concentrations when



(a) *Toroidal environment* **(b)** *Enclosed environment*

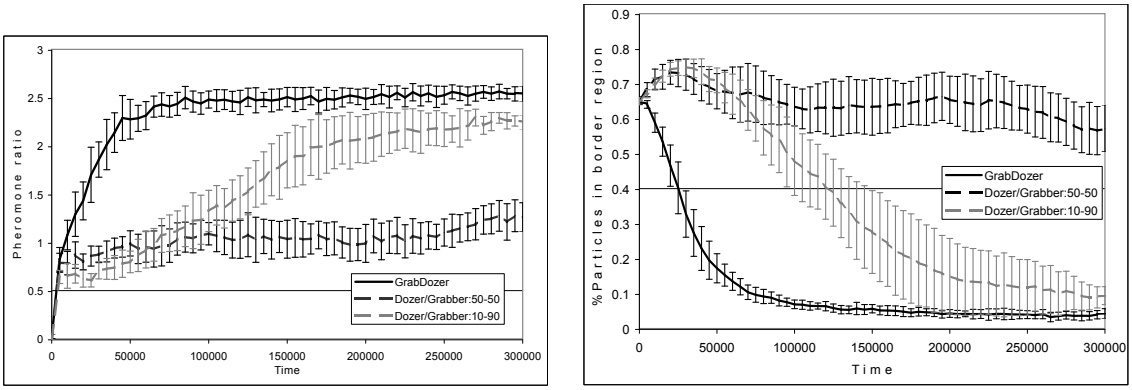
Figure 3.9: *Complexity measurement over time for a GrabDozer group in both environments. Error bars are 97.5% confidence intervals.*

encountering a box, the ratio of grabbing agents to pushing agents varies in accordance to current environmental conditions. To compare the effects of a homogeneous population of flexible, versatile agents (GrabDozers) with those of a heterogeneous population consisting of fixed numbers of inflexible specialists (Grabbers and Dozers), experiments were again run in both an enclosed and a toroidal environment. The same measurements were used as in the previous experiments

As can be seen in Figure 3.9a the homogeneous group of GrabDozers produces patterns in a toroidal world that have comparable complexity values to those brought about by a heterogeneous group consisting of 50% Grabbers. A visual examination of the produced patterns shows that both types of agents produce comparable structures (i.e. the number, size and distribution of chambers and corridors are approximately the same). The group with a Grabber ratio of 90% produces visually the same structures, which is not directly reflected in the complexity values. This again is due to a higher amount of free standing boxes in the environment compared to the other groups, therefore leading to lower values. Also the behaviour over time of GrabDozers matches those of a mixed population of Grabbers and Dozers relating to the measured complexity.

The complexity value of the patterns created by GrabDozers in an enclosed environment is comparable to that in a toroidal world (Figure 3.9b). The values for mixed populations of Grabbers and Dozers in an enclosed environment, however, are significantly lower. Looking at sample pictures reveals that GrabDozers build a single large conglomeration in the centre and a corridor along the wall (example can be seen in Figure 3.11a). The same result was obtained by a heterogeneous group with 90% Grabbers in the earlier experiments. Because GrabDozers in a toroidal world, like a 50% heterogeneous group, do not assemble such central structures, it is of interest to study the effect of borders in more detail.

Our previous findings indicated that higher pheromone concentrations near walls (due to emergent wall following) compared to open areas might be responsible for the creation of



(a) *Ratio of pheromone concentration in the fringe (5 patches wide) of the environment. Y-Axis: average pheromone concentration in fringe divided by average pheromone concentration in environment. Error bars are 97.5% confidence intervals.*

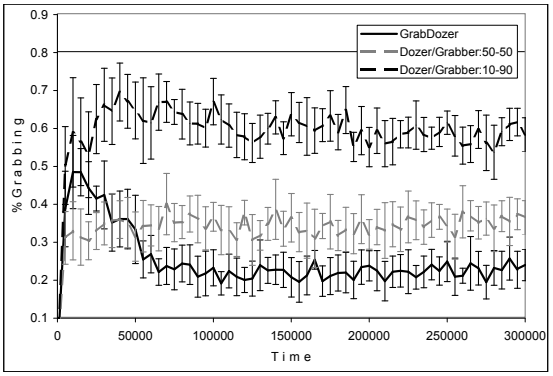
(b) *Percentage of boxes in the fringe (5 quadrants wide) of the environment. X-axis shows time steps of experiment. Y-Axis shows #boxes/#patches. Error bars are 97.5% confidence intervals.*

Figure 3.10: *Measurements of pheromone concentration and concentration of boxes in the fringe area over time.*

a central clump of boxes. To further investigate this, the average amount of pheromone in the fringe region (up to 5 patches from wall) compared to the overall concentration in the environment was measured for all three groups (see Figure 3.10a). It can be clearly seen that in the case of GrabDozers the amount of pheromone in the fringe almost immediately builds up and levels off at 2.5 times the overall average. This coincides with the emergence of a central pile which also stabilises around the same time. In experiments with mixed populations in an enclosed world, the average pheromone concentration in the fringe at first does not differ from the global average. After about 100000 time steps (1/3rd of total time) it rises slowly in the 90% Grabber populations but stays almost constant at a low value in the 50% Grabber runs.

The percentage of patches covered by boxes in the fringe area, strongly correlates to these pheromone measurements (Figure 3.10b). With rising pheromone level, more boxes are moved out of the fringe and into the centre of the environment. This increase of free space along the wall in combination with the wall-following behaviour of the agents and the consequential increase in pheromone concentration creates a positive reinforcement loop. The higher the average pheromone concentration, the more boxes are grabbed and moved, which creates more space along the walls, thus allowing more agents to travel there. This in turn implies that they collide more often with boxes along the wall, which consequently will be removed faster. In mixed populations Dozers push boxes to the wall but do not react to pheromones. They thus counteract the reinforcement outlined above by moving boxes back into newly created free spaces around the boundaries. In the 50% Grabber population, this outwards movement of boxes is so strong that no reinforcing process starts.

(a) *Sample result after 300000 time steps for a GrabDozer group. Grey lines depict walls, white areas boxes and black areas free space.*



(b) *Ratio of agents with grabber behaviour in the fringe (5 quadrants wide) of the environment. X-axis shows time steps of experiment. Y-Axis shows number of agents that are currently carrying a box divided by total number of agents in the fringe. Error bars are 97.5% confidence intervals*

Figure 3.11: *Example of central pile in enclosed environment and number of agents currently exhibiting grabber behaviour in the fringe.*

In the 90% group it slows down the process considerably, but in the end the majority of Grabbers is responsible for a sufficient relocation of boxes away from the borders.

The pushing behaviour of GrabDozers is different to that of Dozers. A GrabDozer that currently pushes a boxes towards a wall, is likely to grab it and carry it away when the local concentration of pheromone position is high enough. This means that a GrabDozer is likely to change its behaviour the closer it gets to a wall, therefore further increasing grabbing activity at the boundary.

In preliminary experiments with GrabDozers it was found that the average number of boxes grabbed in the fringe region was equal or less than performed by heterogeneous populations with more than 40% Grabbers. This contradicts the explanation given above, since the latter would suggest a higher grabbing activity on average along the borders because of higher pheromone concentrations.

To investigate this discrepancy, the grabbing activity (number of agents currently grabbing a boxes in the fringe) over time was recorded (Figure 3.11b). Because the composition is fixed for the heterogeneous populations, the grabbing activity levels off to a value depending on number of Grabbers and the average pheromone concentration. For the GrabDozers, the grabbing activity at first is significantly higher than for a 50% mixed group, but drops soon after to a significantly lower level.

This can be explained as follows: In a GrabDozer group most agents that encounter a particle at or near the rim of the (initially small) conglomeration, will push it because the pheromone concentration tends to be low at that location. The high rate of pushing

behaviour and therefore low rate of grabbing around the rim, keeps the conglomeration together. As a consequence, boxes are transported from the border of the world to the growing pile in the centre; the number of free boxes decrease at the boundaries of the environment and this results in a decreased grabbing activity simply because over time there are fewer boxes to grab. In other words, an initial rise in the grabbing activity of GrabDozers is sufficient to quickly remove boxes from the wall and to assemble a central pile; as soon as this pattern is stabilised it will no longer be destroyed, because grabber activity has declined.

3.3.3 Discussion of GrabDozer Experiments

In these experiments, patterns created by heterogeneous populations of two types of specialised agents (Grabbers and Dozers) were compared with those brought about by populations of versatile, generalised agents (GrabDozers) that dynamically decide on activities in response to changing environmental conditions. GrabDozers build similar structures in a toroidal world as a heterogeneous population composed of equal numbers of Grabbers and Dozers. However, in an enclosed environment the results differ strongly with respect to the resulting structure as well as the temporal dynamics of the building process. In the closed world experiment, no corresponding heterogeneous population composition was found that produces the same output and, at the same time, has comparable temporal dynamics as the GrabDozer population. A mixed population can produce the one-pile pattern, but only towards the end of the experiment.

This leads to the conclusion that in the setting used, a heterogeneous group of agents cannot be converted into a homogeneous population by combining the functions of the different types of specialised agents into a generalized, flexible agent. Vice versa, the dynamics of the assembly process studied cannot be preserved by changing homogeneous agents into a mixed population, splitting the abilities between different types of agents. The main difference between the specialised and versatile agent is the time lag between the end of the pushing to the start of the grabbing. For a versatile agent this is almost zero since the agent is already there and has the box positioned in front of it, ready for grabbing. Within a specialised group, a Grabber has to (randomly, since there is no communication) find the box after it was left by a Dozer, resulting in an often significant lag between pushing and grabbing. Within this lag, the environmental conditions can have changed (pheromone evaporated) or another Dozer can have pushed the box away. Another difference is that GrabDozers have a higher probability (the probabilities of both specialised agent types combined) to perform an action when encountering a box, therefore increasing the chance of altering the environment. These differences are believed to be the reasons for the different temporal dynamics exhibited by the homogeneous groups. The conclusion is therefore, that when the combination of behaviours allows for a different temporal application of the abilities that is not possible (or unlikely) when the behaviours are distributed over specialised agents, a homogeneous group of versatile agents can not be substituted by a

heterogeneous group of specialised agents without changing the overall behaviour.

To create a GrabDozer, the implementation of the two specialists was combined directly and the decision process of a Grabber used to switch between the two (due to the lack of a decision branch within the Dozer implementation). Other combinations are possible, e.g. switching only after failing to push the box any further in Dozer mode and subsequently turning away from the box as compared to being able to switch when traversing an area with high pheromone concentration while pushing. Analysis of the results has shown that the method used gives priority to grabbing behaviour which might have changed the overall behaviour more than other implementations. Nevertheless, it is believed that the differences reported above would hold also with many other implementations unless the switching is geared specifically towards achieving comparable behaviour at group level (i.e. compensating for the advantages of dynamic task switching explained above).

3.4 Conclusion

In this chapter results for heterogeneous groups, consisting of two types of specialised agents, are compared to the outcomes of a homogeneous group of versatile agents that combine the abilities of the two specialised agents. It is shown that even in this simple setting the patterns emerging in different environmental settings cannot be easily predicted by combining the patterns created by homogeneous groups of each agent type. The groups used in this system do not execute a specific task as such. An observer may define a task by observing the system and categorising the outcome as the task the group was trying to execute (e.g. building a large pile in the middle of the environment), but the agents were implemented without such task in mind. The behaviour of the overall group emerges from the individual behaviours of agents and the observed pattern are the result of the interactions of these agents with each other and the environment over time. It was refrained from ascribing tasks to the observed results since this post-hoc categorisation would have been arbitrary and thus only the resulting patterns were analysed to achieve a categorisation of the emerging group behaviour.

Summarising the results it can be said that,

- changes in group composition do not directly lead to changes in the behaviour at group level. As reported in section 3.2 stable behaviour was always found over a range of group compositions. Significant changes in behaviour were expected when approaching group compositions near the extremes, towards homogeneity, but this held true only for the lower bound (0% Grabbers) and the upper bound in the enclosed environment. In the case of a toroidal environment, the pattern converged smoothly towards the Grabber pattern when approaching a homogeneous Grabber setting. The range which produced a stable behaviour was 40% of all compositions which means that the group showed reliable behaviour within this range in terms of robustness to change in agent numbers.

- the ability of individual agents to switch between behaviours can lead to results that a group of two types of specialists, although being able to exhibit both individual behaviours within the group, cannot produce. A group of GrabDozers can be seen as a heterogeneous group with changing group compositions over time. At any given time step, every agent in the group can be seen as either Dozer or Grabber depending on the behaviour exhibited, but compared to the pre-set groups the ratio can change continuously and adapt to the current state of the environment.
- The fact that no group composition of specialised agents could be found that leads to the same results as a group of versatile agents with combined behaviours, leads to the rejection of hypothesis 1 (see section 1.1) for the chosen setting. Nevertheless, it may be possible that a homogeneous group of versatile agents can be substituted by a heterogeneous group of specialists in a different set-up, e.g. when the tasks are completely partitioned or when the separation of behaviours in the versatile agent is more pronounced.

The ability to adapt to an environment is important when the group is placed in a dynamic environment and the efficiency of the group is of interest. In the experiments the environment is constantly changed by the agents leading to a highly dynamic environment where specialists can only execute their task when specific conditions are met. The degree of utilisation within a group of versatile agents is generally higher or equal compared to a pre-set heterogeneous group of specialists due to a higher probability to come across the conditions needed to execute one of their tasks.

This leads to the conclusion that, if task and environmental constraints allow for it, a homogeneous group of versatile agents has a higher aptitude to utilise its agents which can subsequently lead to higher efficiency and adaptability.

Chapter 4

Task Differentiation through Success Feedback

4.1 Introduction

As Labella (2007, chap. 4) shows, the efficiency of a multi-robot system can be increased by using "an optimal number of robots". In a continuously changing environment, the "optimal" number of robots also changes over time and has either to be fixed to an estimated number beforehand by the designer or by giving the robots the ability to adjust their activity through an adaptation process. Since agents in many collective settings have only limited knowledge about their environment and even the combined knowledge of a group is generally not enough to explicitly determine the optimal number, the latter can only be estimated. This estimation of task demand is not trivial and the accuracy of the estimation can strongly affect the efficiency of the group.

Deneubourg et al. (1987) suggested a simple model to explain foraging behaviour in ants without recruitment. They assume a positive feedback loop exists which regulates an ant's activity depending on food availability and leads to "the division of initially identical potential foragers into highly active and largely inactive ones". Furthermore, the same feedback loop affects the selection of possible foraging sites.

Labella et al. (2004) implemented this model in simulations and real robots and confirmed that it leads to successful division of labour between two states "active" and "idle". This adaptation is dependent on the availability of objects and the size of the group and is shown to be robust to changes in the environment. Due to this adaptation, the group is more efficient (more objects collected per time spent active by the whole group) compared to a non-adaptive group. Labella also showed that small hardware differences in real robots get amplified and robots which have a higher performance are more likely to become active. This leads to what he calls "selection of best individuals". The advantage of Labella's model is that it is minimal and requires no communications between agents, no information

centrally shared through the base (as e.g. (Krieger and Billeter 2000)), and no advanced sensing or information processing abilities on the agent. It is minimal in its need of resources and its only requirement is the agent's ability to recognise success and failure, which was the reason to choose it for further investigation.

In this chapter a similar set-up is described with which Labella's results were partly reproduced and features of the system analyzed in more detail. The following section describes a general retrieval task and its subtasks before introducing the two instantiations that were used in this work. In 4.2.6 the control algorithm used by Labella is described and the effects of parameters on the behaviour of the group shown in 4.3. The system is first explored using homogeneous groups only in 4.4. It is shown that, depending on the parameter set used for the agents and the specific task, the range of object densities in which differentiation can occur varies. If a group is used in environments outside this range, the activity of the group is either at minimum or maximum and no differentiation can be expected. In 4.4.3 it is shown that replacing the probabilistic mechanism, which determines when agents leave the base, by a direct calculation of idle time has no effect on the system. Using this direct mapping of internal variable to idle time has no effects on differentiation or the speed of the adaptation. Using this new mechanism, the time an agent will remain in the base is known when it enters the base and could be used in further decision processes (e.g. engaging in different activities in the base, e.g. recharging). The last experiment with homogeneous groups (4.4.4) investigated the efficiency gains of an adaptive group and it is shown that there is no gain but a loss compared to a fully active group in static environments. Despite those losses, it is shown that in dynamic environments, there are efficiency gains in time intervals that include both a low and high density interval.

The second part of this chapter (4.5) shows results from heterogeneous groups and the effects of heterogeneity on the adaptation process. The increased activity of agents with higher performance is confirmed and different parameters that induce heterogeneity are explored. In 4.5.2 it is shown that this increase in activity, or more precisely the lower activity of less agents with lower performance through the VD algorithm leads to efficiency gains. The last experiment then investigates the efficiency gains in a dynamic environment and the results are presented before the chapter is concluded.

4.2 Experimental set-up

4.2.1 Object Retrieval as a Generic Model

Since part of my research extends work done by Labella (2007) the tasks executed by agents in this thesis were based on the same object retrieval task. Retrieval tasks are common in robot experiments (Matarić 1992, Arkin 1992, Krieger and Billeter 2000) and are inspired by animal foraging behaviour. However, object retrieval tasks used in simulations or robot experiments are always an abstraction and lack the properties that are normally implied in the biological counterpart (e.g. objects symbolise food items that provide energy, etc).

Although the term *foraging* is widely used in the literature for both biological behaviour and its metaphor, object retrieval is preferred in this work to encompass a wider group of tasks and it is necessary to explain which features of the task are included in the simulation and which are not.

In general, object retrieval implies that one or more agents have to collect objects in an environment and transport them to a dedicated area. This collective activity, which is modelled on prey retrieval in insect societies, can be used to gain insights on biological foraging as well as develop solutions for many problems in collective robotics, like

- search and rescue, where the agents have to find and retrieve victims
- waste clean up, from simple household dust to toxic waste
- distributed work, where robots have to find work areas and return to a base after completion
- harvesting

All problems in this list share common properties and are similar on an abstract level. We can formulate a general description of object retrieval by splitting it into four serially executed activities:

Search

Searching the environment either individually or cooperatively to find locations within an area that fulfil a condition or set of conditions (e.g. contains object of type X). In this work, search time is characterised by the time it takes an agent to find an object T_s and the time-out TO_s after which the agents aborts the search. T_s is dependent on the agent's abilities (e.g. sensor capabilities and speed but also information transfer between agents) and environmental factors like object density and current distribution. It also includes the time an agent spends avoiding obstacles and other agents while searching.

Execute

Once a target was found, the agent executes one or more subtasks. Depending on the chosen scenario this can either be done individually or require a group of homogeneous or heterogeneous agents and can range from grabbing an item to complex team tasks. Subtasks may require different abilities which can lead to different execution times T_e or completely prevent execution ($T_e = TO_e$). T_e is the time an individual agent spends to execute the subtask until it successfully finishes it or a time-out TO_e is reached.

Return

The agent returns to a base because it failed to execute tasks or because of its own requirements (e.g. energy constraints or carrying capacity). The time from entering the return phase to reaching the base is denoted T_r. This phase can be similar in length to the search phase if the agent has no or little knowledge on how to find its base and is dependent on a homing strategy to find it. It also includes time spent on avoiding obstacles and other agents on the way. Often the agents have a way to find their base through specific environmental clues (e.g. a beacon) or through knowledge acquired in the search phase (i.e. memory), and hence generally $T_s \geq T_r$. Returning to base is generally an individual task but can also require a group of robots (e.g. cooperative retrieval (Groß and Dorigo 2004, Groß et al. 2006)).

Idle

The reasons for the agents to be at a base can be manifold. Depending on the setting the robot may unload objects, recharge, or idle to save energy. The time between entering the base and starting to search again is denoted by T_b.

A task performed by cycling through these subtasks can be seen as a generalisation of the list of tasks similar to prey retrieval and captures many of its properties. In this work two different tasks are used which are instantiations of general object retrieval: An object retrieval and an object consumption task. The algorithms for self-organization of group activity used in this work can be used — in principle — in all scenarios that instantiate the general task.

4.2.2 Implementation of Object Retrieval

The set-up for object retrieval is shown in Figure 4.1. A rectangular area was simulated in NetLogo with a base (or "nest") located on the right, covering one third of the total area, and objects placed randomly outside the base. All agents started at random locations and with random orientation within the base area.

Although the NetLogo grid (i.e. "patches") was used for object placement and collision avoidance, it was not used for agent movements, i.e.agents could move and turn continuously. The area was divided into rectangular cells (41x21) with objects occupying a full cell and agents represented by dots which could move and turn continuously (i.e. in increments $\in \mathbb{R}$) but only one agent could occupy one cell at any given time. Agents had a distance sensor (represented as a line straight ahead) with a given maximum sensing length that reported distances to collectible objects ahead. Throughout this thesis, distance measures are given in cell width, e.g. a sensing range of 3.5 means the agent can detect objects up to 3.5 cells ahead. Other agents and walls could not be distinguished and were only detected when the next move would lead to collision (i.e. end on the same cell), in which case the

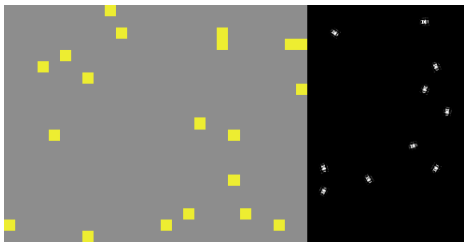


Figure 4.1: *Initial set-up for the object retrieval task. Black area depicts base area with agents starting randomly in the base. Yellow squares depict boxes which are placed randomly outside the base.*

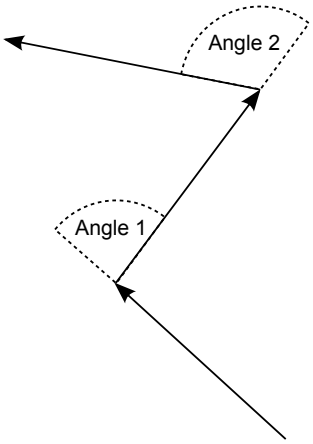


Figure 4.2: *Search movement. Agent moves straight for set time, then turns randomly left or right. The angle is random between 0 and a maximum angle. This is repeated until an object is found or a fixed maximum number of steps is reached. Steps are counted per turn.*

agent turned a random value $\in [-90°, 90°]$.

Agents cycled through the four phases "Search", "Execute", "Return", and "Home" until the end of the experiment. The search phase always started in the base and the agent attempted to find an object within a maximum number of steps following the move pattern as shown in Figure 4.2 with step count incremented by one for each scan turn (avoidance turns were not counted). While turning, the agents used their distance sensor to detect objects and stopped turning when the maximum angle was reached or an object was found (in which case the object could be reached within the next straight move since maximum move distance $\geq$ maximum sensing range). When an object was found, i.e. the agent ended its move on a patch containing an object, the execution phase started in which the agent had to spend a constant time T_e on the object patch after which the object was removed from the environment and the agent started its return to the base. Agents had full knowledge about the direction of the base and moved there in a straight line, i.e. T_r is directly proportional to the distance from the base since agents travelled with constant speed. Upon arrival in the base, the agent entered the "Home" phase and idled for a time proportional to their success in finding objects (see subsection 4.2.6) before starting with a new task cycle. The current number of objects in the environment was kept constant, i.e. a new object was randomly placed when a successful agent entered the "Execute" phase.

If not reported otherwise, agent's parameters were set to the following default values in both the object retrieval and object consumption experiments: Agents travelled straight

for 5 time steps before turning and returned to the base (or stopped at the current location in the OCT setting) after 7 unsuccessful turns. The sensor range was set to 5 patches and after the agent successfully reached an object, it stopped for 100 time steps (= T_e).

This simulated set-up is substantially different to Labella's original experimental set-up. The environment is different in shape, layout and object density and physical properties of the agents as well as movement patterns are not comparable. The aim of this work was to reproduce only the main features in an abstract, simulated environment, which were necessary to qualitatively test the adaptation mechanism used. Agents in this work follow the same subtasks as in the original work, i.e. leaving a base area to search for objects which, if successful within a given time limit, they retrieve. On entering the base success or failure are determined and internal variables changed according to the same algorithm. It will be shown that agents show the same behaviour regarding the adaptation mechanism in the simulated, abstract environment as reported by Labella in the original work.

4.2.3 Object Consumption

A second task was used together with the retrieval for comparison reasons. For this object consumption task also a rectangular area was simulated in Netlogo but in contrast to the retrieval task there was no base area. Agents and objects were initially placed randomly in the available area and agents with random orientation.

"Search" and "Execute" phase were the same as in subsection 4.2.2 with agents following the same search and execution algorithm. In case of $T_s = TO_s$ (i.e. unsuccessful search) the agent didn't return to a base this time but stop on the current patch and immediately entered the "Home" phase. In case the agent was successful, the object was removed and randomly replaced in the environment and after spending T_e on the object's patch the agent immediately entered the "Home" phase and idled on the spot for a time dependent on its success (again see subsection 4.2.6). When finished with the "Home" subtask, agents started their search for objects from the location and orientation they had previously finished the search in.

This task was used in addition to the object retrieval task to investigate differences due to spatial effects of returning and starting from a fixed area.

4.2.4 General System Properties

A few properties of agents, environment and both tasks are important for the analysis and need to be emphasised:

1. **Local knowledge of agents.** Due to their very limited sensor capabilities, the knowledge of the agents is limited to very few aspects of their current, local environment. They have no knowledge about the presence or number of other agents and no means of direct communication. Other agents and walls can only be detected when collision is imminent and objects can only be detected at distance by using the

distance sensor when turning. Objects are found by entering a patch which contains an object.

2. **Single robot tasks.** Although the absence of direct communication would still allow for cooperation between agents (see e.g. Ijspeert et al. (2001)) all tasks in this work are single robot tasks, i.e. a single robot can complete all subtasks. Using a group of robots improves group task performance due to parallel execution of subtasks but is not necessary for success.

3. **Static environment.** The number of objects in the environment was kept constant. Every time an agent found an object, the object was removed and randomly placed in the environment again. There were no dynamic effects (e.g. "growth" of items) and changes in object availability in some experiments were externally induced.

4. **No energy model.** The only function of objects in these experiments was to provide collectible targets for agents and determine their success. Every agent only has access to the number and the type of objects collected by itself and has no access to, nor is it directly influenced by, the number of collected objects by the whole group or the success of other agents. There is no "energy" associated with objects (as e.g. in Liu et al. (2007)) and there is no energy level maintained in an agent or for the group (as e.g. in Krieger and Billeter (2000)).

4.2.5 Success Rate and Efficiency

To compare agents and groups, different measurements were used in this work. The *success rate* s_a of an agent $a \in \mathcal{G}$ in a time interval $\Delta t = [t_1, t_2]$ can be calculated as

$$s_a(\Delta t) \;=\; \frac{O_a(\Delta t)}{C_a(\Delta t)} \tag{4.1}$$

with $O_a(\Delta t)$ being the number of objects retrieved, and $C_a(\Delta t)$ being the number of attempts to retrieve an object by agent a (= number of cycles started by agent a) in the given interval. The success rate can be directly measured by the agent itself and can be used either directly or indirectly in a decision process.

Success rates are an indicator for the performance of an agent and can be used as performance measure (see 2.2.2). With only one task for an agent to execute, the proficiency of an agent therefore equals its success rate. Following the definition from equation 2.6, this in turn means that the success rate also equals the value for specialisation of an agent.

For a group $\mathcal{G}$ of agents, an observer can calculate the arithmetic mean of the individual

success rates to determine the average success rate of all agents in the group:

$$\overline{s}_a(\Delta t) \;=\; \frac{\sum_{a=1}^{|\mathcal{G}|} \frac{O_a(\Delta t)}{C_a(\Delta t)}}{|\mathcal{G}|} \tag{4.2}$$

The arithmetic mean can be calculated for a group when the individual performances are known for the given environment.

The average success rate for a group $\mathcal{G}$ is usually calculated as shown in Equation 4.3 with $O_{\mathcal{G}}(\Delta t)$ being the sum of objects retrieved and $C_{\mathcal{G}}(\Delta t)$ being the sum of attempts to retrieve an object by all agents $a \in \mathcal{G}$ in the given interval. As can be seen, this equates to the weighted average of individual success rates with the number of attempts as weights.

$$s_{\mathcal{G}}(\Delta t) \;=\; \frac{O_{\mathcal{G}}(\Delta t)}{C_{\mathcal{G}}(\Delta t)} \;=\; \frac{\sum_{a=1}^{|\mathcal{G}|} O_a(\Delta t)}{\sum_{a=1}^{|\mathcal{G}|} C_a(\Delta t)} \;=\; \frac{\sum_{a=1}^{|\mathcal{G}|} \frac{C_a(\Delta t) * O_a(\Delta t)}{C_a(\Delta t)}}{\sum_{a=1}^{|\mathcal{G}|} C_a(\Delta t)} \;=\; \frac{\sum_{a=1}^{|\mathcal{G}|} C_a(\Delta t) * s_a(\Delta t)}{\sum_{a=1}^{|\mathcal{G}|} C_a(\Delta t)} \tag{4.3}$$

The success rate of an agent is affected by many parameters of agents (speed and sensing capabilities), the environment (density and distribution of objects, number of agents) and varies over time due to changes in the local environment. For a given parameter set and environment, s_a is independent of group size in this work, because object density in the environment is kept constant. Interference between agents is negligible for group sizes of 10 and the implementation of object avoidance which does not affect the area that is searched. If individual success rates are known for a given environment, this can be used to estimate the success rate $s_{\mathcal{G}}$ of a group.

Results show that $\overline{s}_a(\Delta t) \approx s_{\mathcal{G}}(\Delta t)$ for a homogeneous group with not a single run showing a significant difference. If there is little or no differentiation in the group, differences in $C_a(\Delta t)$ between agents are small and the weighted mean quickly converges to the arithmetic mean. Therefore, the arithmetic and weighted mean can be used interchangeably in homogeneous groups. This does not hold for heterogeneous groups in general because of differences in $C_a(\Delta t)$ in highly differentiated groups.

In order to predict the differentiation in heterogeneous groups, individual success rates gathered in experiments with homogeneous groups composed of the same agents can be used. This is shown in more detail in section 4.5

In experiments with pre-set heterogeneity and stable differentiation the weights used in the weighted mean can be significantly different between agents or subgroups, leading to different values for the means. Highly active agents influence the weighted mean much more than inactive agents leading to differences in the two means if agents have different individual success rates.

Algorithm 3 Adaptation algorithms as proposed by Deneubourg et al. (1987)(left) and Labella (2007)(right). P_i denotes the probability to leave the base and start cycle $i+1$ and it is ensured that $0 < P_{min} \leq P_i \leq P_{max} \leq 1$.

$P_0 = P_{min}$

loop
  if success **then**
    $P_i = min\{P_{max}, P_{i-1} + \Delta^+\}$
  else if failure **then**
    $P_i = max\{P_{min}, P_{i-1} - \Delta^-\}$
  end if
end loop

$P_0 = P_{init},\ fail = 0,\ succ = 0$

loop
  if success **then**
    $succ = succ + 1$
    $fail = 0$
    $P_i = min\{P_{max}, P_{i-1} + succ * \Delta\}$
  else if failure **then**
    $fail = fail + 1$
    $succ = 0$
    $P_i = max\{P_{min}, P_{i-1} - fail * \Delta\}$
  end if
end loop

A group of agents is considered more efficient compared to another group if its agents retrieved more objects while spending less time active (= on duty). The duty time T_D of an agent in a time interval $\Delta t = [t_1, t_2]$ is calculated as

$$T_{D_a}(\Delta t) \ = \ T_s(\Delta t) + T_e(\Delta t) + T_r(\Delta t) \ = \ T_c(\Delta t) - T_b(\Delta t) \tag{4.4}$$

and therefore the efficiency for a group of agents is calculated as

$$E_{\mathcal{G}}(\Delta t) \ = \ \frac{\sum_{a=1}^{|\mathcal{G}|} O_a(\Delta t)}{\sum_{a=1}^{|\mathcal{G}|} T_{D_a}(\Delta t)} \tag{4.5}$$

4.2.6 Success Feedback

Labella (2007, chap. 4) used parts of the model proposed by Deneubourg et al. (1987) (probability to leave the base) to implement a robot control algorithm to test the model in a real robot setting and in simulation. The differences between the original model and Labella's adaptation can be seen in Algorithm 3. In the original model the probability P_i to leave the base was either increased or decreased by a value Δ^- or Δ^+ on returning to the base, depending on failure or success on the last cycle. In his "Variable Delta" (VD) algorithm, Labella used the same Δ for both directions but multiplied by the number of successive failures/successes to allow for faster convergence to the equilibrium point of the system (which was needed due to limited battery life of the robots). P_i was bound by P_{min} and P_{max} with $0 < P_{min} \leq P_i \leq P_{max} \leq 1$. It has to be noted that i denotes the cycle

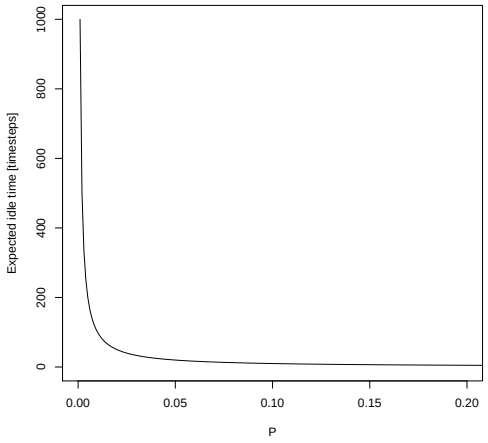


Figure 4.3: *Expected value for the time spent in the resting area depending on the variable P in the "Variable Delta" algorithm. Average of 1000 simulated runs for each value of P.*

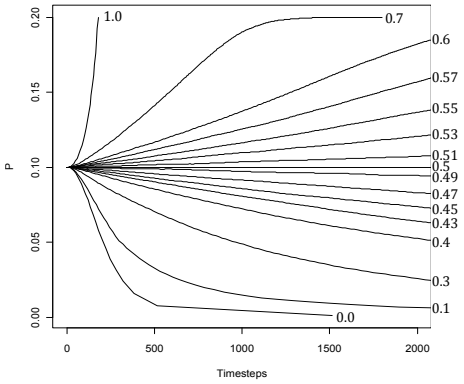


Figure 4.4: *Expected values for P(t) for different average success rates. Lines show average over 1500 random example runs with constant success rate and* $\Delta = 0.002$.

number after which P_i was recalculated, not a time, and $P(t)$ will be used to denote the value of P at time t.

Due to the nature of the feedback loop, the system describing a single agent has two attractors — P_{min} *and* P_{max} — and an unstable equilibrium in between. In the "Variable Delta" algorithm, the value of Δ combined with the success rate of an agent determines the speed of convergence towards these attractors.

Figure 4.3 shows the expected value for the idle time T_b depending on the value of $P(t)$v. P_i is the probability with which a leaving-event takes place during the time interval Δt and as such defines the termination rate λ_b of T_b:

$$\lambda_b = \frac{\text{(Probability to leave base | Agent is in base)}}{\Delta t} \tag{4.6}$$

A well-known result from probability theory states that bouts terminated at random by a constant termination rate are exponentially distributed according to the probability density function $p(t) = \lambda e^{-\lambda t}$ with an expectancy (mean value) of $E(t) = 1/\lambda$ (Feller 1968). The higher T_b, the higher the cycle time T_c and therefore the time between changes in P.

The effects of this can be seen in Figure 4.4 where the expected values of $P(t)$ are plotted for different success rates. Longer cycle times lead to slower average response times, which means that agents with high P can react faster to changes in the environment since they "sample" the environment with higher frequency.

To measure the differentiation of a group of agents, a difference metric $d : A \times A \to \mathbb{R}$ had to be defined (see Equation 2.2). All agents used in the following experiments have the variable P in common, which determines their amount of idle time in the base. The current value of P gives an indication of the agent's success over time and determines its behaviour (in terms of idle times). We therefore define the distance d between two agents a_i and a_j as:

$$d(a_i, a_j) = |P_{a_1} - P_{a_2}| \tag{4.7}$$

with P normalised onto [0,1]. Using this metric which is defined on the 1-dimensional space of P values means that for a group $\mathcal{G}$ of agents, Differentiation($\mathcal{G}$) < 1.0 because there inevitably must be pairs of agents with d < 1.0. The maximum value for Differentiation($\mathcal{G}$) in this work is reached when a group $\mathcal{G}$ splits into two subgroups $\mathcal{G}_1$ and $\mathcal{G}_2$ with $|\mathcal{G}_1| = |\mathcal{G}_2| = |\mathcal{G}|/2$ (or $|\mathcal{G}_1| = |\mathcal{G}_2| = (|\mathcal{G}| - 1)/2$ in case $|\mathcal{G}|$ is odd) and $P_{a_i} = 0, \forall a_i \in \mathcal{G}_1$ and $P_{a_i} = 1, \forall a_i \in \mathcal{G}_2$. From this follows [1] that for a group $\mathcal{G}$ and $a_i, a_j \in \mathcal{G}$:

$$\text{Differentiation}(\mathcal{G}) \leq \begin{cases} \frac{|\mathcal{G}|^2}{4\binom{|\mathcal{G}|}{2}} & \text{if } |\mathcal{G}| \text{ is even,} \\ \frac{(|\mathcal{G}|-1)*(|\mathcal{G}|+1)}{4\binom{|\mathcal{G}|}{2}} & \text{if } |\mathcal{G}| \text{ is odd.} \end{cases} \tag{4.8}$$

For a group of 10 agents as used in this work, the maximum value of Differentiation($\mathcal{G}$) is therefore $0.\bar{5}$. It has to be noted that in case of $|\mathcal{G}|$ being odd, the P value of the remaining agent after an even group split is irrelevant and has no further effect on the value of Differentiation($\mathcal{G}$).

4.3 Effects of System Parameters

An agent is defined by a set of parameters that mainly fall into two categories:

1. Parameters that affect average success rate of agents. The most obvious parameter in this category is the sensor range. Changing the sensor range of an agent has direct effects on its ability to find an object and subsequently changes its success rate. Other

[1] Pairs of agents used in Differentiation($\mathcal{G}$) can be split into 4 categories: (a) pairs within subgroup $\mathcal{G}_1$, (b) pairs within subgroup $\mathcal{G}_2$, (c) pairs between the subgroups for even $|\mathcal{G}|$, and (d) pairs containing last agent in case of odd $|\mathcal{G}|$. Let $|\mathcal{G}|$ be odd and $s = \lfloor |\mathcal{G}|/2 \rfloor$, then:
Differentiation($\mathcal{G}_{odd}$) $= \sum_{i=1}^{s-1} \sum_{j=i+1}^{s} \frac{d(a_i,a_j)}{\binom{|\mathcal{G}|}{2}} + \sum_{i=s+1}^{|\mathcal{G}|-1} \sum_{j=i+1}^{|\mathcal{G}|} \frac{d(a_i,a_j)}{\binom{|\mathcal{G}|}{2}} + \sum_{i=1}^{s} \sum_{j=s+1}^{2s} \frac{d(a_i,a_j)}{\binom{|\mathcal{G}|}{2}} + \sum_{i=1}^{s} \frac{d(a_i,a_{2s+1})}{\binom{|\mathcal{G}|}{2}}$.
With $s = \lfloor |\mathcal{G}|/2 \rfloor = (|\mathcal{G}| - 1)/2$ for odd $|\mathcal{G}|$ and first and second sum evaluating to 0 due to $d(a_i, a_j) = 0$ for agents a_i, a_j from same subgroup:
Differentiation($\mathcal{G}_{odd}$) $= s^2 * \frac{1}{\binom{|\mathcal{G}|}{2}} + s * \frac{1}{\binom{|\mathcal{G}|}{2}} = \frac{((|\mathcal{G}|-1)/2)^2}{\binom{|\mathcal{G}|}{2}} + \frac{((|\mathcal{G}|-1)/2)}{\binom{|\mathcal{G}|}{2}} = \frac{(|\mathcal{G}|-1)^2}{4\binom{|\mathcal{G}|}{2}} + \frac{(2(|\mathcal{G}|-1))}{4\binom{|\mathcal{G}|}{2}} = \frac{(|\mathcal{G}|-1)(|\mathcal{G}|+1)}{4\binom{|\mathcal{G}|}{2}}$.
In case of even $|\mathcal{G}|$, part (d) is dropped which leaves only part (c). With $s = \lfloor |\mathcal{G}|/2 \rfloor = |\mathcal{G}|/2$ for even $|\mathcal{G}|$:
Differentiation($\mathcal{G}_{even}$) $= s^2 * \frac{1}{\binom{|\mathcal{G}|}{2}} = \frac{(|\mathcal{G}|/2)^2}{\binom{|\mathcal{G}|}{2}} = \frac{|\mathcal{G}|^2}{4\binom{|\mathcal{G}|}{2}}$

parameters with the same effect include speed (more area covered per time), average turning angle when scanning for objects and search time-out.

2. Parameters that affect average cycle times of agents. Some parameters have no or only a small effect on the success rate of an agent, but have an impact on the cycle times (i.e. they affect T_s, T_e, T_rorT_b). Parameters that affect T_s often also have an effect on success rate and it becomes difficult to judge which effect has the higher impact on agent behaviour. Changes to T_e, T_r, and T_b usually have no or little impact on the success rate but can affect the feedback loop by changing the time between success evaluations. In this work, T_e is a constant that is set directly by a parameter and T_b is under the control of the VD algorithm and therefore dependent on the parameters P_{min}, P_{max}, and Δ.

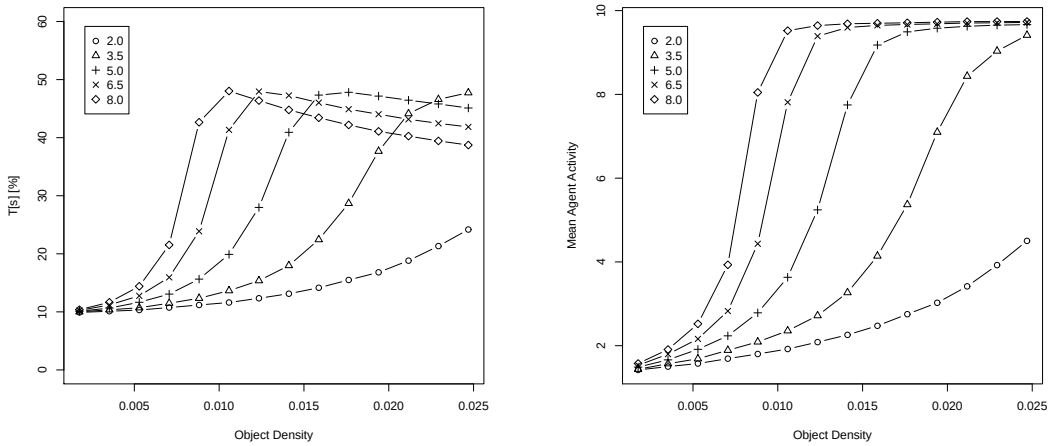
Many of the mentioned parameters would also be parameters in real robots (e.g. VD parameters, time-outs, turning angles) while others are determined by the chosen hardware (e.g. sensor range) and are not easily changed. Since VD parameters would normally be used to adjust the group's response to a specific environment and can easily be changed, other parameters are more dependent on agent's morphology and hardware and are more intuitively linked with heterogeneity.

Two parameters were chosen for investigation — sensor range and execution time — because both have easy to observe effects and are good examples from their respective categories. The sensor range of an agent directly effects success rates since it increases the area that is searched and also affects cycle times since successful cycles have shorter length. Despite the effect on T_s, sensor range was chosen because is only affects cycle times next to success rates. Other parameters, like speed, also have secondary effects (e.g. changing T_r) and effects might not be as easily distinguishable. T_e, which is implemented as a parameter in every agent, was chosen to represent the second category. It only changes cycle times and has no other effect.

To see how strong the effects of these two parameters are on group behaviour, homogeneous groups with different values were put into static environments with different object densities and the effects on average group activity over 50 runs measured. Apart from the parameter in question, default values were used.

4.3.1 Affecting Success Rates

When the density of objects in an environment increases, agents become more active due to increasing success and subsequent increase in P. It can be seen in figure 4.5a that the time spent searching gets proportionally larger with increasing density until a maximum is reached and it decreases thereafter. This decrease can be explained by reduced search times due to increasing density and the increase of execution times with increasing success. Higher T_s and T_e of agents means they spend more time on duty which increases the number of agents that can be found outside (Figure 4.5b). Higher sensor range and thus higher



(a) *Percentage of time spent searching* **(b)** *Number of active agents*

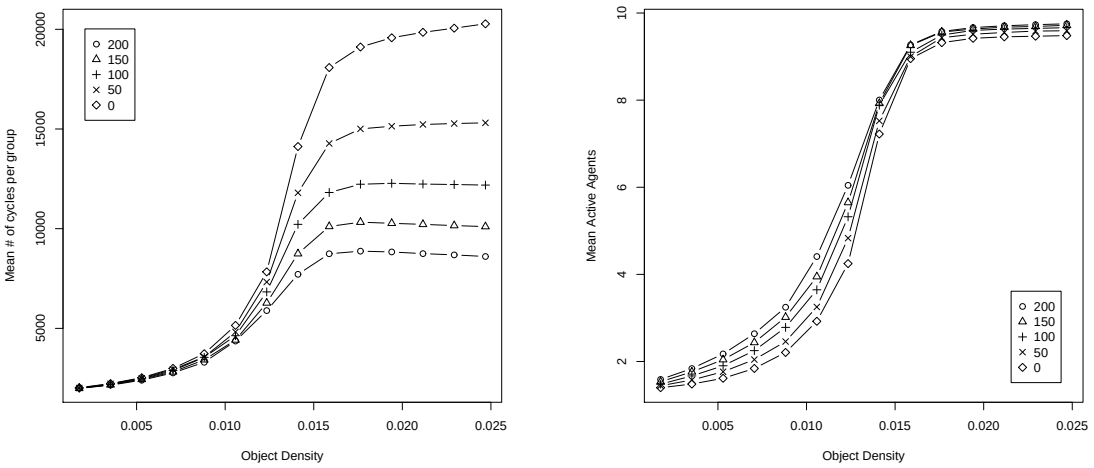
Figure 4.5: *Mean number of active agents and mean T_s of agents as percentage of total time in environments with different object densities. Data points show mean values for homogeneous groups with different sensor ranges ∈ {2.0, 3.5, 5.0, 6.5, 8.0}. Experiments were run 50 times per parameter set and for 200000 time steps.*

success rates means this process is starting earlier when the density in the environment increases.

4.3.2 Affecting Cycle Times

Changing the length of individual cycles mean changing the frequency of changes to P. When a parameter is changed which leads to shorter cycle times, we expect a group to leave the base more often (therefore retrieving more objects) and adjust quicker to an environment. As figure 4.6b shows, this does not generally mean an increase in activity since there is no corresponding change in success rate. When agents are mostly idle, a higher execution time means an agent spends more time outside the base, which explains the increase in activity with higher T_e. In environments for which the group is mostly active, the number of agents outside is lowered with execution times becoming shorter since idle times (which did not change) are now proportionally higher. Although the activity for given environments did not change significantly when cycle times are changed, other properties of the group did. Figure 4.6a shows that the number of times agents left the base increased significantly which lead to a proportionally higher time spent searching and an equally increased count for retrieved objects.

Other parameters that affect cycle times are obviously VD parameters which govern idle



(a) *Number of cycles started by group*

(b) *Number of active agents*

Figure 4.6: *Mean number of active agents and mean number of cycles started by given group in environments with different object densities. Data points show mean values from 50 runs each for homogeneous groups with different execution times ∈ {0, 50, 100, 150, 200} time steps. Experiments were run for 200000 time steps*

times of agents. Increasing idle times leads to a proportional decrease in duty times, thus lowering the number of active agents. By setting the boundary values for P the minimum and maximum duty time can be defined which in turn affects the average number of active agents within a group with P values near the extremes. The effect can be seen in section 4.4.4 where the maximum idle time is lowered and the duty time of a group increased when P values within the group are low. In 4.13 the effects of changes in Δ are shown which defines the speed of changes of P and therefore the speed in which agents adjust the idle times.

4.4 Results from Homogeneous Groups

To understand the properties of the system, experiments with homogeneous groups were performed, thus eliminating one variable: differences between agents. Since all experiments were done in simulation, agents within the same experiment were absolutely identical at the beginning (perfect homogeneity). If not reported otherwise, P_{min} and P_{max} were set to 0.001 and 0.2 (which corresponds to average expected idle times of 1000 and 5 time steps) and $\Delta = 0.003$. Values for P and Δ are always reported normalised onto [0,1] and P was initialized to 0.25 ($P_{min} + 0.25 * (P_{max} - P_{min}) = 0.05075$). This meant an agent needed

between 32 and 500 successful cycles to get from P_{min} to P_{max} depending on how many of them were successive.

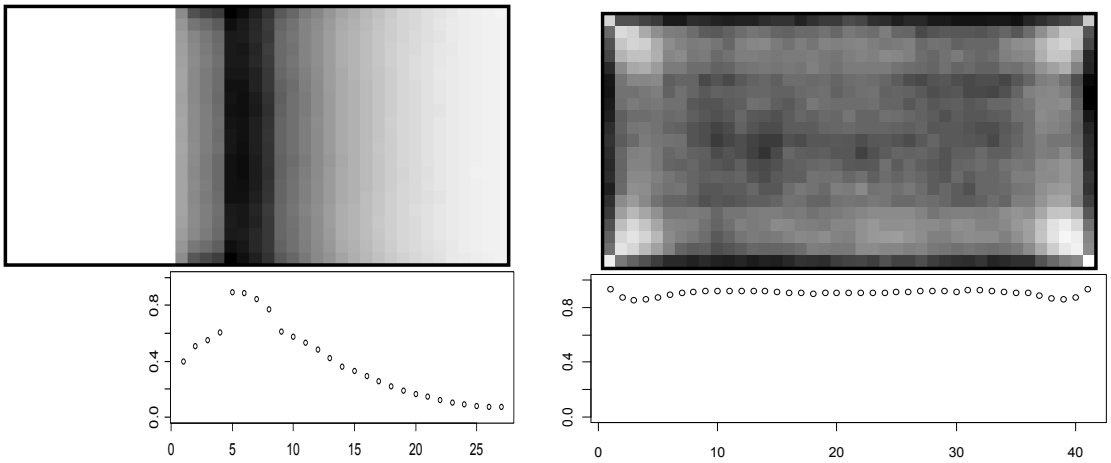
4.4.1 Differences Between ORT and OCT

Although agent parameters are the same in both settings, not having a base to which to return to has some important effects:

- The cycle times T_c of agents are shortened by an amount depending on the average distances of found objects from the base. This can lead to shorter reaction times of the group.
- There are less collisions between agents which are usually most common at the base entrance/exit due to oncoming traffic which again shortens cycle times by shortening both T_s and T_r on average.
- The distribution of objects stays uniform over time. Although objects are placed randomly, the chance of an object to be encountered in the ORT drops with distance from the base as can be seen in Figure 4.7a. This leads to a skewed distribution of objects as shown in Figure 4.10 with objects aggregating further from the base and a therefore decreased overall success rate.
- In addition to more uniformly distributed objects, also agents stay distributed on average for the same reason. As Figure 4.7b shows, this leads to almost equal probabilities for patches to be searched (bar border effects).

For the ORT the frequency with which patches are included in a search, drops with distance. (Figure 4.7a). Since agents in these runs first moved straight out of the base for 5 steps and then scanned an arc with random angle and radius 5 before continuing in the direction they were then facing, the most searched patches have distance 5-8 from the base. In the OCT setting, the frequency with which patches were searched was almost uniform. The only exceptions were patches along the wall (distance 0 from wall) which had a higher probability and patches within a distance of 1 to 4 from a wall, which had a lower probability. The first can be explained by wall avoidance movements which leads to patches along a wall being entered more frequently. After an avoidance turn at a wall, agents often still face along walls which means that patches at walls are visited more often. Furthermore, agents start a search turn by scanning the patches straight ahead before turning in a random direction.

The lower probability near the wall and especially in corners can be explained again by the distance the agents travel straight after turning away from the wall. If an agent hits a wall near a corner, it is very likely to turn towards the second wall which it hits within the first straight movement. Therefore, after a second avoidance turn, it continues to follow the second wall without scanning the patches near the corner. Both effects can also be seen in the ORT setting, albeit less pronounced.



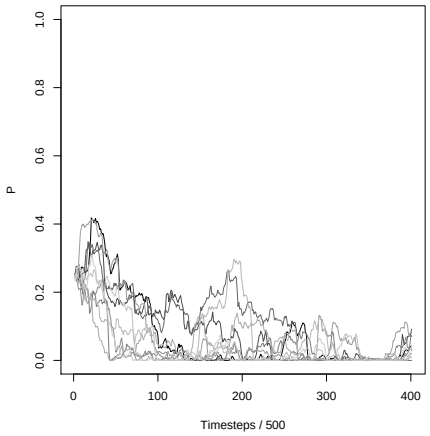
(a) *ORT* **(b)** *OCT*

Figure 4.7: *Search frequency after 1000000 time steps. Shows how often patches (heat map) or vertical strips (graph) were searched, i.e. how often a patch was scanned or entered by agents. Graph shows average of vertical patch columns with same distance from base (ORT) or left wall (OCT). Values were normalised onto and displayed from 0.0/white (≙ lowest observed value) to 1.0/black (≙ highest observed value).*

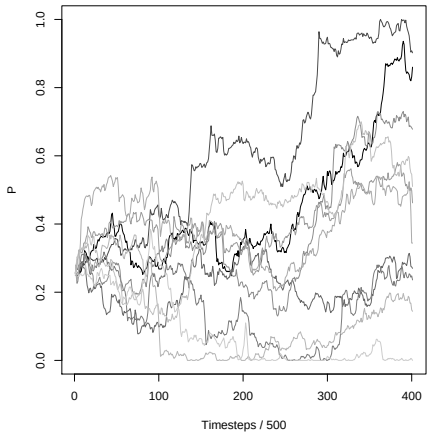
4.4.2 Region of Possible Differentiation

Depending on the agent's success rate, the value P of an agent converges either to P_{min} or P_{max} with the exception of a success rate of 0.5, where P stays constant. The average success rate of agents depends on the overall density of objects in the environment and the agent's abilities, but due to the random distribution of objects and the random paths of agents, it always fluctuates over short time intervals. For a given parameter set, the average success rate is mostly dependent on the object density. The space of possible densities can be split into three sections: If the density is below a certain threshold, the average success rate of all agents is so low that all P values quickly converge to P_{min} since no agent can have continuous success over a longer period of time (for example see Figure 4.8a). Likewise, if the density is above a certain threshold, all P values of a group converge to P_{max} (e.g. Figure 4.8d) Differentiation, which requires different P values within a group, is thus only possible in environments with densities in between those two points. To find this range of environments for later experiments, a group of 10 agents was put into static environments with a varying number of objects and the average number of active agents was recorded.

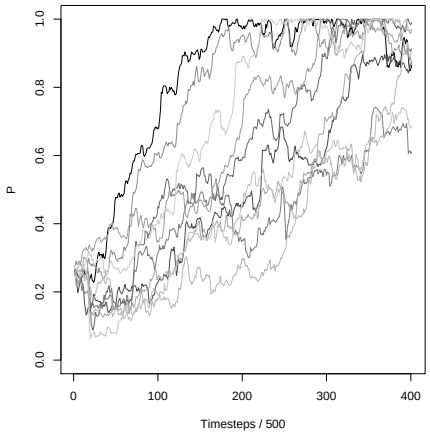
Results for agents with default parameters (sensor range of 5 patches) in both settings are shown in Figure 4.9. Unsurprisingly, for the same densities, agent activity is different for OCT and ORT due to object distribution and the associated higher success rates in the OCT. Due to spatial effects, a group executing the ORT has a lower success rate and



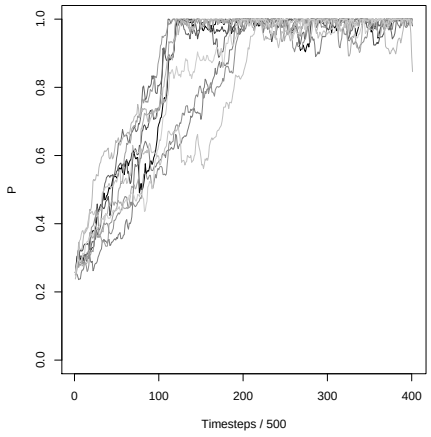
(a) *Density 0.0123,* $\bar{s}_a = 0.46$ *(*$\sigma = 0.0126$*)*



(b) *Density 0.0159,* $\bar{s}_a = 0.51$ *(*$\sigma = 0.0196$*)*



(c) *Density 0.0159,* $\bar{s}_a = 0.54$ *(*$\sigma = 0.0139$*)*



(d) *Density 0.0194,* $\bar{s}_a = 0.59$ *(*$\sigma = 0.0158$*)*

Figure 4.8: *Examples of P over time for agents in ORT setting with default parameters. P is shown normalised to* $[P_{min}, P_{max}]$*. All agents start with P of* $0.25 * (P_{max} - P_{min})$*.* Δ *was set to 0.002. Runs chosen for density of 0.0159 were the runs with the lowest (b) and highest (c) observed* $\bar{s}_a$*. (a) and (d) were chosen randomly. Experiments were run for 200000 time steps.*

therefore lower average activity as compared to a group in a OCT setting with the same object density.

For low densities, all agents in a group spend most of their time idle and only one or two

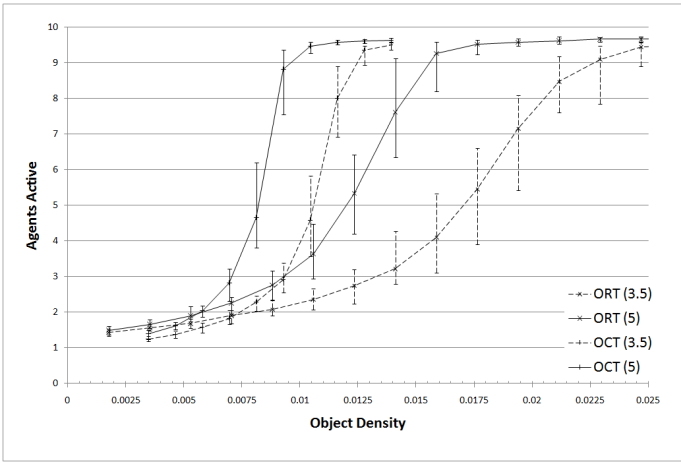
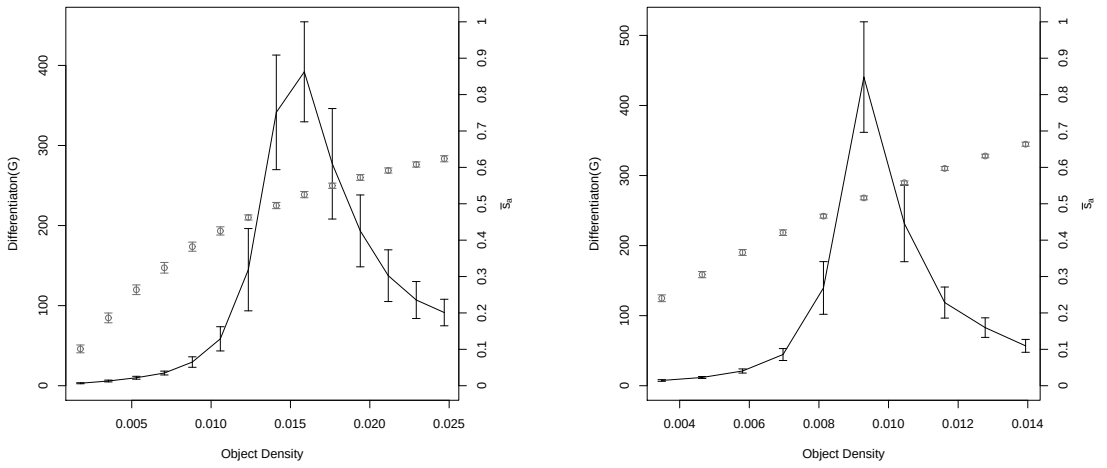


Figure 4.9: *Mean activity of agents in ORT and OCT setting with sensor ranges of 5 and 3.5 patches. Shown is the mean number of active agents over time (agents not in idle status) versus the density of objects in the environment. Experiments were run for 200000 time steps and values averaged over 50 runs per environment. Bars show observed extremes.*

Figure 4.10: *Accumulation of randomly places objects on the left side due to low search frequency by agents. Example of environmental state after initial phase of experiment (2000 time steps) with 10 objects (equates to density of 0.017). Black patches define base, yellow patches denote objects.*

agents are outside the base. Because of $P_{min} > 0$, the average number of active agents is always greater than zero and above one in experiments with default parameters. This is due to the minimum duty time per unsuccessful cycle (≈ 120 for ORT and ≈ 90 for OCT with default parameters) and therefore $\overline{T_D} \approx 1/10 * \overline{T_b}$ at P_{min}. With increasing object density, the success rate of agents that leave the base is also increasing (see Figure 4.9), which means that more and more agents increase their P value and therefore reduce their idle time (even if only temporary). The average activity increases exponentially with success rate up to a success rate of 0.5. With success rates of 0.5 and higher, agents hold or increase their initial P value which, with success rates of 0.6 and higher, quickly leads to almost all agents being active.

Differences within the group appear over time due to differences in the local environment of agents which are reflected in the individual success rates. With proficiency being equal for all agents, the differences in the average individual success rates are small since all agents have the same built-in chance of encountering an object. From this, we can expect the area with maximum differentiation to be around the global success rate of 0.5. With success rates very close to 0.5, agents slowly "drift" away from their initial P, with successes and failures balancing out over time. Since this "drifting" is random and dependent on the current local environment an agent searches, each agent drifts into a random direction, leading to differentiation (as can be seen in Figure 4.8b). Figure 4.11 shows that the environment with the highest observed differentiation was indeed the one with an average



(a) *ORT* **(b)** *OCT*

Figure 4.11: *Accumulated differentiation and average success rate $\bar{s}_a$ for ORT and OCT experiments. Differentiation was calculated every 500 time steps and summarized over one run ($\widehat{=}$400 measurements). Values show averages over 50 runs per environment and error bars show one standard deviation. Success rates are averaged individual success rates from 50 runs ($\widehat{=}$500 samples). and error bars show again one standard deviation.*

success rate of 0.5. Differentiation drops quickly below and above the 0.5 mark as agents without differences in proficiency all fall into the same attractor.

4.4.3 Probability vs. Direct Mapping

In a system that uses the VD algorithm, the idle times of agents in specific cycles can only be estimated by using the expected value. To base idle times on a probability means that the system becomes less predictable for an observer. One possible advantage over a direct mapping of P to idle times might be that agents with low probability sometimes leave the base earlier than expected, leading to a quicker response of the group when facing an increased availability of objects. This is due to the feedback process, where an agent that leaves earlier ($T_b << E[T_b]$) into an environment with high object density could quickly adapt and has a high chance of finishing a number of cycles before another agent, that idles the expected value for T_b, starts its first cycle. Compared to the expected response behaviour, the agent would "jump" to shorter cycle times in the eyes of an observer. This behaviour is outlines in Figure 4.12 where the P values over time are plotted for five agents using the VD algorithm as shown in Algorithm 4. This graph was generated, using a random sequence of successes and failures was calculated corresponding to an overall success

Algorithm 4 Algorithms used by VD (left) and VD* (right) after agent entered the base. First, variable P is adjusted depending on success or failure. Afterwards, the agent idles until a random value between 1 and 0 is smaller than P (VD) or until a calculated amount of time steps have passed (VD*)

Require: Agent enters base	**Require:** Agent enters base
if success **then**	**if** success **then**
$succ = succ + 1$	$succ = succ + 1$
$fail = 0$	$fail = 0$
$P_i = min\{P_{max}, P_{i-1} + succ * \Delta\}$	$P_i = min\{P_{max}, P_{i-1} + succ * \Delta\}$
else if failure **then**	**else if** failure **then**
$fail = fail + 1$	$fail = fail + 1$
$succ = 0$	$succ = 0$
$P_i = max\{P_{min}, P_{i-1} - fail * \Delta\}$	$P_i = max\{P_{min}, P_{i-1} - fail * \Delta\}$
end if	**end if**
	$S = 1/P_i$
while $Random(0,1) > P_i$ **do**	**while** $S > 0$ **do**
	$S = S - 1$
end while	**end while**
Start next cycle phase (Search)	Start next cycle phase (Search)

rate of 0.7 (i.e. 70% success). Then the algorithm was used to calculate P values and sum up idle times T_b, setting $T_s = T_e = T_r = 0$, i.e. only time differences due to different idle times are shown. As can be seen in the five examples for VD, the differences in T_b for low P values might also lead to higher differentiation within the group, since agents with identical initial P values and the same sequence for successes and failures have divergent P values after some time.

To test these hypotheses — quicker response and more differentiation with VD — a modified version of the VD algorithm VD* was used where the idle time T_{b_i} for the i-th cycle is calculated by

$$T_{b_i} = 1/P_i \tag{4.9}$$

i.e. instead of using the variable P as a probability, it is used to calculate the idle times directly (using the expected value, see subsection 4.2.6 and Figure 4.3) to determine the time after which the agent leaves the base. For a given sequence of successes and failures over time, the algorithm produces always the same values of P over time (see Figure 4.12). The transition from idle phase to search phase is now determined by a calculated time, instead of a probabilistic function as in VD. If the first hypothesis is correct, we would expect to see longer response times to changes in the environment when using the VD* algorithm. This should be more pronounced when the environment changes from low to high density since a low P and the corresponding long idle times allow for bigger differences.

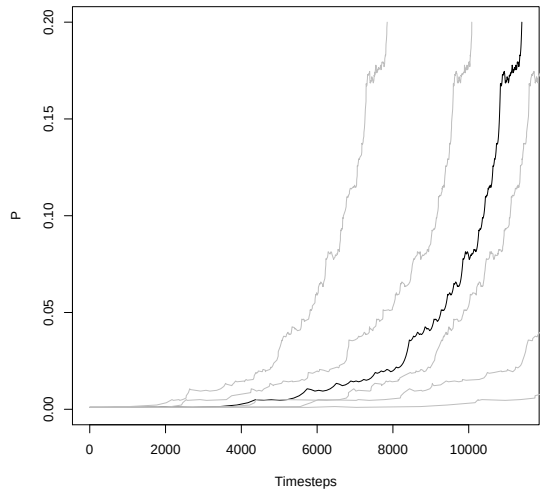


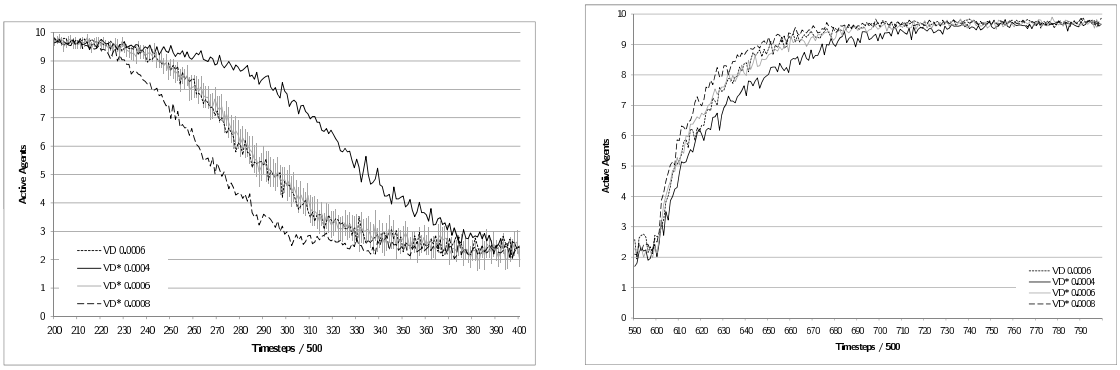
Figure 4.12: *P over time calculated with VD* (black line) and VD (grey lines, 5 examples) algorithm as given in Algorithm 4. The same randomly generated pattern of successes and failures (average success rate of 0.7) was used. Both algorithms used the default value* $\Delta = 0.002$.

To be able to observe response times, experiments were carried out in a changing environment which ran for 500000 time steps. After 100000 time steps (in which the system could settle), the environment was changed to a different density at time step 200000. At 600000 the environment was again set back to its original density at the start of the experiment. In between these external changes, the environment was again kept constant. Three different scenarios were tested:

1. A drop from a high density (0.0247) to a low density (0.0088)
2. A drop from a high density (0.0247) to a density which correlates to an average success rate just below 0.5 (0.0141)
3. An increase from a low density (0.0088) to a density which correlates to an average success rate just above 0.5 (0.0159)

Scenario 1 was also run with $\Delta \in \{0.002, 0.003, 0.004\}$ to study the effects of this parameter on group response times.

Figure 4.13 shows the average number of active agents over time after the drop and increase in density for a Δ of 0.003 in scenario 1. Before the drop, the group settled on the average activity that was typical for that environment. As the density dropped, the group showed a delayed response which was due to the initially slow increase in idle times with decreasing P. As soon as the first agents have a P close to P_{min}, the activity of the

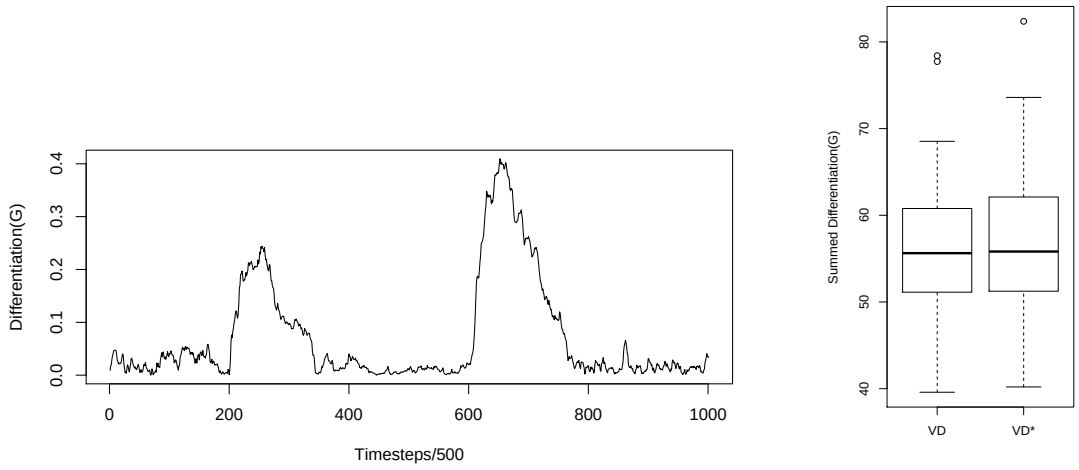


(a) **(b)**

Figure 4.13: *Activity response of group to change in object density from 0.0247 to 0.0088 at x=200 (a) (and back at x=600 (b)). Average number of active agents over 50 runs is shown against time with activity measured every 500 time steps. Experiments were run for different values of* Δ *for VD* and with* $\Delta = 0.003$ *for VD. Error bars in (a) show the 95% confidence interval for VD* with* $\Delta = 0.003$*. Other confidence intervals are omitted to improve readability but are of the same magnitude.*

group drops quickly and converges towards the expected average number of active agents for an environment with that object density. Because of the big changes in idle times near P_{min}, the reaction to a sudden increase in density is much faster but converges slower to the activity level from before the drop. As can be seen in the graph, there is no significant difference in the change of average activity over time between VD and VD*, both for the change from rich to scarce environment and back. A sampling of individual runs showed a tendency of VD runs to have a higher variation in activity, but this did not lead to a significant difference over 50 runs. The same held true for scenarios 2 and 3 in which there was no difference in reaction times between VD and VD*.

The second hypothesis was that there might be a higher degree of differentiation after the changes in the environment due to variations in idle times for agents with the same P value. To test this, differentiation was measured every 500 time steps and summed up from the time of the first change (t=200000) to the end of the experiment. These values were then averaged again over 50 runs. Figure 4.14a shows differentiation over time from a single run for scenario 1. The group differentiates after both external changes to the environment and then became more homogeneous again, which reflects the different individual responses due to local effects. There always was a small peak in the measured success rates when the density increased since the new objects are placed randomly in the environment within one time step. This lead to an unusually high number (for an ORT setting) of objects near the base which temporarily increased the success rate before the objects started to aggregate at the far wall again after a few thousand time steps. This effect lead to agents, which

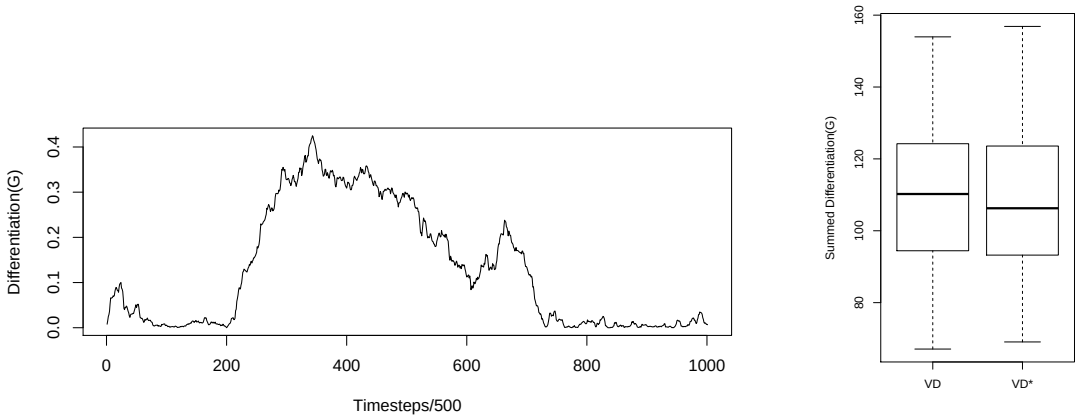


(a) **(b)**

Figure 4.14: *Graph (a) shows differentiation over time in ORT setting with VD algorithm used. The environment changed from density 0.0247 to 0.0088 at time step 100000 (x=200) and back to 0.0247 at time step 300000 (x=600). Shown is one randomly chosen example run with standard parameters and* $\Delta = 0.003$. *Graph (b) shows differentiation summed from time steps 100000 to 500000 in changing environment explained in (a). Sums were averaged over 50 runs for both VD and VD***. Boxes are drawn from 1st quartile to 3rd quartile and whiskers extend to 1.5 interquartile range with outliers shown as circles.*

left the base shortly after, quickly increasing their P value and therefore quickly increasing differentiation. As Figure 4.14b shows, there was no overall difference between VD and VD* in terms of differentiation. There also was no higher variation in Differentiation($\mathcal{G}$) when using VD.

Since scenario 1 was designed as a drop from an upper extreme to a lower, agents quickly went from one attractor for P to the other as response to the environmental change. This meant that the regions of differentiation were short and possible differences perhaps not pronounced enough to be observed. To increase differentiation over time and therefore be able to pick up a difference between the two algorithms, scenario 2 and 3 were designed. A drop from an environment with a high associated success rate to an environment with an average success rate just below 0.5 means that agents would change their P value very slowly towards the lower attractor. This increases the effects of random drift on P and in turn increases differentiation within the group (as shown in subsection 4.4.2). Since success rates near 0.5 lead to differences being sustained, it was thought that differences that arose shortly after a change to such an environment would stay observable for a longer



(a) **(b)**

Figure 4.15: *Graph (a) shows differentiation over time in ORT setting with VD algorithm used. The environment changed from density 0.0088 to 0.0176 at time step 100000 (x=200) and back to 0.0.0088 at time step 300000 (x=600). Shown is one randomly chosen example run with standard parameters and* $\Delta = 0.003$*. Graph (b) shows differentiation summed from time steps 100000 to 500000 in changing environment explained in (a). Sums were averaged over 50 runs for both VD and VD*. Boxes are drawn from 1st quartile to 3rd quartile and whiskers extend to 1.5 interquartile range with outliers shown as circles.*

time. Therefore, if there are differences between VD and VD* in terms of differentiation, the effects should be more pronounced in these scenarios.

As Figure 4.15a shows, differentiation was higher when object density was increased to a medium level and was observable over the whole phase. Sampling of individual runs showed that individual P values of agents drifted slowly into random directions with a tendency towards P_{max}, as expected. But even with the group staying differentiated for longer, there was again no measurable difference between VD and VD* (see Figure 4.15b). Also an in-depth analysis of the activity response and differentiation in small time intervals after the environmental changes showed no significant differences. The same observation held true for scenario 2 and it therefore has to be concluded that there is neither a difference in activity response when using VD as compared to VD*, nor a difference in differentiation within the group even on small time intervals.

4.4.4 Efficiency Increase Through Adaptation

In his original experiment, Labella has shown increased efficiency for groups using the VD algorithm (VD group). As efficiency measure he used $E_{\mathcal{G}}$ as described in section 4.2.5 (total

number of objects retrieved by the group divided by the total amount of time spent outside the base). In order to see differences in efficiency, he compared a group using the VD algorithm, to a control group of the same size which was tested in the same environment. The VD algorithm was rendered inactive in the control group by setting $P_{min} = P_{max} = 1.0$.

Labella showed an efficiency increase for all group sizes in environments where the system did not show saturation effects (i.e. when not all agents were fully active). Since Labella used a regrowth model for objects (regrowth factor was kept constant over the course of the efficiency experiments), the density decreased with increased agent activity and the total number of objects collected stayed constant over different group sizes (due to maximum number of available objects per time). Since the number of collected objects was equal for all groups in the same environment, efficiency differences were only due to different duty times. Therefore, Labella reached the conclusion that reducing the duty time of agents through the VD algorithm lead to the efficiency increase.

One criticism, which Labella himself already mentioned, was the value of 1.0 for P_{max} which lies outside the range of P values the VD agents use ([0.001,0.2]). This means that even in environments where P values of agents in the VD group converge to P_{max}, the control group would still show higher duty times.

Another difference between Labella's and this work was the amount of objects the agents were able to collect. As Labella's data show, there was an easy to reach maximum in the number of objects that could be collected, which explains the constant object count over different group sizes. He only showed one setting in the efficiency experiments where increasing group size also increased the number of objects collected by a group: The increase from single agent to a group of more agents in the environment with the highest density. This shows that for all other environments one agent was already enough to collect all available objects and two were sufficient in the densest environment. As already described, the density is kept constant in the experiments used in this work and which means a constant availability of objects. The theoretical maximum number of objects that a group can collect is only defined by the number of agents in a group, the shortest possible cycle time (i.e. objects all placed at the base) and the length of the experiment. With increasing cycle times, the number of collected objects automatically also drops (due to fewer attempts per time). Hence, efficiency changes are always caused by changes in both variables since changes in the number of collected objects always effect changes in duty times and vice versa.

From the definition of efficiency follows directly that:

$$E_{\mathcal{G}} > E_{\mathcal{C}} \Rightarrow \frac{O_{\mathcal{G}}}{T_{D_{\mathcal{G}}}} > \frac{O_{\mathcal{C}}}{T_{D_{\mathcal{C}}}} \Rightarrow \frac{O_G}{O_C} > \frac{T_{D_{\mathcal{G}}}}{T_{D_{\mathcal{C}}}} \tag{4.10}$$

with $O_{\mathcal{G}}$ being the total number of objects collected by the VD group and $T_{D_{\mathcal{G}}}$ being the total amount of time agents of the VD group were active (respectively O_C and $T_{D_{\mathcal{C}}}$ for the control group). In Labella's experiment, the ratio of $O_{\mathcal{G}}$ to $O_{\mathcal{C}}$ was close to 1.0 which

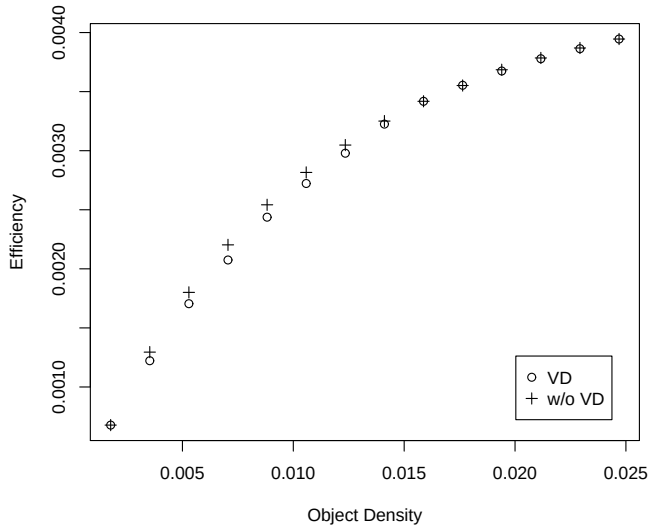


Figure 4.16: *$E_{\mathcal{G}} - E_{\mathcal{C}}$ in percent of $E_{\mathcal{C}}$ for different object densities. Values averaged over 50 runs for each environment and group. Differences between VD and control group with default parameters are significant up to a density of 0.0123 and not significant from 0.014 onwards.*

meant that spending less time outside the base than the control group automatically led to a higher efficiency.

To investigate whether this efficiency increase through the VD algorithm could also be observed in this work, $E_{\mathcal{G}}$ values were calculated for groups in the different static environments (see subsection 4.4.2) and compared to a control group. In order to avoid the aforementioned criticism, P_{min} and P_{max} within the control group were set to P_{max} of the VD groups (0.2). Therefore, the efficiency differences between VD and control group converge to 0 when the VD group approaches maximum activity in an environment.

Figure 4.16 shows the differences between $E_{\mathcal{G}}$ and $E_{\mathcal{C}}$ for different static environments and, contrary to Labella's results, efficiency decreased when group activity was lowered through the VD algorithm (mean activity for these experiments can be seen in figure 4.9). Following equation 4.10 this means that a VD group in those experiments collected proportionally less objects than decreasing duty time.

This finding was interesting since it meant that using the VD algorithm to adapt duty times of agents to the availability of objects does not automatically lead to efficiency increases but is dependent on the type of environment and the agent parameters used. With the average efficiencies measured for static environments, $E_{\mathcal{G}}$ was calculated for the changing environment described in section 4.4.3. When $E_{\mathcal{G}}$ was calculated using the previous data and compared to data from a control group, it turned out that $E_{\mathcal{G}} > E_{\mathcal{C}}$

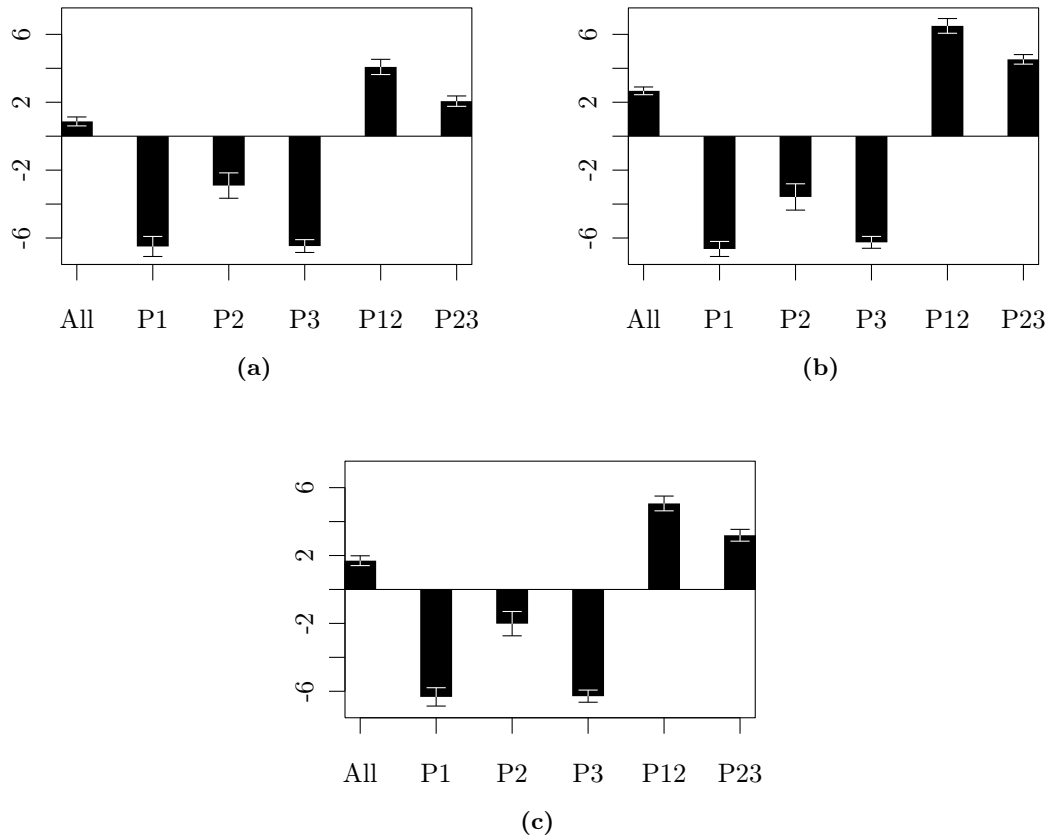


Figure 4.17: *Efficiency gains for experiments with changing environments using ORT setting. Density in time intervals P1 and P3 was 0.0247 and 0.0088 in P2. Data points show efficiency gains of VD group in percent compared to control group. Error bars show 95% confidence interval*

in those experiments. This was peculiar since the agent parameters used, were the same as in the experiments with static environments and the efficiency was expected to be a combination of the corresponding values for the given densities (allowing for differences due to transition times). The experiments were split into three time intervals: P1,P2 and P3 with density(P1) = density(P3) and density(P1) > density(P2)). The time interval P1 was half the length of P3 (100000 and 200000 time steps) and P2 had the same length as P3. Since efficiency is calculated over a time interval, it was possible to calculate efficiencies for individual phases and a combination of those (e.g. P12 = P1 + P2 and thus is the time interval from time step 0 to 299999).

The group $\mathcal{G}_1$ consisted of agents with default values and $\Delta = 0.002$ (i.e. slower response to changes). Compared to the control group, the efficiency $E_{\mathcal{G}_1}$ was higher when calculated

for the full time of the experiments but consistently lower for the individual phases. The efficiency was also higher for any combination that included P2. The combination P1+P3 is not shown but was consistent with the values shown for the static environments with the given density (0.0247). As expected, the efficiency of the control group was always higher in the high density phases, since the adaptive group never reaches full activity (which would only be possible for constant P_{max}). As can be seen, for all three VD groups tested: $E_{\mathcal{G}}^{P1} < E_{\mathcal{C}}^{P1}$ and $E_{\mathcal{G}}^{P2} < E_{\mathcal{C}}^{P2}$ but $E_{\mathcal{G}}^{P12} > E_{\mathcal{C}}^{P12}$ which results in

$$\frac{O_{\mathcal{G}}^{P1}}{O_{\mathcal{C}}^{P1}} < \frac{T_{D_{\mathcal{G}}}^{P1}}{T_{D_{\mathcal{C}}}^{P1}} \text{ and } \frac{O_{\mathcal{G}}^{P2}}{O_{\mathcal{C}}^{P2}} < \frac{T_{D_{\mathcal{G}}}^{P2}}{T_{D_{\mathcal{C}}}^{P2}} \text{ but } \frac{O_{\mathcal{G}}^{P1} + O_{\mathcal{G}}^{P2}}{O_{\mathcal{C}}^{P1} + O_{\mathcal{C}}^{P2}} > \frac{T_{D_{\mathcal{G}}}^{P1} + T_{D_{\mathcal{G}}}^{P2}}{T_{D_{\mathcal{C}}}^{P1} + T_{D_{\mathcal{C}}}^{P2}} \tag{4.11}$$

The low efficiency in P2 of the VD group means that the VD algorithm reduces duty times below the optimum for the given density. A thusly adjusted agent therefore stays longer in the base than necessary for his success rate which results in a lower efficiency for the group. But this "over-adjustment" leads to a higher efficiency when combined with a high-density phase (e.g. P1).

Two more experiments were run to test the effects of parameter changes. Agents in the group $\mathcal{G}_2$ were set to a Δ of 0.004 which (as can be seen in section 4.4.3) leads to a quicker change in P and therefore a quicker response of the group to changes. $\mathcal{G}_2$ therefore settles quicker in P2 than $\mathcal{G}_1$ but the reduction of accumulated duty time in P2 by 21.6% is counterbalanced by a reduction in collected objects of 22.1% and consequently leads to a further decrease in efficiency compared to $\mathcal{G}_1$. But when combining P1 and P2, $\mathcal{G}_2$ collected only 8% less cans than $\mathcal{G}_1$ on average which is now pitted against a reduction of 10% in total duty time, leading to a higher efficiency for $\mathcal{G}_2$ in P12.

To see whether a more optimal adjustment in P2 would lead to an overall increase in efficiency, a third group was compared to $\mathcal{G}_1$. Agents in $\mathcal{G}_3$ used the same Δ as $\mathcal{G}_2$ to adjust P but also were given a higher P_{min} of 0.002, effectively cutting the maximum idle time by half. This does not affect overall activity in P1 and P3 since agent's P values never drop to minimum, but increases activity in P2. As hypothesised before, this increased duty time lessens the "over-adjustment" and indeed increases the efficiency of $\mathcal{G}_3$ in P2 as compared to $\mathcal{G}_1$ and $\mathcal{G}_2$. But this increase in P2 leads to a reduced efficiency in P12 and P23.

4.4.5 Discussion

Labella has shown in his work that by using the VD algorithm, the activity of a group can be adjusted to object availability in the environment. After adopting the algorithm for the system at hand, the aim was to confirm the findings from the original work and then extend it to investigate its effect on heterogeneous groups. Experiments with truly homogeneous groups are only possible in simulation and were initially only planned to test parameter settings for the VD algorithm. After a few runs it became evident that it was harder to adjust the parameters than expected. In the beginning of this section it was shown by the

comparison between the object retrieval and object consumption setting that the existence of a base, to which the agents have to return to, introduces spatial effects that have to be taken into account. The action of returning to base introduces a minimum cycle time which can act as damping for the accelerating increase or decrease through the self-reinforcing feedback. Since all agents start their cycle at the same position, also interference with other agents when exiting and returning to the base further increase cycle times and therefore group responses. Another effect is the distribution of objects. Objects near the base have a much higher chance of being encountered and in combination with random placement accumulate further away from the base over time. The exact placement method in Labella's work is not known, but due to the circular environment and the same effect, a similar density would lead to a different object distribution which makes direct comparison between different set-ups difficult. As an example, the group response in an OCT setting is very different as compared to a group with the same parameters in the ORT set-up. Adjusting parameters of agents executing the object consumption task to get comparable responses to agents executing object retrieval turned out to be more difficult than expected. When trying to select parameters for the VD algorithm, the average success rate of agents has to be proven to be a helpful measure. For homogeneous groups, it can easily be measured for a group by observation and after a short time can give a good estimation on the behaviour of the group in the remainder of the experiment.

Depending on the success rate for the given environment and agent parameters, the VD algorithm can only adjust group activity in a range of environments. When the average success rate was much different from 0.5, all agents' P values fell quickly towards one of the two attractors and the group performed either at maximum or minimum activity. Stable differentiation was not expected in these homogeneous groups, but is was shown that near a average success rate of 0.5, local effects are enough to differentiate agents within a group. The same differentiation has been observed after changes in the environment when the group starts to adjust to the new density. Sample tests have shown that with a higher frequency of changes, these differences between agents were observable for a long time and might be stable under certain conditions (with the required frequencies being dependent on the Δ values of agents).

In the original algorithm, idle times are determined by a probability to leave the base. Although this is easy to implement in artificial agents, it induces randomness into the system and made it difficult for an observer to identify the agent's internal state or predict its behaviour in the coming cycles even when knowing the agent's P value. It was shown that there was no difference in the group's activity response when using a direct mapping to determine idle times. By using the expected value of the original algorithm, idle times can be directly calculated by an agent and counted down and an observer needs only to observe a single cycle to estimate the agent's current state. This can be useful for a group of robots to switch to other tasks within this idle time without changing the dynamics of object retrieval (e.g. maintenance or recharging) or when agents have to determine the state of other agents in their surroundings.

In the last part, Labella's finding of an efficiency increase was tested. Since Labella had found an efficiency increase over a control group in all investigated scenarios, using the VD algorithm seemed like a simply way to improve the performance of a group for a retrieval task. We were able to show that for the setting used in this work, efficiency didn't automatically increase when a VD group adjusted to its environment. In static environments, not only was there no efficiency increase with the given parameter set, but it decreased in the range of environments where the VD algorithm had an effect on group activity. Although it was shown that tuning the parameters of the algorithm can lead to an increase in efficiency for one object density, the same change did decrease it in a dynamic environment, making it difficult to find an optimal parameter set when the environment is not exactly known. This of course depends on the definition of efficiency and the environments used. Since the VD algorithm adjusts duty times, an efficiency measure with higher weight on energy expenditure of an agent would lead better results. Also the constant availability of objects in this work had a huge impact on efficiency. When there is a connection between available objects and agent activity (as there is in the original work), depending on the nature of this connection, efficiency increases might be more common by adjusting duty times. But with the definition of efficiency as given by Labella, the results in this work were highly dependent on the type of environment and the time interval efficiency was calculated for.

4.5 Results from Heterogeneous Groups

In the last section, all agents in a group shared the same parameters within an experiment and only differentiated through the internal variable P over time. This perfect homogeneity can usually only be found in simulation, since hardware differences in real robots introduce heterogeneity to different degrees. These small differences can have an impact on the agent's performance and get reinforced by the VD algorithm. If an agent has a better performance, efficiency of the group can be increased if this agent is more active than others.

Labella has shown that the VD algorithm leads to a higher activity of agents that show a better performance over other agents ("Selection of best individuals"). In 4.5.1, those results are confirmed and it is shown that differentiation correlates with individual success rates for the given environment. This "selection" of better individuals within the VD process in turn means that individuals with lower performance decrease their activity first when the object density decreases. If individuals with lower proficiency become less active, the mean success rate of the group increases since active agents have a higher impact. This should also lead to a higher group efficiency when compared to a fully active group and results can be found in section 4.5.2 below..

A heterogeneous group will also show a different activity pattern over different environments depending on its composition. Results from homogeneous groups show that agents with different success rates become active for different object densities. A group composed

of two types of agents A and B is expected to split into two subgroups — one mainly active, the other inactive — when the success rate of agents A is above 0.5 for an environment and below 0.5 for B. The differentiation in environments can therefore be predicted when individual success rates for agents are known.

4.5.1 Differentiation in Heterogeneous Groups

As shown in 4.3.1, parameters that affect success rates have an effect on the activity pattern of a group. Agents with higher performance become active earlier and are fully active in environments where agents with lower performance are only leaving the base from time to time. This should lead to stable differentiation in environments where some agents show high activity while others are still mostly inactive, and therefore a stable split into active/inactive subgroups over time.

Groups with different parameters that affect cycle times but not success rates are thought to behave like homogeneous groups. Agents in that group show the same average P values over different environments and therefore all have comparable activity. Results reported in 4.3.1 showed that changes in T_e have effects on the number of objects collected and the proportion of T_s in the cycle time. Since these effects lead to observable differences between agents, such a group was also investigated to find out to what degree these differences affect group differentiation.

Heterogeneity Through Sensor Range

To test how a heterogeneous group splits into subgroups, a group of 10 agents was used in a ORT setting with different sensing ranges (2.3, 2.9, 3.5, 4.1, 4.7, 5.3, 5.9, 6.5, 7.1, 7.7). Except for sensor range, all other parameters were set to default values and 50 Experiments were run for 200,000 time steps in an environment with an object density of 0.0123.

Figure 4.18 shows that such heterogeneous group splits over time into two subgroups: One with active ($P > 0.8$) and the second with mostly inactive agents ($P < 0.2$). This split, once reached, is stable until the end of the experiment. The mean fill of the bins in the last time step was (5.54 0.24 0.14 0.12 0.10 0.12 0.10 0.36 0.34 2.94) and it has to be noted that agents are mostly active with $P > 0.5$.

Results from Labella on the selection of best individuals were replicated in these experiments (Figure 4.19) since activity of agents increased with sensor range. The split between active and inactive agents is reflected in the accumulated duty times with three almost fully active and five less active agents.

Individual success rates s_a are shown in figure 4.20. As already shown with homogeneous groups, a high activity of agents correlates with a high mean success rate with a value of 0.5 being a threshold above which agents becomes mostly active. This was confirmed as mean success rates of agents in the two subgroups are above 0.52 for the three most active agents and below 0.48 for the five less active agents. The two agents with mean success

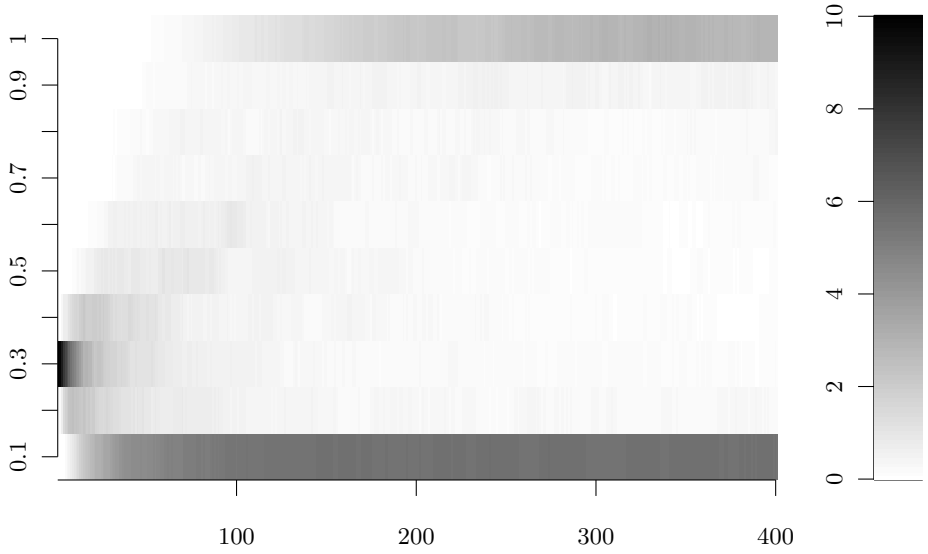


Figure 4.18: *Histogram of P over time for a heterogeneous group (sensing range = (2.3, 2.9, 3.5, 4.1, 4.7, 5.3, 5.9, 6.5, 7.1, 7.7)). Data points were averaged over 50 runs in environment with density 0.0123. 10 bins for P ∈ [0,1] were used with width 0.1. Grey scale shows mean number of agents with P value in given bin at time t.*

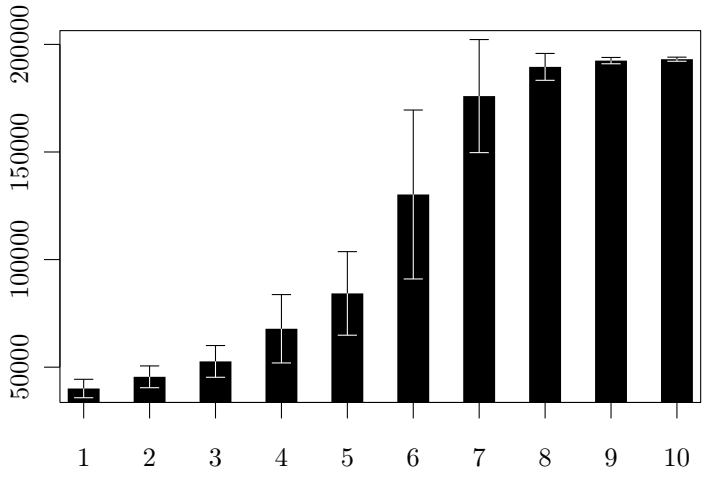


Figure 4.19: *Cumulated duty time for agents in heterogeneous group (sensing range = (2.3, 2.9, 3.5, 4.1, 4.7, 5.3, 5.9, 6.5, 7.1, 7.7). Bars show duty time averaged over 50 runs in environment with density 0.0123 with error bars showing one standard deviation.*

rates of 0.483 and 0.518 show a more fluctuating activity and can not be fully counted to

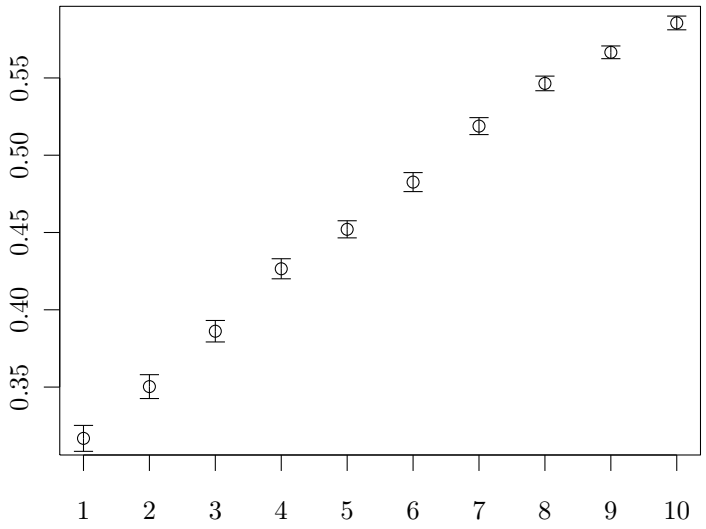


Figure 4.20: *Individual success rates of agents in heterogeneous group (sensing range = (2.3, 2.9, 3.5, 4.1, 4.7, 5.3, 5.9, 6.5, 7.1, 7.7). Data points are mean values of 50 runs in environment with density 0.0123. Error bars show 95% confidence interval.*

their respective subgroups.

The cardinality of the subgroups is therefore dependent on object density and the sensing range of agents. A group starts differentiating when the agent with the highest sensor range starts becoming more active. While this would lead to the whole group becoming active in homogeneous groups, keeping differentiation low, it does not change the behaviour of other agents with lower proficiency. This can be seen in figure 4.21 where differentiation started just before the best individual reached a success rate of 0.5. Differentiation was maximum at density 0.016 where the group split into two almost equal subgroups (theoretic maximum for accumulated differentiation was 220.55). It dropped with higher densities when even more agents became active and reached low levels (< 50) as the last agent reached a success rate of 0.5.

Heterogeneity Through Execution Times

It was already shown that groups of homogeneous agents with different parameter values for T_e show similar activity patterns. The VD algorithm was not affected by changes in cycle times and groups only showed differences in activity that were the direct result from different execution times. A group composed of agents with varying T_e is therefore expected to behave like a homogeneous group in terms of individual agent activity.

To test whether such a group differentiates because of individual differences, a group

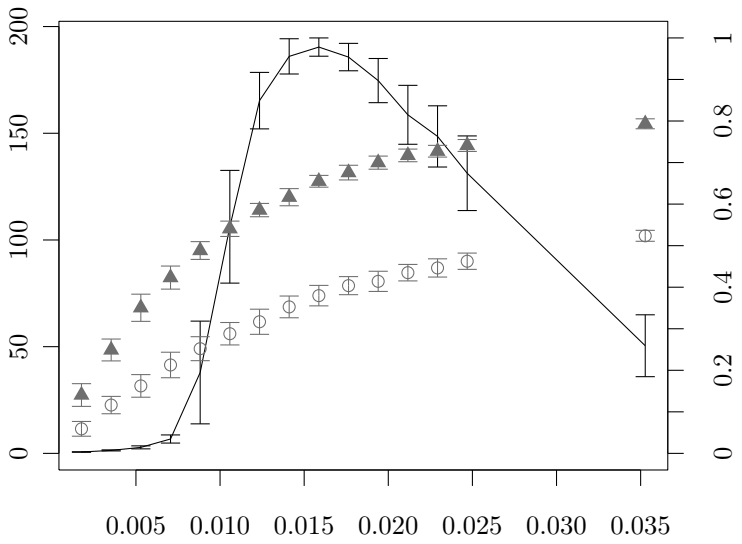


Figure 4.21: *Differentiation and mean success rates in heterogeneous group (sensing range = (2.3, 2.9, 3.5, 4.1, 4.7, 5.3, 5.9, 6.5, 7.1, 7.7). Differentiation is summed over the length of experiments and success rates are of agents with highest (triangle) and lowest (circle) sensing range. Error bars show one standard deviation.*

of 10 agents with T_e values of (190, 170, 150, 130, 110, 90, 70, 50, 30, 10) time steps was tested in environments with different object densities. Except for T_e, parameters were set to default values. All experiments lasted for 200,000 time steps and were repeated 50 times for each environment.

As expected, the level of differentiation (Figure 4.22) was the same as displayed by homogeneous groups with default parameters. Also individual success rates for given environments showed no significant differences between agents. Differentiation was highest for environments around density 0.016 and coincided with success rates of around 0.5, which was already shown to lead to differentiation through drifting P values.

In 4.3.2 it was shown that decreasing T_e decreases cycle times and leads to a proportional increase in T_s. Agents with lower T_e also left the base more often which, with success rates unchanged, means an increase in the number of collected objects per time.

These results can also be observed in the experiments with varying T_e, showing a significant increase in the sum of collected objects, T_r, and T_s with decreasing T_e (figures 4.23b-d). Despite these increases, there was no change in overall duty time for agents with lower T_e (see figure 4.23a). Therefore, parameter differences between agents that lead to observable differences in the number of collected objects and the number of cycles did not affect group behaviour in terms of activity. The decrease in T_e summed over an experiment was balanced by an increased sum of both T_s and T_r. It has to be noted that these increases in the sums are due to an increased frequency of searches, not an increase in individual T_s

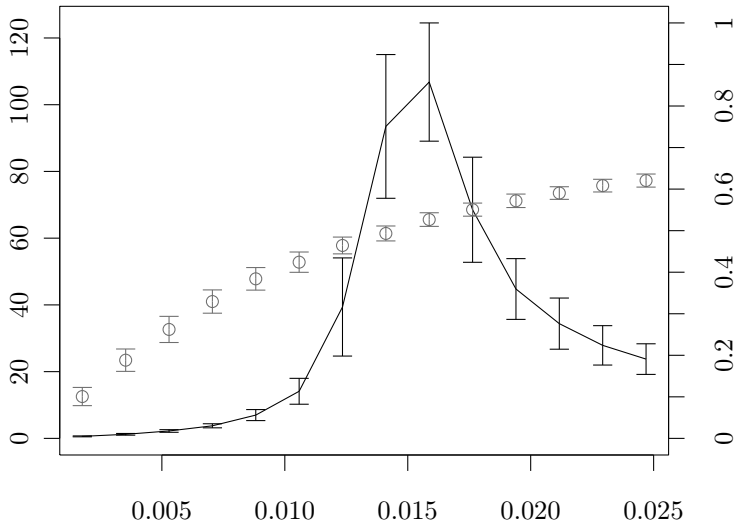


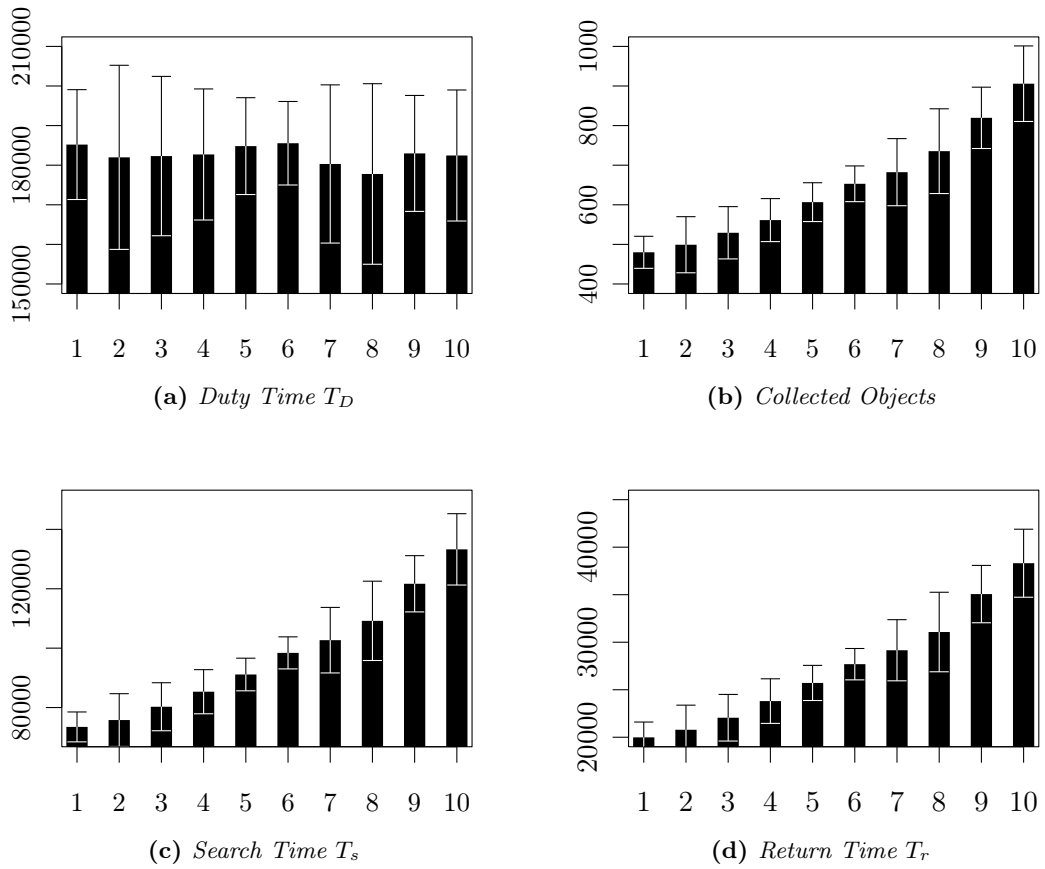
Figure 4.22: *Differentiation and mean success rate in heterogeneous group (T_e = (10, 30, 50, 70, 90, 110, 130, 150, 170, 190). Differentiation is summed over the length of experiments and $\bar{s}_a$ is shown since there was no significant difference between individual success rates and $s_\mathcal{G}$. All values were averaged over 50 runs and error bars show one standard deviation.*

or T_r per cycle The VD algorithm in these experiments was unaffected by the heterogeneity in the group and no differentiation above the level of a homogeneous group was observed.

4.5.2 Efficiency Increase Through Heterogeneity

In homogeneous groups there was no efficiency gain for groups when using the VD algorithm to adjust group activity in static environments. In dynamic environments, the VD group only showed higher efficiency than the control group when the observed time interval included both high and a low density phase. In heterogeneous groups, the VD algorithm increases the activity of agents with high proficiency faster than the activity of other agents and can lead to stable differentiation in certain environments. In these cases, most of the work in the group is done by agents which are better in executing the task (i.e. show higher success rates) which should lead to higher group efficiency. Figure 4.24 shows mean efficiency values for different heterogeneous groups compared to the result from a homogeneous group. In control groups, P_{min} was set to P_{max} again to "freeze" the VD algorithm on minimum idle time.

A homogeneous group has lower efficiency compared to the control group in environments where the VD algorithm lowered group activity. As figure 4.24b shows, a heterogeneous group with sensor ranges (3.65, 3.95, 4.25, 4.55, 4.85, 5.15, 5.45, 5.75, 6.05, 6.35) had less efficiency loss in lower densities and even showed an efficiency gain when almost all agents



(a) *Duty Time* T_D **(b)** *Collected Objects*

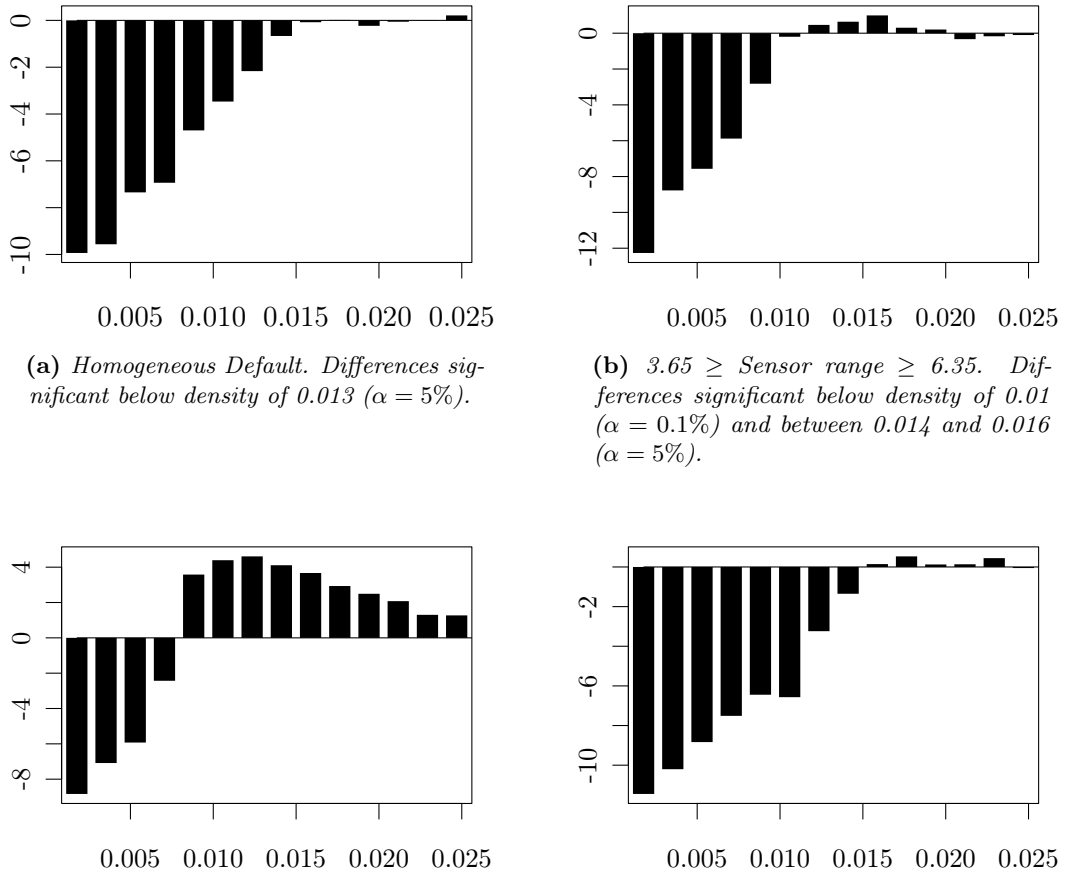
(c) *Search Time* T_s **(d)** *Return Time* T_r

Figure 4.23: *Mean accumulated times* T_D, T_s, T_r *and number of collected objects for agents of heterogeneous group with* T_e *= (190, 170, 150, 130, 110, 90, 70, 50, 30, 10). Data points are averaged over 50 runs in environment with density 0.016 and error bars show one standard deviation.*

were active. A second group with an even wider spread of sensor ranges (2.3, 2.9, 3.5, 4.1, 4.7, 5.3, 5.9, 6.5, 7.1, 7.7) was tested. This group (Figure 4.24c) showed significant efficiency gains in all environments where at least one agent was mostly active (usually the most proficient agent).

For the sake of completeness, efficiency measurements are shown for a heterogeneous group with varying T_e values. As for differentiation in that group, also the efficiency was comparable to a homogeneous group with default parameters.

These results show that in a heterogeneous group, the VD algorithm has an impact on efficiency. The "selection of best individuals" leads to the proficient agents being more active than the others. It can also be seen as a "de-selection" of less proficient agents, which



(a) *Homogeneous Default. Differences significant below density of 0.013 ($\alpha = 5\%$).*

(b) *3.65 $\geq$ Sensor range $\geq$ 6.35. Differences significant below density of 0.01 ($\alpha = 0.1\%$) and between 0.014 and 0.016 ($\alpha = 5\%$).*

(c) *2.3 $\geq$ Sensor range $\geq$ 7.7. All differences were significant ($\alpha = 0.1\%$).*

(d) *10 $\geq T_e \geq$ 190*

Figure 4.24: *Efficiency gain compared to control group in percent. Data points are mean values of 50 repetitions for each group of 10 agents. Parameters were taken from given interval with equal distance.*

leads to efficiency gains in environments where the VD algorithm reduces activity. The gains are therefore higher in groups where the spread of proficiencies is higher since the average group success rate increases more when less proficient agents become idle. This effect should increase the efficiency of a heterogeneous group in dynamic environments which is investigated in the next section.

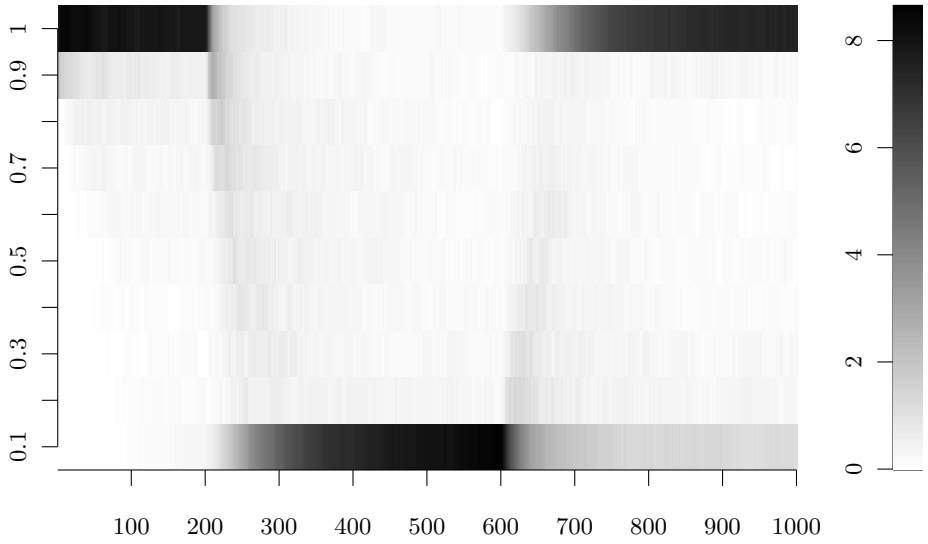


Figure 4.25: *Histogramm of P_i values over time in environment with drop in density from 0.025 to 0.0088. 10 bins were used for P_i with sizes of 0.1 and greyscale shows number of agents within corresponding bin from 1 to 10.*

4.5.3 Heterogeneity and Dynamic Environments

As with homogeneous groups, heterogeneous groups were tested in changing environments. The same basic setting as in 4.4.3 was used. The density in the environment was again dropped from high (0.025) to low (0.0088) after 1/5th of the experiment and restored to high density after 3/5th. The same labels P1,P2, and P3 are used to denote the three time intervals and P12 and P23 for the combined intervals. All agents started at 90% of the P range.

Figure 4.25 shows a histogram of P_i values in the group over the full time of the experiment. The group had the wide spread of sensor ranges as described in the previous section. From experiments in a static environment it was already known that this group usually splits into rough1y one inactive and nine active agents. It can be seen in the histogram that through P1, the group started splitting up like in the static case. After the drop in density, most agents became inactive except for the 1-2 most proficient agents which stayed active. When the density was raised again, around 80% of the group became active again and the group showed the split which was already observed in static environments.

The efficiency of the group for the three time intervals and the combined intervals is shown in 4.26. Unlike in the homogeneous experiments, the heterogeneous group gained efficiency on average when using the VD algorithm even in P1 ($p < 0.074$) and P3 ($p < 5.28E\text{-}10$). Again the efficiency gains were largest when the time interval included P2 and either P1 or P3. The gains were lowest in P1 and P3 since the group converges to

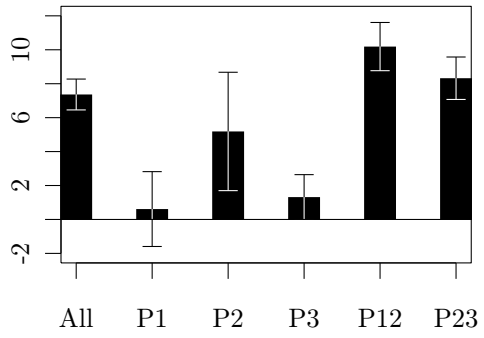


Figure 4.26: *Efficiency gain compared to control group in percent in environment with drop in density from 0.025 (P1,P3) to 0.0088 (P2). Data points are mean values of 50 repetitions for group of 10 agents.*

the efficiency value of the control group when the group is almost fully active. Efficiency gains can only be observed in environments where the VD algorithm reduced overall group activity. In those environments, the group can profit from the specialists in the group.

4.5.4 Discussion

The results show that heterogeneity through parameters that also affect success rates of agents have an effect on differentiation within the group. A group splits up into two subgroups in an environment when parameter sets of agents in the group are such that some agents show a success rate above 0.5 and some below. Success rates of agents with a given parameter set can be estimated from results in homogeneous settings. The number of agents with an estimated success rate below 0.5 for a given object density gives an estimate for the size of the inactive group.

It is also shown that although parameters that affect cycle times have an effect on the number of objects collected or the frequency of searches by an agent, they have no effect on differentiation. Agents that differ because of such parameters show an activity pattern characteristic of an homogeneous group, i.e. all agents change their activity uniformly.

The most interesting aspect of heterogeneity in this setting is the efficiency increase when the VD algorithm is used to control group activity. After it was shown that there was no increase in homogeneous groups in the same environments, results show that the "selection of best individuals" by the VD algorithm had a big impact. In a homogeneous group with almost equal success rates, all agents have on average the same ratio of collected objects to time spent on searching. Deactivating one agent does not change the average ratio in the group. In a heterogeneous group, the success rates of agents are different which leads to different individual efficiencies. Lowering the activity of agents with lower success rate gives the more successful agents higher weights when calculating group efficiency,

hence increasing it. The bigger the difference between agent's success rates, the bigger the efficiency gain.

In the last experiments it was also confirmed that in heterogeneous groups the VD rule adapts the split into subgroups to a dynamic environment. Compared to the results from homogeneous groups in the same dynamic environment, the heterogeneous group shows an overall efficiency gain in all time intervals. The gain was highest in environments where the VD algorithm reduced group activity by lowering the activity of unsuccessful agents.

4.6 Conclusion

Following the hypotheses this chapter was based on, it was shown that

- there is no automatic efficiency gain by adapting the number of active agents in a group. The efficiency gains reported in (Labella et al. 2004) for homogeneous groups could not be reproduced in this work. The major difference was the replacement of objects in the environments. In this work, the object availability was constant which meant that agents leaving the base always had the same chance to encounter an object. In Labella's work, the objects "regrew" with a certain rate which means that the retrieval of an object by one agent temporarily reduces the chance for other agents to encounter an object. Reducing the number of active agents in such an environment seems to have an impact on efficiency even in homogeneous groups.

  Nevertheless, it is shown that there can be an efficiency increase even in homogeneous groups with a constant object density. It was shown that in dynamic environments, reducing the number of active agents in a low density environment can lead to an overall efficiency increase. Combining time intervals with no efficiency gains in the individual intervals can still lead to higher efficiency over combined intervals.

  The original hypothesis had to be rejected since there was no general efficiency gain, but it was shown that efficiency gains can be achieved under certain conditions.

- the higher activity of agents with higher performance in heterogeneous groups was shown to have a significant impact on the efficiency of the group. Results from (Labella et al. 2004) were reproduced and it was shown that the degree of differences between agents directly affects the individual activity. This higher activity of agents with better performance leads to efficiency gains while there is differentiation in the group. This differentiation was also shown to be stable in static environments.

Apart from confirming both hypotheses, it was also shown that

- Depending on the parameter set for the agents and the environment, the range of object densities for which differentiation is possible through the VD algorithm is different. By adjusting parameters that effect success rates of agents, the group can be "tuned" to different ranges.

- Parameters can be classified depending on their effects on the group. The success rate of agents is defined and it is shown that parameters affecting this success rate have an effect on the self-organised activity pattern of the group. Parameters affecting cycle times of individual foraging loops and the proportion an agent spends on given subtasks were shown to have no impact on group coordination.
- It is shown that using P as probability to leave the base or directly calculating idle times following the expected values of the probabilistic mechanism has no impact on the system's behaviour. There was no change in differentiation in the group, nor did it lead to slower adaptation to changes in the environment.

Chapter 5

Conclusion

Designing a group of robots for a given task and implementing a control algorithm which is robust and can be used for different group sizes is no simple undertaking. When designing a group, one of the decisions that has to be made is whether the abilities that are required for the task are built into every agent or whether they can be distributed in the group. This work investigated mainly three hypotheses which were inspired by swarm robotics and analysed in a multi-agent based system. In the first part of this thesis the differences between a homogeneous group of versatile agents and heterogeneous groups of two types of agents are investigated. Heterogeneous groups were composed of different ratios of the two types and used in the same clustering tasks as a group of versatile agents of the same size.

For the heterogeneous groups, it was found that changes in group composition do not directly lead to changes in the behaviour as group. As reported in section 3.2, stable behaviour was always found over a range of group compositions which means that the group showed reliable behaviour within this range in terms of robustness to change in agent numbers. Significant changes in behaviour were expected when approaching group compositions near the extremes, towards homogeneity, but this held true only for the lower bound (0% Grabbers) and the upper bound in the enclosed environment. In the case of a toroidal environment, the pattern converged smoothly towards the Grabber pattern when approaching a homogeneous Grabber setting.

In order to compare heterogeneous to homogeneous groups, a versatile agent was introduced that was able to switch between the two behaviours that the specialised agents exhibited. It was shown that no heterogeneous group could be identified which created the same pattern as the homogeneous group since the ability of individual agents to switch between behaviours can lead to results that a group of two types of specialists, although being able to exhibit both individual behaviours within the group, cannot produce. Even in this simulated set-up, it was not possible to simply split the abilities of a versatile agent between two types of agents so that a mixed group could give the same result. The reasons for that was the ability of the homogeneous group to adapt its "composition" to the current environment, i.e. changing the number of agents exhibiting a particular behaviour. At

any given time step, every agent in the group can be seen as either Dozer or Grabber depending on the behaviour exhibited, but compared to the pre-set groups the ratio can change continuously and adapt to the current state of the environment. This finding led to the rejection of hypothesis 1 for this setting. Nevertheless, it may be possible that a homogeneous group of versatile agents can be substituted by a heterogeneous group of specialists in a different set-up, e.g. when the tasks are completely partitioned or when the separation of behaviours in the versatile agent is more pronounced.

In more complex tasks, the organisation within the group and its adaptation to the current environment is subject of much research. Decentralised, self-organised division of labour, i.e. the self-organised allocation of agents to tasks uses less resources and can be achieved without inter-agent communication. These algorithms do not need much information or resources and generally scale well with group size. In chapter 4 of this thesis one of these approaches, the VD algorithm introduced in Labella et al. (2004), is explored in more detail. The original work showed that the activity of agents can be adjusted using a simple success feedback loop. The algorithm was chosen due to its minimal resource requirements in terms of sensors and computational power and especially because it does not require communication between agents or information sharing between agent and nest.

After the behaviour of agents and the dynamics of the VD adaptation mechanism were qualitatively reproduced in environments with constant object density, it was shown that there was no automatic efficiency gain by by adapting the number of active agents in a group. The major difference between the abstract and original experimental setting was the replacement of objects in the environments. In this work, the object availability was constant which meant that agents leaving the base always had the same chance to encounter an object. In the original work, the objects "regrew" with a certain rate which means that the retrieval of an object by one agent temporarily reduces the chance for other agents to encounter an object. Reducing the number of active agents in such an environment seems to have an impact on efficiency even in homogeneous groups. Nevertheless, it was shown that there can be an efficiency increase even in homogeneous groups in an environment with constant object density. It was shown that in dynamic environments with changing object densities, reducing the number of active agents in a low density environment can lead to an overall efficiency increase. Combining time intervals with no efficiency gains in the individual intervals can still lead to higher efficiency over combined intervals. The original hypothesis 2 had to be rejected since no general efficiency gain could be shown, but it was shown that efficiency gains can be achieved under certain conditions.

Labella et al. (2004) also showed that the VD algorithm lead to robust differentiation and the "selection of the best individual" which was reproduced in this work for heterogeneous groups. It was shown that agents with a high individual performance were more active over time compared to agents with lower performance (in terms of collected cans per time). The degree of difference between two agents had a direct effect on the differences in individual activity, i.e. time spent outside the base area. It was shown that this difference in individual activity is the reason for group efficiency gains in heterogeneous groups due to an

increased activity of agents with higher individual performance or, in turn, the low activity of badly performing agents. Since agents with low individual performance decreased their activity first when object density in the environment decreased, the average performance of the group was increased. The efficiency gains are increasing with increasing degree of heterogeneity in the group. This confirmed hypothesis 3 that stated the efficiency increase would be due to heterogeneity in the group.

Additionally, it was shown that the original probabilistic method to determine when agents leave the base can be exchanged by a direct calculation of idle times without any observable effect. Directly determining the time the agent will spend in the base area leads to more predictable idle times which can be of use when determining whether the agent can execute other tasks while in the base (e.g. recharging a robot when time allows, without changing the groups gathering behaviour). The extension VD* to the original VD adaptation algorithm was validated in the abstract agent-based system but can be of use in robotic settings, where the knowledge about resting times of agents may be used to efficiently combine different tasks or different mechanisms.

5.1 Future Work and Possible Extensions

This thesis has explored a possible mechanism for self-organised division of labour and although results from previous work were replicated and the features of the mechanism explored in more detail, some questions remain unanswered. There are also possible extensions the the VD algorithm which has to be left unexplored in this work.

Confirmation of results with real robots Due to problems with a real robot set-up, the results presented in this thesis could not be confirmed outside simulation, yet. Due to inevitable differences between the abstract simulation and a set-up with real robots, it can not be guaranteed that the findings from this thesis are valid outside the multi-agent, abstract simulation environment used. Due to hardware differences between robots and the difficulties in achieving full homogeneity, the experimental set-up has to be adjusted to account for possible effects of a small degree of heterogeneity.

Efficiency gain in environment with "regrowing" objects It was hypothesised that the efficiency gains in homogeneous groups reported by Labella were due to the differences in object replacement. In this thesis, the density of objects was kept constant and only changed through external intervention. In the original experiments, object were replaced following a regrowth model. A certain number of objects was introduced into the environment over time and the replacement speed was constant. This meant that an agent removing an object from the environment temporarily changed the density for the remaining agents. This difference in the object densities might be responsible for the observed efficiency gains when using the VD algorithm. To test this hypothesis, the simulation used in this thesis has to

be changed to the original replacement mechanism and the experiments with homogeneous groups repeated.

Extension to multiple tasks Currently only the activity of agents for one task is regulated through the VD algorithm. Since most real-world scenarios include multiple tasks, it would be of interest to explore possible extensions to the VD mechanism. Apart from regulating activity for each task, the number of agents engaging in each task could also be regulated by success feedback. The VD mechanism was shown to increase the activity of agents with higher performance for a task. If this effect also holds in a multi-task setting, efficient distribution of agents to tasks in a heterogeneous group could be achieved without communication. The individual agents would not need information on which task they are best suited for.

Dynamic specialisation Following an idea that was already proposed in (Magg and te Boekhorst 2008) the degree of heterogeneity could be changed over the lifetime of a group. As was shown, the degree of heterogeneity has a significant impact on the division of labour in the group. A second feedback loop could be used to change parameters of agents (e.g. speed) over a longer time scale, thus changing the heterogeneity in the group. Since individual success rates can be measured by each agent, this second loop could change parameters in a way to keep those success rates near a desired point. This would in effect change the region of possible differentiation of the group.

Bibliography

Agassounon, W. and Martinoli, A.: 2002, Efficiency and robustness of threshold-based distributed allocation algorithms in multi-agent systems, *AAMAS '02: Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, ACM, New York, NY, USA, pp. 1090–1097.

Arkin, R.: 1987, Motor schema-based mobile robot navigation, *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 264–271.

Arkin, R.: 1992, Cooperation without communication: Multiagent schema-based robot navigation, *Journal of Robotic Systems* **9**(3), 351–364.

Arkin, R., Balch, T. and Nitz, E.: 1993, Communication of behavorial state in multi-agent retrieval tasks, *Robotics and Automation, 1993. Proceedings., 1993 IEEE International Conference on*, IEEE, pp. 588–594.

Baker, G. and Gollub, J.: 1990, *Chaotic dynamics*, Cambridge University Press.

Balch, T.: 1999, The impact of diversity on performance in multi–robot foraging, *AGENTS '99: Proceedings of the third annual conference on Autonomous Agents*, ACM, New York, NY, USA, pp. 92–99.

Balch, T.: 2000, Hierarchic social entropy: An information theoretic measure of robot group diversity, *Autonomous Robots* **8**, 209–237.

Balch, T. and Arkin, R. C.: 1994, Communication in reactive multiagent robotic systems, *Autonomous Robots* **1**, 27–52. 10.1007/BF00735341.

Beckers, R., Holland, O. and Deneubourg, J.: 1994, From local actions to global tasks: Stigmergy and collective robotics, *Artificial life IV*, Vol. 181, p. 189.

Beni, G.: 1988, The concept of cellular robotic system, *Intelligent Control, 1988. Proceedings., IEEE International Symposium on*, pp. 57 –62.

Beni, G.: 2004, From swarm intelligence to swarm robotics, *Swarm Robotics*, pp. 1–9.

Beni, G. and Wang, J.: 1989, Swarm intelligence in cellular robotic systems, *Proceedings of NATO Advanced Workshop on Robots and Biological Systems*, Vol. 102.

Beshers, S., Huang, Z., Oono, Y. and Robinson, G.: 2001, Social inhibition and the regulation of temporal polyethism in honey bees, *Journal of Theoretical Biology* **213**(3), 461–479.

Beshers, S. N. and Fewell, J. H.: 2001, Models of division of labor in social insects, *Annu Rev Entomol* **46**, 413–440. 0066-4170 Journal Article Review.

Bonabeau, E., Dorigo, M. and Theraulaz, G.: 1999, *From Natural to Artificial Swarm Intelligence*, Oxford University Press.

Braitenberg, V.: 1986, *Vehicles: Experiments in synthetic psychology*, The MIT press.

Brooks, R. A.: 1986, A robust layered control system for a mobile robot, *Robotics and Automation, IEEE Journal of* **2**(1), 14–23.

Brooks, R. A.: 1991, Intelligence without representation, *Artificial Intelligence* **47**(1-3), 139–159.

Brooks, R., Maes, P., Mataric, M. and More, G.: 1990, Lunar base construction robots, *Intelligent Robots and Systems' 90.'Towards a New Frontier of Applications', Proceedings. IROS'90. IEEE International Workshop on*, IEEE, pp. 389–392.

Caloud, P., Choi, W., Latombe, J. C., Pape, C. L. and Yin, M.: 1990, Indoor automation with many mobile robots, *International Conference on Intelligent Robots and Systems*.

Camazine, S., Deneubourg, J.-L., Franks, N., Sneyd, J., Theraulaz, G. and Bonabeau., E.: 2001, *Self-Organisation in Biological Systems*, Princeton University Press, NJ, USA.

Cao, Y. U., Fukunaga, A. S. and Kahng, A. B.: 1997, Cooperative mobile robotics: Antecedents and directions, *Autonomous Robots* **4**, 226–234.

Şahin, E.: 2005, Swarm robotics: From sources of inspiration to domains of application, *in* E. Sahin and W. Spears (eds), *Swarm Robotics*, Vol. 3342 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 10–20.

Deneubourg, J., Goss, S., Franks, N., Sendova-Franks, A., Detrain, C. and Chrétien, L.: 1991, The dynamics of collective sorting robot-like ants and ant-like robots, *Proceedings of the first international conference on simulation of adaptive behavior on From animals to animats*, pp. 356–363.

Deneubourg, J.-L., Goss, S., Pasteels, J. M., Fresneau, D. and Lachaud, J.-P.: 1987, Self-organization mechanisms in ant societies (ii): Learning in foraging and division of labor, *in* J. M. Pasteels and J.-L. Deneubourg (eds), *From Individual Characteristics to Collective Organisation: The Example of Social Insects*, Vol. 54 of *Experientia supplementum*, Birkhäuser, Basel, Switzerland, pp. 177–196.

Dorigo, M., Maniezzo, V., Colorni, A., Dorigo, M., Dorigo, M., Maniezzo, V., Maniezzo, V., Colorni, A. and Colorni, A.: 1991, Positive feedback as a search strategy, *Technical Report 91-016*, Politecnico di Milano, Italy.

Dorigo, M. and Stützle, T.: 2004, *Ant Colony Optimization*, Bradford Company, Scituate, MA, USA.

Dudek, G., Jenkin, M., Milios, E. and Wilkes, D.: 1996, A taxonomy for multi-agent robotics, *Autonomous Robots* **3**(4), 375–397.

Dudenhoeffer, D. D. and Jones, M. P.: 2000, A formation behavior for large-scale micro-robot force deployment, *Proceedings of the 32nd conference on Winter simulation*, WSC '00, Society for Computer Simulation International, San Diego, CA, USA, pp. 972–982.

Eberhart, R. C. and Kennedy, J.: 1995, A new optimizer using particle swarm theory, *Proceedings of the Sixth International Symposium on Micromachine and Human Science*, Nagoya, Japan, pp. 39–43.

Feller, W.: 1968, *An Introduction to Probability Theory and Its Applications. Volume I.*, Wiley, John & Sons, Inc.

Garnier, S., Gautrais, J. and Theraulaz, G.: 2007, The biological principles of swarm intelligence, *Swarm Intelligence* **1**, 3–31. 10.1007/s11721-007-0004-y.

Gerkey, B. P. and Matarić, M. J.: 2004, A formal analysis and taxonomy of task allocation in multi-robot systems, *International Journal of Robotic Research* **23**(9), 939–954.

Goldberg, D. and Matarić, M. J.: 2003, Maximizing reward in a non-stationary mobile robot environment, *Autonomous Agents and Multi-Agent Systems* **6**, 287–316. 10.1023/A:1022935725296.

Grassé, P.: 1959, La reconstruction du nid et les coordinations interindividuelles chezbellicositermes natalensis etcubitermes sp. la théorie de la stigmergie: Essai d'interprétation du comportement des termites constructeurs, *Insectes sociaux* **6**(1), 41–80.

Groß, R. and Dorigo, M.: 2004, Group transport of an object to a target that only some group members may sense, *in* X. Yao, E. Burke, J. A. Lozano, J. Smith, J. J. Merelo-Guervós, J. A. Bullinaria, J. Rowe, P. Tiňo, A. Kabán and H.-P. Schwefel (eds), *Parallel Problem Solving from Nature - 8th International Conference - PPSN VIII*, Vol. 3242 of *Lecture Notes in Computer Science*, Springer Verlag, Berlin, Germany, pp. 852–861.

Groß, R., Tuci, E., Dorigo, M., Bonani, M. and Mondada, F.: 2006, Object transport by modular robots that self-assemble, *Proceedings of the 2006 IEEE Int. Conf. on Robotics and Automation, ICRA 2006*, IEEE Computer Society Press, Los Alamitos, CA, pp. 2558–2564.

Holland, O. and Melhuish, C.: 1999, Stigmergy, self-organization, and sorting in collective robotics, *Artificial Life* **5**(2), 173–202.

Howard, A., Parker, L. E. and Sukhatme, G. S.: 2006, Experiments with a large heterogeneous mobile robot team: Exploration, mapping, deployment and detection, *Int. J. Rob. Res.* **25**(5-6), 431–447.

Ijspeert, A. J., Martinoli, A., Billard, A. and Gambardella, L. M.: 2001, Collaboration through the exploitation of local interactions in autonomous collective robotics: The stick pulling experiment, *Auton. Robots* **11**(2), 149–171.

Jardine, N. and Sibson, R.: 1971, *Mathematical taxonomy [by] Nicholas Jardine and Robin Sibson*, Wiley London, New York,.

Kennedy, J. and Eberhart, R. C.: 1995, Particle swarm optimization, *Proceedings of IEEE International Conference on Neural Networks*, Piscataway, NJ, pp. 1942–1948.

Krieger, M. J. B. and Billeter, J.-B.: 2000, The call of duty: Self-organised task allocation in a population of up to twelve mobile robots, *Robotics and Autonomous Systems* **30**(1-2), 65–84.

Kube, C. R. and Zhang, H.: 1993, Collective robotics: From social insects to robots, *Adaptive Behavior* **2**(2), 189–218.

Kube, R. C. and Zhang, H.: 1992, Collective robot intelligence, *Proceedings of the second international conference on From animals to animats 2 : simulation of adaptive behavior*, MIT Press, Cambridge, MA, USA, pp. 460–468.

Labella, T., Dorigo, M. and Deneubourg, J.-L.: 2004, Efficiency and task allocation in prey retrieval, *in* A. Ijspeert, D. Mange, M. Murata and S. Nishio (eds), *Proceedings of the First International Workshop on Biologically Inspired Approaches to Advanced Information Technology (Bio-ADIT2004)*, Lecture Notes in Computer Science, Springer Verlag, Heidelberg, Germany, pp. 32–47.

Labella, T. H.: 2007, *Division of Labour in Groups of Robots*, PhD thesis, Universite Libre de Bruxelles.

Lein, A. and Vaughan, R. T.: 2008, Adaptive multirobot bucket brigade foraging, *In Proceedings of the Eleventh International Conference on Artificial Life (ALife XI*, MIT Press, pp. 337–342.

Liu, W.: 2008, *Design and modelling of Adaptive Foraging in Swarm Robotic Systems*, PhD thesis, Univeristy of the West of England, Bristol, UK.

Liu, W., Winfield, A. F. T., Sa, J., Chen, J. and Dou, L.: 2007, Towards energy optimization: Emergent task allocation in a swarm of foraging robots, *Adaptive Behavior* **15**(3), 289–305.

Magg and te Boekhorst, R.: 2008, Task allocation by dynamic specialisation, *German Workshop on Artificial Life*, Leipzig, Germany, pp. 91–100.

Maris, M. and te Boekhorst, R.: 1996, Exploiting physical constraints: Heap formation through behavioral error in a group of robots, *IROS'96, IEEE/RSJ International Conference on Intelligent Robots and Systems*, Senri Life Science Center, Osaka, Japan.

Martinoli, A., Franzi, E. and Matthey, O.: 1998, Towards a reliable set-up for bio-inspired collective experiments with real robots, *Experimental Robotics V* pp. 595–608.

Matarić, M.: 1992, Minimizing complexity in controlling a mobile robot population, *International Conference on Robotics and Automation*.

Melhuish, C., Holland, O., Hoddell, S. and Lane, C.: 1998, Collective sorting and segregation in robots with minimal sensing, *From Animals to Animats 5: Proceedings of the Fifth International Conference on Simulation of Adaptive Behavior*, MIT Press, Cambridge, MA, pp. 465–470.

Melhuish, C., Wilson, M. and Sendova-Franks, A.: 2001, Patch sorting: multi-object clustering using minimalist robots, *Advances in Artificial Life* pp. 543–552.

Nilsson, N. J.: 1984, Shakey the robot, *Technical Report 323*, AI Center, SRI International, 333 Ravenswood Ave., Menlo Park, CA 94025.

Noreils, F.: 1990, Integrating multirobot coordination in a mobile-robot control system, *Intelligent Robots and Systems '90. 'Towards a New Frontier of Applications', Proceedings. IROS '90. IEEE International Workshop on*, Vol. 1, pp. 43–49.

Ostergaard, E., Sukhatme, G. and Matari, M.: 2001, Emergent bucket brigading: a simple mechanisms for improving performance in multi-robot constrained-space foraging tasks, *Proceedings of the fifth international conference on Autonomous agents*, ACM, pp. 29–30.

Parker, L.: 1994, *Heterogeneous Multi-Robot Cooperation*, PhD thesis, MIT EECS Department.

Parker, L.: 1998, Alliance: an architecture for fault tolerant multirobot cooperation, *Robotics and Automation, IEEE Transactions on* **14**(2), 220–240.

Schneider-Fontán, M. and Matarić, M.: 1998, Territorial multi-robot task division, *Robotics and Automation, IEEE Transactions on* **14**(5), 815–822.

Sendova-Franks, A. and Franks, N.: 1993, Task allocation in ant colonies within variable environments (a study of temporal polyethism: experimental), *Bulletin of mathematical biology* **55**(1), 75–96.

Sharkey, A. J. C.: 2006, Robots, insects and swarm intelligence, *Artificial Intelligence Review* **26**(4), 255–268.

Simmons, R., Singh, S., Hershberger, D., Ramos, J. and Smith, T.: 2000, Coordination of heterogeneous robots for large-scale assembly, *Proceedings of the International Symposium on Experimental Robotics (ISER), Hawaii*.

Steels, L.: 1990, Cooperation between distributed agents through self-organisation, *Intelligent Robots and Systems '90. 'Towards a New Frontier of Applications', Proceedings. IROS '90. IEEE International Workshop on*, pp. 8–14 supl.

Theraulaz, G., Bonabeau, E. and Deneubourg, J.: 1998, Response threshold reinforcements and division of labour in insect societies, *Proceedings of the Royal Society of London. Series B: Biological Sciences* **265**(1393), 327.

Webb, B.: 2001, Can robots make good models of biological behaviour?, *Behavioral and brain sciences* **24**(06), 1033–1050.

Wilson, M., Melhuish, C., Sendova-Franks, A. and Scholes, S.: 2004, Algorithms for building annular structures with minimalist robots inspired by brood sorting in ant colonies, *Autonomous Robots* **17**(2), 115–136.

Winfield, A. F.: 2009, Foraging robots, *Encyclopedia of Complexity and System Science* pp. 3682–3700.