Universite Libre de Bruxelles
Universite d'Europe
Faculté des Sciences Appliquées
IRIDIA, Institut de Recherches
Interdisciplinaires
et de Développements en Intelligence Artificielle

Teamwork in a Swarm of Robots – An Experiment in Search and Retrieval

Shervin Nouyan

Promoteur:
Prof. Marco Dorigo

Thése présentée en vue de l'obtention du
titre de Docteur en Sciences Appliquées

Année académique 2007–2008

Abstract

In this thesis, we investigate the problem of path formation and prey retrieval in a swarm of robots. We present two swarm intelligence control mechanisms used for distributed robot path formation. In the first, the robots form linear *chains*. We study three variants of robot chains, which vary in the degree of motion allowed to the chain structure. The second mechanism is called *vectorfield*. In this case, the robots form a pattern that globally indicates the direction towards a goal or home location. Both algorithms were designed following the swarm robotics control principles: simplicity of control, locality of sensing and communication, homogeneity and distributedness.

We test each controller on a task that consists in forming a path between two objects—the *prey* and the *nest*—and to retrieve the prey to the nest. The difficulty of the task is given by four constraints. First, the prey requires concurrent, physical handling by multiple robots to be moved. Second, each robot's perceptual range is small when compared to the distance between the nest and the prey; moreover, perception is unreliable. Third, no robot has any explicit knowledge about the environment beyond its perceptual range. Fourth, communication among robots is unreliable and limited to a small set of simple signals that are locally broadcast.

In simulation experiments we test our controllers under a wide range of conditions, changing the distance between nest and prey, varying the number of robots used, and introducing different obstacle configurations in the environment. Furthermore, we tested the controllers for robustness by adding noise to the different sensors, and for fault tolerance by completely removing a sensor or actuator. We validate the chain controller in experiments with up to twelve physical robots. We believe that these experiments are among the most sophisticated examples of self-organisation in robotics to date.

Acknowledgments

The work presented in this thesis was financially supported by the SWARM-BOTS project, a project funded by the Future and Emerging Technologies programme of the European Commission, under grant IST-2000-31010.[†] The SWARM-BOTS project started in October 2001, and I joined the project in October 2002, sharing with many valuable researchers the opportunity to propose innovative ideas, study new solutions and see them eventually realised. In addition to the SWARM-BOTS project, I was supported by the "ANTS" project, an "Action de Recherche Concertée" funded by the Scientific Research Directorate of the French Community of Belgium.

Apart from the financial support, this thesis could not have been finished without the support of many people. Above all, I wish to thank my supervisor, Professor Marco Dorigo (papa P). Our first contact was established approximately one year before the start of the Ph.D., when I was still studying in Germany and looking for a place abroad to produce my Master Thesis. With a quick and uncomplicated phone call everything was settled and he offered me a computer, a desk and an interesting project to work on. Since that day I have got to know Professor Dorigo as a person that is always honest, open, straightforward, and above all a very good advisor in both professional and personal situations.

Professor Dorigo gave me the possibility to work at the 'Institut de Recherches Interdisciplinaires et de Développements en Intelligence Artificielle' (IRIDIA), which will always keep a special place in my heart. At IRIDIA I had the opportunity to meet great people from various countries and cultures, some of which I consider as close friends now. First of all, I wish to thank my fellow colleagues from the robotics group: Alexandre Campo (crazy P), who has the most exceptional rhythm of all people I know, Anders Christensen, Carlo Pinciroli, Christos Ampatzis (BTP), who is as good a cook as a friend (despite his female characteristics), Elio Tuci (HP), who is always honest in expressing his opinion, Erol Şahin, who always lost against me in Ping-Pong when I needed some motivation during my Master Thesis, Giovanni

[†]The information provided in this thesis is the sole responsibility of the authors and does not reflect the Community's opinion. The Community is not responsible for any use that might be made of data appearing in this document.

Pini, who is a great person to argue with, Halva Labella (Mama P), the biggest teddy bear alive, Rehan O'Grady, Roderich Groß (Alcoholic P), who is helpful and friendly, although very ambitious, Vito Trianni (sublime stinky P), a very good friend (and cook) with whom I shared lots of joy and beer, and who eventually gave in to my superior minigolf skills. Among other fellow iridians, I wish to thank Álvaro Gutiérrez, Bruno Marchal, a beautiful mind with the most wonderful English accent, Carlotta Piscopo, the flower of IRIDIA, Christophe Philemotte, Federico Vicentini, Hugues Bersini, Joshua Knowles, Jean-Louis Deneubourg, who gave invaluable insights into the secret world of an ant's brain (in comparison to the stupid robot controllers we developed), Marco Montes, who is always helpful and never stressed, Mauro Birattari, the analytical mind, Max Manfrin, the most helpful person I ever met, Muriel Decreton, Paola Pellegrini, Prasanna Balaprakash, who always smiles, Tom Lenaerts, who gives great advice with a very positive and enthusiastic attitude, Thomas Stützle, my favourite sex symbol Xico, who never stops to impress me with his lovely mind and his extraordinary achievements (especially concerning the coffee machine).

Among the projects I worked on at IRIDIA one of my favourites was definitely the e-puck student project. I enjoyed a lot to work with Antoine, Francesco, Giovanni, Mouhcine, Laurent and Olivier (thanks again for all the boxes).

I also wish to thank the proof-readers who (hopefully) found all the little typos in my thesis: Alex, Christos, Giovanni, Halva, Marco Montes, Prasanna, Rodi and Vito. Thanks a lot for your help.

Among the non-work related people I met during my stay in Brussels I wish to thank most of all my football buddies: Ben, Bram, Chris, David, Joris, Laurent, Quoc, Sam, Sven, Yves and Wim. I loved to play football with you guys and I loved at least as much to share one or many beers with you afterwards.

Of course, I also wish to thank my family. Thank you Mami, Shadi, La, Dai and Chale. It's really great to have you!

Last but not least, I want to thank my own little family: Isabelle and Reza. You two are the most important people in my life. Thank you for teaching me what really counts in life, for your love and for being there.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this first chapter, we start by discussing the content of this thesis and specify the contributions and the related publications in Section 1.1. A brief overview of the context is given in Section 1.2, where we discuss the concepts of robots, swarms and teams. Finally, in Section 1.3 we present the thesis layout.

1.1 Content

In this thesis, we study the behaviour of large groups of robots while performing a task that requires path formation, self-assembly and group transport. We emphasize the cooperation and collectivity of the robot group, and rely on principles such as simplicity, homogeneity, distributedness of control, and locality of communication and information. While the algorithms used for path formation are original contributions, those used for self-assembly and group transport have originally been developed by Groß et al. (2006a,b), and were modified in order to integrate them into one controller.

In the considered problem, the robots are initially randomly scattered in a bounded arena that contains two objects—the *prey* and the *nest*. The task is to retrieve the prey to the nest. The following constraints are given:

- C_1: the prey requires concurrent, physical handling by multiple robots to be moved,
- C_2: each robot's perceptual range is small when compared to the distance between the nest and the prey; moreover, perception is unreliable,
- C_3: no robot has any (explicit) knowledge about the environment beyond its perceptual range,

- C_4: communication among robots is unreliable and limited to a small set of simple signals that are locally broadcast.

These constraints have implications on the division of labour within the group. Some robots are required to engage in the physical handling of the prey (constraint C_1). To do so, they self-assemble into physically connected pulling structures. While pulling the prey, the robots neither perceive the nest (constraint C_2) nor have any knowledge about its location in the environment (constraint C_3). In principle, they could transport the prey in a random direction. However, this has no practical value in large arenas or open space. Our solution to the problem is to have some robots establish a path between the nest and the prey (constraints C_3 and C_4), along which the transport is guided towards the nest.

1.1.1 Original Contributions and Related Publications

There are three main contributions of this thesis. First, we propose two novel control mechanisms for navigation and path formation in a swarm of robots: chains with cyclic directional patterns and vectorfield. Both control mechanisms emphasize the use of local information and communication, and the behaviours are achieved by following simple rules rather than relying on complex mechanisms such as building up a representation of the environment (e.g. a map). Similarities and differences from other approaches are discussed along with the related work in Chapter 2. Second, we conduct an extensive analysis to compare the two mechanisms, and to show that they achieve what is often claimed as a major motivation to use swarm robotics control mechanisms: a high degree of scalability, robustness, and fault tolerance. The results from simulation with these two control mechanisms are presented in Chapters 5 and 8 and were in part published in:

- S. Nouyan. Path formation and goal search in swarm robotics. Technical Report TR/IRIDIA/2004-14, Université Libre de Bruxelles, Belgium, September 2004. DEA Thesis

- M. Dorigo, E. Tuci, R. Groß, V. Trianni, T. H. Labella, S. Nouyan, C. Ampatzis, J.-L. Deneubourg, G. Baldassarre, S. Nolfi, F. Mondada, D. Floreano, and L. M. Gambardella. The SWARM-BOTS project. In E. Sahin and W.M. Spears, editors, *Swarm Robotics: SAB 2004 International Workshop*, volume 3342 of *Lecture Notes in Computer Science*, pages 31–33. Springer Verlag, Berlin, Germany, 2004

- S. Nouyan and M. Dorigo. Chain based path formation in swarms of robots. In M. Dorigo et al., editors, *Ant Colony Optimization and Swarm Intelligence: 5th International Workshop, ANTS 2006*, volume 4150 of *Lecture Notes in Computer Science*, pages 120–131. Springer Verlag, Berlin, Germany, 2006

- M. Dorigo, E. Tuci, V. Trianni, R. Groß, S. Nouyan, C. Ampatzis, T. H. Labella, R. O'Grady, M. Bonani, and F. Mondada. SWARM-BOT: Design and implementation of colonies of self-assembling robots. In G. Y. Yen and D. B. Fogel, editors, *Computational Intelligence: Principles and Practice*, chapter 6, pages 103–135. IEEE Computational Intelligence Society, New York, 2006

- S. Nouyan, A. Campo, and M. Dorigo. Path formation in a robot swarm. *Swarm Intelligence*, 2(1), in press

The mechanism to indicate a direction was originally used in the work on negotiation of a goal direction for the transport of heavy objects and published in:

- A. Campo, S. Nouyan, M. Birattari, R. Groß, and M. Dorigo. Negotiation of goal direction for cooperative transport. In M. Dorigo et al., editors, *Ant Colony Optimization and Swarm Intelligence: 5th International Workshop, ANTS 2006*, volume 4150 of *Lecture Notes in Computer Science*, pages 191–202. Springer Verlag, Berlin, Germany, 2006b

- A. Campo, S. Nouyan, M. Birattari, R. Groß, , and M. Dorigo. Negotiation of goal direction for cooperative transport. In *Proceedings of the 18th Belgium-Netherlands Conference on Artificial Intelligence*, pages 365–366. University of Namur, Namur, Belgium, 2006a

The third contribution lies in the integration of different controllers to form a path between a nest and a prey, to assemble to the prey, and to transport it back to the nest. To the best of our knowledge, the three different tasks of path formation, self-assembly, and group transport have been tackled only separately with real robots so far. We present the first attempt to solve these three tasks as parts of an integrated scenario, using a robot team that is homogeneous both in hardware and control. This work is based on the chaining algorithm used for path formation, and a modified version of the controllers for self-assembly and group transport developed by Groß et al. (2006a,b). The results of the integration of these three controllers are presented in Chapter 6 and were published in:

- S. Nouyan, R. Groß, M. Bonani, F. Mondada, and M. Dorigo. Group transport along a robot chain in a self-organised robot colony. In *Proc. of the* 9^{th} *Int. Conf. on Intelligent Autonomous Systems*, pages 433–442. IOS Press, Amsterdam, The Netherlands, 2006

- R. Groß, S. Nouyan, M. Bonani, F. Mondada, and M. Dorigo. Division of labour in self-organised groups. In *From Animals to Animats 10. Proceedings of the Tenth International Conference on Simulation of Adaptive Behavior (SAB08)*, pages 426–436. Springer Verlag, Berlin, Germany, 2008

- S. Nouyan, R. Groß, M. Bonani, F. Mondada, and M. Dorigo. Teamwork in self-organised robot colonies. *IEEE Transactions on Evolutionary Computation*, accepted for publication

Finally, even if not reported in this thesis, we have conducted research in the area of scheduling problems using a distributed swarm intelligence algorithm based on the methodology of division of labour in social insects, resulting in the following publications:

- S. Nouyan. Agent-based approach to dynamic task allocation. In M. Dorigo, G. Di Caro, and M. Sampels, editors, *Ant Colony Optimization and Swarm Intelligence: 3rd International Workshop, ANTS 2002*, volume 2463 of *Lecture Notes in Computer Science*, pages 28–39. Springer Verlag, Berlin, Germany, 2002

- S. Nouyan, R. Ghizzioli, M. Birattari, and M. Dorigo. An insect-based algorithm for the dynamic task allocation problem. *Künstliche Intelligenz*, 4(5): 25–31, 2005

1.2 Context

The first part of the thesis' title contains three concepts: robots, swarms and teamwork. In this section, we will discuss these three concepts and the motivations that led us to study them. The first concept, *robots*, denotes the general area of research within which we conduct our work. A complete overview of robotics would go beyond the scope of this work. Instead, we briefly give a historical background and shortly discuss areas of application in which robots are used nowadays in Section 1.2.1. In particular, we are interested in studying the control of *swarms of robots*, which leads us to *swarms*, the second concept. Swarms and swarm robotics are the subject of Section 1.2.2. Finally, the concept *teamwork* denotes the structural organisation of individuals when performing a task. This organisation and the particular task we study are explained in Section 1.2.3.

1.2.1 Robots and Robotics

Historically the term *robot* comes from the Slavonic word *robota*, which literally means "work". It became known to the public by the play "*Rossum's Universal Robots*", written by Karel Capek in 1920. In this context, the term robot was used for artificially created people that were built to serve, which is closer to the modern idea of androids. There is in general no particular definition of robot that satisfies everyone. However, a robot typically has some or all of the following properties:

- it is not "natural", that is, it is artificially created,

- it can sense its environment, and manipulate or interact with things in it,
- it has some ability to make choices based on its perception of the environment, often using automatic control or a pre-programmed sequence,
- it moves with one or more axes of rotation or translation,
- it appears to have intent.

The first robot of the modern time was a humanoid robot known as *Elektro*, built by the Westinghouse Electric Corporation in the late 1930's (see Figure 1.1a). It was built for the New York World Fair in 1939, was over 2 m tall and weighed 120 kg. It could walk by voice command, talk, smoke cigarettes, blow up balloons, and move its head and arms.

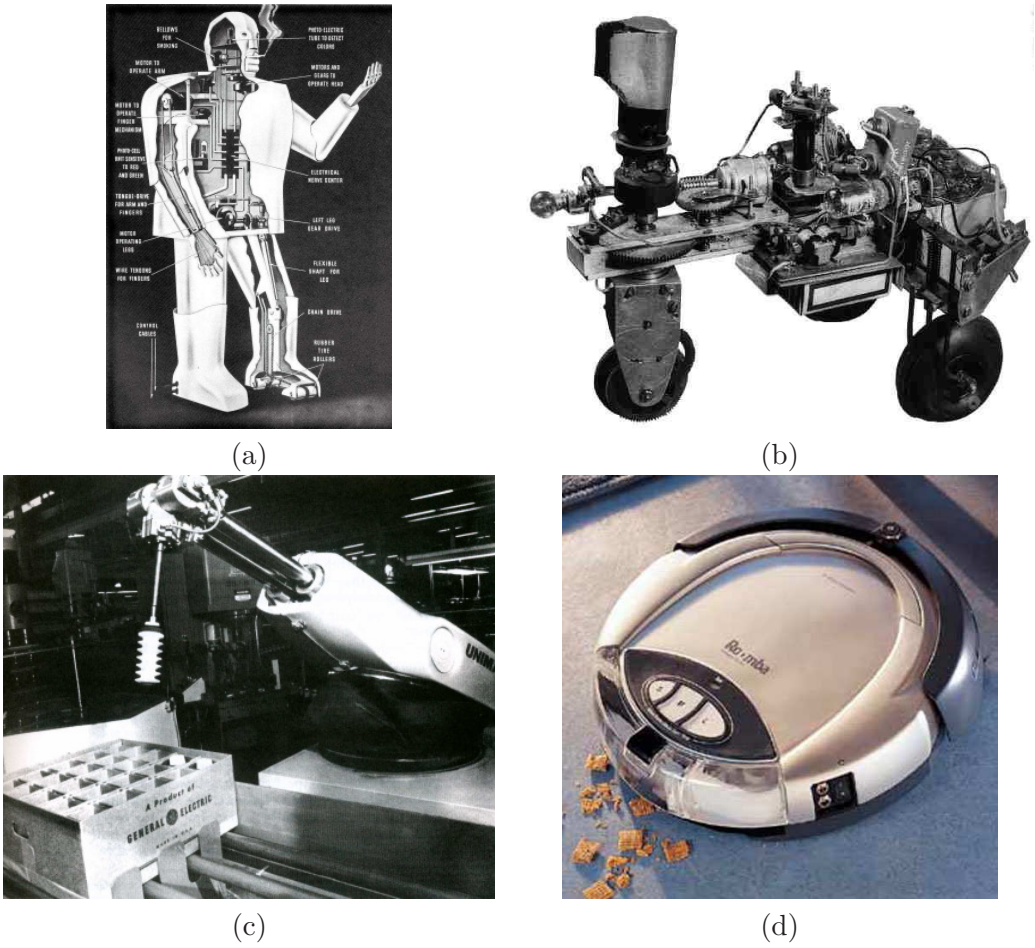
Around one decade later, William Grey Walter of the Burden Neurological Institute at Bristol created the first electronic autonomous robots named *Elmer and Elsie* (see Figure 1.1b), which could sense light and contacts with objects, and use this sensory information to navigate.

The first industrial robot was built by George Devol in 1954 and was called Unimate (see Figure 1.1c). The first Unimate was sold to General Motors in 1960 and was used to lift hot pieces of metal.

As the performance and computational ability of robots rises, and their price falls, they become affordable for a wide range of applications. For instance, robots are used in military as unmanned autonomous vehicles. The DARPA Grand Challenge, organized by the research division of the Pentagon, is stimulating researchers and car companies to develop cars that can drive autonomously in cities. In medicine—even if not completely autonomous—robots are controlled remotely to conduct surgery at a level of precision far beyond the one that can be reached by a human being. In Japan, robots are starting to be used to take care of children and of the elderly. The latter is particularly important given the demographic change with ageing populations in most developed countries. Robots are also conquering everybody's home. With more than 2 million copies, the individually most sold robot to date is a domestic vacuum cleaner robot called *Roomba* (see Figure 1.1d), which was built by the company iRobot at the beginning of this decade.

1.2.2 Swarms and Swarm Robotics

In this thesis, we make use of a particular robot, called *s-bot*. Due to its innovative features and design, it can be used for the study of *swarm robotics*. One of the most innovative features of the *s-bot* is its abilility to connect to other robots, as displayed in Figure 1.2. Before explaining what swarm robotics exactly denotes, let us discuss the term *swarms*, which is the second important concept given in this thesis' title.



(a) (b)

(c) (d)

Figure 1.1: Historical robots: (a) Elektro, a humanoid robot built in the 1930's for the world fair, (b) Elsie, the first electronic autonomous robot built in 1948, (c) Unimate, the first industrial robot built in 1954, (d) Roomba, a domestic vacuum cleaner robot built by iRobot which is, with over two million copies, the most sold robot to date.

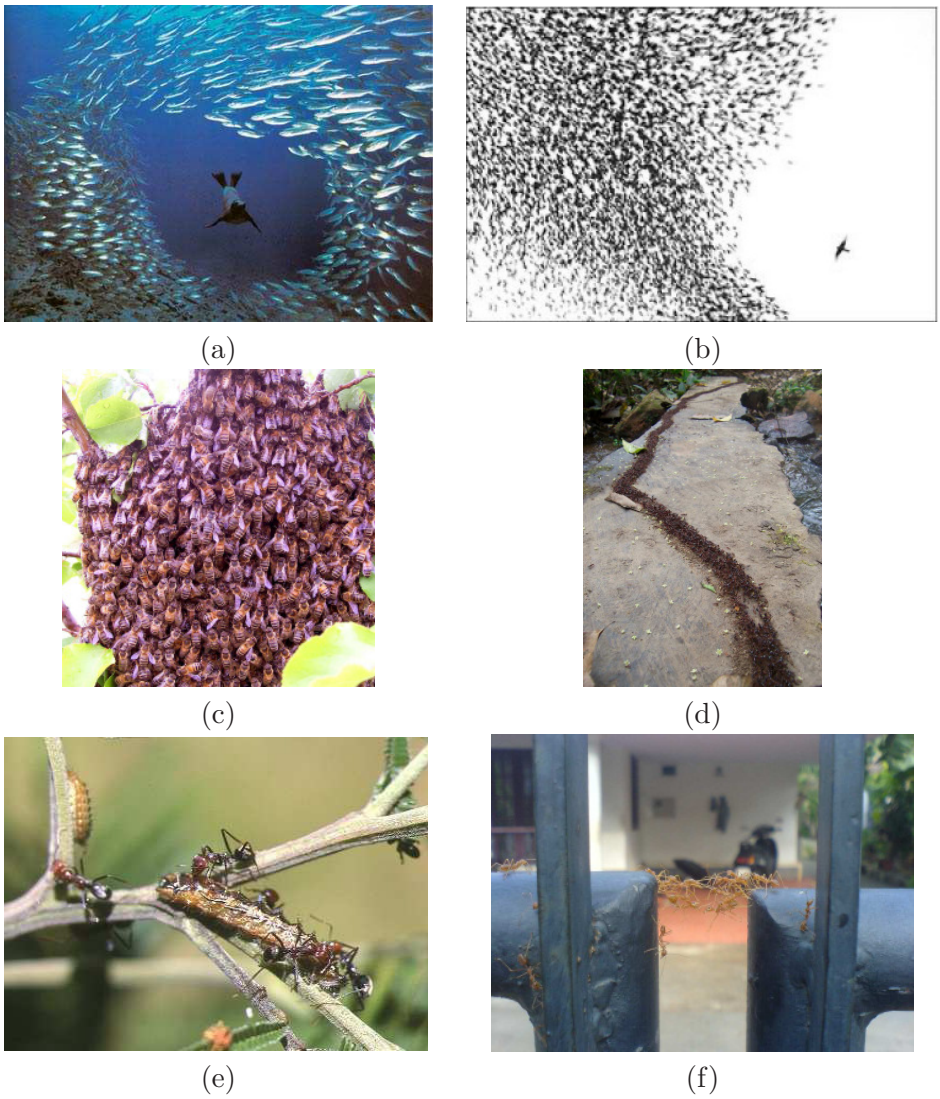
Figure 1.2: A group of *s-bots* performing self-assembly. The *s-bot* is the robot platform used in this thesis. The image was exposed during approximately 37 sec. In this way the image can show the movement of three robots while attempting to form a physical connection with the robot on the right.

The term swarm is used to refer to large groups of individuals that behave coherently and form a clearly distinguishable unit. Examples are social insect colonies, fish schools, bird flocks and microorganisms such as bacteria. Group size is a major aspect when identifying a group of individuals as a swarm. In addition to a large group size, it is the behaviour of the individuals and of the group as a whole that is important. In a swarm everybody is equal. Even though there is no hierarchy, no leader and no centralization, swarms of animals are highly organized and act like single entities with abilities that go far beyond those of the single individual. Very interestingly, this is achieved by following very simple and above all local rules. For instance, the collective movement in a fish school or bird flock, as those displayed in Figure 1.3a and b, is performed through local coordination of an individual with its neighbours. When a predator approaches, it is locally avoided by the individuals that are closest to it, in this way forming a hole in the structure, and collectively avoiding the predator. To move in group is an effective strategy to protect from predators, who struggle to catch an individual because it can only see the swarm as a whole, but has difficulties to identify an individual.

In order to coordinate their activities, swarms can exploit a particular form of indirect communication called *stigmergy*. Stigmergy is usually based on modifications of the environment. These modifications lead to a positive or negative feedback to other individuals. A typical example is given by several termite species building a large structure in an environment with some building material consisting of pellets of soil and excrements without any order (Wilson, 1971; Camazine et al., 2001). Different phases of building up a structure can be distinguished. At the beginning the workers pass through a state in which their work seems rather uncoordinated. A pellet placed at one position by a worker is often quickly picked up by another worker. After some time, seemingly by chance, some pellets get stuck on top of each other and thereby the behaviour of the workers changes very fast. The little cluster of pellets is much more attractive to the termites than single pellets, so that they quickly begin to add more pellets to that cluster. Clusters become pillars, then walls, then chambers, until the nest is built. In this way the environment is "cleaned" from single pellets and large structures are built up without any direct communication among the individuals, but only by modifications of the environment.

The term stigmergy (from the Greek *stigma*: sting, and *ergon*: work) was introduced by Grassé (1959). Grassé describes the indirect communication that he observed in two species of termites: *Bellicositermes natalensis* and *Cubitermes*. His original definition of stigmergy was: "Stimulation of workers by the performance they have achieved".

Another example of indirect communication can be found in the mechanism that leads to prey retrieval in ants. When foraging for food, ants of many species lay trails of pheromone, a chemical substance that attracts other ants. If the ants are pre-



(a) (b) (c) (d) (e) (f)

Figure 1.3: Swarms in nature: (a) a fish school, (b) a bird flock, (c) a swarm of bees, (d) an ant "highway", (e) a group of ants performing collective transport, and (f) a group of ants forming a bridge.

sented multiple paths of different lengths to a food source, the laying of pheromone trails allows them to select the shortest path, as was shown by Deneubourg et al. (1990). In their experiment the ants are presented one short and one long path. At the beginning, the ants choose either of the two paths with equal probability. After some time, the shorter of the two paths has a higher concentration of pheromone because the ants pass it faster. Therefore, the shorter path is selected in most cases. This collective decision-making process is achieved through *self-organization*, which is at heart of a wide range of collective behaviours in social insects. Garnier et al. (2007) name four basic ingredients that self-organization relies on:

i. Positive feedback that results from the execution of simple and local behavioural rules. In the case of the pheromone trails, the positive feedback can be found in the attraction of other ants to follow the trail.

ii. Negative feedback, that leads to stabilization. In the case of the pheromone trails sources of negative feedback are for instance the limited number of foragers or the evaporation of pheromone.

iii. Randomness of the individual decision making process, that allows the colony to find new solutions.

iv. Multiple direct or stigmergic interactions among individuals that lead to the appearance of a global and enduring structure.

In robotics, by benefiting from the development of ever cheaper and smaller components, the study of multi-robot systems has received increasing attention over the last few decades. Using a group of robots instead of a single one can have several advantages, such as increase in capabilities or efficiency, or increase of redundancy and fault tolerance. However, also new challenges arise. For example, when the number of robots becomes large, traditional approaches that rely on a centralised management of the robots' activities and on excessive information exchange rapidly reach limits, for instance, because of the risk of individual failure or of limits in the communication bandwidth.

To overcome similar problems, researchers in robotics begun to draw inspiration from decentralised, self-organising biological systems in general and from the collective behaviour of social insects in particular. Giving birth to the swarm robotics domain, swarm robotic systems are typically composed of robots that, at the individual level, offer relatively limited task solving abilities and that have only limited knowledge about their environment. Still, the overall system can exhibit complex behaviour. This is realized in a bottom-up fashion; complexity arises from numerous interactions among the robots and between the robots and their environment. The general paradigm is often referred to as *swarm intelligence* (Bonabeau et al., 1999; Garnier et al., 2007; Dorigo and Birattari, 2007).

Presently, little is known about how to design swarm intelligence systems. Thus, it is not surprising that the complexity exhibited in current implementations does neither come close to the complexity of biological systems, nor does it come close to the complexity of systems men built following the more traditional top-down approach. Dorigo and Şahin (2004) introduce the topic of swarm robotics in the following way:

> Swarm robotics can be loosely defined as the study of how collectively intelligent behaviour can emerge from local interactions of a large number of relatively simple physically embodied agents. Swarm robotics studies are often inspired by the observation of social insects—ants, termites, wasps and bees—which stand as fascinating examples of how collectively intelligent systems can be generated from a large number of simple individuals.

They identify four criteria to distinguish swarm robotics research from other multi-robot studies:

i. Large number of robots: Swarm robotics aims at studying the coordination of large numbers of robots with the goal of reaching a high degree of scalability, that is, ideally the efficiency of the system should grow with the number of robots.

ii. Homogeneity: A swarm robotics system should consist of one or relatively few homogeneous groups. Each of these groups should consist of a large number of robots.

iii. Cooperation: A task solved by a swarm robotics system should require the cooperation of multiple robots, or at least the performance should be comparably low when robots do not cooperate to solve the task.

iv. Locality: The robots should have very limited sensing and communication abilities that allow them to sense and communicate only in their local vicinity.

These four criteria—even if not meant to be used as a checklist—help to identify whether a study is located within swarm robotics or not. These characteristics can be observed in social insects, such as ants, bees or termites, which therefore often serve as a source of inspiration.

1.2.3 Groups, Teams and Teamwork

One of the aspects investigated in this thesis are the conditions under which complexity can "emerge" in swarm intelligence systems. We believe that the design

and study of such systems is relevant not only for advancing the state of the art in robotics and similar technologically driven disciplines, but it may also provide valuable insights to other disciplines such as biology, economics, and social sciences.

One way of measuring the complexity that "emerges" in a swarm intelligence system is to look at the structural organisation of individuals when performing a task. In an insect colony, various organisational levels can be observed. Both behaviours at the individual level as well as at the colony level have been extensively studied (Hölldobler and Wilson, 1990). "However, between these two extremes, numerous functional adaptive units, or 'parts' exist" (Anderson and McShea, 2001). These *intermediate-level* parts comprise *groups* and *teams*.

Teamwork is widely observed in vertebrates. Here, *individual recognition* is believed to be an important factor (Wilson, 1975). Fewer examples of teamwork are known in invertebrates. Oster and Wilson (1978) argue that members of social insect colonies can not form teams as a consequence of their low grade of discrimination: social insects can discriminate "nest mates from aliens, [and] members of one caste as opposed to another" (Oster and Wilson, 1978), however, "there is very little evidence that social insects can recognise each other as individuals (but see Tibbetts (2002))" (Anderson and Franks, 2004a). In contrast, in the recent literature (Franks, 1986; Hölldobler and Wilson, 1990; Anderson and Franks, 2004a), biologists suggest that teams are indeed formed in social insects, and do not require individual recognition. Another aspect that is subject of the ongoing debate is whether inter-individual differences (e.g., members of different castes) are fundamentally required in teamwork (Hölldobler and Wilson, 1990; Beshers and Fewell, 2001; Anderson and Franks, 2001). Can a team be composed of interchangeable individuals of a monomorphic society? In general, several models of division of labour have been proposed (Beshers and Fewell, 2001). Some models suggest some proximate causes of division of labour: "Two general patterns of division of labour are recognised in social insects: temporal polyethism, or age-correlated patterns of task performance, and morphological polyethism, in which a worker's size and/or shape is related to its performance of tasks" (Beshers and Fewell, 2001).

Anderson and Franks (2004b) list a number of misconceptions about teamwork (from their point of view): "groupwork is synonymous with teamwork", "teamwork requires inter-individual differences", "teamwork requires individual recognition", "some tasks are inherently team tasks", "efficient teamwork requires direct communication", "teams require a leader", and "team members need to know the state and goals of other members".

One of the merits of studying robotic systems is that the individual morphology and behaviour are system variables that are controlled. Therefore, we can investigate whether tasks that require a complex division of labour fundamentally require individual recognition or inter-individual differences. We illustrate the methods and

results of a series of experimental works in which a set of "identical" robots is required to perform the complex, cooperative task described at the beginning of this chapter.

In the following we use the terms groups and teams as defined by Anderson and Franks (2001). In particular, a *group* is a set of individuals that tackle a group task; a *team* is a set of individuals that tackle a team task. A *group task* is a task that "requires multiple individuals to perform the same activity concurrently"; a *team task* is a task that "requires different subtasks to be performed concurrently". Furthermore, a *partitioned task* is "a task that is split into two or more subtasks that are organised sequentially (Jeanne (1986); reviewed in Ratnieks and Anderson (1999); Anderson and Ratnieks (2000))" (Anderson and Franks, 2004a). Anderson and Franks (2001, 2004a), and Anderson and McMillan (2003) found that the definition of teamwork, developed primarily from studies of social insects, also applies more generally to societies of other animals, including humans, and robots.

Figure 1.4 illustrates the division of labour present in our system. The overall task can be considered a partitioned task comprising three subtasks that are organised sequentially: (i) *path formation* (left part of Figure 1.4) requires robots to explore the environment and form a path in between the nest and the prey; (ii) *assembly* (centre part of Figure 1.4) requires robots to follow the path from the nest to the prey and then assemble either to the prey directly or to another robot that already gripped to it; (iii) *transport* (right part of Figure 1.4) requires some robots to transport the prey back to the nest. The previously formed path guides them to find the nest. *Path formation* itself is a group task, because only a group of robots can establish a path. Similarly, *path maintenance* and *path decomposition* are group tasks. *Assembly* is a team task, because it requires two different subtasks to be performed concurrently—*path maintenance* and *path following & grasp*, where the latter is an individual task.[1] *Transport* is a team task as some robots have to engage in *group transport*, while others, at the same time, have to reside in the path to guide the transport robots towards the nest.

1.3 Thesis Layout

This thesis is organized into nine chapters. In Chapter 2, we present an overview of related work. We describe the current state of the art in exploration, navigation and path formation in single and multi robot systems, and describe similarities and differences to other approaches in order to situate our research. Furthermore, we give a brief overview of self-assembly and group transport in robotics. In Chapter 3, we present the task, a high-level description of the two control approaches used, and

[1] It needs to be performed by multiple individuals (constraint C_1), however, they do not have to act concurrently.

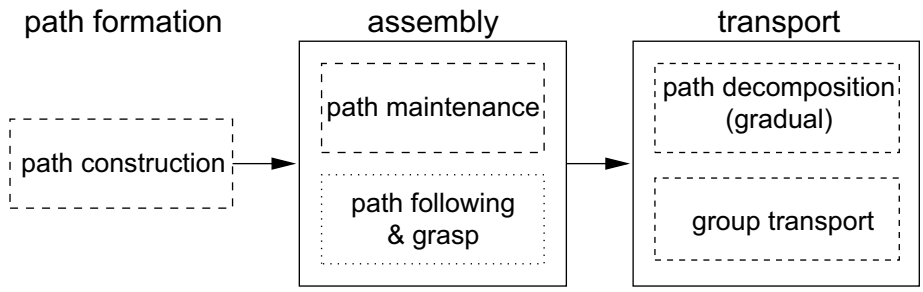


Figure 1.4: Illustration of the division of labour to accomplish the foraging task under constraints C_1, C_2, C_3 and C_4. The overall task is a partitioned task. It splits into three (sub-)tasks—*path formation* (left part), *assembly* (centre part) and *transport* (right part)—that are organised sequentially (indicated by arrows). Once that a path is formed between nest and prey, robots are recruited to the prey by following the path and then have to assemble to it or to another robot that already gripped onto it. *Transport* requires those robots that are gripped onto the prey and/or to each other, to transport the prey back to the nest. The path then guides the transporting group to find the nest. Individual tasks, group tasks and team tasks are framed respectively by dotted, dashed and solid lines.

the hardware and simulator used. In Chapter 4, we describe the chain controller, give a general description of the resulting group behaviour, explain the different variants, and detail the different behaviours and rules. We then present the results obtained with the chain controller in simulation (Chapter 5) and on the real robots (Chapter 6). Keeping the same structure used for the chain controller, Chapters 7 and 8 describe the algorithm and results in simulation of the vectorfield controller. Finally, in Chapter 9, we summarize our work, draw conclusions and discuss possible future directions.

Chapter 2

Related Work

In this chapter we give an overview of the research related to this thesis. This comprises the three subjects of path formation, self-assembly and group transport. We put the emphasis on path formation and give an overview of the state of the art in single- and multi-robot systems in Section 2.1. Afterwards, we briefly discuss related works in self-assembly in Section 2.2 and group transport in Section 2.3.

2.1 Exploration, Navigation and Path Formation

The term navigation originates from nautics and refers to the science and skill of sailing from one place to another. The navigator of a ship has to determine the ship's position, relate it to the desired destination, and accordingly set an adequate course for the ship. This description has entered into the domain of robotics nearly unchanged. For instance, Levitt and Lawton (1990) define navigation by the following three questions: (i) "Where am I?"; (ii) "Where are other places relative to me?"; (iii) "How do I get to other places from here?".

The first question refers to the problem of localization, which is the process of identifying the robot's specific position. Answering this question does not necessarily have to yield the specific position within a global reference frame, but may more generally let the robot identify certain characteristics of its position. The second question denotes the process of putting the current position within a global representation of the environment. The answers to these two questions lay the basis for extracting the required actions to move towards a desired position, which is the object of the third question.

This interpretation of navigation is used by many robot navigation systems (Kuipers and Byun, 1988). However, none of these systems has yet reached the flexibility and performance of animals such as bees, ants, birds or fish (Franz and Mallot, 2000). This has led robotics researchers to investigate more closely the nav-

igation mechanisms applied in biological systems, which gave birth to the research field of *biomimetic robot navigation*. Navigation mechanisms in animals, the main source of inspiration for biomimetic navigation, do not necessarily rely on answering all or even any of the three questions mentioned above. On the other hand, the most important issue appears to be the identification of how to reach the goal, which does not always require a localization or planning process.

In Section 2.1.1, we introduce the hierarchy of biologically inspired navigation strategies as defined by Trullier et al. (1997) and extended by Franz and Mallot (2000). The discussed strategies refer to the single robot domain. Therefore, in Section 2.1.2 we discuss some implementations of exploration and navigation strategies in the multi-robot domain.

2.1.1 Single Robot Systems

Table 2.1 summarizes the six strategies in the navigation hierarchy according to their behavioural prerequisites and navigation competences. The table is split into *local navigation* strategies and *way finding* strategies. Local navigation strategies have also been called tactics (Werner et al., 1997) or local control strategies (Kuipers and Byun, 1988). An agent chooses its action on the basis of current sensory or internal information only, without representing any objects outside the current sensory horizon. Way finding strategies, on the other hand, also store and make use of global information.

Random Search: In the simplest form of navigation, a robot randomly explores the environment. A robot only requires the basic competences of locomotion and goal recognition. Compared to the strategies presented in the following sections, random search typically requires a larger amount of time to detect the goal, but can be used as a backup strategy when the agent is not able to detect the goal.

Target approaching: Navigation would not be possible without the basic ability of approaching a perceived object. In biology, target approaching can be observed in most animals that are capable of locomotion. For a robot, to approach a target is a basic navigational requirement. To do so, the sensory information has to be used in order to orient the robot in the direction of the goal, often referred to as "body alignment". The robot must then be able to move towards the goal.

Braitenberg (1984) shows that minimal sensory information and a very simple controller suffice to approach a target. Several studies address the target approaching behaviour in insects. For instance, Lund and Webb (1997); Webb (1995) developed a controller that mimics the sound approaching behaviour observed in female crickets, by using a mechanism that is able to discriminate the relative phase and the different travel times of incoming sound signals. Webb implemented this on a mobile robot that was able to find an artificial sound source (Webb, 1995). This system was later extended so that the robot was able to find real crickets (Lund and Webb, 1997).

Table 2.1: The hierarchy of biologically inspired navigation strategies. Six strategies are classified with respect to the information they store and their characteristics.

Strategy	Used information	Characteristics
Local Navigation Strategies		
Random search	Goal recognition	Backup strategy
Target approaching	Goal recognition for body alignment	Basic requirement for navigation
Guidance	Extraction of goal direction from local landmark-configuration	Local navigation
Way Finding Strategies		
Recognition-triggered response	Set of landmark-configurations for each sub-goal	Global guidance
Topological navigation	Set of landmark-configurations linked by topological relationships	Topological detours
Metric navigation	Set of landmark-configurations linked by metrical relationships	Metrical detours Metric shortcuts

Guidance: Guidance is the process of extracting the direction towards a goal from the local landmark-configuration.[1] Bees and ants are able to use visual guidance to find a goal location which is only defined by an array of locally visible landmarks (for a review see Collett (1992)). Experiments suggest that to do this, an insect needs to memorize a snapshot of the spatial relationship between itself and the landmarks when it is located at the goal position. Later, when it is searching the goal position, it attempts to move so as to replicate this view.

This simple form of guidance has inspired several robot implementations as it enables a robot to find a goal that cannot be directly perceived, without requiring a complex representation of the environment. For instance, Franz et al. (1998) applied a snapshot-based guidance method using a miniature robot with a conical mirror camera. Robust performance was shown in a number of experiments in a realistic low contrast environment. Möller et al. (1998) successfully implemented a similar method using the Sahabot 2 robot on a flat plane in the Sahara desert with four

[1]Landmarks, also referred to as beacons, are usually tall objects that can be perceived from comparably far distances, or even globally in the environment.

black cylinders as landmarks.

Recognition Triggered Response: Guidance is a local navigation strategy as it requires only to process the locally available information. On the other hand, recognition triggered response, requiring the global localization of the robot, is a way finding method. Recognition triggered response is in many ways similar to guidance, as it relies on the perceived landmark configuration. It can be considered as an extension of the simple guidance strategy as, instead of just memorizing one landmark configuration, a set of landmark configurations is saved, each one connecting two locations by means of local navigation. In order to associate the appropriate local navigation method with the current landmark configuration, this method not only involves the recognition of the goal, but also of the starting location. The sequence of recognition triggered responses leads an agent to follow a route step by step, where the arrival at one sub-goal triggers the next step. In this way, a robot can navigate between locations that cannot be reached by local navigation methods alone.

Insects can associate movement decisions with visual landmarks. Ants, for instance, may learn to always pass a landmark on the right side (Collett et al., 1986). This association persists even when the order of the landmarks or their relative positions to the nest are changed. Bees are able to learn routes, that is, a sequence of recognition triggered responses (Collett et al., 1993).

Recognition triggered response has been used for numerous robotic navigation systems. Gaussier and Zrehen (1995), for instance, presented a robot that learned associations between compass directions and landmark configurations. The landmark configurations were extracted from panoramic images obtained from a rotating camera. A place was characterized by a sequence of local landmark views and bearings connected by camera movements. The system could find its goal from any position inside an office room. Other implementations of recognition triggered response methods can be found in (Nelson, 1991; Recce and Harris, 1996).

Map-Based Navigation: Map-based navigation seems quite natural to humans because using a map is a very convenient way to describe an environment and to share it with other people. However, the human use of a map requires a lot of high-level cognitive processes to interpret the map and to relate it to the real world. Map-based navigation relies on three processes (Levitt and Lawton, 1990; Balakrishnan et al., 1999; Filliat and Meyer, 2003; Meyer and Filliat, 2003):

- Map-learning, which consists in memorizing the data collected by a robot during exploration into a map.
- Localization, which consists in deriving the current position of the robot within the map.
- Path planning, which consists in choosing the appropriate actions to reach a goal destination from its current position.

In the following, we give a brief description of the two main categories of maps: topological maps where only particular locations and their relative positions are stored, and metric maps, where objects are stored in a common reference frame.

- **Topological maps:** The recognition triggered response method is only capable to lead an agent always through the same sequences of locations. There is no planning involved in the navigation process, possibly causing problems, for instance, if a part of the route is blocked by an obstacle. In that case the robot would have to perform a random search to find a known place again. This can be avoided if the spatial representation of the environment is goal-independent. To do so, a robot needs to be able to detect whether different routes pass through the same place, and in case they do, merge them by *route integration*. Integrated routes then become a topological global representation of the environment, which can be expressed as a graph with vertices representing locations, and edges representing the local navigation method to connect two vertices. Typically stored are the locations of objects, corridors, rooms and entrances to such rooms. By planning alternative routes, an agent using topological maps can dynamically adapt its route when encountering obstacles.

  Biological systems seem to construct topological representations by integrating routes in a bottom-up manner (Lieblich and Arbib, 1982). This ability has been observed in many animals, ranging from honeybees (Dyer, 1991) to humans (Gillner and Mallot, 1998). Implementations on robots mostly follow such a bottom-up approach and mainly differ in the place recognition, local navigation and route integration strategies used.

  Matarić (1990), for instance, developed a behaviour-based controller for topological navigation. In contrast to most other approaches, the recognition of places in the environment was only determined by their context, that is, by the sequence of actions preceding the current one. The only information stored in the topological graph representation were actions, not place descriptions. The robot was capable of acquiring routes autonomously by following the walls of the experimental room. Routes were integrated as soon as the robot encountered previously visited locations. Mallot et al. (1995) used a miniature robot to explore hexagonal mazes. Between junctions, the robot travelled by means of corridor following using infrared proximity sensors. Mallot *et al.* did not integrate views into a common place representation. Instead, the view graph was learned by a neural architecture that associated sequences of views with movement decisions.

- **Metric maps:** While for topological navigation a robot only memorizes key locations in the environment, metric navigation requires the robot to learn all known places and their position in a global reference frame. In contrast

to topological navigation, where the spatial relations are known between two directly connected locations only, in metric navigation the spatial relationship between any two locations can be extracted. An agent using metric navigation is able to find new paths through unknown terrain, as the integration of the current location into the reference frame allows it to deduce the spatial relations to previously visited locations. This includes, for example, shortcuts and detours around obstacles.

2.1.2 Multi Robot Systems

In this section we discuss the state of the art in multi robot systems exploring the environment. We will first discuss approaches in which the robots do not use a map. A path is then formed either by the robots themselves, or by immobile devices in the environment. Afterwards, we review map-based approaches.

Traditional approaches to environment navigation are often based on an internal map-like representation of the environment. Such approaches do not scale well for large groups of agents, where a distributed control strategy may be better suited. When approaching the problem of controlling swarms of robots, researchers often take inspiration from social insects and sometimes directly refer to the term pheromone (Mamei and Zambonelli, 2005; Payton et al., 2001, 2004), or to ants (Svennebring and Koenig, 2004).

All these approaches employ distributed control mechanisms, and mostly use simple strategies and local information. We can roughly distinguish between two categories of distributed multi-agent path planning, based on whether the path is formed by immobile, or mobile devices:

- **The path is formed by a network of immobile devices.** The devices are placed either a priori at fixed positions, or by the robots themselves. An individual network node is usually very limited in its sensing and computing capabilities. Robots can locally communicate with the network to find a path in the environment. Due to their simplicity, network nodes have low power consumption and are relatively cheap to produce, which makes them ideally suited for large scale experiments. For instance, O'Hara and Balch (2004) use a sensor network with up to 156 Gnats sensor nodes that compute the shortest path using the distributed Bellman-Ford algorithm (Bellman, 1957), and test the impact of different configurations of sensors placement. Li et al. (2003) use a similar approach with 50 sensors of the Mote platform and take into account so called danger zones which have to be avoided. Batalin and Sukhatme (2002) study a sensor network in the context of terrain coverage and navigation. A robot action is computed based on transition probabilities between the nodes. They use the Pioneer mobile robot and 9 nodes.

In the simplest case, network nodes do not have any sensory capabilities at all and are used as landmarks or as a medium for indirect, so called *stigmergic*, communication. Promising examples for this are RFID-based devices. Mamei and Zambonelli (2005) use such passively powered RFID tags in an office environment to mark fixed locations such as a door or a table, and to identify objects that may move around, such as keys or pencils. In their experiments, robots can manipulate the RFID tags and leave a trail which enables other robots to find particular objects. They encountered some problems due to the very limited storage capacity. Nevertheless, the general idea of using RFID technology is very appealing as RFID can be produced very cheaply, and will probably soon be found everywhere.

In addition to their low production cost, such devices have the advantage of being more robust than robots. However, they have to be placed in the environment a priori, or by the robots. This is not required if the robots form the path themselves.

- **The robots serve as landmarks or beacons themselves**. This is the case for our approach. When designing our controllers, we took inspiration from Goss and Deneubourg (1992), who have studied robot chains for a prey retrieval task. In their approach, every robot in a chain emits a signal indicating its position in the chain. Similar systems were implemented by Drogoul and Ferber (1992), and by Cohen (1996). In the latter case the robot group is heterogeneous, such that there are two groups of robots, one taking care of path formation and the other exploiting the formed path to follow it to a goal location. All these works were carried out in simulation, and differ from our approach to chains because robots in a chain structure need to transmit as many signals as there are robots. This leads to an increasing degree of complexity for growing group sizes. In our approach the number of different signals is independent of the number of robots. For the chains, three colours for nest and chain, and one colour for the prey are required. For the vectorfield two colours for nest and prey, and one pattern for direction indication suffice.

  Werger and Matarić (1996) use real robots to form a chain in a prey retrieval task. In their case the chains are not visually connected. Rather, they rely on physical contact: one robot in the chain has to regularly touch the next one in order to maintain the chain.

  Payton et al. (2001, 2004) study robot networks which can be used to represent a path as well. To build up the robot network different strategies are proposed. A gas expansion model leads to a uniform distribution similar to our vectorfield. A group of robots first spreads in the environment using simple attraction/repulsion mechanisms. Afterwards the robots communicate three

different sorts of pheromone to select the shortest of the many different possible paths. In our case, the network is built up incrementally, and the robots do not need to communicate at all with each other, except for indicating a direction. Another strategy to form the network is referred to as guided growth, and results in less branched structures such as our chains. One robot is selected to be the leader. The other robots follow this leader and in this way the robot structure stretches to form a line. The leading robot can for instance be designated by the user or by some rule.

Unlike the previously mentioned approaches, map-based navigation usually relies on a centralized control mechanism, in which the robots share a map.

Rekleitis et al. (1997, 1998, 2001) focus on the problem of reducing the odometry error during exploration. They separate the environment into stripes that are explored successively by the robot team. Whenever one robot moves, the other robots are kept stationary and observe the moving robot, a strategy similar to the one presented by Kurazume and Shigemi (2001). Whereas this approach can significantly reduce the odometry error during the exploration process, it is not designed to distribute the robots over the environment. Instead, the robots stay within visibility range.

Koenig et al. (2001) analyze different terrain coverage methods for ants which are simple robots with limited sensing and computational capabilities. They consider environments that are discretized into equally spaced cells. Instead of storing a map of the environment, the ants leave markers in the cells they visit. Two different strategies for updating the markers are considered. The first is similar to the approach of Yamauchi (1998) and leads the robots greedily to the closest unexplored area. In the second approach the ants simply count the number of times a cell has been visited.

Billard et al. (2000) use a probabilistic model to simulate a team of mobile robots that explores and maps locations of objects in a circular environment, and demonstrate the correspondence of their model with the behaviour of a team of real robots.

Balch and Arkin (1994) analyze the effects of different kinds of communication on the performance of robot teams for object search or terrain coverage.

In the context of cleaning tasks, Kurabayashi et al. (1996) propose an approach in which the map of the environment is given a priori, and in which the time to cover a known environment by a robot team is minimized.

Yamauchi (1998); Yamauchi et al. (1999) present a technique to learn maps with a robot team, where information about the map is exchanged and continuously updated among the robots. To explore the environment all robots move to the closest frontier cell.

Singh and Fujimura (1993) propose an approach that addresses the problem of

heterogeneous robot systems. While exploring, a robot identifies so called *tunnels* to unexplored areas of the environment. In case a robot is too big to pass through a tunnel it informs other robots better suited for the task.

Zlot et al. (2002) propose a control concept that is based on market economy. Potential target locations are considered for each robot and tasks are traded using auctions.

2.2 Self-Assembly

Following Whitesides and Grzybowski (2002), self-assembly can be defined as a process by which pre-existing discrete components organise into patterns or structures without human intervention.

Self-assembly is widely observed in social insects (Sendova-Franks and Franks, 1999; Anderson et al., 2002). Via self-assembly, ants, bees, and wasps can organise into functional units at an intermediate level between the individual and the colony. Anderson et al. (2002) identify 18 distinct types of self-assembled structures that insects build. The function of self-assemblages "can be grouped under five broad categories which are not mutually exclusive: (1) defence, (2) pulling structures, (3) thermoregulation, (4) colony survival under inclement conditions, and (5) ease of passage when crossing an obstacle". Anderson et al. (2002) claim that in almost all of the observed instances, the function could not be achieved without self-assembly.

Self-reconfigurable robots (Yim et al., 2002a; Rus et al., 2002) hold the potential to self-assemble and thus to mimic the complex behaviour of social insects. In current implementations (Murata et al., 2002; Yim et al., 2002a; Rus et al., 2002; Jørgensen et al., 2004), however, single modules usually have highly limited autonomous capabilities (when compared to an insect). Typically, they are not equipped with sensors to perceive the environment. Nor, typically, are they capable of autonomous motion. These limitations, common to most self-reconfigurable robotic systems, make it difficult to let separate modules, or groups of modules, connect autonomously. In some systems, self-assembly was demonstrated with the modules being pre-arranged at known positions (Yim et al., 2002b; Zykov et al., 2005). Some instances of less constrained self-assembly are reported (for an overview see Groß et al. (2006a)). Fukuda et al. (1988, 1995) demonstrate self-assembly among robotic cells using the CEBOT system (Fukuda and Ueyama, 1994). In the experiment, a moving cell approached and connected to a static cell. The moving cell was controlled with a finite-state automaton. Rubenstein et al. (2004) demonstrate the ability of two modular robots to self-assemble. Each robot consisted of a chain of two linearly-linked CONRO modules (Castano et al., 2002). The robot chains were set up at distances of 15 cm, facing each other with an angular displacement not larger than 45 degrees. The control was heterogeneous, both at the level of individual modules within each robot

and at the level of the modular makeup of both robots. Recently, self-assembly has been demonstrated with the swarm-bot system (Mondada et al., 2005). Experiments were conducted on different terrains and with up to 16 robots (Groß et al., 2006a).

Self-assembly is a particularly interesting mechanism in social insects (Anderson et al., 2002). Insects physically connect to each other to form aggregate structures with capabilities exceeding those of an individual insect. Some observed uses have strong implications for robotic system design (e.g., the formation of pulling structures (Hölldobler and Wilson, 1990)).

Most modular robotic systems are not capable of self-assembly—modules are pre-assembled by the experimenter or by a separate machine (Yim et al., 2002a). Other systems can self-assemble if the modules are pre-arranged in specific patterns. Rare instances of less constrained self-assembly with up to three robots have been reported (Fukuda and Ueyama, 1994; Rubenstein et al., 2004).

2.3 Group Transport

Group transport can be defined as the "conveyance of a burden by two or more individuals" (Moffett, 1992). In the biological literature, group transport is almost exclusively reported in the context of ants. In fact, Moffett (1992) claims that group transport "is better developed in ants than in any other animal group".

In most studies of transport with robotic groups, the robots move an object by pushing it. Pushing strategies have the advantage that they allow the robots to move objects that are hard to grasp. In addition, multiple objects can be pushed at the same time. On the other hand, it is difficult to predict the motion of the object and of the robots, especially if the ground is not uniform.[2] Therefore, the control typically requires sensory feedback. Most studies consider two robots pushing a wide box simultaneously from a single side (Matarić et al., 1995; Sugie et al., 1995; Donald et al., 1997; Parker, 1999; Gerkey and Matarić, 2002). To coordinate the robots' actions, robots are specifically arranged (Matarić et al., 1995; Donald et al., 1997; Parker, 1999; Gerkey and Matarić, 2002), control is synchronised (Matarić et al., 1995), relative positions are known (Donald et al., 1997; Parker, 1999), explicit communication is used (Matarić et al., 1995; Parker, 1999), and/or individual tasks are generated by a designated leader agent (Gerkey and Matarić, 2002; Sugie et al., 1995).

Only few studies consider more than two robots, pushing a box simultaneously (Kube and Zhang, 1993; Yamada and Saito, 2001; Kube and Zhang, 1997; Kube and Bonabeau, 2000; Tuci et al., 2006; Groß et al., 2006b,c). In these cases, the control is homogeneous, decentralised, and the robots make no use of explicit communication. Kube and Zhang (1997) and Kube and Bonabeau (2000) reported

[2]For a theory on the mechanics of pushing see Mason (1986).

that if the box is small compared to the size of the pushing robots the performance decreases drastically with group size as the box offers only limited contact surface.

Many studies consider the transport of an object by multiple mobile robots grasping and/or lifting it. In these studies, typically 2–3 robots are manually attached to the object (Desai et al., 1996; Kosuge and Oosumi, 1996; Aiyama et al., 1999; Sugar and Kumar, 2002; Miyata et al., 2002; Wang et al., 2003). To coordinate the robots' actions, robots often have knowledge of their relative positions. In some systems the desired trajectories are given prior to experimentation to all robots of the group. The object is transported as each robot follows the given trajectory by making use of dead-reckoning (Desai et al., 1996). In other systems, the manipulation is planned in real-time by an external workstation, which communicates with the robots (Miyata et al., 2002). Often, instead of an external computer, a specific robot called the *leader* knows the desired trajectory or the goal location. The leader robot can send explicit high- or low-level commands to the *followers* (Sugar and Kumar, 2002; Wang et al., 2003). However, in many leader-follower systems explicit communication is not required (Kosuge and Oosumi, 1996; Aiyama et al., 1999).

Chapter 3

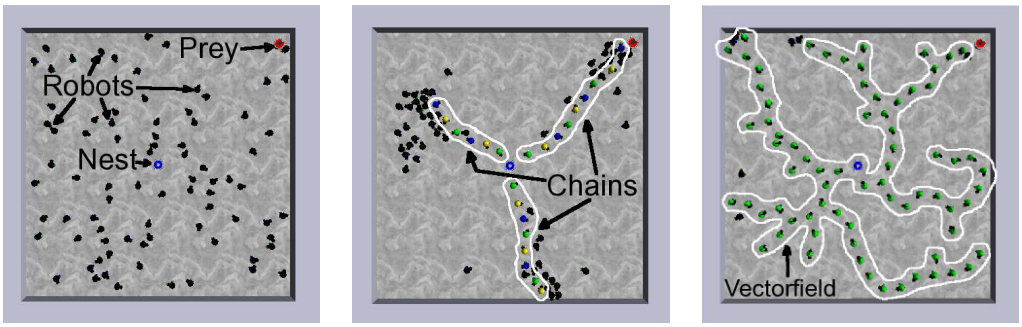
Methods

In this chapter we describe the task studied and the approach followed in this thesis. We start by illustrating the task along with giving a general idea of the two control algorithms in Section 3.1, and then give a detailed description of the robot platform and simulation environment on which we tested our controllers in Section 3.2.

3.1 Task and Control Approach

The task that we have chosen as a test-bed for our control algorithms is illustrated in Figure 3.1. A group of robots has to form a path between two objects—denoted as nest and prey. The robots have no a priori knowledge about the dimensions and the position of any object within the environment, and a robot's perception range is small when compared to the distance between the nest and the prey. The difficulty of the task can be varied by changing the distance between nest and prey, and by placing obstacles in the environment.

Initially, as shown in Figure 3.1a, all robots are placed at random positions. They search the nest, and once they perceive it, they start to self-organize into chains (Figure 3.1b) or into a vectorfield (Figure 3.1c). In both cases robots act as trail markers and attract other robots. Neighbouring robots within the path forming structure have to be able to sense each other in order to assure the connectivity. As the robots have no knowledge about the position of the prey, the structures are oriented in random directions. A self-organized process in which robots leave the structure and join it again at a different position leads to a continuous exploration of the environment until the prey is discovered. A path is then formed, and can be used by other robots to navigate between the nest and the prey, or to transport the prey to the nest. When controlled by the chain mechanism, robots in the path signal one out of three colours. The sequence of these colours gives directionality to the chain. In the vectorfield controller (Figure 3.1c) the directionality is not given



(a) (b) (c)

Figure 3.1: Simulation snapshots from the initial situation (a), and a typical outcome when employing the chain (b) and the vectorfield (c) controllers. 80 robots are indicated by small black circles. The task is to form a path between the blue nest in the centre of the arena, and the red prey in the top right corner. No obstacles are employed. When a robot in a chain or in a vectorfield perceives the prey, a path is formed that can be used to transport the prey to the nest.

by a sequence of colours, but each robot explicitly indicates a direction. In addition to these two control algorithms we employ a mechanism that we call prey extension. By default, a robot controlled by the chain or vectorfield controller does not react in case it perceives the prey unless it also perceives a robot that is part of the path forming structure. However, when the prey extension mechanism is employed, a robot that perceives the prey and no path has a given probability to activate its LEDs in the colour of the prey, in this way extending the area in which the prey can be perceived. Details about the two control mechanisms and the prey extension mechanism will be given in Chapter 4 for the chain controller, and in Chapter 7 for the vectorfield controller.

3.2 Hardware and Simulator

The experiments presented later in this thesis have been conducted either on the robot platform *s-bot*, or in simulation. Our simulation platform, called TwoDee (Christensen, 2005), is a multi-robot simulator based on a custom high-level dynamics engine. It has been optimized for the use with the *s-bot*,[1] and controllers developed in simulation have been successfully ported to the real robot for several tasks (Christensen and Dorigo, 2006; Christensen et al., 2008, 2007; Nouyan et al.,

[1]The *s-bot* was developed within the SWARM-BOTS Project, a Future and Emerging Technologies project funded by the European Commission (see `www.swarm-bots.org`).

2006). In this section, we give a description of the *s-bot* robot, and explain the mechanisms employed to ensure a realistic simulation. For a more comprehensive description of the *s-bot*'s hardware see Mondada et al. (2005), and for the TwoDee simulator see Christensen (2005).

3.2.1 The *S-bot* Robot

Figure 3.2a shows the physical implementation of an *s-bot*. It has a diameter of 12 cm and weighs approximately 700 g. In the following we briefly overview the actuators and sensors relevant to this study.

The robot's traction system consists of a combination of tracks and two external wheels, called *treels*. The *s-bot* is capable of a maximum speed of 30 cm/s. For the chaining mechanism, we used a maximum speed of 13 cm/s on the real robot. This speed corresponds to a maximum angular velocity of 97.6 deg/s when turning on the spot.

For the purpose of communication, the *s-bot* has been equipped with eight RGB LEDs distributed around the robot. In particular, this LED-ring is used by robots in a chain to activate the LEDs with the colours blue, green and yellow, and by robots in a vectorfield to activate a pattern which may be used to indicate a direction, as shown in Figure 3.2b.

In order to perceive the LED-ring, a VGA camera is mounted on top of the *s-bot* and is directed towards a spherical mirror, in this way providing an omni-directional view. The camera is used to perceive the nest, the prey, and other *s-bots* emitting a colour with their LED-ring. A snapshot taken from an *s-bot*'s camera is shown in Figure 3.2c. Given that the spherical mirror is mounted at 10 cm on top of a robot, another robot does not entirely block the view of the camera. However, obstacles, such as those employed in our experiments, do block the view. Due to differences among the robots' cameras, there are some variations in the perceptual ranges. The software we use on the real robot to detect coloured objects allows a recognition of the red coloured prey up to a distance of $70 - 90$ cm, and of the three colours blue, green and yellow, up to $35 - 60$ cm (depending on which robot is used). Due to the spherical shape of the mirror, the distance to close objects can be approximated with good precision up to a distance of 30 cm, but it becomes increasingly difficult to deduce the distance for objects that are further away. The direction to other objects is perceived quite precisely, and the precision increases with growing distance.

The *s-bot* has 15 infra-red proximity sensors distributed around its turrets, used for obstacle avoidance. Using these sensors the *s-bot* can recognise another object when its distance is less than 15 cm.

Figure 3.2d shows the *s-toy*, an object which we use either as nest or as prey (depending on its colour). It has a diameter of 20 cm and, like the *s-bot*, it is equipped with RGB LEDs. The nest is immobile. The prey weighs 800 g and requires the

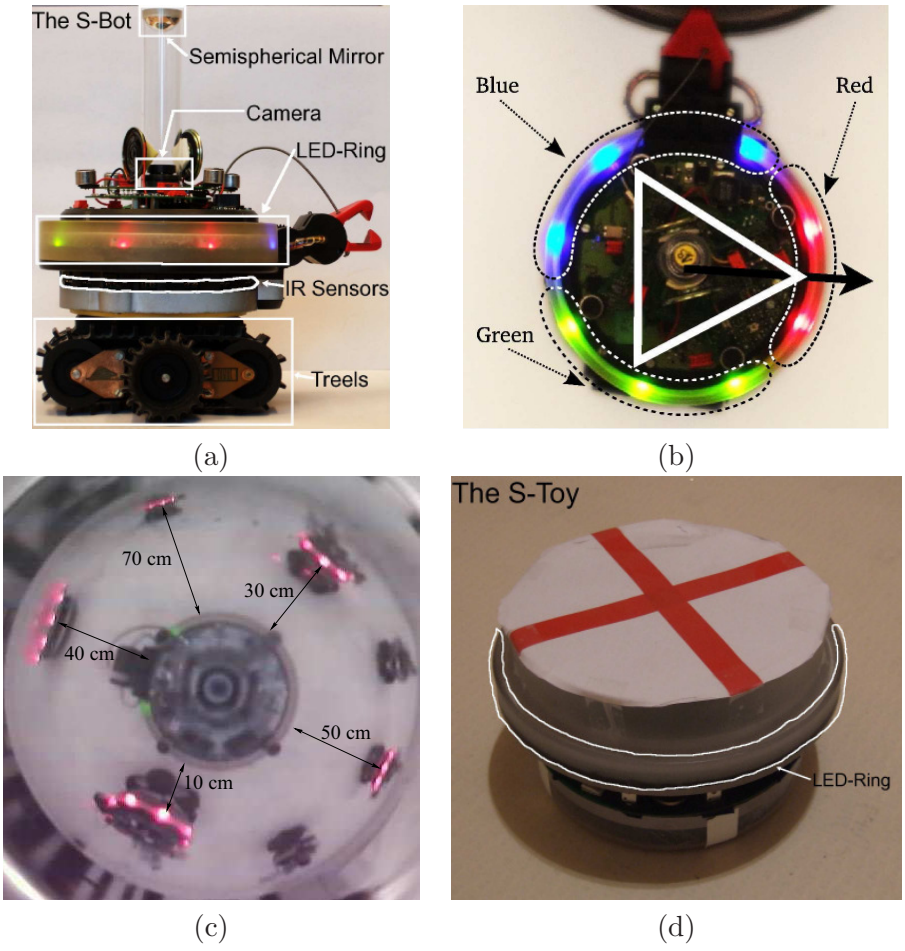


Figure 3.2: The hardware. (a) The *s-bot*. (b) A robot activating its LEDs to indicate a direction as employed by the vectorfield controller. (c) An image taken with the omni-directional camera of the *s-bot*. It shows other *s-bots* and an *s-toy* activating their red LEDs at various distances. (d) The *s-toy*, which is used both as nest and as prey.

cooperative effort of two or more *s-bots* to be moved.

3.2.2 The TwoDee Simulator

In TwoDee, the *s-bot* is modelled as an object composed of a cylinder with the diameter of the *s-bot* body, and of a cuboid with the dimensions of the *s-bot* gripper. The nest and the prey are represented by coloured cylinders of the size of an *s-toy*. Despite the efforts to devise a precise simulation, some characteristics of the robots and of the robot-environment interaction may escape the modelling phase. For this reason, noise is used to ensure that the behaviour developed in simulation will cope with differences between simulation and reality (Jakobi, 1997; Jakobi et al., 1995). Noise is simulated for both actuators and sensors, adding a random value uniformly distributed in a given range. The noise distribution from the real robots is modelled by a uniform noise distribution. The bounds of the added random values are specified in the following and are in general higher than the standard error observed on the real robots.

The tracks have been simulated by two active wheels. The speed of each wheel is set individually. We adopted the values of maximum speed and angular velocity as reported above. When setting the speed of a wheel to v_0, we add a noise value in the range $[-0.1 \cdot v_0; 0.1 \cdot v_0]$.

The camera and the proximity sensors have been modelled in the simulator trying to closely match their physical counterpart. A sampling technique was employed using samples from the corresponding devices recorded from the real robot (Miglino et al., 1995). These samples are collected in a matrix of activation values that can afterwards be used in the simulation to characterise the sensor activation for a given situation.

For the camera, we recorded 100 samples from camera images for 36 angles and 22 distances in the range $[5\,\text{cm}; 100\,\text{cm}]$ with respect to a prey, a nest, or another *s-bot*. When calculating distance and direction to another object in simulation, we take the median values from the collected samples for the given situation, and add noise values in the ranges $[-10\,\text{cm}, 10\,\text{cm}]$ for the distance and $[-18^o, 18^o]$ for the direction. Concerning the perception of LED patterns indicating a direction, as used by the vectorfield, we add a noise value in the range $[-36^o, 36^o]$ to the median value taken from the samples. The differences in the perception of the different colours and the differences between the robots are taken into account in simulation as well: each robot is given a different set of perceptual ranges for the four colours, and each value is chosen randomly from the ranges mentioned above.

The proximity sensors, like the camera, have been modelled by recording 100 samples from the proximity sensor activation for 36 angles and 18 distances in the range $[1\,\text{cm}; 20\,\text{cm}]$ with respect to either another *s-bot* or to a wall. To calculate the value of a proximity sensor in simulation, we take the median value from the

collected samples for the given situation, and add a noise value in the range $[-0.2 \cdot prox_{max}, 0.2 \cdot prox_{max}]$, where $prox_{max}$ is the proximity sensor saturation value.

Chapter 4

Chain Controller

In this chapter we describe the chain controller. We first give a general description of the overall behaviour in Section 4.1 and of the different variants used in Section 4.2. In order to integrate the different behaviours required to solve the overall task, we follow a behaviour based approach (Arkin, 1998) as it allows us to comfortably merge different approaches and sub-controllers into one structure. Each behaviour consists of a collection of motor schemas, that is, low level control mechanisms. The motor schemas used are described in Section 4.3. Afterwards, the different behaviours are explained in Section 4.4, and the rules that trigger a transition from one behaviour to another are detailed in Section 4.5.

4.1 General Description

The controller that we designed and implemented to run our experiments consists of eight behaviours, each of which is designed to achieve a specific goal. The overall task can be split into the subtasks path formation, assembly and transport. The individual behaviours for the path formation subtask, which is the focus of this thesis, are implemented using the motor schema paradigm. For the assembly and transport subtasks the behaviours are based on the work of Groß et al. (2006a); Groß and Dorigo (2004) and Groß et al. (2006b), and rely on neural networks or simple hand written commands. The eight behaviours are detailed in Section 4.4.

At the beginning of a trial, the robots are located at random positions. If a robot does not perceive the nest or a chain, it performs a random walk (`Search Chain` behaviour) until it perceives one of them. Note that the nest is considered as the root of each chain. A robot that finds the nest starts either to explore the environment around the nest or to follow an existing chain (`Explore Chain` behaviour). In the latter case, when it reaches the tail of the chain, it joins the chain with probability P_{in} per time step (`Join Chain` behaviour). Robots that are part of a chain cannot

leave it unless they are situated at the chain's tail, in which case they leave it with probability P_{out} per time step. The process of probabilistically joining/leaving a chain is at the basis of the exploration of the environment as it allows the formation of new chains in unexplored areas.

If a chain member perceives the prey, it does not leave the chain, so that when a chain encounters the prey the formed path becomes stable. At this point there are two possibilities: If the prey is far, other robots can still join the chain to make a connection that is closer to the prey; if the prey is close, the subtask of path formation is successfully accomplished. Once a path is formed, it is maintained and in this way automatically recruits other robots to assemble to the prey (`Assemble` behaviour). If a robot that tries to assemble to the prey does not succeed within a certain time it gives up, moves back to the nest and rests for a while (`Recovery-A` behaviour). When a sufficient number of robots has assembled to the prey, the prey transport starts. Robots assembled to the prey transport it by moving towards the closest perceived member of a chain (`Transport Target` behaviour). When the prey pulling structure moves close to a chain member, the latter leaves the chain and moves back to the nest to rest for a while (`Recovery-P` behaviour). In this way the pulling structure of robots is guided from node to node of the chain to eventually reach the nest. A robot leaving the chain to rest at the nest emits a sound signal for a period of 30 s. A robot transporting the prey and perceiving this sound signal reacts to it by pausing the transport. Otherwise, if it continues moving towards the sound emitting robot, there is a risk that the robot is blocked from moving back to the nest. This situation can occur when the pulling structure of transporters touches the respective robot. No other robots react to the sound signal.

The directionality in our chains relies on the concept of *cyclic directional patterns* (Figure 4.1). Each robot emits one out of three signals (i.e., LED colours) depending on its position in the chain. By taking into account the sequence of the signals, a robot can determine the direction towards the nest, or towards the prey. The prey and the nest can be recognised by their colour: the nest is blue, and the prey is red.

4.2 Variants

We implemented three variants of the chain formation mechanism. These variants differ by the degree of motion allowed to the chain structure. In the simplest case, referred to as static strategy, there is no motion at all. In the second case, referred to as aligning strategy, the chains as a whole align. Finally, in the third case, referred to as moving strategy, the chains perform a circular movement around the nest.

In addition to this, we implemented a prey extension mechanism. When this mechanism is active, robots that perceive the red prey, but no chain, activate their red LEDs with a certain probability, in this way increasing the area in which the

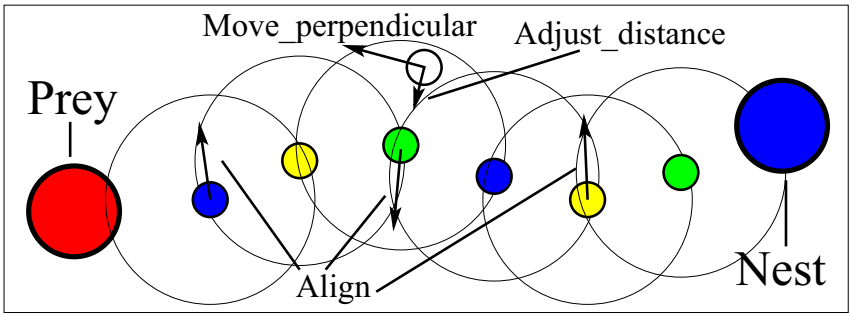


Figure 4.1: Chains with cyclic directional patterns (CDP). The small coloured circles represent robots that have formed a CDP-chain that connects the nest with the prey. Three colours are sufficient to give directionality to the chain. The large circles surrounding the robots indicate their perceptual range. The small uncoloured circle represents an explorer robot following the chain towards the prey. The three vectors drawn on top of the chain members represent the motor schema that leads to an overall alignment of the chain. The two vectors drawn on top of the explorer robot represent the motor schemas that lead to a tangential trajectory along the chain.

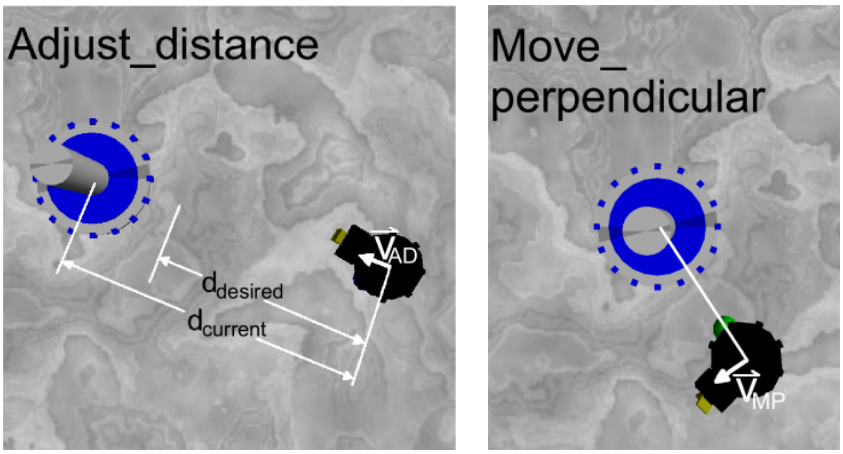
prey can be perceived by the chain (`Extend Prey` behaviour). This mechanism potentially speeds up the path formation process because a second path, starting from the prey, is formed in parallel to the chains.

4.3 Motor Schemas

The behaviours used for path formation are realized following the motor schema paradigm (Arkin, 1989, 1992). A motor schema couples perception to action without the use of abstract representations. The motor schemas can be considered as basic building blocks for a behaviour. Each motor schema outputs a vector denoting the desired direction of motion. For each behaviour, a set of motor schemas is active in parallel. Active motor schemas are added and translated into motor activation at the beginning of each *control time step*.[1] Common to all behaviours is a motor schema for collision avoidance. In the following, we detail the five employed motor schemas as shown in Figure 4.2 and Figure 4.3.

- **Adjust_distance**(α, $d_{current}$, $d_{desired}$): returns a vector $\overrightarrow{v_{AD}}$ that points towards an object at angle α if the current distance to the object $d_{current}$ is

[1] A control time step has a length of 120 ms in simulation. On the real robot this value is not constant because it depends on the time required for image processing.



(a) (b)

Figure 4.2: Employed motor schemas. (a) Adjust_distance returns a vector pointing towards an object if the current distance $d_{current}$ is larger than the desired distance $d_{desired}$, and away from it otherwise. (b) Move_perpendicular returns a vector directed perpendicularly with respect to an object.

(a) (b) (c)

Figure 4.3: Employed motor schemas. (a) Avoid_collisions returns a vector pointing away from objects that are close. (b) Move_straight returns a vector that points forward. (c) Align returns a vector that leads to the alignment of a chain member with respect to its neighbours.

larger than the desired distance $d_{desired}$, and in the opposite direction otherwise. The length of the returned vector is proportional to the value of $\|d_{current} - d_{desired}\|$. In order to avoid an oscillating behaviour, the vector is set to zero if $\|d_{current} - d_{desired}\| < 5\,\text{cm}$ (see Figure 4.2a):

$$\overrightarrow{v_{AD}} = \begin{cases} (d_{desired} - d_{current}) \cdot \begin{pmatrix} cos(\alpha) \\ sin(\alpha) \end{pmatrix} & , \|d_{desired} - d_{current}\| \geq 5\,\text{cm} \\ 0 & , \|d_{desired} - d_{current}\| < 5\,\text{cm}. \end{cases}$$

- **Move_perpendicular**(α, *clockwise*): returns a unit vector $\overrightarrow{v_{MP}}$ that is perpendicular to an object at angle α. The boolean parameter *clockwise* determines whether the vector is perpendicular in a clockwise sense or not (see Figure 4.2b):

$$\overrightarrow{v_{MP}} = \begin{cases} \begin{pmatrix} -sin(\alpha) \\ cos(\alpha) \end{pmatrix} & , clockwise = 1 \\ \begin{pmatrix} sin(\alpha) \\ -cos(\alpha) \end{pmatrix} & , clockwise = 0. \end{cases}$$

- **Avoid_collisions**($IR\_sensors$): returns a vector $\overrightarrow{v_{AC}}$ that takes into account each activation IR_j of a proximity sensor j that is above a threshold Θ (an activation of a proximity sensor above the value Θ is reached for distances smaller than 5 cm). The direction of the vector is opposed to the source of activation α_j (i.e. the direction of proximity sensor j with respect to the robot's heading), and its length is proportional to the difference between the activation and the threshold (see Figure 4.3a):

$$\overrightarrow{v_{AC}} = -\sum_{j=1}^{numProx} max\{0, IR_j - \Theta\} \cdot \begin{pmatrix} cos(\alpha_j) \\ sin(\alpha_j) \end{pmatrix}.$$

- **Move_straight**: returns a unit vector $\overrightarrow{v_{MS}}$ that points forward (see Figure 4.3b):

$$\overrightarrow{v_{MS}} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

- **Align**($\alpha_{previous}$, α_{next}): returns a vector $\overrightarrow{v_{AL}}$ that leads to the alignment between the previous and the next chain neighbour which are perceived at the angles $\alpha_{previous}$ and α_{next}. The length of the vector is proportional to the value of $180^o - |\alpha_{previous} - \alpha_{next}|$. In order to avoid an oscillating behaviour, the vector is set to zero if $|\alpha_{previous} - \alpha_{next}| > \alpha_{align}$ (α_{align} is set to 170^o for the

aligning strategy, and to 175^o for the moving strategy, with 180^o representing perfect alignment, see Figure 4.3c):

$$\overrightarrow{v_{AL}} = \begin{cases} \begin{pmatrix} cos(\alpha_{previous}) + cos(\alpha_{next}) \\ sin(\alpha_{previous}) + sin(\alpha_{next}) \end{pmatrix} & , |\alpha_{previous} - \alpha_{next}| \leq \alpha_{align} \\ 0 & , |\alpha_{previous} - \alpha_{next}| > \alpha_{align}. \end{cases}$$

The active motor schemas are summed up, weighting each one with a gain value g_i. The individual gain values are given in the next section and were found through trial and error. Once the active motor schemas have been summed up, the resulting vector $\overrightarrow{v_{RES}}$ has to be translated into movement of the two wheels. This is done by the following function:

$$\begin{pmatrix} lSpeed \\ rSpeed \end{pmatrix} = \begin{cases} \begin{pmatrix} cos(2 \cdot \alpha_{RES}) \\ 1 \end{pmatrix} & , 0 \leq \alpha_{RES} < \frac{\pi}{2} \\ \begin{pmatrix} cos(2 \cdot \alpha_{RES} - \pi) \\ -1 \end{pmatrix} & , \frac{\pi}{2} \leq \alpha_{RES} < \pi \\ \begin{pmatrix} -1 \\ -cos(2 \cdot \alpha_{RES}) \end{pmatrix} & , \pi \leq \alpha_{RES} < \frac{3 \cdot \pi}{2} \\ \begin{pmatrix} 1 \\ -cos(2 \cdot \alpha_{RES} - \pi) \end{pmatrix} & , \frac{3 \cdot \pi}{2} \leq \alpha_{RES} < 2 \cdot \pi \end{cases} ,$$

where $lSpeed$ and $rSpeed$ denote the normalized speed of left and right wheel, and α_{RES} is the desired direction of movement with respect to the current heading. The resulting speed of the wheels is independent from the length of the summed vector v_{RES}, and depends on the maximum allowed velocity v_{max}, which is set to $12.37 \frac{cm}{sec}$, as on the *s-bot* for the given length of a time step this was the maximum speed found to result in a stable behaviour.

4.4 Behaviours

The behaviours and the rules that trigger a transition from one behaviour to another are illustrated by the state diagram in Figure 4.4. Each state corresponds to a robot behaviour, and arrows connecting states represent behaviour transitions. Using the aforementioned motor schemas as basic building blocks, the individual behaviours used for path formation were implemented as follows:

- **Search Chain**: perform a random walk by moving straight until an obstacle is detected in the front. Then turn on the spot for a random angle. LEDs are off. Active motor schemas: Move_straight (gain value $g_{MS} = 1$), Avoid_collisions($g_{AC} = 0.1$).

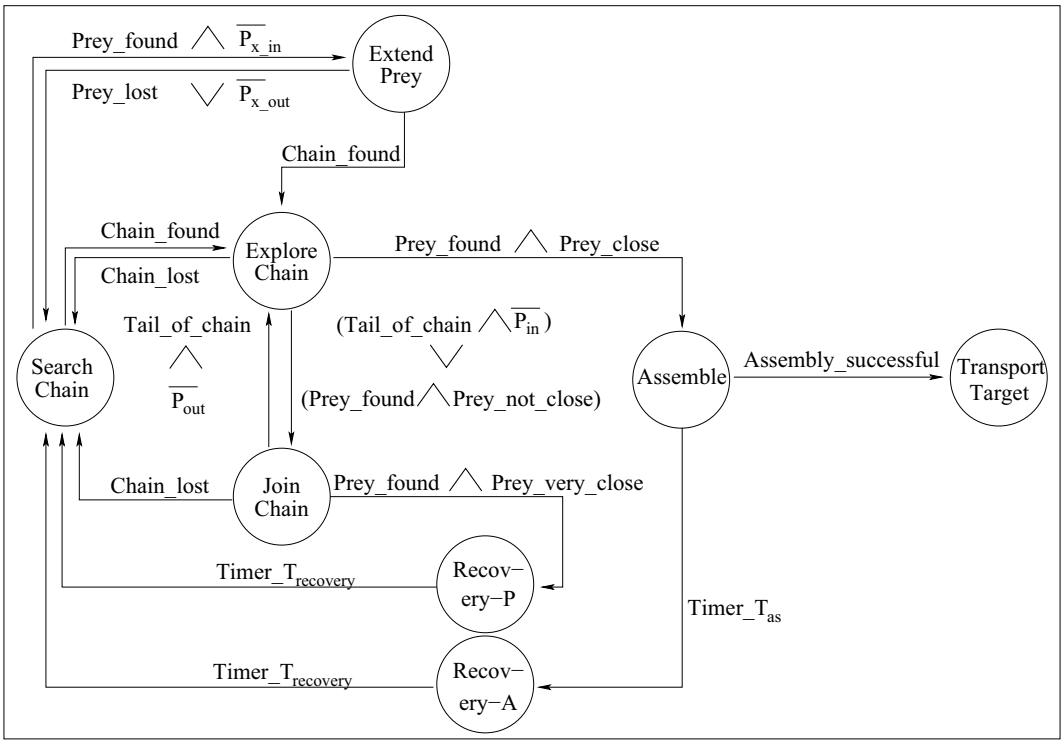


Figure 4.4: State diagram of the finite state machine that controls each robot. Circles represent states (i.e., behaviours). Edge labels specify conditions that trigger transitions between the corresponding states. The initial state is `Search Chain`. $\overline{P_{in}}$, $\overline{P_{out}}$, $\overline{P_{x-in}}$ and $\overline{P_{x-out}}$ are boolean variables, which are set to *True* with the probabilities P_{in}, P_{out}, P_{x-in} and P_{x-out}, and *False* otherwise. The value of P_{in} (P_{out}) determines the rate at which robots join (leave) a chain. The value of P_{x-in} (P_{x-out}) determines the rate at which robots join (leave) the prey extending structure.

- **Explore Chain**: move along a chain towards its tail or towards the nest. By default, an explorer moves towards a chain's tail. In case a robot becomes an explorer by leaving a chain, it first moves back to the nest, and then turns around the nest to follow a (possibly) different chain or probabilistically decides to start a new chain by itself. LEDs are off. Active motor schemas: Move_perpendicular($g_{MP} = 1$), Adjust_distance($g_{AD} = 10$), Avoid_collisions($g_{AC} = 0.1$).

- **Join Chain**: the LEDs are activated with the appropriate colour, depending on the colour of the previous chain neighbour. For what concerns the movement of a chain member, we employ three different strategies:

  - Static strategy: In this simplest case a chain member does not move at all. Active motor schemas: none.
  - Aligning strategy: A chain member avoids collisions, aligns itself with respect to its neighbours (see Figure 4.1) and adjusts its distance with respect to its previous neighbour to roughly 27 cm. The combination of the last two motor schemas improves the length of the chains and guarantees a visual connection of the chain neighbours. A side effect of this is that loops within the chain are avoided. Active motor schemas: Adjust_distance($g_{AD} = 10$), Align($g_{AL} = 1$), Avoid_collisions($g_{AC} = 0.1$).
  - Moving strategy: The moving strategy is an extension of the aligning strategy. One motor schema is added, and affects only the last member in a chain. While for the aligning strategy the last chain member only adjusts its distance with respect to its precedent, in the moving strategy the motor schema to move perpendicularly is employed in addition. In this way the last chain member turns around the previous one. As the rest of the chain continuously tries to align itself, the movement of the last member results in a clockwise movement of the whole chain around the nest. The last chain member acts as a kind of leader that triggers the chain as a whole to move in a circle around the nest. The angular speed of a chain is determined by the speed of its last member and its length. Active motor schemas: Move_perpendicular($g_{MP} = 1$), Adjust_distance($g_{AD} = 10$), Align($g_{AL} = 1$), Avoid_collisions($g_{AC} = 0.1$).

- **Recovery-P**: Move back to the nest and rest. Emit a sound signal for 30 s. LEDs are off. Active motor schemas: Move_perpendicular($g_{MP} = 1$), Adjust_distance($g_{AD} = 10$), Avoid_collisions($g_{AC} = 0.1$).

- **Extend Prey**: The LEDs are activated with red. Keep a distance of 60 cm to the closest red object perceived. This is either the prey or another robot in the

prey extending structure. Active motor schemas: Adjust_distance($g_{AD} = 10$), Avoid_collisions($g_{AC} = 0.1$).

For assembly and transport the following behaviours are used:

- **Assemble**: two different behaviours are used, depending on whether the behaviour is executed on the real robot or in simulation:
  - Real robot: A feed-forward artificial neural network—a single-layer perceptron—takes input from the camera as well as sensor readings from the left-front and right-front proximity sensors. The network's output is used to control the speed and status of the tracks and the connection mechanisms. The network is trained to let the robot approach and grasp nearby objects that have activated their LEDs in red. Initially, the prey is the only object having activated its LEDs in red. Upon connection, a robot activates its own LEDs in red. Therefore, it becomes itself an object with which to establish a connection. A detailed description of the behaviour can be found in (Groß et al., 2006a).
  - Simulation: As the connection mechanism of the robot was not accurately simulated at the time we conducted our experiments, we used a simple hand written control algorithm that provides a similar behaviour as the neural network stated above. The simulated robot approaches the object towards which it attempts a connection, tries to connect, and given the distance and the angle towards the object it has a given probability that the assembly is successful. If the attempt to connect is not successful it moves backwards, and tries to connect again.
- **Transport**: if a sound signal is perceived the robot rests. Otherwise, if a chain member is perceived, the robot orients its chassis towards the closest chain member, which indicates the direction to the nest, and starts pulling. A detailed description of the behaviour can be found in (Groß et al., 2006b). If no chain member is perceived we use two different behaviours on the real robot and in simulation:
  - Real robot: A simple recurrent neural network is fed with input from the robot's force and torque sensors. The force sensor indicates the mismatch between the robot's own direction of motion and the motion of other robots it is connected with; moreover, it is influenced by the prey if it is moved by other robots. The torque sensor indicates whether *stagnation* is present. The network's output is used to control the speed of the tracks and the desired orientation of the chassis. A detailed description of the behaviour can be found in (Groß and Dorigo, 2004; Groß et al., 2006b).

  - Simulation: As these forces could not be accurately simulated, a robot in simulation that does not perceive the path does not move at all.

Finally, a recovery behaviour, similar to the one used for the path formation, is used for the assembly as well. The only difference is that no sound signal is emitted.

- **Recovery-A**: Move back to the nest and rest. LEDs are off. Active motor schemas: Move_perpendicular($g_{MP} = 1$), Adjust_distance($g_{AD} = 10$), Avoid_collisions($g_{AC} = 0.1$).

4.5 Behaviour Transitions

The following set of conditions trigger behaviour-transitions:

- **Search Chain → Explore Chain**: if a chain member (this includes the nest) is perceived. Note that a robot in the `Search Chain` state does not react to the perception of the prey, unless the prey extension mechanism is used.
- **Search Chain → Extend Prey**: if the prey, but no chain member is perceived, the robot joins the prey extending structure with probability P_{x-in} per time step.
- **Explore Chain → Search Chain**: if no chain member is perceived any more.
- **Explore Chain → Join Chain**: (i) if the prey is not perceived and the tail of a chain is reached (i.e., only one chain member is perceived), the robot joins the chain with probability P_{in} per time step, or (ii) if the prey is perceived at a distance $> 35\,\text{cm}$.
- **Explore Chain → Assemble**: if the prey is detected at a distance $< 35\,\text{cm}$.
- **Join Chain → Search Chain**: if the previous chain neighbour is no longer perceived.
- **Join Chain → Explore Chain**: if a chain member is situated at the tail of a chain, it leaves the chain with probability P_{out} per time step.
- **Join Chain → Recovery-P**: if the prey is perceived at a very close distance (i.e., less than 5 cm), which only occurs if the prey is transported towards the chain member.
- **Extend Prey → Search Chain**: if no red object is perceived any more, or with probability P_{x-out} per time step.

- **Extend Prey → Explore Chain**: if a chain member is perceived.
- **Recovery-P → Search Chain**: if $T_{recovery} = 30\,\mathrm{s}$ has elapsed.
- **Assemble → Recovery-A**: if the robot does not succeed in connecting to an object within $T_{as} = 90\,\mathrm{s}$.
- **Assemble → Transport-Target**: if the robot succeeds in connecting to an object.
- **Recovery-A → Search Chain**: if $T_{recovery} = 30\,s$ has elapsed.

Chapter 5

Chain Controller: Experiments in Simulation

In this chapter we present the results obtained in the TwoDee simulation environment for the chain controller. This includes the three chain strategies static, aligning and moving, as well as a version of each of these with the prey extension mechanism. After describing the experimental setup in Section 5.1, we explain the method used for parameter selection and give an overview of the performance in Section 5.2, and then present the results with the selected parameters for the different tests performed in Section 5.3.

5.1 Experimental Setup

We employ a bounded arena of size $5\,\mathrm{m} \times 5\,\mathrm{m}$. The task consists in forming a path between two locations in the environment, the nest and the prey, to assemble to the prey and to transport it back towards the nest. The nest is placed in the centre of the arena, and the prey is placed towards one of the corners. Obstacles are cubes with a side length of $0.5\,\mathrm{m}$ (i.e., one obstacle occupies 1% of the arena). An instance of the task is defined by the triplet (N, D, O), where:

- N is the robot group size,
- D is the distance between nest and prey (in meters),
- O is the number of obstacles in the environment.

The initial position and orientation of the robots, as well as the positions of the obstacles, are chosen randomly.

The primary performance measure is the *completion time*. We distinguish the three completion times for the three subtasks path formation (T_p), assembly (T_a),

and transport (T_t). For practical reasons, we allow a maximum completion time of 10,000 seconds. If this time does not suffice to establish a path, the trial is stopped and considered to be a failure. As a second performance measure we use the *success rate*, which we define as the ratio of successful trials. Again we distinguish three success rates, one for each subtask.

5.2 Parameter Selection

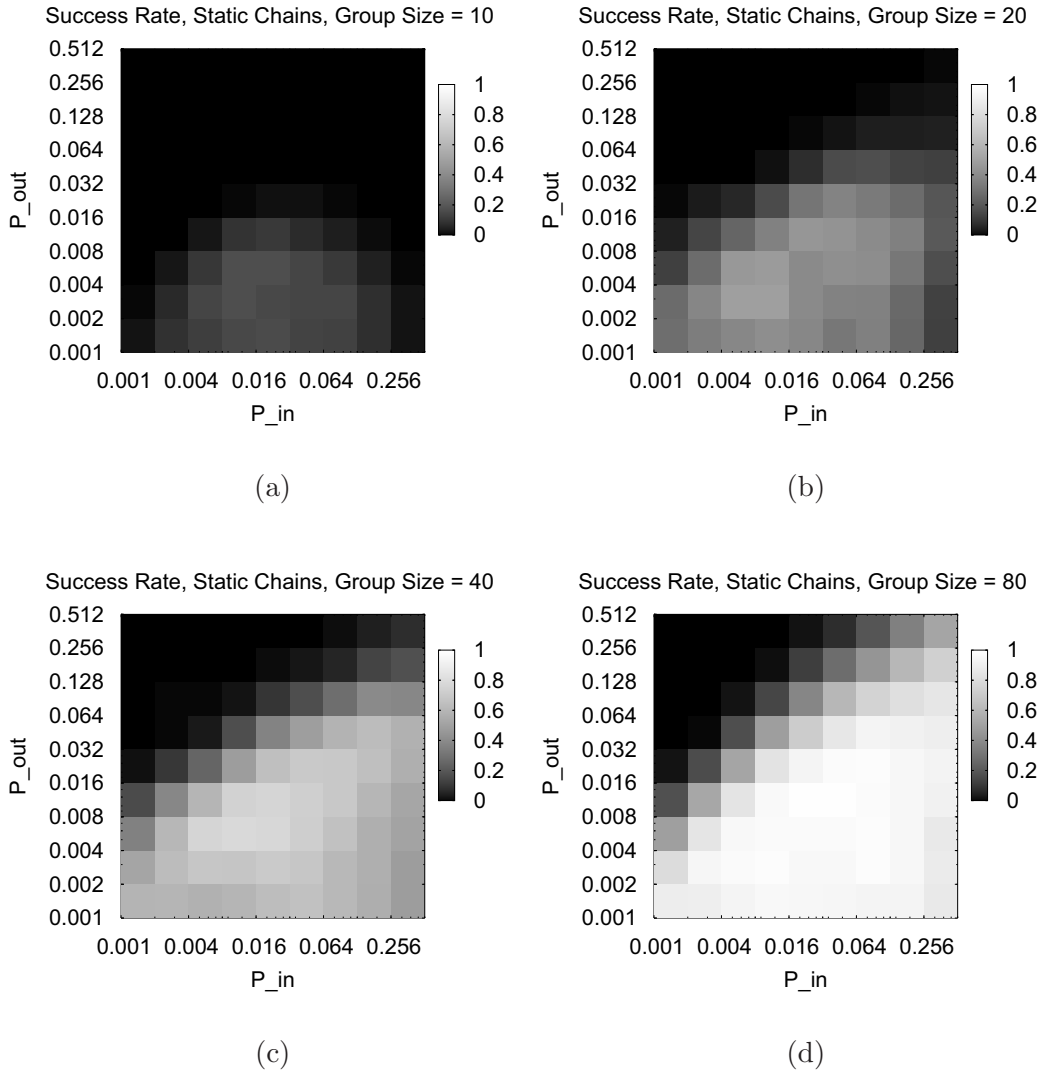
The overall behaviour of our controllers is a function of the two parameters P_{in} and P_{out}, and in the cases where the prey extension mechanism is employed, additionally of the two parameters P_{x-in} and P_{x-out}. The values of these determine the probability per time step at which the robots join or leave the path forming and prey extending structures respectively. To assess the general impact of these probabilities we have conducted a parameter study. For each probability we examined ten values defined by $0.001 * 2^x$, with $x \in \{0, 1, 2, 3, \ldots, 9\}$, resulting in 10 candidates in the range $[0.001, 0.512]$. We first discuss the parameter landscape and select a parameter set for the probabilities of P_{in} and P_{out} in Section 5.2.1. Then, based on these two selected parameters, we discuss the parameter landscape and select a parameter set for the probabilities of P_{x-in} and P_{x-out} in Section 5.2.2.

5.2.1 Parameters P_{in} and P_{out}

Figures 5.1, 5.2 and 5.3 show surface plots of the success rate of the P_{in}/P_{out} parameter landscapes for a group of (a) 10 robots, (b) 20 robots, (c) 40 robots and (d) 80 robots. All trials were conducted in an environment without obstacles and a nest to prey distance of 3 meters.

The parameter landscapes are qualitatively similar for the three chain variants, and the aligning and moving chains reach higher success rates than the static chains. In general, the success rate is higher when $P_{in} > P_{out}$. Low values of P_{in} result in a rather patient behaviour: in most cases a single chain is formed slowly. For rather difficult setups with a small group size and a large distance between nest and prey, the highest success rates are reached for low values P_{in}, because this allows the formation of a single chain, which is the only possibility to solve the task. For high values of P_{in}, several chains are formed fast and in parallel. The second parameter, P_{out}, determines the stability of the formed chains, directly influencing their lifetime and the frequency of chain disbandment. High values of P_{out} lead to an impatient behaviour where robots joining a chain quickly leave it. In general, the results are qualitatively similar for other group sizes, distances, and when obstacles are added to the environment.

The study of the parameter landscapes gives a general idea of the impact of the parameters and of the overall performance of the controllers. To further analyse



(a) (b)

(c) (d)

Figure 5.1: Surface plots of the success rates of the parameter landscapes when changing the two probability parameters P_{in} and P_{out} (100 observations per parameter combination), for static chains and a group of (a) 10 robots, (b) 20 robots, (c) 40 robots and (d) 80 robots. All experiments were conducted in an environment without obstacles and a nest to prey distance of 3 meters. The axes of the parameters are plotted in logarithmic scale. The lighter the surface the higher is the success rate.

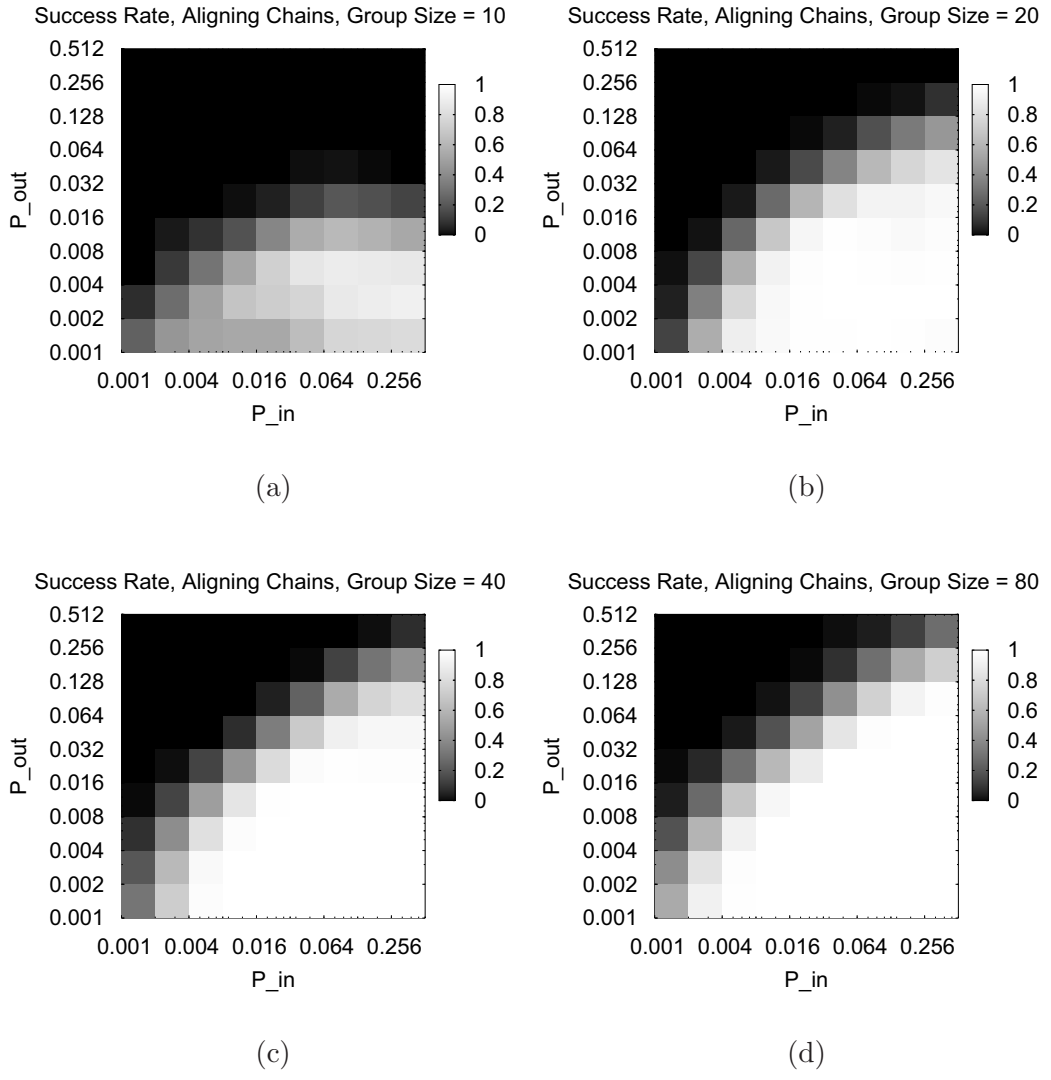


Figure 5.2: Surface plots of the success rates of the parameter landscapes when changing the two probability parameters P_{in} and P_{out} (100 observations per parameter combination), for aligning chains and a group of (a) 10 robots, (b) 20 robots, (c) 40 robots and (d) 80 robots. All experiments were conducted in an environment without obstacles and a nest to prey distance of 3 meters. The axes of the parameters are plotted in logarithmic scale. The lighter the surface the higher is the success rate.

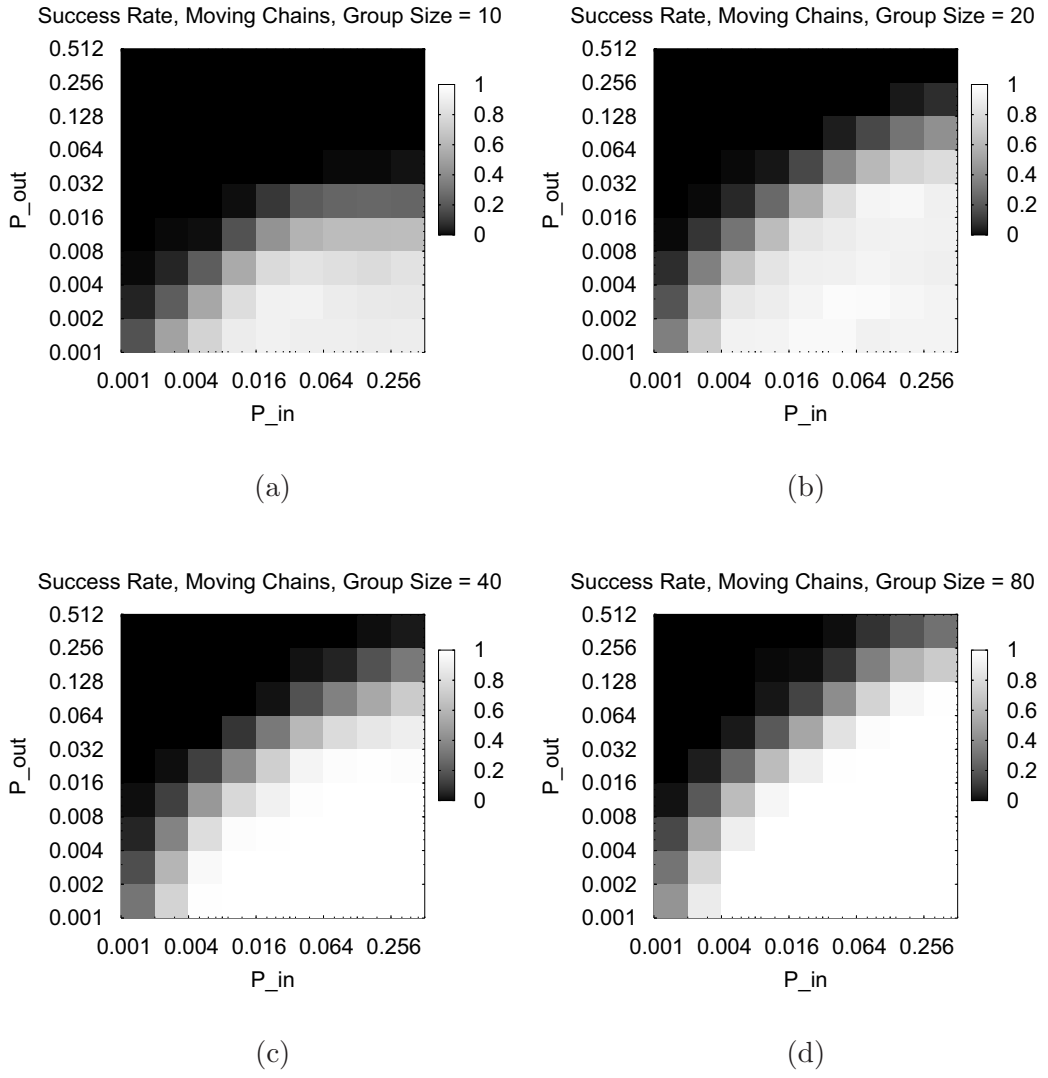


Figure 5.3: Surface plots of the success rates of the parameter landscapes when changing the two probability parameters P_{in} and P_{out} (100 observations per parameter combination), for moving chains and a group of (a) 10 robots, (b) 20 robots, (c) 40 robots and (d) 80 robots. All experiments were conducted in an environment without obstacles and a nest to prey distance of 3 meters. The axes of the parameters are plotted in logarithmic scale. The lighter the surface the higher the success rate.

the performance under a wide range of experimental settings we have to select one parameter combination for each controller. For this purpose, we employed the racing method by Birattari et al. (2002); Birattari (2005). This method is an efficient way to determine a good parameter set: it sequentially evaluates a set of candidate parameter configurations, discarding the worst performing candidates as soon as statistical evidence is gathered against them. The process is stopped when only one candidate is left, or when the candidates have been tested on all problem instances.

In our case, a candidate configuration is a set of two values for the parameters P_{in} and P_{out}. We considered the same ten values defined by $0.001 * 2^x$, with $x \in \{0, 1, 2, 3, \ldots, 9\}$, resulting in 100 candidates. Candidate configurations were tested on 27 experimental setups obtained considering all the possible combinations of values for N, D, and O, with $N \in \{10, 20, 40\}$, $D \in \{2, 2.5, 3\}$, and $O \in \{0, 10, 20\}$. Each setup is initialized in 100 different ways, obtained varying the initial positions of the robots and the obstacle configuration. It is important to note that while for $N = 20$ and $N = 40$ a solution exists for any value of D and O, this is not necessarily the case for $N = 10$. In this case, in fact, the 10 robots can form a linear structure of approximately 3 m. Therefore, when $O = 0$ a solution exists, while when $O = 10$ or $O = 20$ the existence of the solution depends on the actual disposition of the obstacles in the arena.

For each controller there were between three and five candidates left for which no statistically significant difference was found based on the completion time. Among these candidates we chose those with the highest success rates. The performance obtained with these candidates is reported in Table 5.1.

Confirming our previous observation, the aligning and moving chains clearly outperform the static ones, and successfully form a path in most problem instances where this is possible. A success rate of 100% is not reachable because the problem mix includes tasks that cannot be solved by 10 robots.

The lower success rate of the static chains is mainly due to three reasons. First, the static chains are not straight. Therefore, in general, they cover shorter distances from the nest than the two dynamic chain strategies whose structures can stretch

Table 5.1: The selected parameter sets for P_{in} and P_{out} based on the outcome of a racing algorithm on 27 experimental setups obtained considering all the possible combinations of values for N, D, and O, with $N \in \{10, 20, 40\}$, $D \in \{2, 2.5, 3\}$, and $O \in \{0, 10, 20\}$. Each setup was initialized in 100 different ways.

Strategy	P_{in}	P_{out}	Success Rate	Median Completion Time
Static	0.064	0.008	62.2 %	4142 seconds
Aligning	0.128	0.004	89.7 %	1066 seconds
Moving	0.128	0.004	91,9 %	1181 seconds

over longer distances thanks to the aligning mechanism which is used in both of them. Second, the two dynamic chain strategies allow some motion to the chain members which leads to an exploration of the arena even of chains that are already formed. Third and last, the static chains have a higher risk to create loops in the form of a successive order of three chain members. The risk for this to happen is much lower for the other two chain strategies as they lead the chains to align.

5.2.2 Parameters P_{x-in} and P_{x-out}

Given the parameter selection for P_{in} and P_{out} we performed a similar study for the parameters P_{x-in} and P_{x-out}. While P_{in} and P_{out} represent the rate at which robots join and leave a chain, P_{x-in} and P_{x-out} determine the rate at which robots join and leave the prey extending structure.

Figures 5.4, 5.5 and 5.6 show surface plots of the success rate of the P_{x-in}/P_{x-out} parameter landscapes for a group of (a) 10 robots, (b) 20 robots, (c) 40 robots and (d) 80 robots. All trials were conducted in an environment without obstacles and a nest to prey distance of 3 meters.

The parameter landscapes are qualitatively similar for the three chain variants. Again, the aligning and moving chains reach higher success rates than the static chains. In general, the performance is high throughout all tested parameters and the landscapes are flat when compared to those presented in the previous section. This is due to the fact that the parameters P_{in} and P_{out} are fixed according to the selected values in Table 5.1. Therefore, the basic performance is in general good. The following observations can be made concerning the effect of the parameters:

- For the most successful parameter combinations the value of P_{x-in} exceeds the one of P_{x-out}. The ratio of $\frac{P_{x-in}}{P_{x-out}}$ is in the range $[2, 16]$. A higher rate to join the prey extending structure rather than leaving it is beneficial because otherwise the structure is unstable as robots are leaving it faster than they join it.

- The performance drops for values of $P_{x-in} > 0.128$ because that would lead a robot to join the prey extending structure virtually immediately when perceiving the prey or a prey extending robot. If too many robots join the prey extending structure then the path from the nest formed by chains is neglected.

- The performance drops for values of $P_{x-out} > 0.032$ because the prey extending structure becomes unstable.

- The performance drops for values of $P_{x-in} < 0.004$ because then robots join the prey extending structure too rarely.

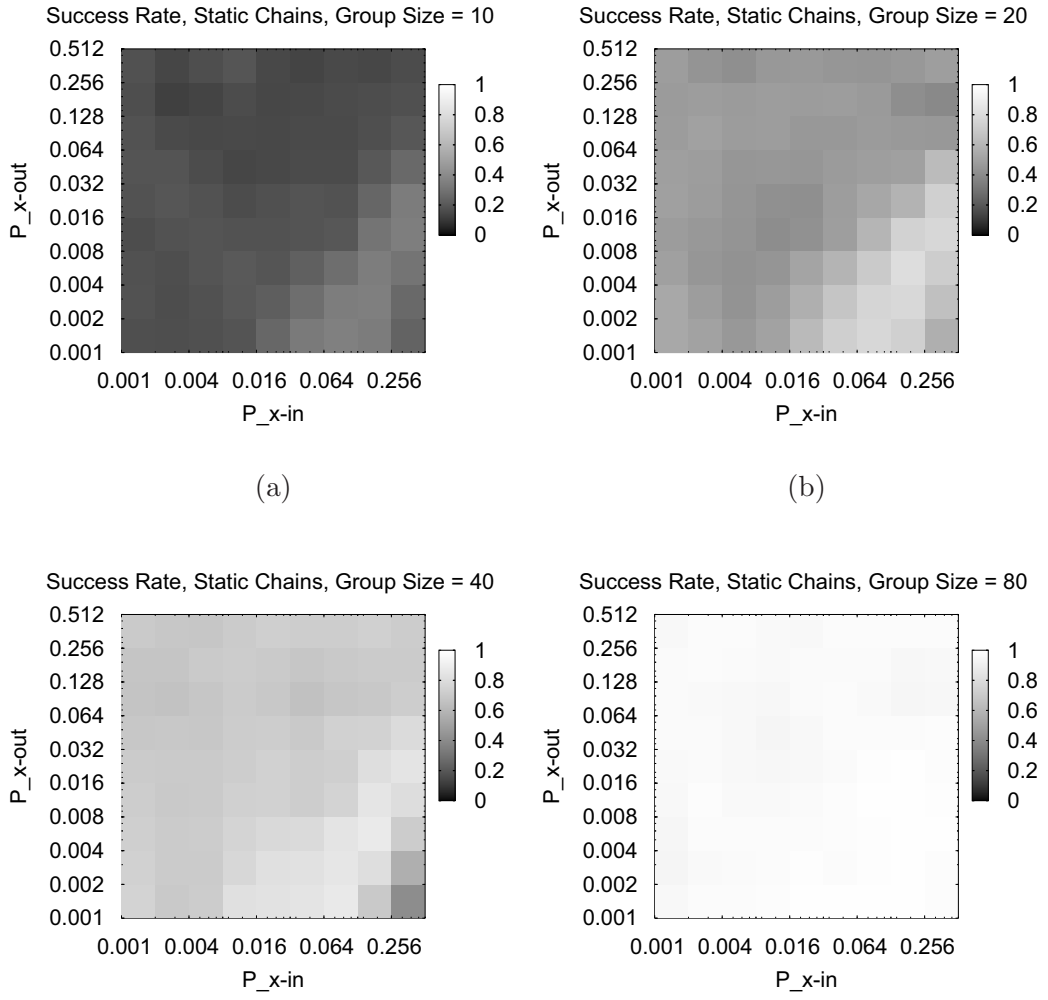


Figure 5.4: Surface plots of the success rates of the parameter landscapes when changing the two probability parameters P_{x-in} and P_{x-out} (100 observations per parameter combination), for static chains and a group of (a) 10 robots, (b) 20 robots, (c) 40 robots and (d) 80 robots. All experiments were conducted in an environment without obstacles and a nest to prey distance of 3 meters. The axes of the parameters are plotted in logarithmic scale. The lighter the surface the higher is the success rate. The parameters P_{in} and P_{out} are fixed according to Table 5.1.

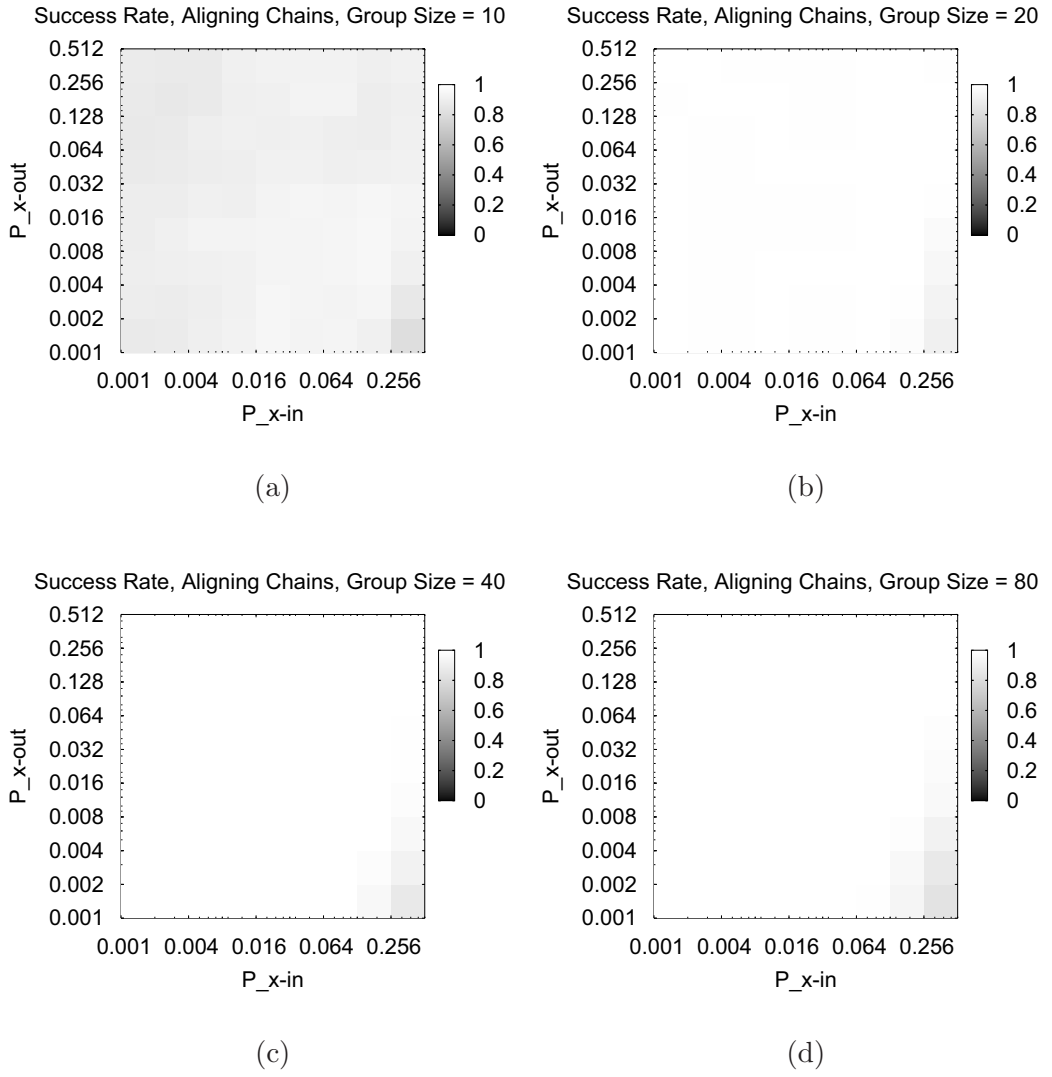


Figure 5.5: Surface plots of the success rates of the parameter landscapes when changing the two probability parameters P_{x-in} and P_{x-out} (100 observations per parameter combination), for aligning chains and a group of (a) 10 robots, (b) 20 robots, (c) 40 robots and (d) 80 robots. All experiments were conducted in an environment without obstacles and a nest to prey distance of 3 meters. The axes of the parameters are plotted in logarithmic scale. The lighter the surface the higher is the success rate. The parameters P_{in} and P_{out} are fixed according to Table 5.1.

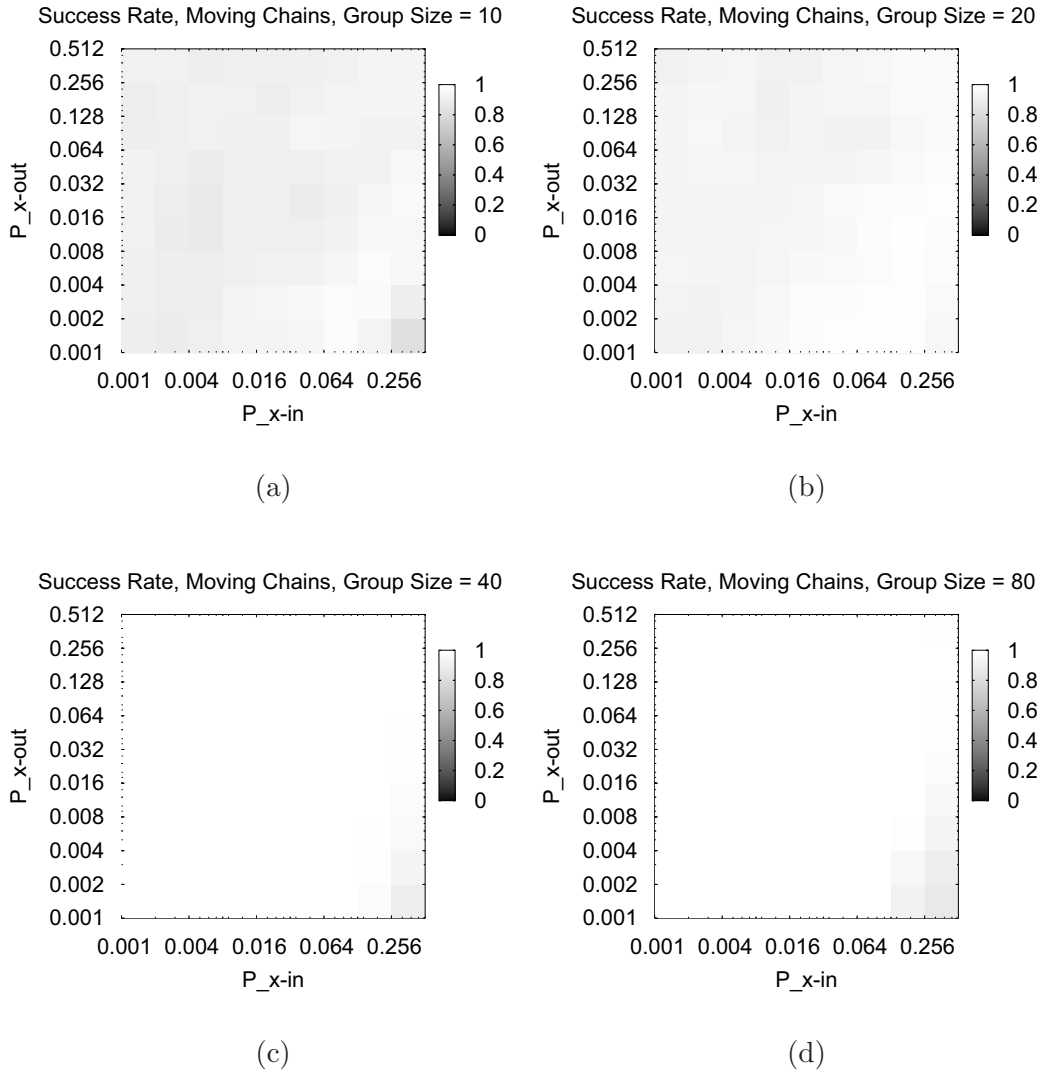


Figure 5.6: Surface plots of the success rates of the parameter landscapes when changing the two probability parameters P_{x-in} and P_{x-out} (100 observations per parameter combination), for moving chains and a group of (a) 10 robots, (b) 20 robots, (c) 40 robots and (d) 80 robots. All experiments were conducted in an environment without obstacles and a nest to prey distance of 3 meters. The axes of the parameters are plotted in logarithmic scale. The lighter the surface the higher is the success rate. The parameters P_{in} and P_{out} are fixed according to Table 5.1.

Table 5.2: The selected parameter sets for P_{x-in} and P_{x-out} based on the outcome of a racing algorithm on 27 experimental setups obtained considering all the possible combinations of values for N, D, and O, with $N \in \{10, 20, 40\}$, $D \in \{2, 2.5, 3\}$, and $O \in \{0, 10, 20\}$. Each setup was initialized in 100 different ways. The parameters P_{in} and P_{out} are fixed according to Table 5.1.

Strategy	P_{x-in}	P_{x-out}	Success Rate	Median Completion Time
Static	0.004	0.002	63.1 %	3156 seconds
Aligning	0.008	0.002	93.9 %	762 seconds
Moving	0.008	0.001	94.3 %	548 seconds

The parameters selected from the racing algorithm for further evaluation are summarized in Table 5.2.

5.3 Performance Evaluation

In the previous section we selected parameter sets for the three different chain controllers and for the prey extension mechanism. Based on these parameters we performed extensive tests under various conditions to evaluate the performance of the control mechanisms at hand. These experiments are the subject of this section. First, we give an overview of the performance in Section 5.3.1, and then show the results of systematic experiments under various conditions:

- In a difficulty test we vary the distance D between the nest and the prey in the range $[1\,\mathrm{m}, 3\,\mathrm{m}]$ (Section 5.3.2).
- In a scalability test we vary the number of robots N in the range $[10, 200]$ (Section 5.3.3).
- In an obstacle test we vary the number of obstacles O in the range $[0, 30]$ and additionally test two predefined obstacle environments (Section 5.3.4).
- In a set of robustness tests we vary the noise of various sensors (Section 5.3.5).
- In a set of fault tolerance tests we vary the fraction of robots that suffer from individual failure by disabling various sensors or actuators (Section 5.3.6).

5.3.1 General Evaluation

In this section we discuss the general performance of the three chain strategies and the prey extension mechanism. We describe first the general behaviour of the three chain strategies and the prey extension mechanism in Section 5.3.1.1. Then we

discuss the branching, that is, the number of formed chains in Section 5.3.1.2, and finally the environment exploration in Section 5.3.1.3.

5.3.1.1 Behaviour

To illustrate the behaviour of the chain strategies Figures 5.7, 5.8 and 5.9 display a sequence of snapshots from typical simulation trials for the static, aligning and moving strategy respectively.[1]

A group of $N = 20$ robots is randomly distributed in the environment and the nest to prey distance is $D = 2\,\text{m}$. In all three cases the robot group succeeds to form a path, assemble to the prey, and transport it back to the nest. In the following we give a short explanation of the observed behaviours:

- **Static strategy** (Figure 5.7): After an initial random walk several robots discover the nest and quickly form three chains simultaneously (b,c). The chains have an arbitrary form. Usually they are not straight, as can in particular be seen by the chain of eleven robots that form a path from nest to prey (d). At the time that the path is formed there is one other chain that remains. Given that there is no explicit communication among the robots about the successful formation of a path there is no explicit mechanism to disaggregate the other chain. However, following the normal probabilistic mechanism, robots leave the chain, move back to the nest, and follow the path forming chain towards the prey and connect to the prey (e). Once that two robots are connected to the prey the transport can start (f). The transport is made more difficult as one of the robots in the path forming chain is very close to the wall (g). Therefore, the transport takes a rather long time and the structure of prey transporting robots grows continually (h) up to eleven robots once the prey is successfully transported to the nest (i).

- **Aligning strategy** (Figure 5.8): Again three chains are formed simultaneously (b). The aligning strategy leads the chain members to align with respect to their neighbours, which results in the chains to be straight when compared to the static strategy. The path forming chain (c) consists of five robots, six less than for the trial with the static strategy, and equal to the minimum number of robots required for the given distance of $D = 2\,\text{m}$ between nest and prey. Once that two robots are assembled to the prey the transport starts (e). Due to the reduced number of robots in the path the distance that the prey has to be transported is effectively reduced as well. Therefore, the transport is very fast when compared to the static strategy. Seven robots are assembled to the prey when the transport is finished (i).

[1] A selection of movies can be found at `http://iridia.ulb.ac.be/supp/IridiaSupp2008-014`.

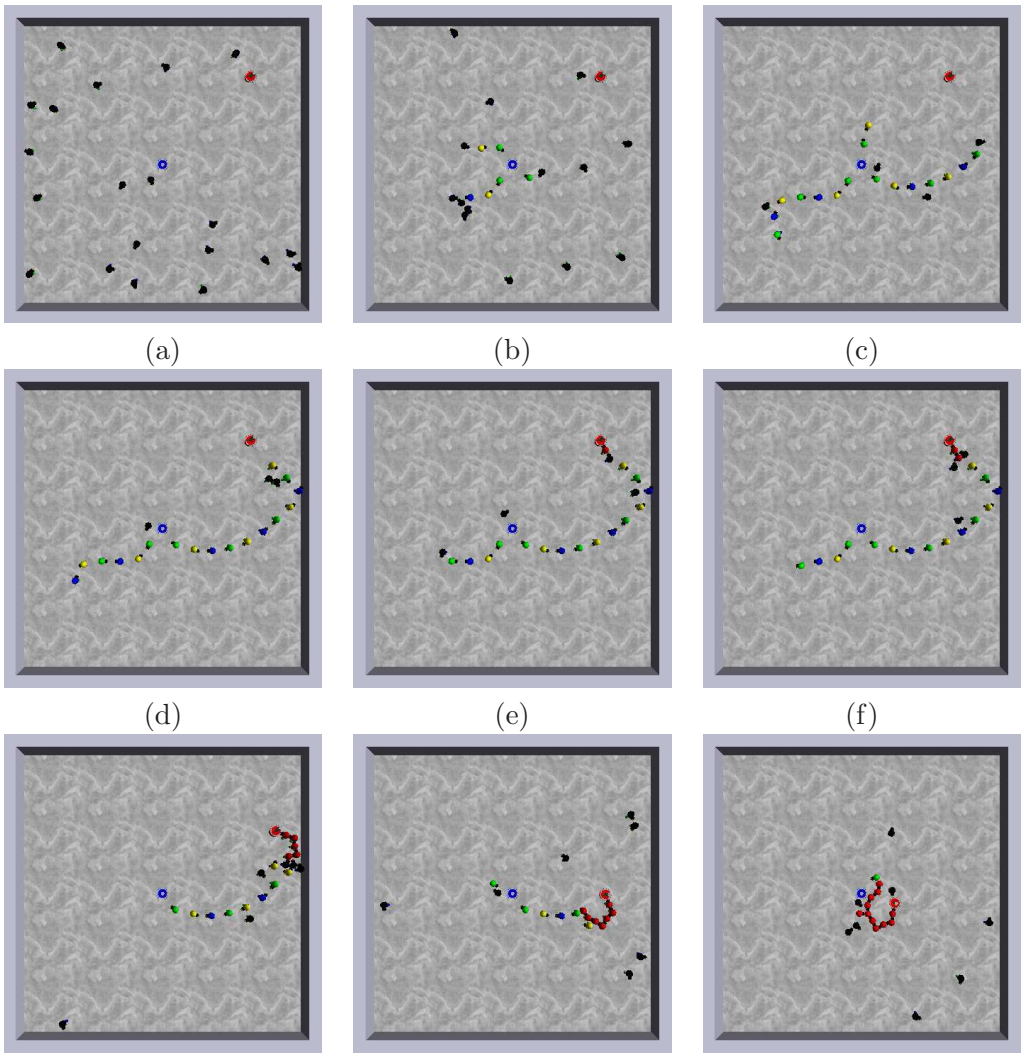


Figure 5.7: Sequence of images taken for a simulation trial with group size $N = 20$ *s-bots* and distance $D = 2\,\mathrm{m}$ between the nest (blue cylindrical object in the centre) and the prey (red cylindrical object on the top right), when using the chain controller with the static strategy: (a) $t = 0\,\mathrm{s}$, (b) $t = 54\,\mathrm{s}$, (c) $t = 231\,\mathrm{s}$, (d) $t = 395\,\mathrm{s}$, (e) $t = 462\,\mathrm{s}$, (f) $t = 528\,\mathrm{s}$, (g) $t = 808\,\mathrm{s}$, (h) $t = 1025\,\mathrm{s}$, and (i) $t = 1387\,\mathrm{s}$.

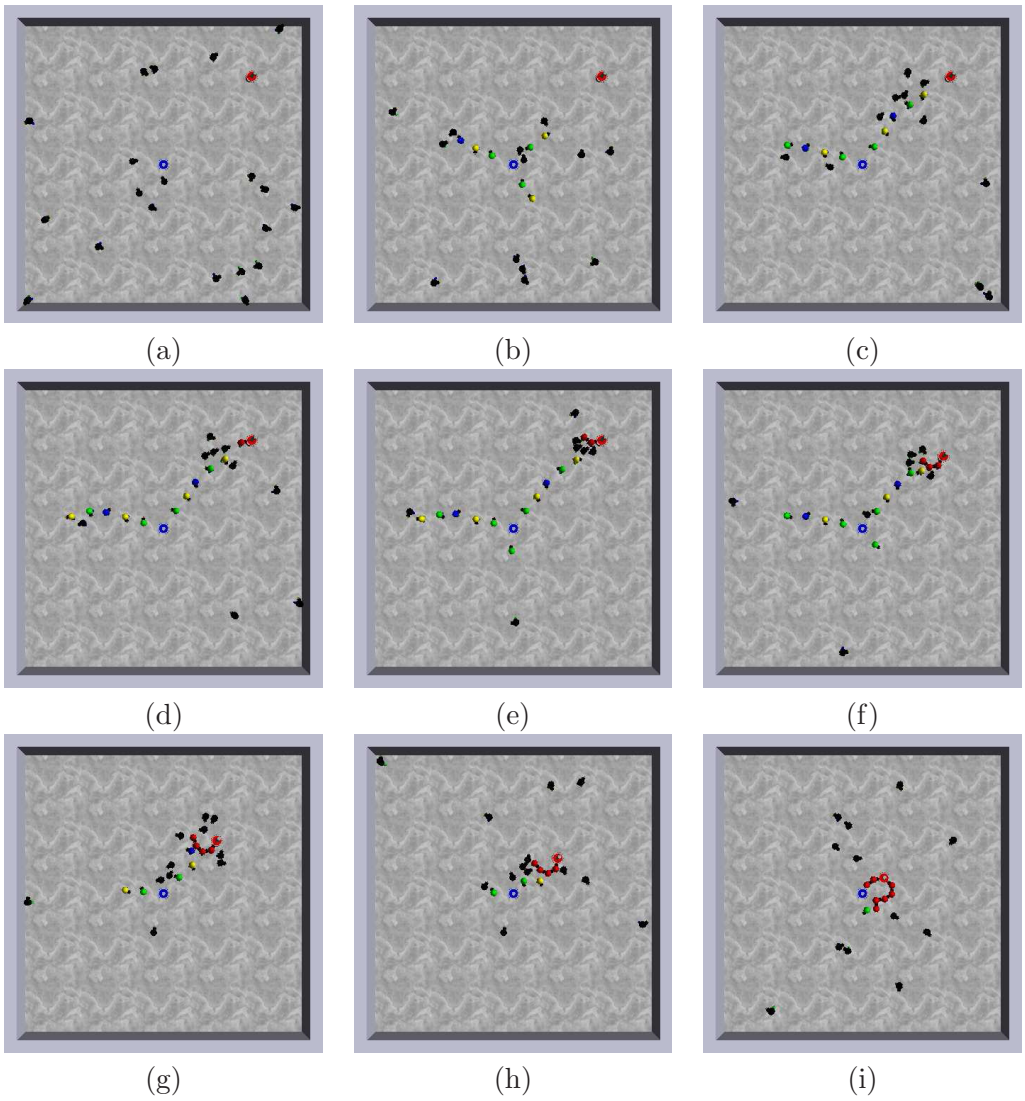


Figure 5.8: Sequence of images taken for a simulation trial with group size $N = 20$ *s-bots* and distance $D = 2\,\mathrm{m}$ between the nest (blue cylindrical object in the centre) and the prey (red cylindrical object on the top right), when using the chain controller with the aligning strategy: (a) $t = 0\,\mathrm{s}$, (b) $t = 88\,\mathrm{s}$, (c) $t = 131\,\mathrm{s}$, (d) $t = 141\,\mathrm{s}$, (e) $t = 168\,\mathrm{s}$, (f) $t = 217\,\mathrm{s}$, (g) $t = 284\,\mathrm{s}$, (h) $t = 309\,\mathrm{s}$, and (i) $t = 374\,\mathrm{s}$.

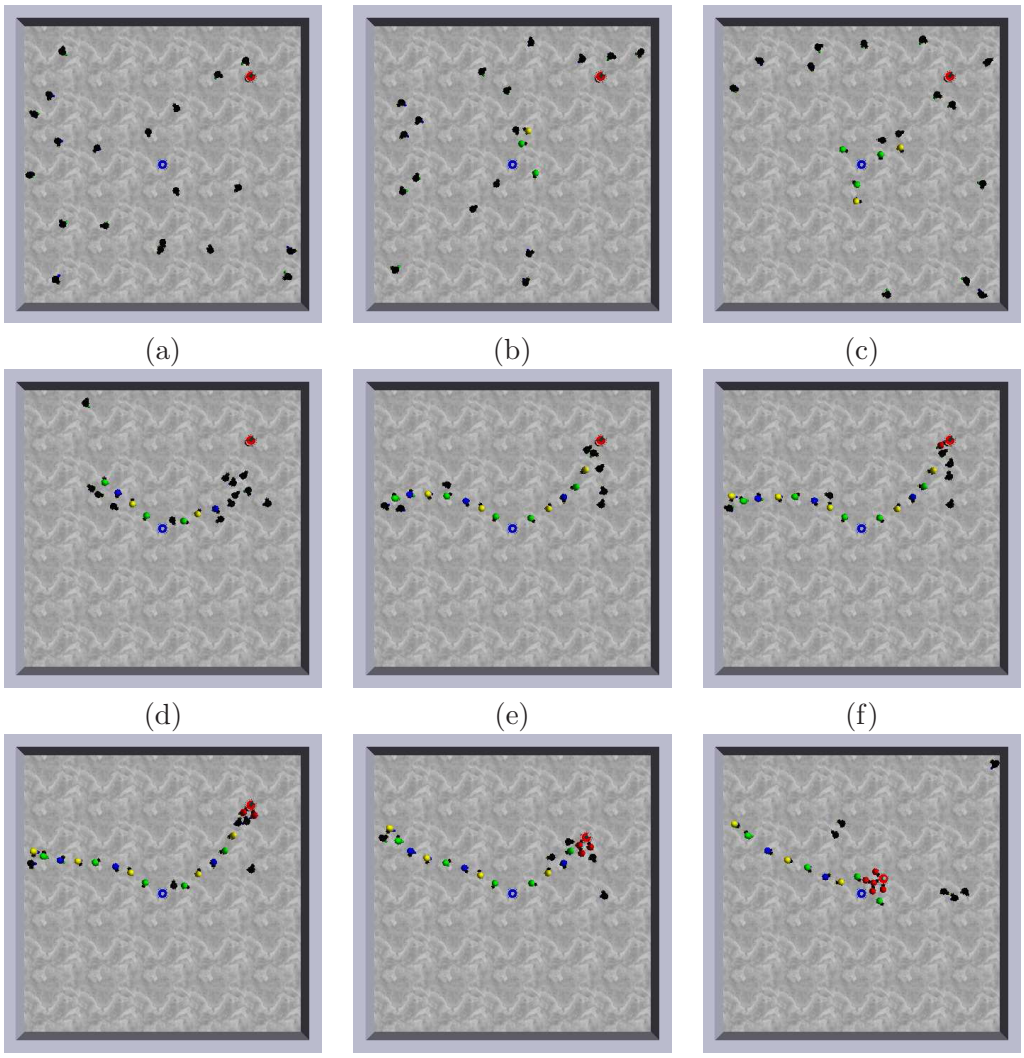


Figure 5.9: Sequence of images taken for a simulation trial with group size $N = 20$ *s-bots* and distance $D = 2\,\mathrm{m}$ between the nest (blue cylindrical object in the centre) and the prey (red cylindrical object on the top right), when using the chain controller with the moving strategy: (a) $t = 0\,\mathrm{s}$, (b) $t = 35\,\mathrm{s}$, (c) $t = 90\,\mathrm{s}$, (d) $t = 190\,\mathrm{s}$, (e) $t = 235\,\mathrm{s}$, (f) $t = 244\,\mathrm{s}$, (g) $t = 255\,\mathrm{s}$, (h) $t = 283\,\mathrm{s}$, and (i) $t = 345\,\mathrm{s}$.

- **Moving strategy** (Figure 5.9): Initially there are two (b), and then three chains (c) that are formed. Similarly to the aligning strategy, chain members align with respect to their neighbours, and therefore the chains are straight when compared to the static strategy. The moving strategy employs a stricter threshold angle of the aligning motor schema than the aligning strategy. This is required for chain members to follow the overall movement of a chain, and leads the chains to be straighter than is the case for the aligning strategy. The movement of the chains leads to a "scanning" of the environment until the prey is found and a path to it is formed (e). As for the aligning strategy, this path consists of five robots. The speed of the transport is similar to the one of the aligning strategy. Five robots are assembled to the prey when the transport is finished (i).

Figures 5.10, 5.11 and 5.12 display similar sequences of snapshots from typical simulation trials for the static, aligning and moving strategy respectively when the prey extension mechanism is employed.

Again, a group of $N = 20$ robots is randomly distributed in the environment and the nest to prey distance is $D = 2\,\mathrm{m}$. Also in these three cases the robot group succeeds in forming a path, assembling to the prey, and transporting it back to the nest.

The basic chain formation algorithm and the individual strategies are the same as when the the prey extension mechanism is not used. Prey extension is a mechanism that functions in parallel to chain formation. The part of the controller that is responsible for forming chains remains completely unchanged. Therefore, the differences between the three individual strategies remain the same as stated above for the case when no prey extension mechanism is employed.

The prey extension mechanism works in a very simple way. A robot that searches for a chain and finds the prey (or a robot activating its LEDs in the same colour as the prey) has a fixed probability to activate its LEDs to enhance the area from which the prey can be perceived. For a robot that is part of a chain or exploring one it makes no difference whether it actually perceives the prey or just another robot that activates its LEDs in the colour of the prey: It is attracted and forms a connection. In this way paths are in fact formed from both the nest as well as the prey.

When the prey extension mechanism is used, the aligning and the moving strategies have an important advantage when compared to the static one: In case the prey extending structure is encountered the chains can follow it until they actually encounter the prey. Chains controlled by the static strategy cannot do that as the individual chain members do not move at all.

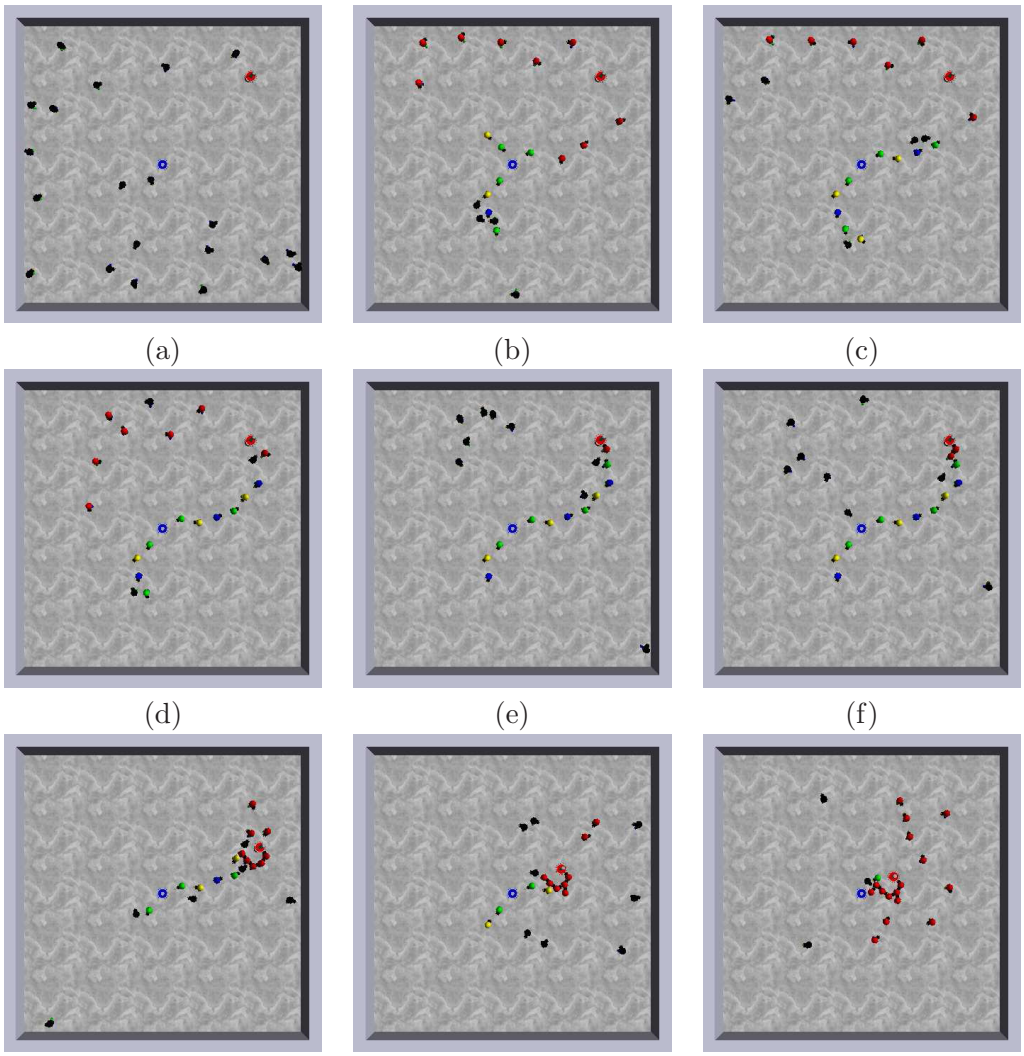


Figure 5.10: Sequence of images taken for a simulation trial with group size $N = 20$ *s-bots* and distance $D = 2\,\text{m}$ between the nest (blue cylindrical object in the centre) and the prey (red cylindrical object on the top right), when using the chain controller with the static strategy and the prey extension mechanism: (a) $t = 0\,\text{s}$, (b) $t = 60\,\text{s}$, (c) $t = 130\,\text{s}$, (d) $t = 256\,\text{s}$, (e) $t = 315\,\text{s}$, (f) $t = 326\,\text{s}$, (g) $t = 488\,\text{s}$, (h) $t = 590\,\text{s}$, and (i) $t = 772\,\text{s}$.

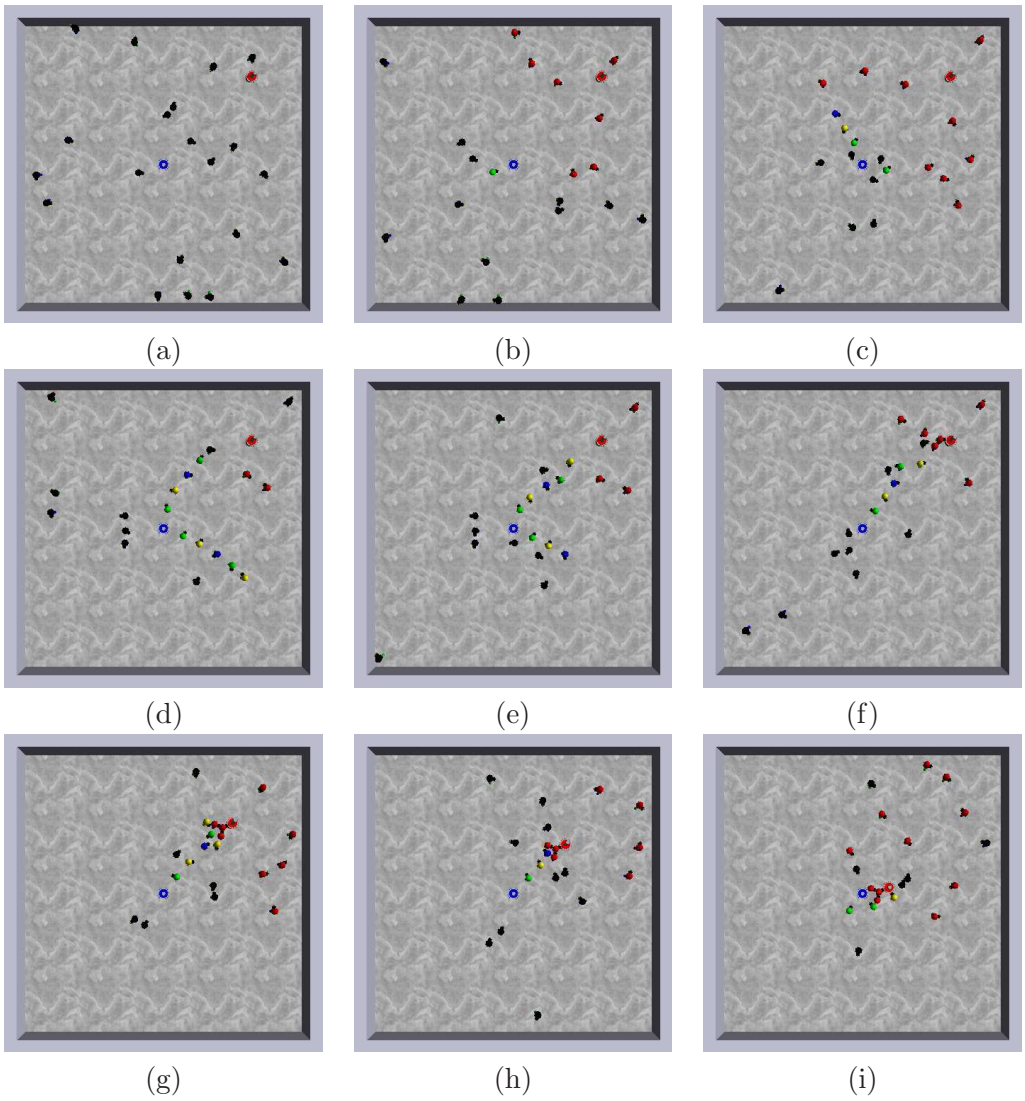


Figure 5.11: Sequence of images taken for a simulation trial with group size $N = 20$ *s-bots* and distance $D = 2\,\text{m}$ between the nest (blue cylindrical object in the centre) and the prey (red cylindrical object on the top right), when using the chain controller with the aligning strategy and the prey extension mechanism: (a) $t = 0\,\text{s}$, (b) $t = 11\,\text{s}$, (c) $t = 140\,\text{s}$, (d) $t = 249\,\text{s}$, (e) $t = 286\,\text{s}$, (f) $t = 385\,\text{s}$, (g) $t = 510\,\text{s}$, (h) $t = 646\,\text{s}$, and (i) $t = 704\,\text{s}$.

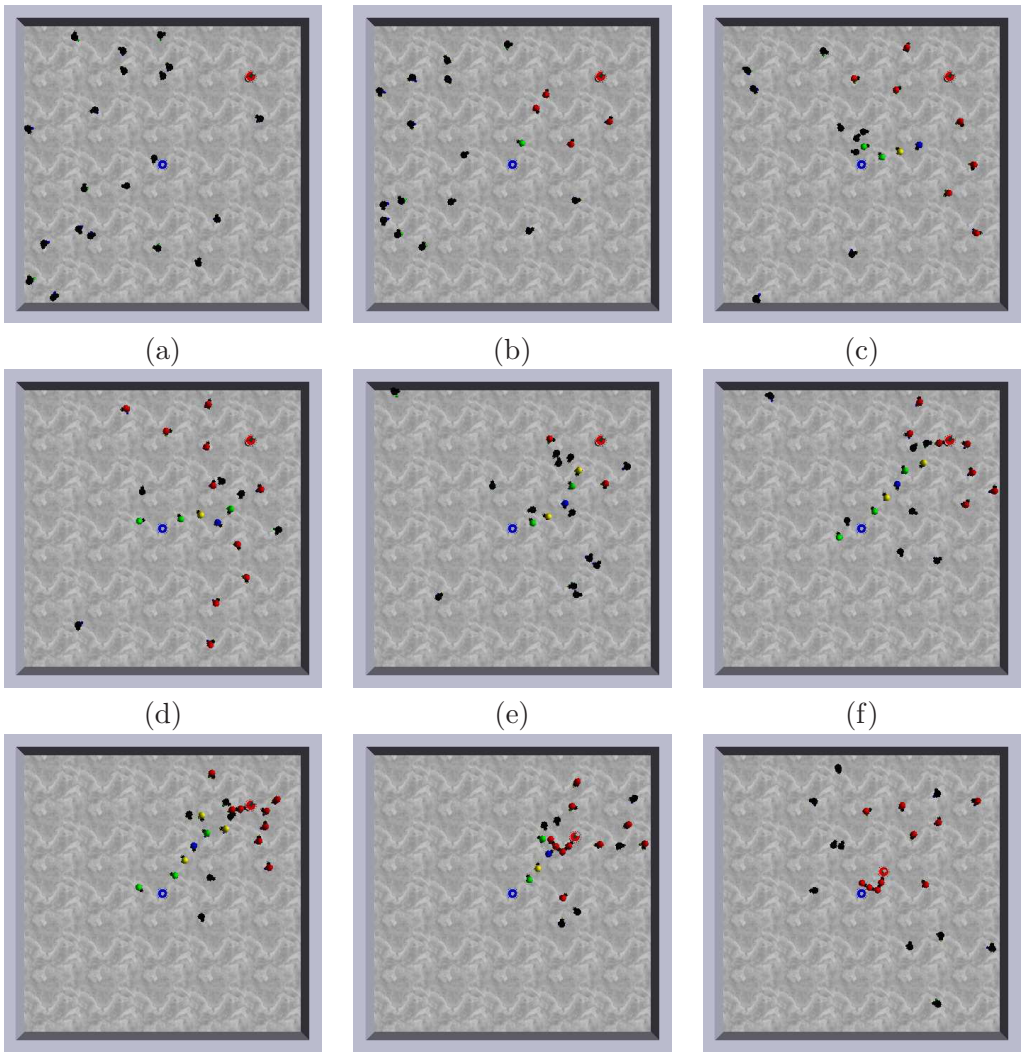


Figure 5.12: Sequence of images taken for a simulation trial with group size $N = 20$ *s-bots* and distance $D = 2\,\mathrm{m}$ between the nest (blue cylindrical object in the centre) and the prey (red cylindrical object on the top right), when using the chain controller with the moving strategy and the prey extension mechanism: (a) $t = 0\,\mathrm{s}$, (b) $t = 24\,\mathrm{s}$, (c) $t = 82\,\mathrm{s}$, (d) $t = 146\,\mathrm{s}$, (e) $t = 176\,\mathrm{s}$, (f) $t = 261\,\mathrm{s}$, (g) $t = 328\,\mathrm{s}$, (h) $t = 413\,\mathrm{s}$, and (i) $t = 533\,\mathrm{s}$.

5.3.1.2 Branching

For the selected parameter values, Figure 5.13 shows the number of chains formed simultaneously in an environment without obstacles and without prey at time $t = 3600\,\text{sec}$.

For group sizes with $N < 40$ in most cases there are two chains formed simultaneously. For the static strategy this is still true for larger groups, but for the other two strategies, three or four chains are formed more frequently. Note that the number of simultaneously formed chains depends not only on the parameters selected, but also on the size of the nest. A new chain can only be formed if there is enough room. If a robot is situated at the nest and perceives a chain it will follow the chain instead of forming a new one. For the given diameter of the nest there are not more than three, and in very rare cases four chains formed simultaneously. There is simply not enough space for more. However, this is not a general limitation of the chain algorithm. By employing a larger nest diameter this limitation could be overcome. Figure 5.13 also displays the degree of branching. For group sizes $N = 10$ and $N = 20$ there is a very low degree of branching, so that a chain starting from the nest usually has one sole ending. For larger group sizes the branching is more pronounced. Each chain splits approximately once into two branches.

5.3.1.3 Exploration

The basic requirement for the chains to find the prey and to connect it to the nest is to explore the environment. In order to investigate the exploration capabilities of the chain formation algorithm we performed a test in an environment without prey. As for the other trials, the robots are initially randomly positioned. We define a position in the environment to be explored if it is within a radius of 40 cm of an explorer or chain robot. Initially the arena is completely unexplored. Figures 5.14, 5.15 and 5.16 show the percentage of the explored arena for the three chaining strategies.

Among the three strategies, the static one performs worst, and the two others reach a similar performance. For all three strategies the exploration rate grows continuously, indicating that the mechanism of randomly leaving a chain and joining one at a different position leads to a continuous exploration so that unexplored areas of the environment are still found after hours of experimentation. This is particularly important for smaller group sizes. For larger group sizes the environment gets quickly explored already for the fact that the density of the robots is sufficiently high to cover a high percentage of the environment. For the two dynamic strategies and for group sizes of $N \geq 40$ the environment gets entirely explored for all trials.

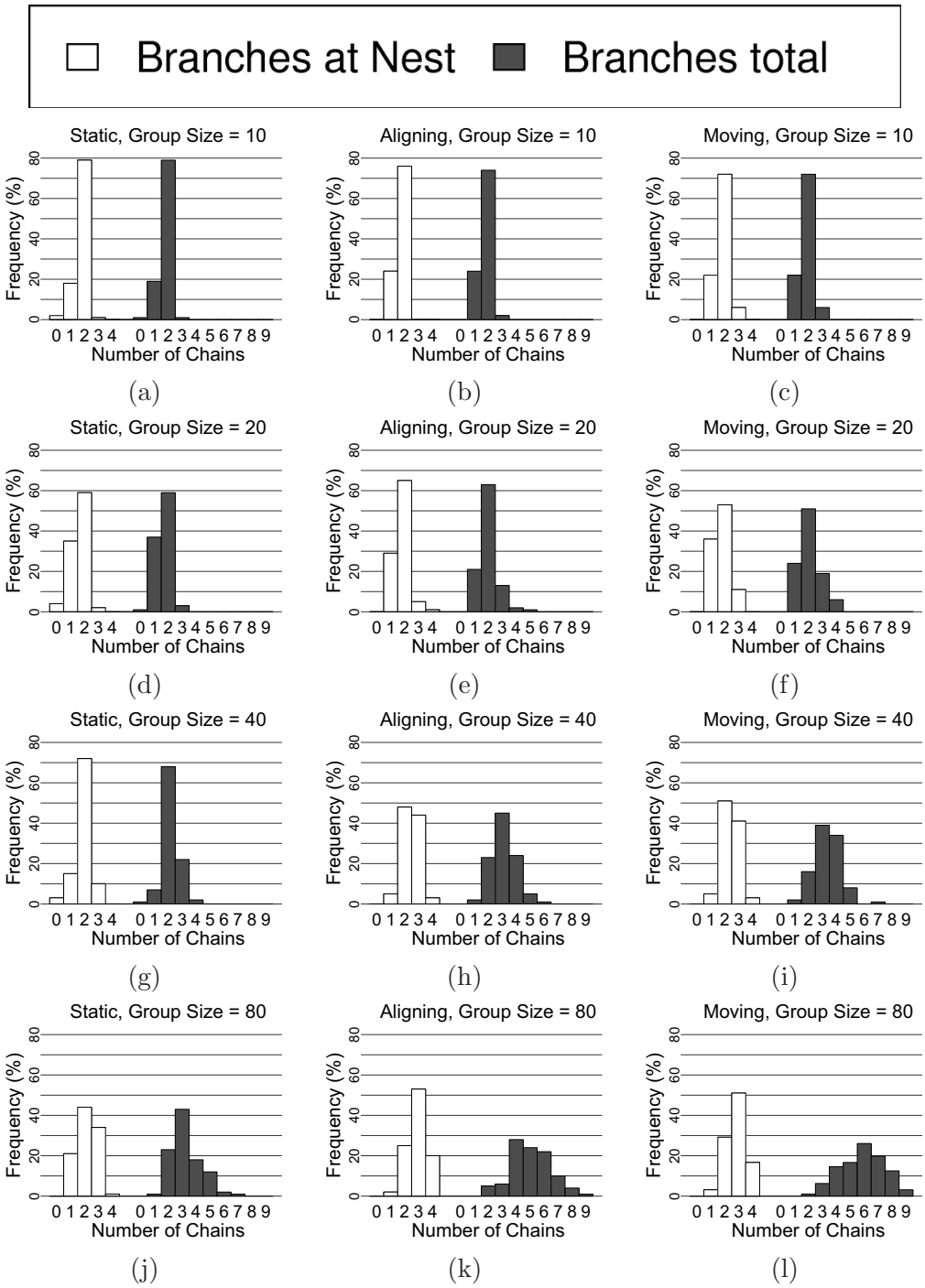


Figure 5.13: The histograms show the frequency of the number of chains formed for the static (left), aligning (centre) and moving (right) chain strategies and a group size of (abc) 10, (def) 20, (ghi) 40 and (jkl) 80 robots in 100 evaluations. The results are collected at $t = 3600$ sec in an obstacle free environment with no prey. The white bars on the left of each histogram indicate the number of chains formed directly from the nest. The grey bars on the right also include splits in chains that lead to different branches of the same chain.

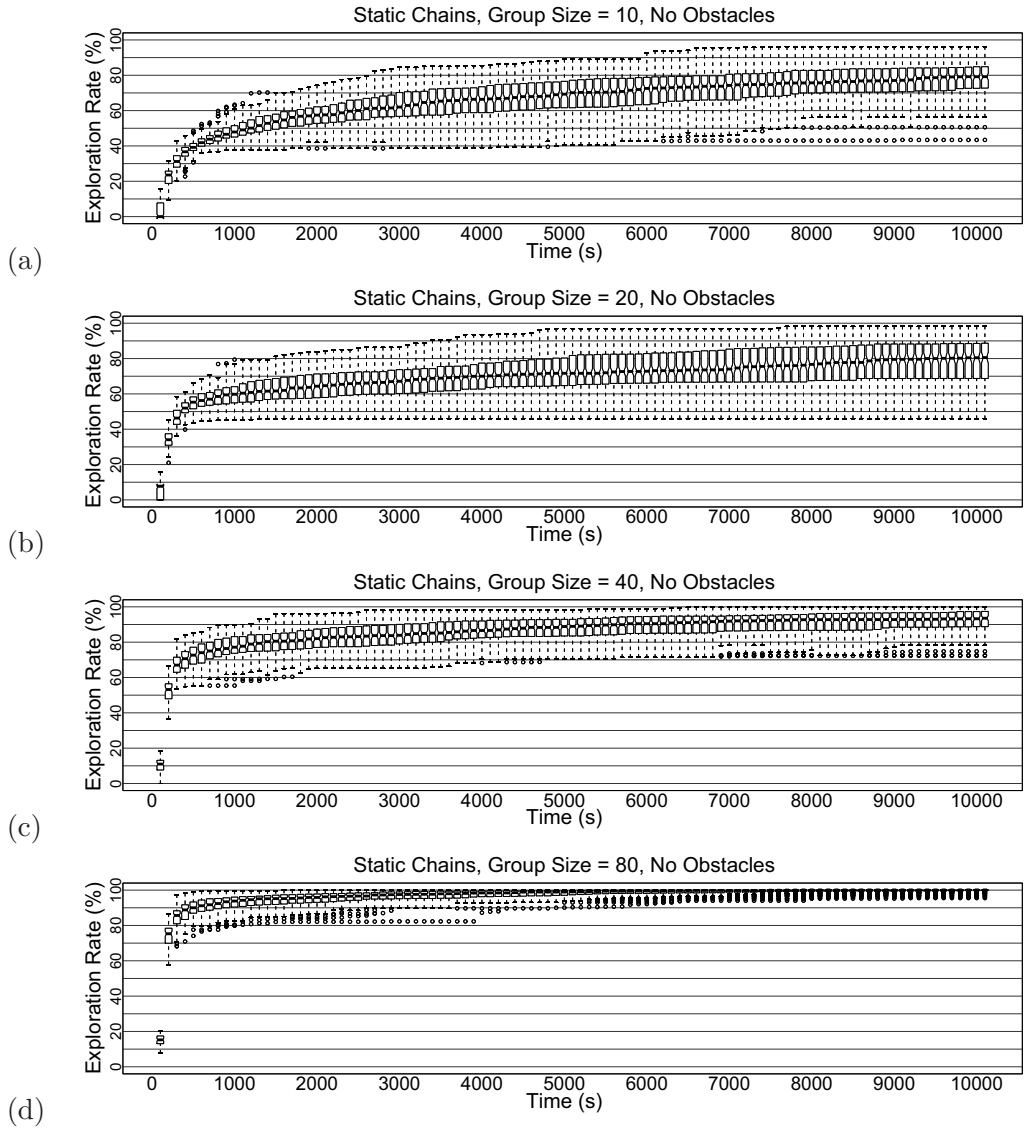
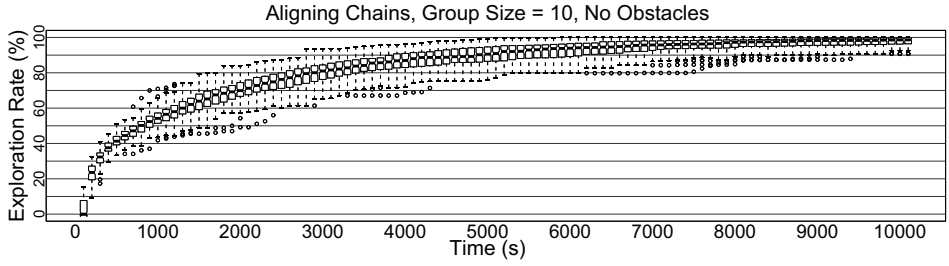
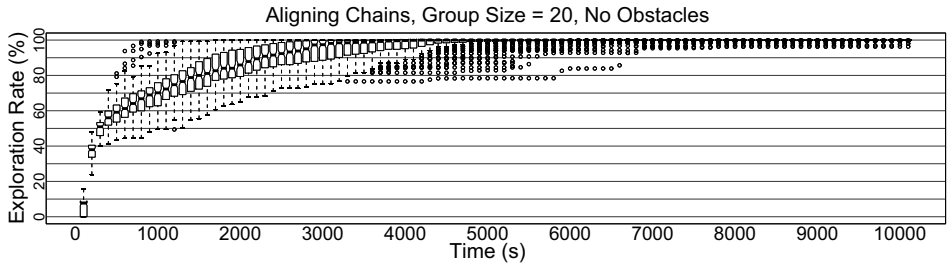


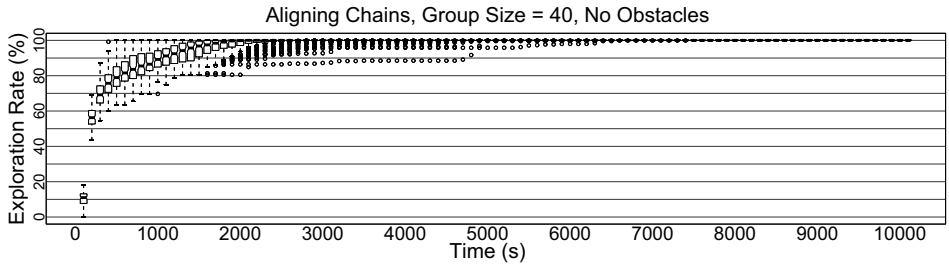
Figure 5.14: The box-and-whisker-plots (Becker et al., 1988) show 100 evaluations of the exploration rate for a group of (a) 10 robots, (b) 20 robots, (c) 40 robots and (d) 80 robots using the static chain controller in an obstacle free environment. Given that no prey is present in the arena the results do not rely on the prey extension mechanism. Boxes represent the inter-quartile range of the data, while the horizontal bars inside the boxes mark the median values. The whiskers extend to the most extreme data points within 1.5 of the inter-quartile range from the box. The empty circles mark the outliers.



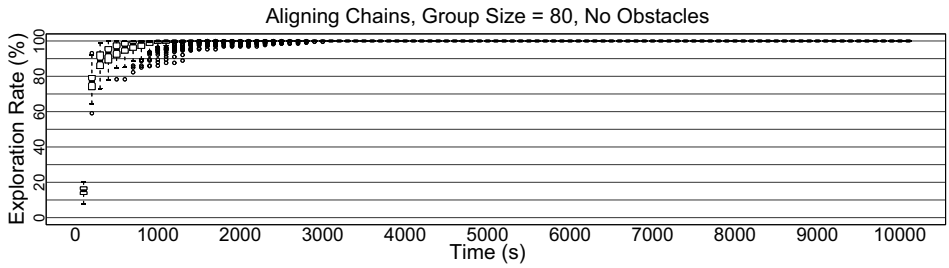
(a)



(b)



(c)



(d)

Figure 5.15: The box-and-whisker-plots (Becker et al., 1988) show 100 evaluations of the exploration rate for a group of (a) 10 robots, (b) 20 robots, (c) 40 robots and (d) 80 robots using the aligning chain controller in an obstacle free environment. Given that no prey is present in the arena the results do not rely on the prey extension mechanism. See the caption of Figure 5.14 for an explanation of the box-and-whisker plots.

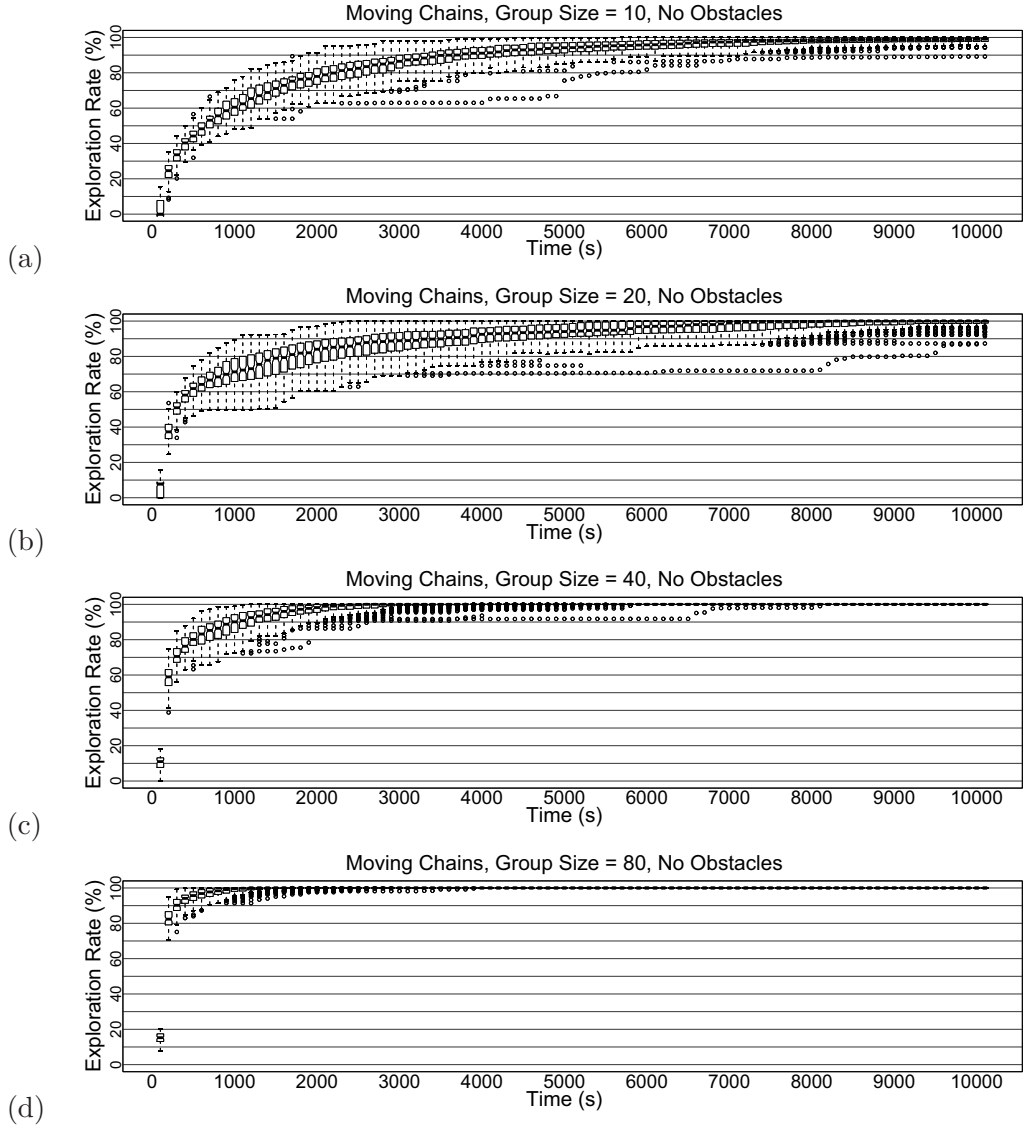


Figure 5.16: The box-and-whisker-plots (Becker et al., 1988) show 100 evaluations of the exploration rate for a group of (a) 10 robots, (b) 20 robots, (c) 40 robots and (d) 80 robots using the aligning chain controller in an obstacle free environment. Given that no prey is present in the arena the results do not rely on the prey extension mechanism. See the caption of Figure 5.14 for an explanation of the box-and-whisker plots.

Table 5.3: Success rates for all strategies for the difficulty test of the chain controller and a group size of $N = 20$ robots. The three values represent percentages of the success rate from 100 trials for path formation, assembly and transport in this order.

D	Static	Aligning	Moving	Static-X	Aligning-X	Moving-X
1.0	100/100/99	100/100/100	100/100/100	100/100/98	100/100/100	100/100/100
1.2	96/95/87	100/100/100	100/100/99	98/98/95	100/100/100	100/100/100
1.4	94/91/83	100/100/100	100/100/100	98/98/89	100/100/100	100/100/100
1.6	83/77/68	100/100/98	100/100/99	100/98/82	100/100/100	100/100/100
1.8	83/79/68	100/100/100	100/100/99	96/95/84	100/100/99	100/100/100
2.0	83/79/63	100/100/99	100/100/100	91/90/73	100/100/99	100/100/99
2.2	72/64/50	99/99/98	99/99/97	91/91/73	100/100/98	100/100/99
2.4	65/60/40	100/100/100	97/97/95	82/78/67	100/100/98	100/100/98
2.6	64/59/45	100/100/95	97/94/85	76/71/47	100/100/95	97/97/93
2.8	48/44/29	100/98/90	94/93/81	80/73/59	100/100/96	98/98/95
3.0	40/38/28	100/100/95	91/90/84	61/60/34	100/100/94	99/99/96

5.3.2 Difficulty Test

We test the ability of the chaining algorithm to cope with changes in the difficulty for the task by varying the distance D between nest and prey in the range $[1\,\mathrm{m}, 3\,\mathrm{m}]$. Figure 5.17 shows the completion times of the subtasks (a) path formation, (b) assembly, and (c) transport, and Figure 5.18 shows the normalized completion times, that is, the completion time divided by the prey distance ($\frac{T}{D}$), for the same subtasks and situations. Table 5.3 displays the success rates.

For all three subtasks the aligning and moving strategies outperform the static one no matter if the prey extension mechanism is used or not. The reasons for this are the same for subtasks assembly and transport as those previously cited for subtask path formation: For the static strategy the formed path is not straight. Therefore, the path is effectively longer and robots take longer to move along the path to assemble to the prey, and to transport it back to the nest.

For subtask path formation the prey extension mechanism appears to be particularly useful when the task is more difficult. While the strategies without prey extension mechanism are faster for small distances, the prey extension mechanism decreases the completion time for path formation when the prey distance is large. For subtasks assembly and transport the prey extension mechanism does not have a significant influence on the completion times.

For growing prey distances, the time to form a path grows more than linearly (Figure 5.18a). This is due to the fact that the area to be explored grows quadratically with the distance. The time to assemble to (Figure 5.18b) and transport (Figure 5.18c) the prey grows linearly with the prey distance as the difficulty for these subtasks grows linearly as well.

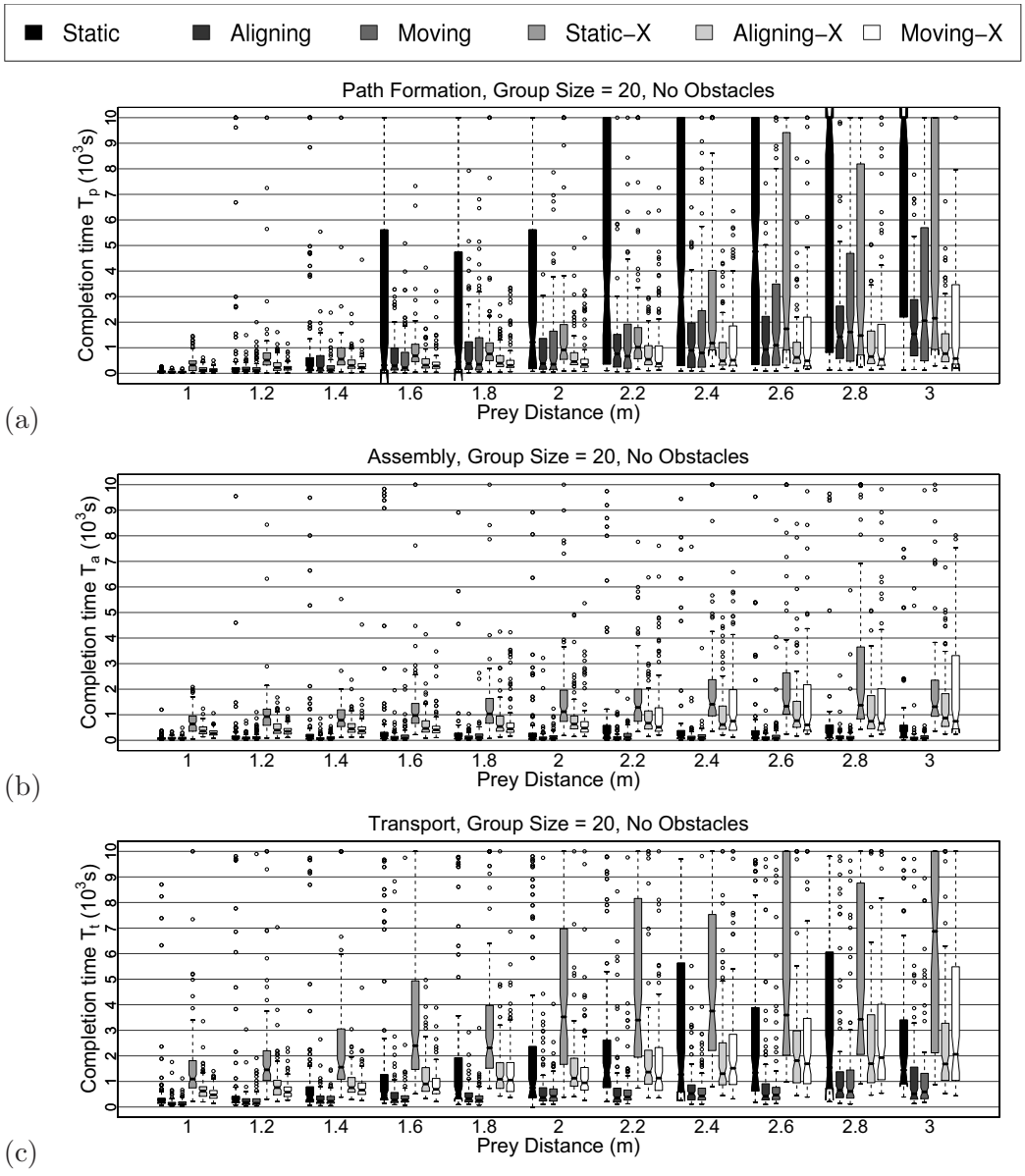


Figure 5.17: The box-and-whisker-plots (Becker et al., 1988) show the results taken from 100 trials of the completion time for subtasks (a) path formation, (b) assembly, and (c) transport when changing the nest to prey distance for a group 20 robots using the different chain strategies with and without prey extension mechanism in an environment without obstacles. Note that subtask assembly is only taken into account in case a path has been formed, and respectively subtask transport is only taken into account in case the assembly was successful. See the caption of Figure 5.14 for an explanation of the box-and-whisker plots.

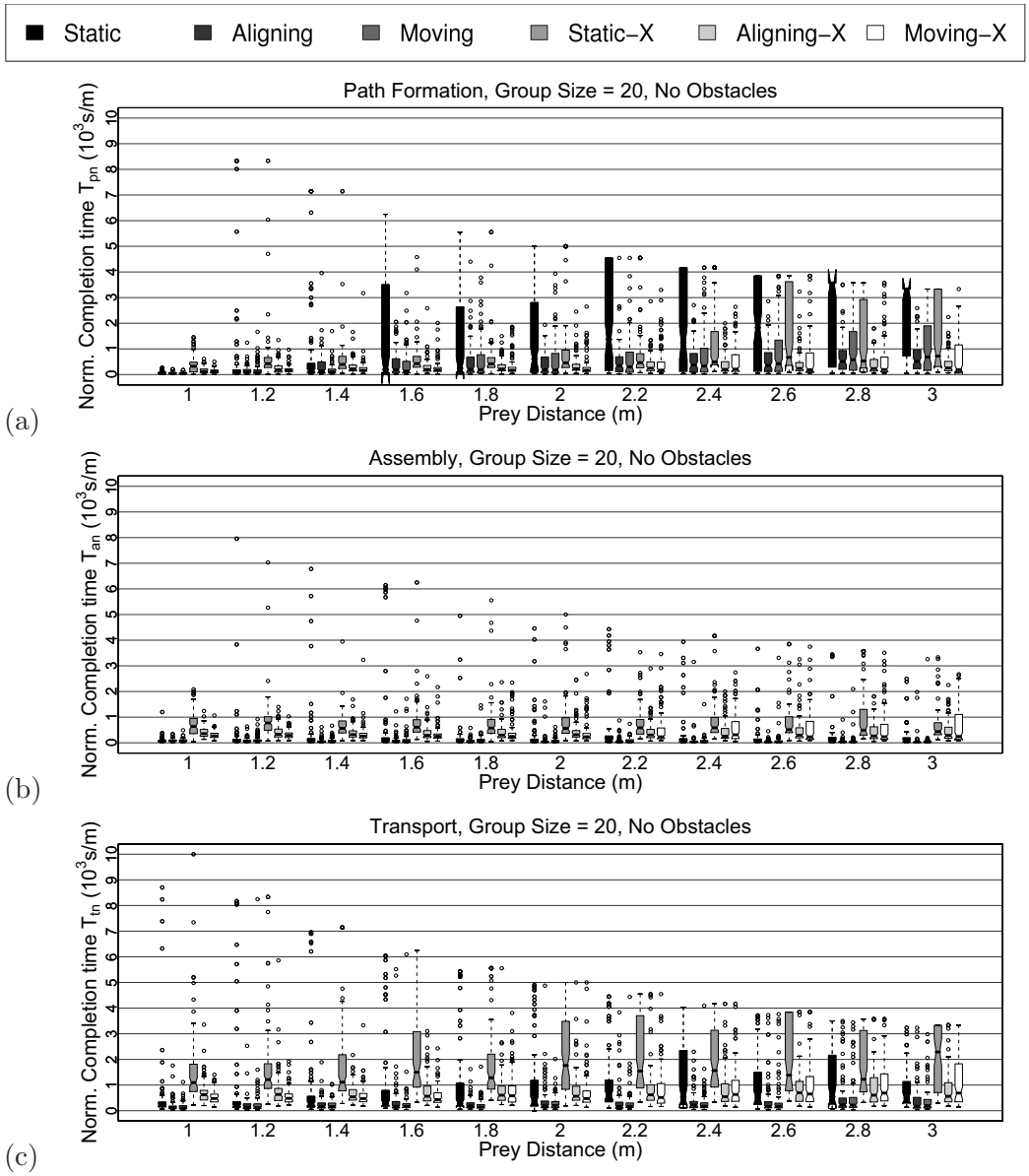


Figure 5.18: The box-and-whisker-plots (Becker et al., 1988) show the results taken from 100 trials of the normalized completion time (i.e. the completion time divided by the distance $\frac{T}{D}$) for subtasks (a) path formation, (b) assembly, and (c) transport when changing the nest to prey distance for a group 20 robots using the different chain strategies with and without prey extension mechanism in an environment without obstacles. Note that subtask assembly is only taken into account in case a path has been formed, and respectively subtask transport is only taken into account in case the assembly was successful. See the caption of Figure 5.14 for an explanation of the box-and-whisker plots.

Table 5.4: Success rates for the scalability test of the chain controller with and without prey extension mechanism for selected group sizes N and a prey to nest distance of $D = 3\,\mathrm{m}$. The three values represent percentages of the success rate from 100 trials for path formation, assembly and transport in this order.

N	Static	Aligning	Moving	Static-X	Aligning-X	Moving-X
10	16/3/0	89/34/9	91/46/13	41/2/0	96/50/19	94/56/14
12	35/18/4	93/93/75	96/90/71	59/39/11	97/96/80	98/96/81
14	42/26/9	93/91/81	89/88/82	60/43/20	95/95/84	93/93/84
16	37/23/7	100/99/93	84/83/77	64/57/30	100/99/94	88/86/81
18	30/25/17	100/100/91	84/84/74	60/57/40	100/100/92	93/93/86
20	40/38/28	100/100/95	91/90/84	61/60/34	100/100/94	99/99/96
25	51/46/30	100/99/93	99/98/94	76/71/52	100/100/93	100/100/98
30	57/49/34	100/100/94	100/100/93	82/79/61	100/100/99	100/100/99
40	71/69/43	100/100/95	100/99/90	82/79/58	100/100/100	100/100/98
50	69/66/50	100/100/94	100/100/94	95/93/71	100/100/97	100/100/98
60	85/83/62	100/100/100	100/100/95	98/98/81	100/100/100	100/100/98
80	93/93/76	100/100/97	100/100/99	99/99/87	100/100/99	100/100/100
100	100/100/82	100/100/100	100/100/100	100/100/86	100/100/95	100/100/93
140	100/100/92	100/100/99	100/100/99	100/100/85	100/100/96	100/100/99
200	100/100/99	100/100/100	100/100/96	100/100/98	100/100/98	100/100/100

5.3.3 Scalability Test

In a scalability test we analyse the performance when varying the group size. In this test we keep the prey at a distance of $D = 3\,\mathrm{m}$, and use an obstacle free environment. A summary of the results is given in Table 5.4 for the success rates. Figures 5.19 and 5.20 show the completion time and the overall effort for subtasks (a) path formation, (b) assembly, and (c) transport. The overall effort is the product of completion time and robot group size. This measure is a good indicator of scalability and can be used to investigate super-linearity in the system. The usual way to do this is to divide—rather than multiply—performance by the robot group size. However, as in our case the specific performance metric of completion time has to be minimized (rather than maximized), a multiplication is required to obtain a normalized measure representing the cumulated effort of all robots. A decrease for growing group sizes means that the added resources lead to a more than proportional decrease in completion time. The results for the three subtasks can be summarized as follows:

- **Path formation:** The performance of all tested controllers improves with the group size (see Figure 5.19a). In particular, this is true when the prey extension mechanism is not employed. In this case the overall effort decreases up to 100 robots, that is, the efficiency grows super-linearly, and then remains roughly constant (see Figure 5.20a). The observed super-linear effect is possible in

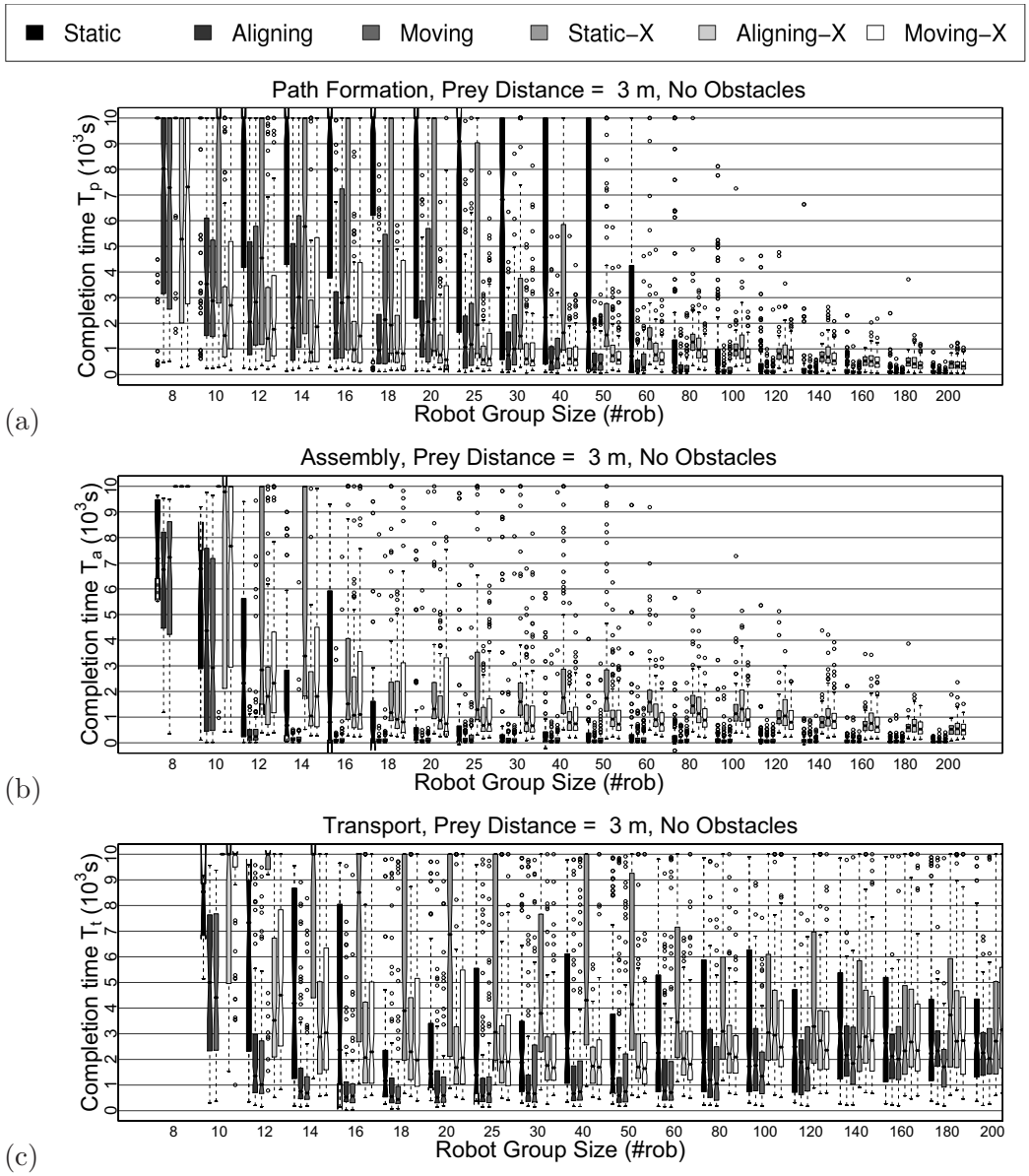


Figure 5.19: The box-and-whisker-plots (Becker et al., 1988) show the results taken from 100 trials of the completion times for subtasks (a) path formation, (b) assembly, and (c) transport when changing the robot group size using the different chain strategies with and without prey extension mechanism in an environment without obstacles. Note that the subtask assembly is only taken into account in case a path has been formed, and respectively the subtask transport is only taken into account in case the assembly was successful. See the caption of Figure 5.14 for an explanation of the box-and-whisker plots.

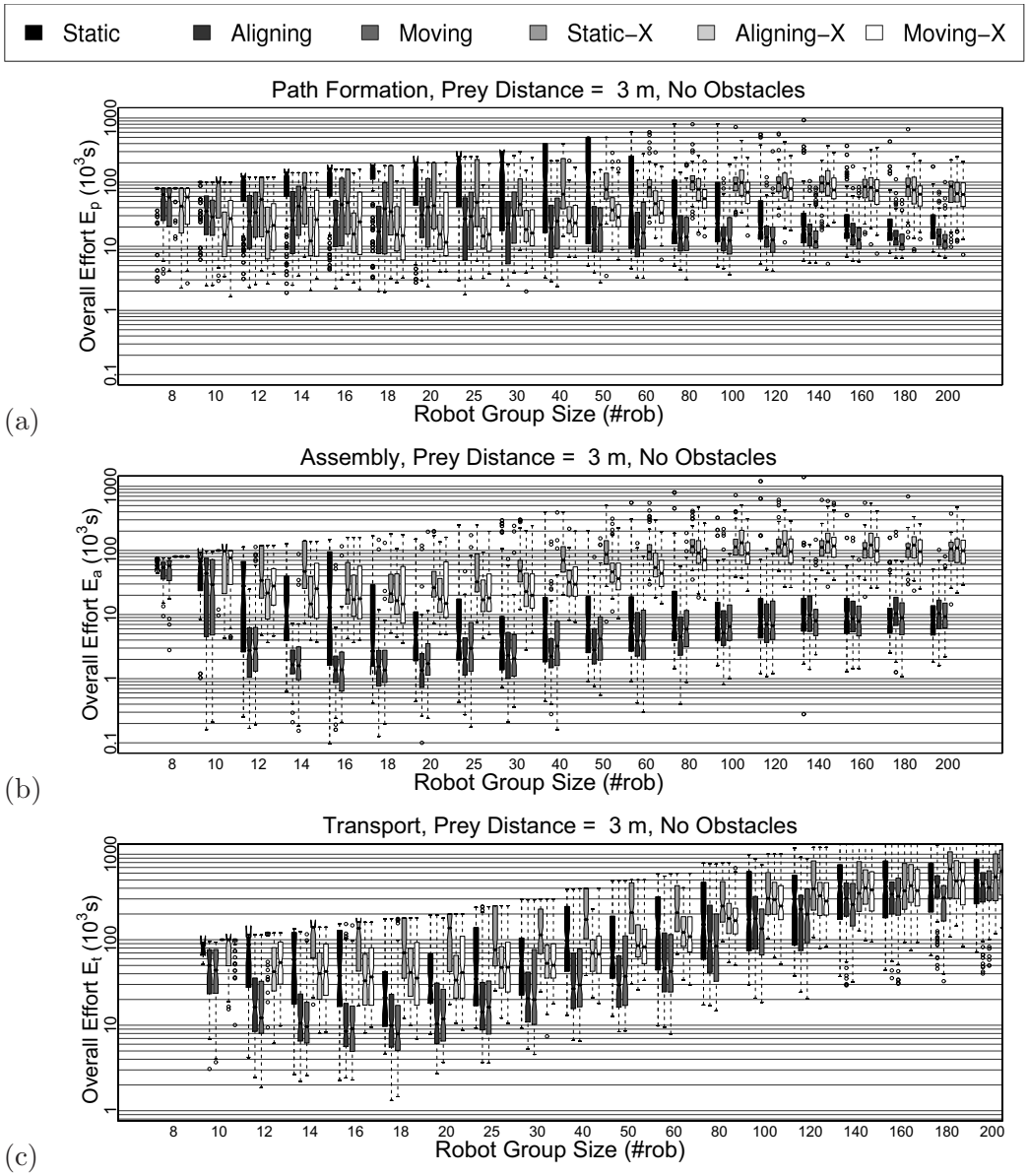


Figure 5.20: The box-and-whisker-plots (Becker et al., 1988) show the results taken from 100 trials of the overall efforts for subtasks (a) path formation, (b) assembly, and (c) transport when changing the robot group size using the different chain strategies with and without prey extension mechanism in an environment without obstacles. Note that the subtask assembly is only taken into account in case a path has been formed, and respectively the subtask transport is only taken into account in case the assembly was successful. See the caption of Figure 5.14 for an explanation of the box-and-whisker plots.

our experimental conditions as there is no overcrowding, which would cause physical interferences between the robots. The performance of the static chains remains below the one of the dynamic chains throughout all group sizes.

While the combination of chain and prey extension mechanism performs better for small group sizes, it is less effective for larger group sizes. This is the case because for large group sizes the prey extension mechanism may lead to some confusion in the behaviour of the robots. When there is a high density of robots the prey extension mechanism leads to a fast formation of a network of prey extending robots covering the arena even at areas that are far away from the nest. Therefore, it takes some additional time until a chain has found the real prey instead of a robot activating its LEDs in the colour of the prey. However, when the group size is small, the prey extension mechanism improves the performance because it effectively leads to the formation of a path from both nest and prey, and thereby makes a more efficient use of the resources.

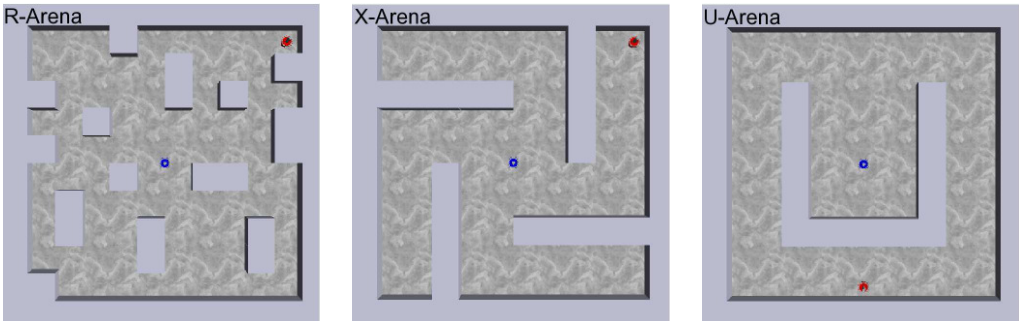
- **Assembly:** As for path formation, also for subtask assembly the performance of all tested controllers increases with the group size (see Figure 5.19b). The overall effort decreases up to a group size of 20 robots, and then increases again, suggesting that for this particular setup the optimal group size for assembly is 20 robots. If more robots are used, there are usually several robots attempting to assemble at the same time, in this way disturbing each other to succeed.

  Throughout all group sizes the prey extension mechanism has a negative impact on the assembly performance. This is the case because prey extending robots attract nearby robots to assemble to them, thereby hindering them from assembling to the prey.

- **Transport:** Unlike the completion times of the other two subtask, the completion time of subtask transport does not decrease for growing group sizes. It decreases up to a group size of roughly 20 robots, and then increases again. Larger group sizes have a negative impact on the transport performance. If the transporting structure contains many robots, some of these robots are not able to perceive the goal direction, that is, a chain robot. Therefore, they do not contribute to the transport and instead act as an additional weight to be transported. As is the case for subtask assembly, the prey extension mechanism has a negative impact on the performance. In the case of the transport subtask, this is due to chains being distracted by prey extending robots.

5.3.4 Obstacle Test

To assess the performance in presence of obstacles, we tested our controllers in three types of obstacle environments. In addition to the standard arena with a random



(a) (b) (c)

Figure 5.21: The three different types of arena used for the obstacle test. (a) The R-arena has a random positioning of the obstacles. In this case 20 obstacles were included. (b) The X-arena has four corridors. The prey is hidden behind one of them. (c) The U-arena, where the prey is positioned behind a U-shaped obstacle.

configuration of obstacle cubes (R-arena, Figure 5.21a), we also used two predefined arenas with a fixed configuration of obstacles: the X-arena (Figure 5.21b) and the U-arena (Figure 5.21c).

Figures 5.22, 5.23, and 5.24 show the results of the individual subtasks for the obstacle test for group sizes of (a) 10, (b) 20, (c) 40, and (d) 80 robots, and Table 5.5 shows the success rates for six selected obstacle environments and the four group sizes. The prey distance is 3 m for all cases except for the U-arena, where it is placed behind a long corridor at a distance of 2.12 m.

By adding obstacles to the environment, the task becomes more difficult in several ways. First, the presence of obstacles increases the difficulty of navigation, which is required for all three subtasks. Second, finding the nest or the prey becomes more difficult because they might be hidden behind the obstacles. Third, it might be impossible to form a straight path connecting nest and prey. This increases the length of the shortest path, and implies a particular difficulty for the aligning and the moving strategies. In fact, these two strategies attempt to align the chains. Due to this alignment, a chain can be broken up in cases where the line of sight of two neighboured chain members is blocked by an obstacle. Finally, the obstacles pose a difficulty for the transport subtask because the prey or the robots transporting it might get stuck at an obstacle. The difficulties of the environments X and U lie more in the particular configuration of the obstacles, than in the number of obstacles employed. In the case of the X-arena, 20 obstacles are employed. The difficulty lies in the fact that there are four corridors, and the prey is hidden behind one of them. A minimum number of twelve robots are required to form a path. For the U-arena,

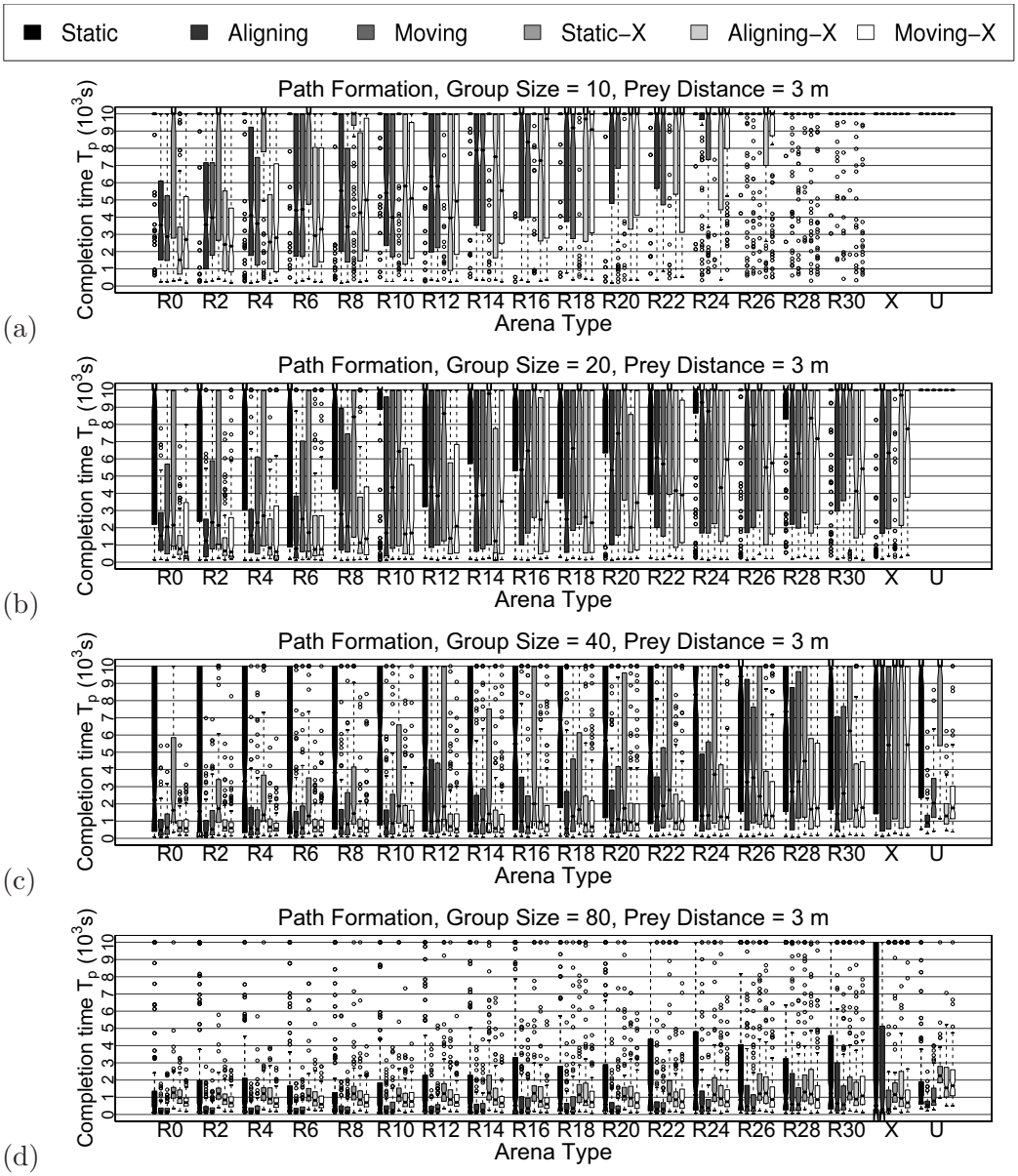


Figure 5.22: The box-and-whisker-plots show 100 evaluations per box of the completion time for subtask path formation when changing the number and configuration of obstacles in the environment for a nest to prey distance of 3 meters and a group of (a) 10 robots, (b) 20 robots, (c) 40 robots and (d) 80 robots using the different chain strategies with and without prey extension mechanism. For arenas of type R the number of obstacles is indicated. See the caption of Figure 5.14 for an explanation of the box-and-whisker plots.

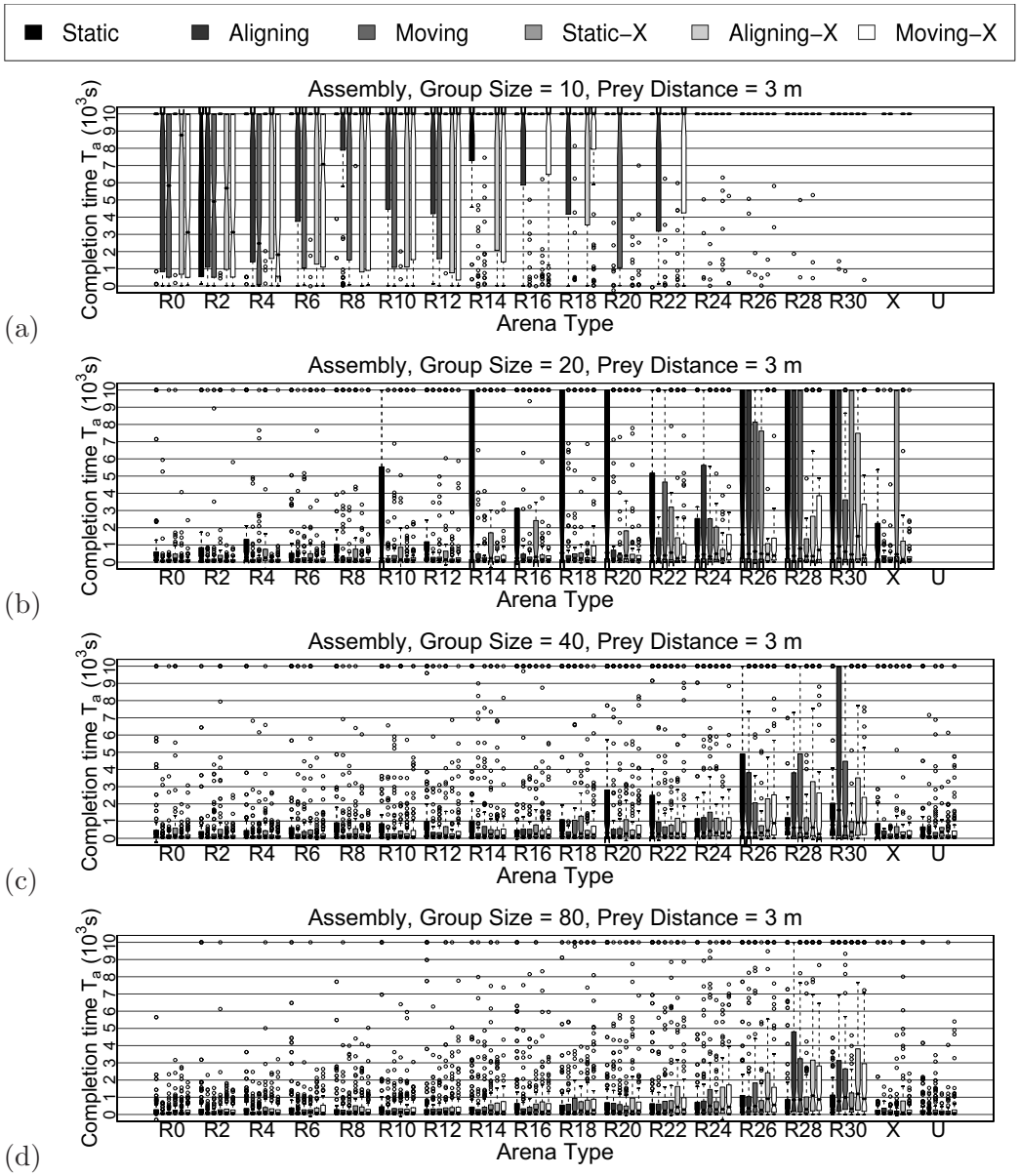


Figure 5.23: The box-and-whisker-plots show the results from 100 trials of the completion time for subtask assembly when changing the number and configuration of obstacles in the environment for a nest to prey distance of 3 meters and a group of (a) 10 robots, (b) 20 robots, (c) 40 robots and (d) 80 robots using the different chain strategies with and without prey extension mechanism. For arenas of type R the number of obstacles is indicated. Note that a trial is only taken into account in case a path has been formed. See the caption of Figure 5.14 for an explanation of the box-and-whisker plots.

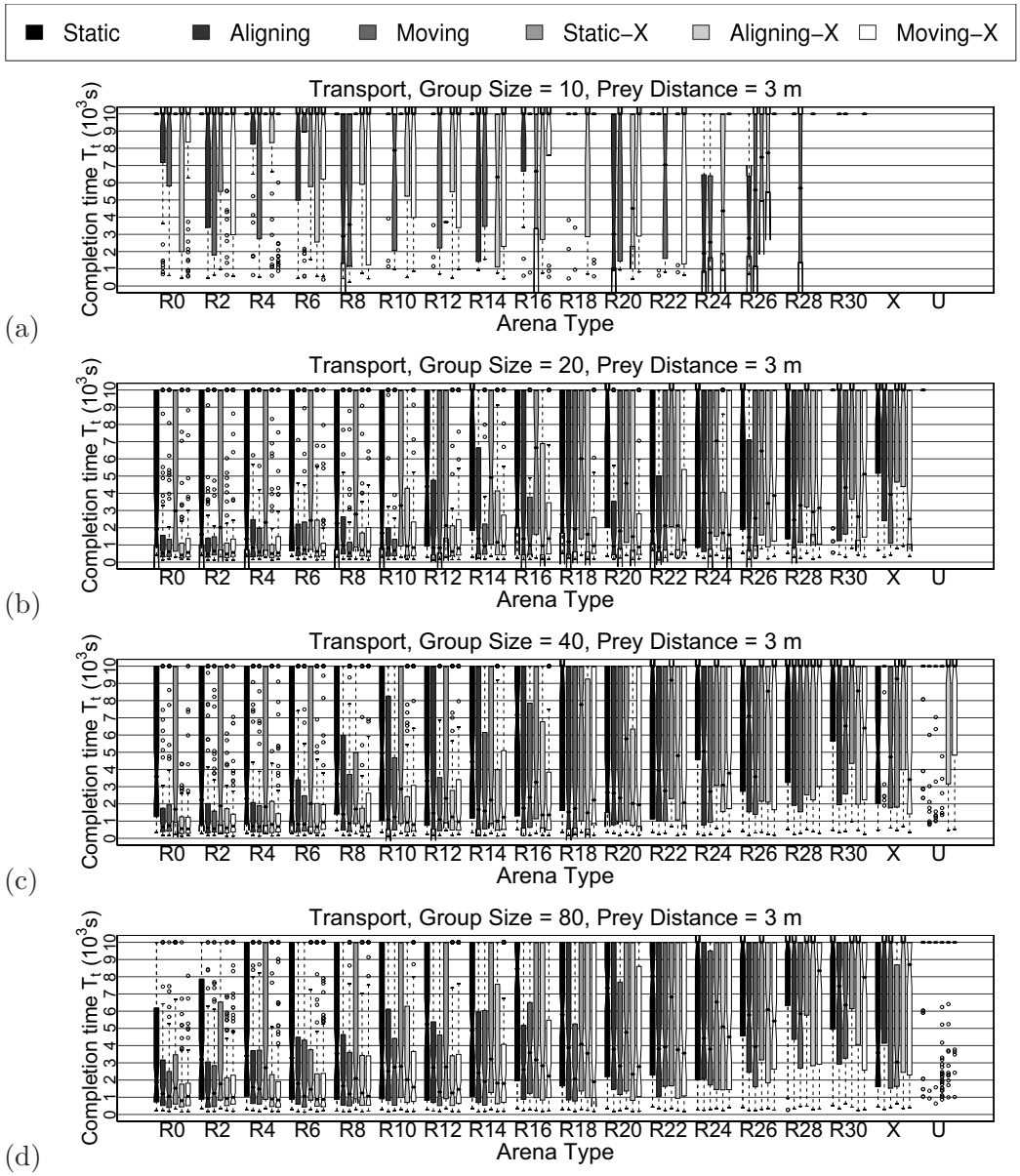


Figure 5.24: The box-and-whisker-plots show the results from 100 trials of the completion time for subtask transport when changing the number and configuration of obstacles in the environment for a nest to prey distance of 3 meters and a group of (a) 10 robots, (b) 20 robots, (c) 40 robots and (d) 80 robots using the different chain strategies with and without prey extension mechanism. For arenas of type R the number of obstacles is indicated. Note that a trial is only taken into account in case the assembly was successful. See the caption of Figure 5.14 for an explanation of the box-and-whisker plots.

Table 5.5: Success rates for all strategies for selected setups of the obstacle test of the chain controller. The three values represent percentages of the success rate from 100 trials for path formation, assembly and transport in this order.

Arena	Static	Aligning	Moving	Static-X	Aligning-X	Moving-X
$N = 10$						
R0	16/3/0	89/34/9	91/46/13	41/2/0	96/50/19	94/56/14
R10	12/0/0	70/19/3	70/24/12	19/3/0	66/27/9	79/27/11
R20	5/0/0	37/8/5	31/12/5	10/0/0	50/12/7	43/9/4
R30	2/0/0	12/2/0	13/1/0	2/0/0	15/0/0	17/1/0
X	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0
U	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0
$N = 20$						
R0	40/38/28	100/100/95	91/90/84	61/60/34	100/100/94	99/99/96
R10	27/20/13	76/74/65	70/68/59	52/48/27	81/78/61	85/83/77
R20	32/21/7	59/51/41	55/50/36	39/31/17	77/74/52	69/65/56
R30	23/14/3	49/32/13	49/39/24	29/19/6	64/48/32	64/51/27
X	22/19/8	46/45/18	51/49/30	24/16/5	48/44/15	54/53/34
U	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0
$N = 40$						
R0	71/69/43	100/100/95	100/99/90	82/79/58	100/100/100	100/100/98
R10	61/54/31	98/96/73	98/98/79	81/74/49	100/100/91	99/97/85
R20	55/46/26	92/84/57	91/87/60	77/69/39	97/94/71	95/92/68
R30	50/44/12	80/50/23	78/63/35	55/49/18	85/75/39	84/74/39
X	38/36/22	48/41/9	53/50/32	49/46/23	50/48/18	59/52/34
U	49/44/5	100/100/9	100/98/9	35/32/6	100/100/39	98/96/27
$N = 80$						
R0	93/93/76	100/100/97	100/100/99	99/99/87	100/100/99	100/100/100
R10	97/95/68	100/100/82	100/100/89	97/97/69	100/100/81	100/100/94
R20	93/88/47	100/99/68	99/98/80	98/96/56	100/97/72	100/99/75
R30	87/80/27	92/81/43	95/88/50	90/83/25	99/88/38	97/89/45
X	69/68/47	78/69/22	80/79/37	85/85/65	85/80/32	87/87/45
U	98/97/5	100/100/2	100/100/2	99/99/24	100/99/11	99/99/5

16 obstacles are employed, and configured such that they are U-shaped. In this case at least 22 robots are required to form a path. In the following, we give a summary of the results for each subtask: c

- **Path formation:** There is a clear performance drop when the environment contains obstacles. The intensity of performance drop depends on the group size. For $N = 10$ the resources are scarce as at least eight robots are required to form a path in an environment without obstacles. Therefore, if the configuration of the obstacles is such that a straight path can not connect nest and prey, ten robots are often not sufficient to form a path. In the case of the aligning and moving strategies, the success rate decreases from roughly 90% for arena R0, to roughly 15% for arena R30. The performance drop is less

steep for growing group sizes. For $N = 80$ the success rate for environment R30 is 87% for the static strategy, and is even higher for the other strategies.

For the X-arena, the success rate is 0 for $N = 10$, as the number of robots is too small to form a path. For $N = 20$, the aligning and moving strategies succeed to form a path in roughly every second, and the static strategy in roughly every fourth case. For $N = 40$, the success rate of the static strategy increases to 49% when the prey extension mechanism is used, and to 38% otherwise. Interestingly, the success rates of the other two strategies remain approximately at the same level as for $N = 20$. The reason that the success rate hardly grows beyond 50% is the combination of the particular obstacle configuration for the X-arena, and the number of chains formed simultaneously. For a group size of $N = 40$, there are in most cases two chains formed simultaneously (see Figure 5.13). In the X-arena, there are four corridors, and respectively four paths that a chain normally takes. Once that a chain has chosen one of the four corridors, it usually remains stuck there. For a group size of 40 or higher there is a very low probability that the chain gets completely dissolved as there is an overcrowding of robots at the end of the chain trying to join it. The observed success rate of 50% represents the probability that one of the two chains follows the one of the four corridors that leads to the prey. For a group size of 80 robots the situation is different. There are more chains formed simultaneously. For the aligning and moving strategies there are in most cases three chains. Furthermore, due to the large group size there is a higher probability for a split of one of these chains. Therefore, the success rate grows to roughly 80%.

The difficulty of the U-arena is different than the one of the X-arena. A chain can in principle follow two paths, both of which eventually lead to the prey. The difficulty is mainly that at least 22 robots have to join the chain connecting nest and prey. This is reflected by the results. For $N = 10$ and $N = 20$ the success rate is 0, and for N=40 the success rate for the aligning and moving strategies is 100% in nearly all cases. For the static strategy, the success rate is below 50% for $N = 40$, but nearly reaches 100% for $N = 80$.

The prey extension mechanism in general has a positive effect in the presence of obstacles. As the prey might be hidden behind an obstacle, to extend its perception is an effective method to speed up the search.

- **Assembly:** The performance of subtask assembly is in general diminished by the presence of obstacles. This is due to the fact that a connection to the prey might be restricted from several directions. However, if there is a sufficient amount of resources, the robots usually succeed to assemble to the prey in case a path has been formed.

- **Transport:** Also the performance of subtask transport is in general diminished when the environment contains obstacles. In particular, the performance is weak for the X- and the U-arenas. There are two reasons for this: First, the distance of the path, and therefore also the distance of the transport is longer. The path has a length of approximately 4.2 m for the X-arena, and of approximately 7.8 m for the U-arena. On the one hand this increases the time it takes to transport the prey. On the other hand it increases the probability that too many robots connect to the prey (or to a robot connected to the prey), which leads the transport to get stuck. Second, for the X- and U-arenas the chains have to make a very steep turn around a corner. This poses a particular problem for the transport as the prey transporting structure can easily get stuck at such a corner. Interestingly, this appears to pose a bigger problem to the aligning and moving strategies, than to the static one, where the success rates for the X- and U-arenas are higher at least for $N = 80$. Due to the aligning mechanism the chain gets closer to a corner. The static strategy does not employ the aligning mechanism, and therefore usually keeps a higher distance to corners, thereby facilitating the transport task.

5.3.5 Robustness Tests

To analyse how the performance of our controllers changes in the presence of noisy conditions, we have conducted a series of robustness tests in which we vary the noise of the various sensors. The noise is calculated at each time step as a uniformly random value within the range $[-noise_{max}; noise_{max}]$, and is added to the considered sensor value. We varied the noise of the direction to objects[2] perceived by the camera (Section 5.3.5.1), the distance to objects perceived by the camera (Section 5.3.5.2), and the proximity sensor (Section 5.3.5.3).[3]

5.3.5.1 Camera Direction

The direction perceived with the camera is on the basis of most decisions that concern the navigation of and near a chain, as it is the chain members activating their LEDs that are perceived with the camera. Two examples for this are (i) the alignment of a chain member with respect to its neighbours as used by the aligning and moving strategies, and the (ii) motion of a robot along a chain to explore its vicinity. To decide where exactly to move it is crucial to know the direction at which a chain member is perceived.

[2] An object can be another robot, the nest or the prey. Obstacles can not be perceived with the camera.

[3] As mentioned in Section 3.2.2, the values used in all experiments for these four noise levels are 18^o, 10 cm and 0.2. In the robustness tests we manipulate one value at a time.

We test the robustness to the noise of this information by adding maximum noise levels in the range $[0^o, 180^o]$. Table 5.6 shows the achieved success rates for the different strategies and subtasks.

The results show a limited robustness of the controllers with respect to the noise applied. The larger the group size, the higher is the robustness achieved. In general the performance decreases when increasing the level of injected noise. This is always the case when the prey extension mechanism is used. However, when it is not used, surprisingly, the performance increases for the highest tested noise level of 180^o with respect to the noise levels 108^o and 144^o, as becomes particularly clear for group size $N = 40$.

The reason for this can be found by observing the following behaviour. When noise is injected the information available to the camera gets more random. Therefore, also the decisions based on this information become more random. The higher the noise, the higher is the risk that a wrong decision of motion disrupts a chain. In the case of a noise level of 180^o the information is in fact completely random. The robot then knows that there is an object, but not where this object is. The result in the behaviour of the aligning and moving strategies is that for robots in a chain it is not possible to execute the aligning behaviour, because it tries to align with respect to two objects that permanently appear to change positions. So, once a robot joined a chain it effectively stops moving: The chains become static, explaining the similar results of the three strategies for the highest noise level. By being static, the chains are prevented getting disrupted, as is the case for lower noise levels. If the density of robots in the environment is sufficiently high, there is a high probability that the prey is found by this immobile network of robots.

This positive effect does not occur when the prey extension mechanism is used. What can be observed is that the higher the level of the injected noise, the more robots join the prey extending structure instead of the chain, thereby taking away the resources required to form a path. The reason for this is that the connectivity of the network of prey extending robots relies on simpler rules than the network of a chain. In a chain there are three colours. Whether a robot may leave the chain or not depends on its own colour and on the colours it perceives. In a prey extending structure the robots all have the same colour and already the perception of this colour yields a stimulus to remain within the structure. Therefore, when the direction information from the camera is very noisy, the prey extending structure becomes more stable than the chains, in this way claiming the majority of the robots, and making the successful formation of a path very difficult.

The accomplishment of subtasks assembly is very sensitive to the injected noise. With noise in the perception of direction, the task of assembling becomes very difficult, as a successful assembly relies on precise information of the direction to the object towards which the assembly is attempted. For instance, for a group size

Table 5.6: Success rates of the chain controller for the robustness test on the perception of direction using the camera. The three values represent percentages of the success rate from 100 trials for path formation, assembly and transport in this order.

Noise	Static	Aligning	Moving	Static-X	Aligning-X	Moving-X
$N = 10$						
0^o	14/2/0	91/33/8	90/44/13	41/2/0	96/33/7	91/42/11
36^o	10/0/0	73/49/12	61/36/11	18/1/0	81/57/11	72/39/8
72^o	0/0/0	3/1/0	7/1/0	1/0/0	11/4/0	5/0/0
108^o	0/0/0	0/0/0	1/0/0	0/0/0	0/0/0	1/0/0
144^o	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0
180^o	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0
$N = 20$						
0^o	39/31/23	100/100/91	91/89/80	69/59/48	100/100/92	98/98/93
36^o	39/37/26	85/78/73	81/78/70	55/52/43	95/95/86	97/97/90
72^o	16/12/7	66/43/13	42/31/11	25/21/8	80/70/22	58/41/12
108^o	0/0/0	4/0/0	3/0/0	0/0/0	14/4/0	6/3/0
144^o	0/0/0	0/0/0	0/0/0	1/0/0	1/0/0	0/0/0
180^o	1/0/0	1/0/0	3/0/0	0/0/0	0/0/0	0/0/0
$N = 40$						
0^o	68/62/42	100/100/93	100/100/96	83/83/55	100/100/98	100/100/98
36^o	57/54/39	98/98/91	100/100/93	85/83/67	99/99/97	100/100/99
72^o	40/34/12	88/73/22	76/61/15	65/58/17	99/95/41	92/81/40
108^o	6/2/0	33/13/0	20/5/0	8/1/0	50/21/0	42/17/1
144^o	3/0/0	10/0/0	3/0/0	6/1/0	33/1/0	18/0/0
180^o	31/0/0	35/0/0	22/0/0	0/0/0	0/0/0	0/0/0
$N = 80$						
0^o	99/99/74	100/100/96	100/100/95	99/99/81	100/100/97	100/100/97
36^o	95/94/73	100/100/98	100/100/99	100/99/78	99/99/98	100/100/99
72^o	80/76/31	100/97/24	100/96/36	83/75/28	99/97/29	98/96/29
108^o	50/17/0	76/34/0	71/29/0	59/24/0	91/69/0	89/60/0
144^o	24/1/0	58/3/0	54/1/0	49/5/0	54/3/0	66/5/0
180^o	83/0/0	85/0/0	86/0/0	0/0/0	0/0/0	1/0/0

of $N = 20$ and a noise level of 144^o on, even if a path is formed in roughly one out of two trials, only in very rare occasions do the robots succeed to assemble to the prey.

The noise has an even more disruptive effect on subtask transport. The cooperative nature of the subtask requires the robots to agree on the direction towards which the prey is pulled or pushed. Given that there is no direct communication between the robots to negotiate the goal direction they can only transport the prey in case their perception of the goal direction is sufficiently precise.

5.3.5.2 Camera Distance

In addition to the direction towards an object, the camera also informs about the respective distance. This information is very important in order to adjust the distance, which is for instance required by explorers to navigate along a chain, or by chain members to adjust the distance among each other (at least for the aligning and moving strategies). As is the case for the direction, the knowledge about the distance to an object is crucial to make decisions on where to move.

We test the robustness to the noise of this information by adding maximum noise levels in the range $[0\,\text{cm}, 50\,\text{cm}]$. Table 5.7 shows the achieved success rates for the different strategies and subtasks.

To add noise to the perception of the distance has a disruptive effect on the stability of the chains. With increasing levels of noise, it becomes increasingly difficult for the chain members to adjust their distance with respect to each other, and the probability increases to get out of sight of each other and in this way break up a chain. In particular, and similarly to the noise in direction, when the prey extension mechanism is used, a high level of noise leads to the majority of the robots to join the prey extending structure, as it has a higher degree of stability. Furthermore, a robot in a chain leaves it if the prey (or a robot that activates its LEDs in the same colour as the prey) is perceived at a close distance. This is often and especially for large group sizes falsely the case if the prey extension mechanism is used. Therefore, for large group sizes the performance decrease is stronger for high levels of noise when the prey extension mechanism is used.

The static strategy seems to be less disrupted by high levels of noise, in some cases even reaching a higher success rate with noise than without. One reason is that the chains are static, that is, they do not move. Therefore, once a chain is formed it will not get broken up due to neighbouring chain members adjusting their distance with one another. Another reason is that the noise can lead to a higher distance between two robots in a chain, in this way increasing the reach of the chain. A robot joins a chain when it perceives a chain member at a given distance. Due to the noise this distance can actually be higher than the distance perceived by the robot, leading the robot to join the chain at a further distance.

Table 5.7: Success rates of the chain controller for the robustness test on the perception of distance using the camera. The three values represent percentages of the success rate from 100 trials for path formation, assembly and transport in this order.

Noise	Static	Aligning	Moving	Static-X	Aligning-X	Moving-X
$N = 10$						
0 cm	18/1/0	75/9/2	89/7/2	37/6/1	85/11/7	90/8/3
10 cm	16/3/0	89/34/9	91/46/13	41/2/0	96/50/19	94/56/14
20 cm	7/3/1	58/48/23	81/73/27	38/24/2	83/81/43	86/78/43
30 cm	0/0/0	12/0/0	10/0/0	18/1/0	44/1/0	26/0/0
40 cm	1/0/0	0/0/0	0/0/0	4/0/0	8/0/0	5/0/0
50 cm	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0
$N = 20$						
0 cm	40/34/23	100/100/94	95/92/81	67/66/54	100/100/94	97/96/90
10 cm	40/38/28	100/100/95	91/90/84	61/60/34	100/100/94	99/99/96
20 cm	25/25/14	96/92/60	99/97/78	71/59/39	100/100/83	100/100/86
30 cm	21/0/0	48/0/0	52/4/2	72/1/0	92/9/0	95/5/0
40 cm	11/0/0	25/0/0	12/0/0	12/0/0	37/0/0	43/0/0
50 cm	0/0/0	7/0/0	4/0/0	2/0/0	0/0/0	9/0/0
$N = 40$						
0 cm	66/61/35	100/100/87	100/100/98	84/80/57	100/100/97	100/100/97
10 cm	71/69/43	100/100/95	100/99/90	82/79/58	100/100/100	100/100/98
20 cm	56/51/42	100/98/90	100/100/92	94/90/71	100/100/98	100/100/100
30 cm	73/8/1	99/20/4	99/9/3	98/16/5	100/24/7	100/19/3
40 cm	87/0/0	94/0/0	78/0/0	6/0/0	36/0/0	64/0/0
50 cm	59/0/0	83/0/0	67/0/0	1/0/0	4/0/0	9/0/0
$N = 80$						
0 cm	95/93/48	100/100/96	100/100/95	98/98/74	100/100/97	100/100/98
10 cm	93/93/76	100/100/97	100/100/99	99/99/87	100/100/99	100/100/100
20 cm	94/91/83	100/100/79	100/100/80	100/99/86	100/100/97	100/100/96
30 cm	100/58/30	100/59/16	100/35/12	99/41/9	100/20/1	100/26/5
40 cm	100/0/0	100/2/0	100/0/0	36/0/0	41/0/0	61/0/0
50 cm	100/0/0	100/0/0	100/0/0	1/0/0	2/0/0	6/0/0

For assembly and transport, from noise levels of 30 cm the performance decreases significantly. The assembly behaviour requires precise information of the distance towards an object in order to be able to connect to it at the right position. If the information is too noisy, the robot will try to assemble when it is too far or too close to grip. A transporting robot chooses the direction towards the closest chain member as the goal direction. Therefore, a high level of noise may lead it towards the wrong direction. However, for the aligning and moving strategies, group size $N = 10$ and noise levels of 10 cm and 20 cm the success rate to assemble and transport the prey is increased. This has two reasons. First, the distances between two neighbouring robots in the chain may be larger in the presence of noise. If the noise is not too high, the chain is not broken up. In the end fewer robots might be required in order to form a path, leaving more to assemble to the prey. Secondly, when a robot is near the prey and perceives the chain, it either joins the chain or assembles to the prey. The decision depends on the distance at which the prey is perceived. If it is close the robot assembles, and if it is far the robot joins the chain to make a closer connection. A low level of noise increases the probability that the prey is falsely perceived at a close distance, which leads it to assemble to the prey.

5.3.5.3 Proximity Sensors

In the final robustness test, we investigate different levels of noise added to the proximity sensors. The level of noise injected is normalized by the maximum activation of a proximity sensor, obtained when a robot is placed directly next to a wall, an obstacle, or another robot. We tested six different ratios of noise in the range $[0, 1]$. The achieved success rates for the different strategies and subtasks are reported in Table 5.8.

The proximity sensors are used for obstacle avoidance. When a high level of activation of an individual proximity sensor above a given threshold is reached, this leads to the activation of a motor schema that points away from the source of activation. On the one hand, noisy information of the proximity sensor leads to more collisions because a nearby object might not be perceived. On the other hand, it leads to the activation of the avoidance behaviour even in cases when in fact it is not required. At the same time the robots may try to avoid non existent obstacles. When observing the behaviour with a high level of injected noise, the robot movements look more random.

Nevertheless, as can be seen from the results, there is hardly any decrease in performance for any of the three subtasks. Indeed, for subtasks assembly and transport the proximity sensors are not used at all, and for subtask path formation the obstacle avoidance behaviour is not a requirement. It is rather a basic behaviour to protect the robots from getting damaged.

Table 5.8: Success rates of the chain controller for the robustness test on the perception of obstacles using the proximity sensors. The three values represent percentages of the success rate from 100 trials for path formation, assembly and transport in this order.

Noise	Static	Aligning	Moving	Static-X	Aligning-X	Moving-X
$N = 10$						
0%	18/3/0	93/42/16	88/46/12	41/4/0	94/52/20	93/55/17
20%	11/4/0	95/35/14	83/42/10	25/0/0	92/47/10	90/57/8
40%	13/0/0	91/41/16	92/52/16	43/4/1	95/49/12	91/47/26
60%	11/3/0	91/31/12	82/38/15	28/3/0	92/40/20	89/47/13
80%	12/1/0	86/44/22	86/43/19	30/2/0	93/50/31	90/56/31
100%	10/2/0	84/59/35	76/51/27	22/3/0	89/70/37	79/54/26
$N = 20$						
0%	43/41/28	100/100/92	92/89/79	73/68/48	100/100/92	96/96/94
20%	43/38/23	100/100/96	91/89/83	80/76/56	99/99/95	95/95/85
40%	47/46/31	100/100/96	89/88/79	74/71/46	100/100/95	95/94/86
60%	39/37/22	99/98/90	93/93/86	68/64/47	100/100/94	96/96/90
80%	48/43/21	100/99/88	92/90/83	73/71/50	100/100/96	100/100/91
100%	45/42/27	97/96/84	88/85/75	56/49/32	100/100/93	95/95/89
$N = 40$						
0%	67/64/45	100/100/95	100/100/98	81/78/61	100/100/96	100/100/97
20%	68/66/43	100/100/95	100/100/97	87/82/64	100/100/96	100/100/98
40%	70/69/49	100/100/93	100/100/94	81/80/63	100/100/96	100/100/98
60%	72/71/52	100/100/88	100/100/95	83/83/61	100/100/99	100/100/99
80%	71/66/39	100/100/90	100/100/94	83/81/67	100/100/95	100/100/98
100%	76/72/45	100/100/92	98/98/90	83/80/62	100/100/96	100/100/95
$N = 80$						
0%	97/97/69	100/100/98	100/100/98	100/96/80	100/100/98	100/100/98
20%	96/94/76	100/100/99	100/100/100	99/99/77	100/100/96	100/100/97
40%	98/97/74	100/100/97	100/100/98	100/99/75	100/100/97	100/100/97
60%	99/98/77	100/100/99	100/100/98	100/100/83	99/98/97	100/100/98
80%	96/96/79	100/100/95	100/100/98	100/100/84	100/100/99	100/100/97
100%	96/95/80	100/100/96	100/100/96	100/100/86	98/92/78	100/97/92

5.3.6 Fault Tolerance Tests

In a series of fault tolerance tests we analyse our controllers' ability to cope with individual failure by deactivating a sensor or actuator for varying fractions of the robot group in the range $[0\%, 100\%]$. We disabled either the camera (Section 5.3.6.1), the LEDs (Section 5.3.6.2), the proximity sensors (Section 5.3.6.3), or the tracks (Section 5.3.6.4).

5.3.6.1 Camera

In the robustness tests we varied the noise injected to the camera with respect to the perception of direction and distance. The results show that in particular for a noisy perception of the direction the behaviour is very sensitive. When the camera is completely disabled a robot can neither join the path forming structure, nor assemble to the prey or transport it, and therefore it cannot contribute to the solution of the problem at all. Instead, it just performs a random walk and can be considered as a mobile obstacle.

This is confirmed by our results as reported in Table 5.9. The robots for which the cameras were disabled do not contribute to the problem solution. However, if

Table 5.9: Success rates of the chain controller for the fault tolerance test on disabled cameras. The three values represent percentages of the success rate from 100 trials for path formation, assembly and transport in this order.

Fraction	Static	Aligning	Moving	Static-X	Aligning-X	Moving-X
$N = 10$						
0%	11/4/0	95/35/14	83/42/10	25/0/0	92/47/10	90/57/8
20%	10/0/0	67/0/0	66/0/0	21/0/0	88/1/0	87/2/0
40%	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0
60%	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0
80%	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0
100%	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0
$N = 20$						
0%	43/38/23	100/100/96	91/89/83	80/76/56	99/99/95	95/95/85
20%	38/33/22	99/99/93	91/88/79	64/58/39	99/99/94	94/94/88
40%	41/20/4	94/94/80	97/95/87	62/43/19	95/95/86	96/95/87
60%	10/0/0	74/0/0	79/0/0	34/0/0	92/2/0	94/8/2
80%	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0
100%	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0
$N = 40$						
0%	68/66/43	100/100/95	100/100/97	87/82/64	100/100/96	100/100/98
20%	67/65/52	100/100/91	100/100/94	80/78/60	100/100/93	100/100/99
40%	54/50/33	100/100/90	99/99/90	76/73/60	100/100/98	99/99/95
60%	53/42/26	99/99/95	90/87/80	56/52/39	100/100/95	94/93/84
80%	13/0/0	76/0/0	75/0/0	31/0/0	93/0/0	99/0/0
100%	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0
$N = 80$						
0%	96/94/76	100/100/99	100/100/100	99/99/77	100/100/96	100/100/97
20%	96/96/66	100/100/95	100/100/97	98/97/81	100/100/98	100/100/98
40%	91/89/71	100/100/100	100/100/100	98/98/85	100/100/99	100/100/99
60%	75/71/56	100/100/97	100/100/94	86/84/75	100/100/92	100/99/96
80%	49/45/31	97/97/91	90/89/82	73/70/52	99/99/94	94/93/91
100%	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0

the number of remaining robots is sufficiently high, they can complete the task. Consequently, the tolerance to individual failure is considerably higher for larger group sizes, so that for the two more dynamic strategies at $N = 80$ there is still a very good performance even when for 80% of the robots the camera is disabled. The static strategy requires more robots to form a path. Therefore, the performance decrease is stronger.

5.3.6.2 LEDs

Without LEDs robots are not able to signal their state to the other robots in case they joined a chain or assembled to the prey. Therefore, they do not contribute to the formation of a path and can be considered as a mobile obstacle. In fact, for what concerns path formation, a robot without LEDs can even be considered slightly more disruptive than a mobile obstacle. It is able to perceive the other robots, the nest and the prey. It tries to join the path forming structure, thereby getting in the way of other robots whose LEDs do work.

This is reflected by our results as shown in Table 5.10. The performance of subtask path formation is in most cases below the one of the previously studied test where the cameras are disabled. The robots with disabled LEDs do indeed disturb the other robots from forming a path. In case of the two dynamic strategies a high fraction of erroneous robots also has a disruptive effect on existing chains because the robots in the chains try to avoid the many erroneous robots and thereby may break up a chain.

However, once a path is formed, the robots can contribute by assembling and transporting the prey, even if they may hinder other robots from assemblage as they do not activate their LEDs. Nevertheless, also for these two subtasks the success rates are in most cases below the ones when the camera is disabled. This is due to the fact that usually all erroneous robots try to assemble to the prey at the same time, in this way disturbing each other in doing so.

5.3.6.3 Proximity Sensors

Our robustness study on injecting noise to the proximity sensors showed that the performance stays very high even in the presence of very noisy proximity information. This suggests that the proximity sensors are not crucial for the accomplishment of the task.

Indeed, our results, as reported in Table 5.11 show a lower degree of performance decrease than the previous two fault tolerance tests. Nevertheless, there is a decrease in performance which is more significant for smaller group sizes. In fact, when the proximity sensors are disabled, a robot does not perceive any obstacles and may therefore get stuck in a corner of the environment. However, if it perceives a part

of the path forming structure it can join it. Therefore, a robot without proximity sensors can still contribute to form a path and the performance stays high even when the proximity sensors of most robots are disabled.

5.3.6.4 Tracks

With disabled tracks a robot is not able to move and therefore can be considered as an immobile obstacle. As shown in Table 5.12, the performance remains high as long as there are a sufficient number of robots that remain fully functional. Removing the tracks of a robot leads to a similar performance as removing the robot. Nevertheless, a robot without tracks can in principle still contribute to form a path if, by chance,

Table 5.10: Success rates of the chain controller for the fault tolerance test on disabled LEDs. The three values represent percentages of the success rate from 100 trials for path formation, assembly and transport in this order.

Fraction	Static	Aligning	Moving	Static-X	Aligning-X	Moving-X
$N = 10$						
0%	11/4/0	95/35/14	83/42/10	25/0/0	92/47/10	90/57/8
20%	10/0/0	65/28/15	56/17/10	20/3/0	86/41/17	88/51/21
40%	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0
60%	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0
80%	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0
100%	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0
$N = 20$						
0%	43/38/23	100/100/96	91/89/83	80/76/56	99/99/95	95/95/85
20%	49/42/27	100/99/93	88/83/77	75/73/56	100/100/92	96/96/91
40%	26/18/13	94/92/84	85/83/70	62/58/43	99/99/97	94/93/85
60%	8/5/0	49/37/30	39/24/20	22/12/7	82/75/72	76/67/62
80%	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0
100%	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0
$N = 40$						
0%	68/66/43	100/100/95	100/100/97	87/82/64	100/100/96	100/100/98
20%	74/64/43	100/100/87	100/100/94	82/80/66	100/100/93	100/100/97
40%	57/43/27	100/99/79	98/97/79	78/75/57	100/100/97	100/100/96
60%	46/28/7	98/86/49	91/83/38	60/50/34	100/100/93	100/99/89
80%	14/5/0	9/3/2	9/1/0	22/9/1	35/20/15	24/11/8
100%	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0
$N = 80$						
0%	96/94/76	100/100/99	100/100/100	99/99/77	100/100/96	100/100/97
20%	91/87/67	100/100/84	100/100/89	99/99/90	100/100/99	100/100/96
40%	94/85/55	100/100/64	100/100/69	100/97/73	100/100/91	100/100/89
60%	84/68/16	100/96/15	100/98/26	96/90/44	100/100/65	100/100/76
80%	56/31/0	79/22/0	74/17/0	69/39/3	96/58/6	74/35/2
100%	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0

Table 5.11: Success rates of the chain controller for the fault tolerance test on disabled proximity sensors. The three values represent percentages of the success rate from 100 trials for path formation, assembly and transport in this order.

Fraction	Static	Aligning	Moving	Static-X	Aligning-X	Moving-X
$N = 10$						
0%	11/4/0	95/35/14	83/42/10	25/0/0	92/47/10	90/57/8
20%	8/0/0	73/4/0	65/6/3	20/2/0	88/18/5	89/11/2
40%	1/0/0	18/0/0	19/4/1	4/0/0	38/0/0	38/2/1
60%	0/0/0	7/1/0	3/0/0	0/0/0	12/0/0	4/0/0
80%	0/0/0	1/0/0	0/0/0	0/0/0	2/0/0	3/0/0
100%	0/0/0	0/0/0	0/0/0	0/0/0	1/0/0	0/0/0
$N = 20$						
0%	43/38/23	100/100/96	91/89/83	80/76/56	99/99/95	95/95/85
20%	39/35/21	99/99/97	82/82/75	70/60/34	100/100/96	95/95/83
40%	35/31/12	95/95/87	84/84/69	68/58/34	98/98/91	96/96/91
60%	27/17/8	90/89/71	85/81/65	54/41/21	93/91/74	88/85/65
80%	7/2/0	53/45/27	54/39/24	32/22/6	77/71/50	73/57/39
100%	0/0/0	11/11/5	14/10/5	8/5/1	38/29/19	32/25/14
$N = 40$						
0%	68/66/43	100/100/95	100/100/97	87/82/64	100/100/96	100/100/98
20%	72/72/51	100/100/93	100/100/96	93/92/69	100/100/98	100/100/100
40%	64/62/48	100/100/95	100/100/92	84/82/67	100/100/94	100/100/99
60%	61/58/32	100/99/94	93/92/82	81/77/59	100/100/95	100/100/98
80%	38/34/22	96/96/89	98/98/84	82/79/52	99/99/95	98/98/94
100%	25/21/10	73/71/54	68/63/49	67/63/37	98/97/73	91/89/73
$N = 80$						
0%	96/94/76	100/100/99	100/100/100	99/99/77	100/100/96	100/100/97
20%	96/95/77	100/100/98	100/100/98	99/99/83	100/100/97	100/100/98
40%	97/96/68	100/100/99	100/100/93	99/99/77	100/100/100	100/100/98
60%	90/88/60	100/100/96	100/100/97	99/99/90	100/100/99	100/100/98
80%	91/90/51	100/100/91	100/99/91	99/99/80	100/100/98	100/100/96
100%	87/87/52	99/98/72	93/90/78	98/97/77	100/100/97	100/100/91

it happens to be positioned at the right place. And indeed, for group size $N = 80$, when there is a high density of robots, a path can be formed in a few trials even if no single robot is able to move.

A robot without tracks can of course not assemble to or transport the prey. In fact, if it is part of the path connecting nest and prey, it will block the transport as it can not move and therefore can not release itself from the chain.

5.4 Conclusions

In this chapter we presented the results obtained from experiments in simulation for the chain controller on the task of forming a path between nest and prey, assembling

Table 5.12: Success rates of the chain controller for the fault tolerance test on disabled tracks. The three values represent percentages of the success rate from 100 trials for path formation, assembly and transport in this order.

Fraction	Static	Aligning	Moving	Static-X	Aligning-X	Moving-X
$N = 10$						
0%	11/4/0	95/35/14	83/42/10	25/0/0	92/47/10	90/57/8
20%	2/0/0	54/0/0	58/0/0	15/0/0	80/0/0	90/0/0
40%	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0
60%	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0
80%	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0
100%	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0
$N = 20$						
0%	43/38/23	100/100/96	91/89/83	80/76/56	99/99/95	95/95/85
20%	43/33/19	98/98/88	84/80/69	74/70/45	100/99/92	94/94/86
40%	32/13/8	87/81/57	75/70/61	58/42/21	87/86/79	85/83/79
60%	16/0/0	46/0/0	49/0/0	34/0/0	74/0/0	72/0/0
80%	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0
100%	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0
$N = 40$						
0%	68/66/43	100/100/95	100/100/97	87/82/64	100/100/96	100/100/98
20%	68/66/46	99/99/95	100/100/92	88/83/62	100/100/97	100/100/99
40%	59/55/34	97/95/89	97/96/89	80/74/57	99/99/94	99/99/95
60%	41/32/16	83/78/68	83/74/63	74/68/49	96/94/84	92/90/82
80%	31/0/0	50/0/0	49/0/0	30/0/0	73/0/0	64/0/0
100%	0/0/0	0/0/0	0/0/0	1/0/0	0/0/0	0/0/0
$N = 80$						
0%	96/94/76	100/100/99	100/100/100	99/99/77	100/100/96	100/100/97
20%	94/93/69	100/100/96	100/100/96	97/97/78	100/100/98	100/100/97
40%	98/94/71	100/100/92	100/100/100	99/97/78	100/100/99	100/100/98
60%	86/82/53	99/98/86	100/100/87	86/76/57	99/97/88	100/99/94
80%	73/63/38	85/79/60	91/80/62	51/36/23	82/67/53	92/83/72
100%	7/0/0	6/0/0	6/0/0	4/0/0	5/0/0	2/0/0

to the prey, and transporting it back to the nest. We tested three strategies of the chain algorithm: Static, aligning and moving. In addition we tested the prey extension mechanism, a control module which can be mounted on top of the chain controller.

For what concerns path formation, we found that the more dynamic aligning and moving strategies outperform the static one. This is due to three reasons. First, the static chains are not straight, as opposed to the other two strategies, and therefore cover shorter distances from the nest. Second, the two dynamic chain strategies allow some motion to the chain members, which leads to an exploration of the arena even of chains that are already formed. Third and last, the static chains have a higher risk to create loops in the form of a successive order of three chain members.

The risk for this to happen is much lower for the other two chain strategies as they lead the chains to align.

The prey extension mechanism was found to be particularly useful for difficult tasks with small resources (i.e. robots) as in this way the resources are used more efficiently. However, for high densities of robots the prey extension mechanism was found to be less efficient, as in this case the arena gets quickly covered with the prey extending structure. Therefore, the chains take longer until they find the real prey instead of a robot activating its LEDs in the colour of the prey.

As shown in the difficulty test, the time to form a path grows more than linearly with the distance between prey and nest. This is due to the fact that the area that has to be explored to find the prey grows quadratically with the distance. Opposed to this, for subtasks assembly and transport, the completion time grows only linearly with the distance.

The chain controller exhibits good scalability characteristics with all three tested strategies. When the prey extension mechanism is not used the overall effort of the system, that is, the product of group size and completion time, decreases up to a group size of $N = 100$ robots, and then remains roughly constant. This means that the system's efficiency grows super-linearly. Due to the reasons stated in the above, the scalability is higher when the prey extension mechanism is not used. If it is used, the highest degree of efficiency is observed for a group size of $N = 20$ robots. For subtask assembly, the performance increases, but the efficiency, as measured through the overall effort, reaches its maximum for a group size of $N = 20$ robots. If more robots are used, there are usually also more robots trying to assemble at the same time, in this way disturbing each other from doing so. For subtask transport, the situation is similar. If the size of the pulling structure of transporting robots is large, there are several robots that can not perceive the goal direction (i.e. a chain member). Therefore, they can not contribute to the transport and even have a negative impact on the performance as they act as an additional weight that has to be transported.

We tested the chain controller in three different obstacle environments, and showed that, even though the performance drops, the chain controller is in principle able to form a path also if the environment contains obstacles. The performance drop is mainly due to three reasons. First, the presence of obstacles increases the difficulty of navigation, which is required for all three subtasks. Second, finding the nest or the prey becomes more difficult because they might be hidden behind the obstacles. Third, it might be impossible to form a straight path connecting nest and prey. This increases the length of the shortest path, and implies a particular difficulty for the aligning and the moving strategies. In fact, these two strategies attempt to align the chains. Due to this alignment, a chain can be broken up in cases where the line of sight of two neighboured chain members is blocked by an

obstacle. For subtask assembly, the performance drops because a connection to the prey can be restricted from several directions. For subtask transport, obstacles pose a particular problem as robots trying to move the prey can get stuck at an obstacle. Also in this case, the aligning and moving strategies have disadvantages to the static one because due to the aligning mechanism of the chains the path followed by the transporting robots is usually closer to corners, and can therefore more easily lead to a situation where the transporting structure gets stuck.

In a series of robustness tests, we showed that the chain controller can cope with a noisy perception of sensory data. We varied the noise of the direction at which objects are perceived using the camera, the distance at which objects are perceived using the camera, and of the proximity sensors. In general, the achieved robustness increases with the group size. The more robots there are, the better the system as a whole can cope with noisy conditions. Among the three kinds of sensory data, the performance of the chain controller was diminished most for the direction at which objects are perceived. This is the case for all three subtasks. While the performance also decreases when injecting noise to the distance perception of objects, the effect is less disruptive. In both of these cases of noise, the performance decrease was stronger if the prey extension mechanism was used. In general, the stability of the chain structure suffers more from noisy conditions than the stability of the structure of prey extending robots, because the stability of the latter relies on simpler rules. Therefore, a high level of noise leads to a higher degree of robots joining the prey extending structure, and therefore not leaving enough robots to form a path in a chain structure. For what concerns the proximity sensors, the chain controller showed a high degree of robustness. The proximity sensor is mainly used for avoiding collisions with objects in the environment, but it is not necessary for completing the task.

Finally, in a series of fault tolerance tests we analysed the ability of the chain controller to cope with individual failure by deactivating the camera, the LEDs, the proximity sensors, or the tracks of a given fraction of robots in the group. In general, as also observed for the robustness test, a large group size also leads to a higher degree of fault tolerance. Among the four tests performed, the proximity sensor again led to the lowest degree of performance decrease. By disabling the camera a robot acts like a mobile obstacle and can not contribute to any of the three subtasks. By disabling the LEDs the effect on the behaviour is even more pronounced. A high degree of erroneous robots then has a disruptive effect on existing chains. In the case of disabled tracks a robot acts like an immobile obstacle. However, if by chance it happens to be positioned at the right place, it can in principle contribute to form a path, but not to any of the other two subtasks.

Chapter 6

Chain Controller: Experiments with Robots

In this chapter we present the experiments performed on the *s-bot* robot platform. In Section 6.1 we describe the experimental setup. In Section 6.2 we present the results, and in Section 6.3 we draw some conclusions.

6.1 Experimental Setup

We have tested all three strategies of the chain controller examined in the previous chapter, and we have chosen the aligning strategy for our experiments because it was the most stable one. After preliminary tests we have set the probabilities per time step to join and leave a chain to $P_{in} = 0.14$, and $P_{out} = 0.007$.

The experiments take place in a bounded arena of size 500 cm × 300 cm. The nest is positioned in the centre of the arena. The prey is put at distance D from the nest towards one of the four corners. N *s-bots* are positioned on a grid composed of 60 points uniformly distributed in the arena. The initial position of each *s-bot* is assigned randomly by uniformly sampling without replacement. An *s-bot*'s initial orientation is chosen randomly from a set of 12 possible directions.

We conduct two sets of experiments. In the first set we examine three setups (N, D), with a linear relationship between group size N and distance D: $(2, 30)$, $(4, 60)$ and $(8, 120)$, where distances are expressed in cm. For each of the three setups we conduct 10 independent trials. In the second set of experiments we study a wider range of experimental setups, with group sizes $N = 1$, 2, 3, 4, 5, 6, 7, 8, 10 and 12, and distances (in cm) $D = 60$, 90, 120, 150, 180, 210 and 240. For each of these 70 setups we conduct a single trial.

The number of *s-bots* required to form a path connecting the prey with the nest depends on the initial distance between the two objects. To calculate lower

Table 6.1: Number of *s-bots* required to accomplish subtasks *path formation* (N_p), *assembly* (N_a) and *transport* (N_t) for different initial distances (D in cm) between the nest and the prey.

D	30	60	90	120	150	180	210	240
N_p	0	1	2	3	4	6	7	8
N_a	2	3	4	5	6	8	9	10
N_t	2	3	4	5	6	8	9	10

bounds for the number of *s-bots*, we assume the *s-bots* to be organised in a single chain, which is perfectly linear and directed towards the prey. The lower bound values can be calculated from the distances between the individual nodes forming the path and the distance of the last chain member with the prey (all distances are measured from centre to centre). The distance between neighbouring nodes forming the path is programmed to be constant. The actual distances vary slightly due to imprecision in the *s-bots*' perception. The average distance observed between neighbouring chain members (i.e., *s-bots*) is 27 cm. The average distance observed between the first chain member and the nest is 30.5 cm, and the distance between the last chain member and the prey is at most 38.5 cm.[1] The lower bound values so computed are given in Table 6.1. For the accomplishment of the overall foraging task, two additional *s-bots* are required (at the same time) to engage in transport.

During experimentation, the *s-bots* are fully autonomous. The only exception to this is when an *s-bot* topples over. To protect its hardware from being damaged (e.g., the camera mirror), we then remove the *s-bot* manually from the arena, and do not return it until the end of the trial.

We distinguish three levels of success which are satisfied respectively if subtasks *path formation*, *assembly* and *transport* are completed. The three different levels of success are satisfied, if

i. *path formation*: a path connecting nest and prey has been formed and can be traversed in both directions,

ii. *assembly*: two or more *s-bots* have been recruited and are physically assembled with the prey so that the transport can start,

iii. *transport*: the transport has been completed, that is, the prey, or an *s-bot* transporting it, is in physical contact with the nest.

[1] The diameter of the nest is 3.5 cm(= 30.5 cm − 27 cm) bigger than the diameter of the *s-bot*. The diameter of the prey is 3.5 cm bigger than the diameter of the *s-bot*; the remaining distance of 35 cm to the last chain member corresponds to the threshold that controls the transition *explore chain* → *join chain*.

Variables T_p, T_a and T_t denote the completion times (in s) for subtasks *path formation*, *assembly* and *transport*.

6.2 Results

We conduct two sets of experiments. In the first set we examine three setups (N, D), with a linear relationship between group size N and distance D: $(2, 30)$, $(4, 60)$ and $(8, 120)$, where distances are expressed in cm. For each of the three setups we conduct 10 independent trials. In the second set of experiments we study a wider range of experimental setups, with group sizes $N = 1$, 2, 3, 4, 5, 6, 7, 8, 10 and 12, and distances (in cm) $D = 60$, 90, 120, 150, 180, 210 and 240. For each of these 70 setups we conduct a single trial.

6.2.1 First Set of Experiments

We performed 30 trials in total. In 29 trials the overall task was successfully completed, that is, all three levels of success were satisfied. In the remaining trial, which belongs to setup $(N, D) = (8, 120)$, only the first two success levels were satisfied. The system failed to complete subtask *transport* as an *s-bot* incorrectly assumed that it was part of the transport structure.

Figure 6.1 shows a series of images taken during trial 8 of the setup $(N, D) = (8, 120)$.[2] Within the first 120 s, four *s-bots* find the nest (Figures 6.1a, b and c), and establish a path to the prey (Figure 6.1c). At time $t = 160$ s the first of the four remaining *s-bots* is successfully recruited, and thus has gripped onto the prey and activated its LEDs in red (Figure 6.1d). This *s-bot* alone is not strong enough to pull the prey. However, shortly after, another *s-bot* becomes part of the pulling structure (Figure 6.1e). The so formed group of two *s-bots* starts moving the prey. The transport group follows the path which gradually decomposes as the prey approaches the nest. The overall task is completed at time $t = 318$ s (Figure 6.1f).

Table 6.2 lists the measured completion times T_p, T_a and T_t for the different subtasks. In the trials with setups $(N, D) = (2, 30)$ and $(4, 60)$ subtask *path formation* is accomplished faster than any other subtask. In setup $(2, 30)$, no path needs to be formed; in setup $(4, 60)$, a path requires only one *s-bot* to find the nest and to form a chain in the direction of the prey (see Table 6.1). Most time was spent for subtask *assembly*, on average 211.9 s and 133.3 s for setups $(2, 30)$ and $(4, 60)$, respectively. Recall that all *s-bots* start from random positions in the arena and initially search the nest by performing a random walk. As the arena is large when compared to the *s-bot*'s perceptual range, it can take a considerable amount of time until 2 out of 2,

[2] A selection of movies can be found at `http://iridia.ulb.ac.be/supp/IridiaSupp2008-014`.

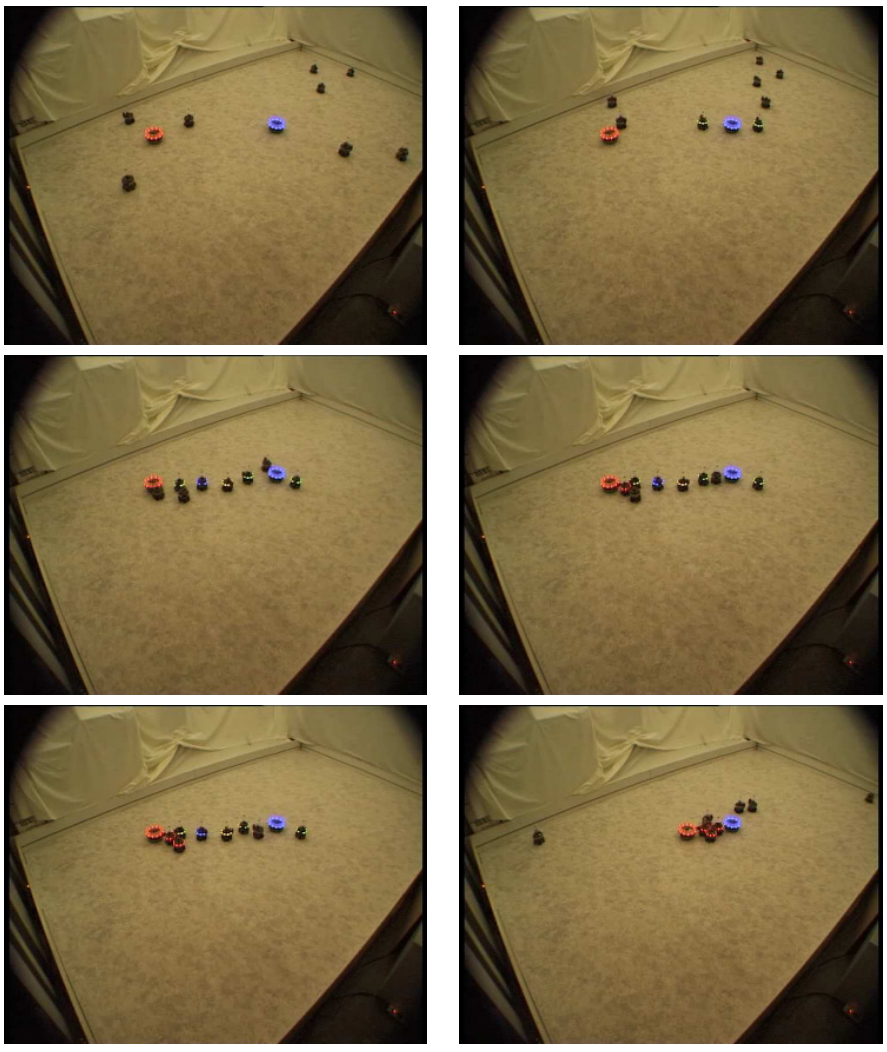


Figure 6.1: Sequence of images taken for a trial with group size $N = 8$ *s-bots* and distance $D = 120\,\text{cm}$ between the nest (blue cylindrical object) and the prey (red cylindrical object): (a) $t = 0\,\text{s}$, (b) $t = 55\,\text{s}$, (c) $t = 115\,\text{s}$, (d) $t = 160\,\text{s}$, (e) $t = 214\,\text{s}$, and (f) $t = 318\,\text{s}$.

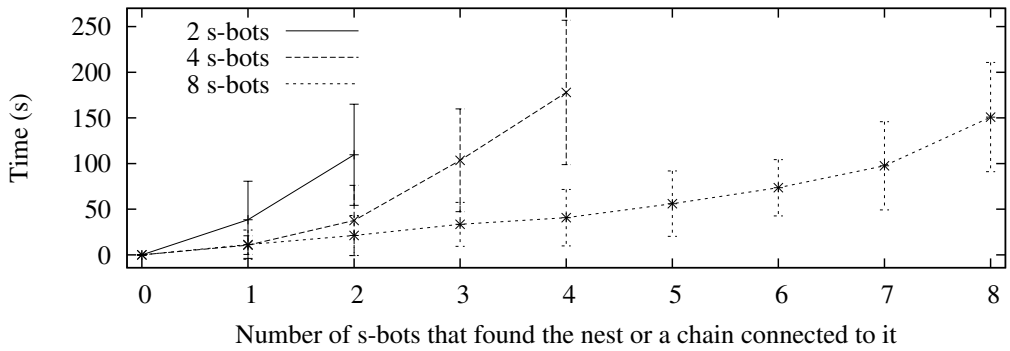


Figure 6.2: Time until the n-th *s-bot* finds either the nest or a chain (which provides a path to the nest) for the first set of experiments with $(N, D) = (2, 30)$, $(4, 60)$ and $(8, 120)$.

or 3 out of 4 *s-bots* have encountered the area from which they can perceive either the nest or a chain connected to it.

The situation is different for the setup with 8 *s-bots*. Only 87.4 s were spent on average for subtask *assembly*, which is significantly less than the times observed for group sizes 2 or 4, respectively (two-sided Mann-Whitney, 5% significance level). Also the time until a sufficient number of *s-bots* have found the nest drops from 109 s (103 s) for group size 2 (4) to 56 s for group size 8 (see Figure 6.2). One possible explanation is the higher degree of redundancy: in the system with 8 *s-bots*, five of them are candidate for assembly (while only 2 are required), whereas in the system with 2 *s-bots* (4 *s-bots*) only 2 (4) are candidate for assembly (see Table 6.1). Another possible explanation is linked to an effect of the formed chain: the larger the group size, the more *s-bots* take part in the chain formation process, in this way extending the area from which a path to the nest can be found by those *s-bots* still performing the initial random walk. This accelerates the process of gathering *s-bots* at the nest.

The time spent during transport, T_t, grows approximately linearly with the distance between nest and prey: 23.9 s, 51.8 s, and 95.6 s are required for the three setups with $D = 30, D = 60$ and $D = 120$. This suggests that for the transport it is not beneficial to increase the number of *s-bots*. Indeed, we observed that a pulling structure of 2–3 *s-bots* seems to be the optimal configuration for this particular transport task.

Table 6.2: Summary of the results for the first set of experiments. We investigated three setups (N, D) with a linear relationship between group size N and distance D: $(2, 30)$, $(4, 60)$ and $(8, 120)$. The value of T_p denotes the time it takes the *s-bots* to form the path, T_a denotes the time it takes the first two *s-bots* to be recruited, and T_t denotes the time it takes the *s-bots* to transport the prey to the nest. All results are given in seconds.

Trial	2 *s-bots*				4 *s-bots*				8 *s-bots*			
	T_p	T_r	T_t	$\sum T$	T_p	T_r	T_t	$\sum T$	T_p	T_r	T_t	$\sum T$
1	0	78	20	98	30	128	26	184	39	96	67	163
2	0	169	21	190	27	306	51	384	105	217	43	365
3	0	354	19	373	22	85	47	154	214	48	57	323
4	0	148	32	180	12	119	27	158	28	43	178	249
5	0	209	34	243	20	59	37	116	107	80	129	316
6	0	135	24	159	10	195	154	359	76			
7	0	394	14	408	29	106	25	160	69	86	86	241
8	0	414	25	439	28	65	82	175	112	56	150	318
9	0	132	23	155	48	119	28	195	72	154	49	285
10	0	86	114	144	19	151	41	211	114	42	91	247
Mean	0 (0%)	211.9 (89.9%)	23.9 (10.1%)	235.8 (100%)	24.5 (11.7%)	133.3 (63.6%)	51.8 (24.7%)	209.6 (100%)	93.6 (33.8%)	87.4 (31.6%)	95.6 (34.6%)	278.6 (100%)
Std. Dev.	0	127.5	6.3	125.1	10.8	72.8	39.8	89.3	51.9	59.7	46.6	60.3

Table 6.3: Overall level of success achieved for setups (N, D) in the second set of experiments: no success (0), subtask *path formation* accomplished (1), subtask *assembly* accomplished (2), and subtask *transport* accomplished (3). Entries in parentheses denote setups that were not tested as the number of *s-bots* N is clearly not sufficient to solve the task. Grey levels of cells represent the best achievable level of success (see Table 6.1): White denotes no success, light grey denotes success level 1, medium grey denotes success level 2, and dark grey denotes success level 3.

D / N	1	2	3	4	5	6	7	8	10	12
60	1	1	3	3	3	3	3	3	3	3
90	0	1	1	3	3	3	3	3	3	3
120	0	0	1	1	3	3	2	3	3	3
150	0	0	0	1	1	1	3	2	3	3
180	(0)	(0)	0	0	1	1	1	3	3	3
210	(0)	(0)	(0)	0	0	1	1	0	3	2
240	(0)	(0)	(0)	(0)	(0)	0	0	0	3	3

Table 6.4: Number of *s-bots* that are part of the path formed between nest and prey (if any) in setups (N, D) of the second set of experiments.

D / N	1	2	3	4	5	6	7	8	10	12
60	1	1	1	1	1	1	1	1	1	1
90		2	2	2	2	2	2	3	2	2
120			3	3	3	4	4	4	4	3
150				4	5	5	4	5	5	5
180					5	6	7	6	6	6
210						6	7		7	7
240									8	8

6.2.2 Second Set of Experiments

We examine the system under a wide range of group sizes (N) and prey distances (D). We conducted 70 trials, one for each different setup.[3]

Tables 6.3 and 6.4 give an overview of the level of success reached, and of the number of *s-bots* that formed the path connecting nest and prey.

In 46 out of the 70 setups a path can in principle be formed (see Table 6.1). In 44 out of the corresponding 46 trials the *s-bots* succeeded in forming a path. Only in

[3]Given that the limited amount of experimental time available allowed us to conduct max 70 trials, we had the choice between conducting them on 70 different setups, or to reduce the number of setups to obtain some mean performance on each of them. Although most researchers would say that the second choice should be followed, the optimal experimental design is the one that we have chosen Birattari (2005, 2004a,b).

two setups a path was not formed even though there were enough *s-bots*. For setup $(N, D) = (8, 210)$ two *s-bots* failed to join the chain as explorers, thereby making it impossible to form a path. For setup $(N, D) = (8, 240)$ a path requires all 8 *s-bots* to form one linear chain in the direction of the prey. This would take a long time as chains form into random directions, and several chains can form simultaneously. The trial was stopped at time $t = 3600\,s$ because of empty batteries of some *s-bots*.

For setups $(N, D) = (5, 180)$ and $(6, 210)$ a path was formed even though the number of *s-bots* was thought to be insufficient. A path of five (six) *s-bots* has a maximum predicted length of $30.5 + 4 \cdot 27 + 38.5 = 177\,\text{cm}$ ($30.5 + 5 \cdot 27 + 38.5 = 204\,\text{cm}$), which is 3 cm (6 cm) less than the distance that needs to be covered, and therefore still within the range of perceptual error of the camera.

In 33 out of the 46 setups, also subtasks *assembly* and *transport* can in principle be accomplished by the given number of *s-bots*. In 29 out of these 33 setups, the *s-bot* group was able to do so, thereby the entire task was completed. In three setups, however, although a path was formed and two or more *s-bots* were recruited (and gripped onto the prey), the transport back to the nest was not successful. For setups $(N, D) = (7, 120)$ and $(8, 150)$ the gripper of one of the *s-bots* in the pulling structure opened during the transport phase, in this way blocking the transport. For setup $(N, D) = (12, 210)$ the trial was stopped when 7 *s-bots* formed a linear structure to pull the prey. In such a long structure most members cannot perceive the path, thus the prey could not be moved any more. For setup $(N, D) = (6, 150)$ the formed path was not linear and thus required one additional *s-bot* (five in total). The remaining *s-bot* was not capable of transporting the prey alone.

Table 6.5 lists the completion times for each of the three subtasks *path formation*, *assembly* and *transport*. Table 6.6 lists the overall completion time.

Figure 6.3 shows state diagrams for four selected setups (N, D): $(12, 150)$, $(12, 240)$, $(7, 150)$ and $(7, 240)$. In the first three cases the task was successfully accomplished. In the last case the system failed as the number of *s-bots* was too small to form a path, and thus also too small to accomplish the task. In the following the four setups are discussed in more detail:

- $(N, D) = (12, 150)$: All *s-bots* start in state `search chain`. Once the nest has been found, they aggregate into chains. At $t = 78\,s$, a path to the prey consisting of five chain members is established. Even though a path to the prey is formed, other *s-bots* that find the nest self-organise into an additional chain. Recall that the formation of the path is not explicitly communicated among the *s-bot* group. However, as the *s-bots* in the newly formed chain leave this chain with a constant positive probability, after some time only the chain forming the path remains. At time $t = 128\,\text{s}$ a first *s-bot* is recruited and grasps the prey, joined by a second *s-bot* about 16 s later. While the prey is transported towards the nest, the chain gradually dissolves. During

Table 6.5: Completion times (in s) of subtasks *path formation* (T_p), *assembly* (T_a), and *transport* (T_t) in setups (N, D) of the second set of experiments. If no value is given the respective subtask was not successfully completed.

T_p: time required for *path formation*										
D / N	1	2	3	4	5	6	7	8	10	12
60	82	144	14	64	14	15	11	12	19	1
90		76	45	23	99	28	32	35	14	8
120			192	88	174	483	160	88	97	65
150				662	337	486	32	379	511	78
180					317	1975	902	541	222	1649
210						2135	988		2370	810
240									827	335
T_a: time required for *assembly*										
D / N	1	2	3	4	5	6	7	8	10	12
60			286	62	67	104	257	135	79	61
90				59	272	69	159	168	57	41
120					181	198	281	458	35	110
150							59	635	314	66
180								616	97	729
210									176	170
240									229	138
T_t: time required for transport										
D / N	1	2	3	4	5	6	7	8	10	12
60			17	121	246	276	33	47	24	183
90				41	23	158	80	20	545	129
120					56	84		245	144	604
150							201		123	168
180								64	146	170
210									582	
240									258	442

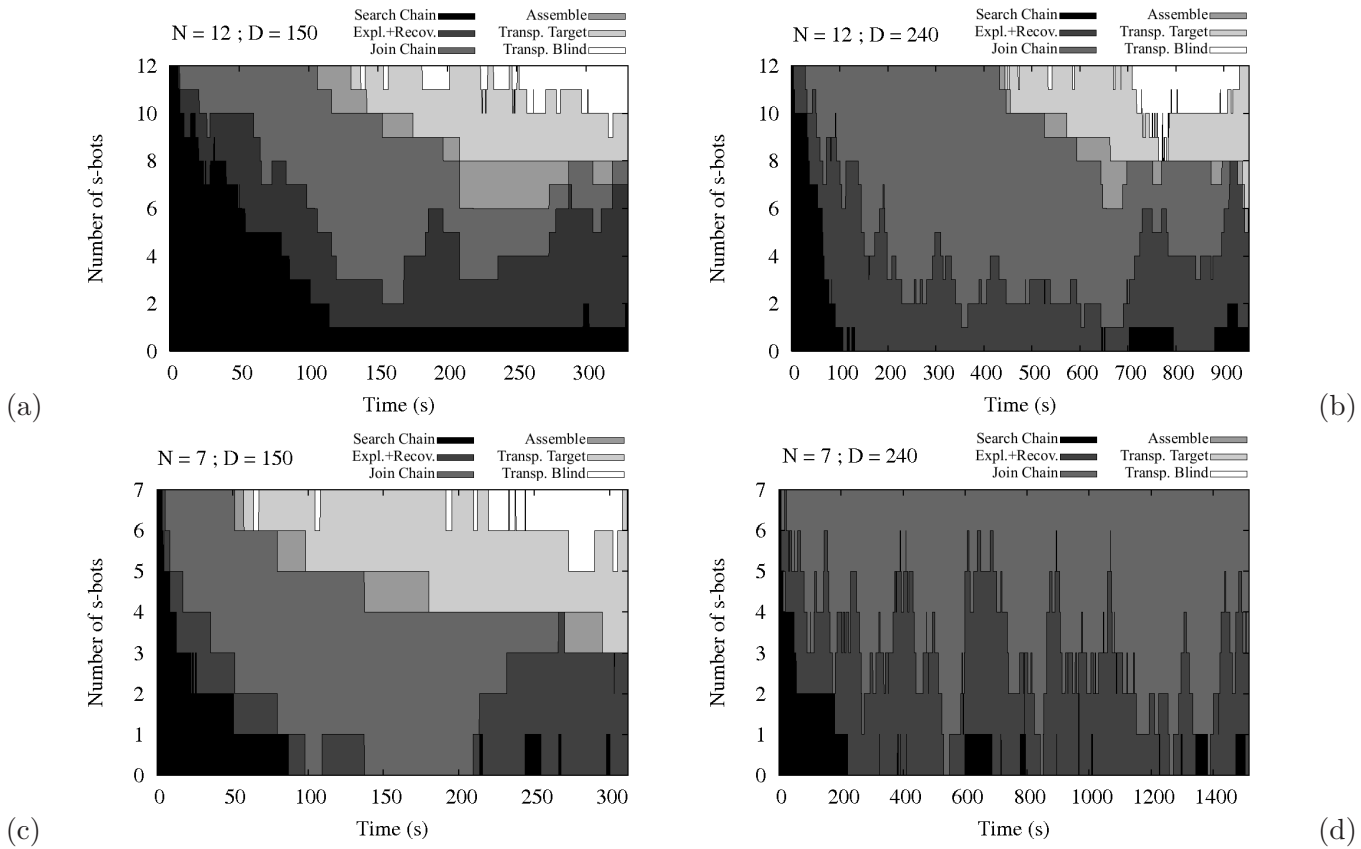


Figure 6.3: State diagrams for four selected setups (N, D) from the second set of experiments: (a) $(12, 150)$, (b) $(12, 240)$, (c) $(7, 150)$ and (d) $(7, 240)$. The respective grey levels indicate the number of *s-bots* in states `search chain`, `explore chain` and `recovery`, `join chain`, `assemble`, `transport target` and `transport blind`.

Table 6.6: Completion time (in s) of the overall foraging task in setups (N, D) of the second set of experiments. If no value is given the overall task was not successfully completed.

D / N	1	2	3	4	5	6	7	8	10	12
60			327	247	327	395	301	194	122	245
90				123	394	710	271	223	616	178
120					411	765		791	276	779
150							292		948	312
180								1221	465	2548
210									3126	
240									1314	915

the transport, additional *s-bots* try to assemble with the pulling structure. Two of them succeed, whereas others fail because the pulling structure is in motion. By looking at the state diagram in Figure 6.3a, one can see that some of the *s-bots* engaged in transport are not capable of perceiving the path (see white area). Thus, we have an example where the *s-bots* exhibit a hierarchy of teamwork: the group of *s-bots* that cannot perceive the path need to interact with the group of *s-bots* that can perceive the path, and thereby form a team. This team, which is composed of all transport *s-bots* can be considered a higher order entity. It forms part of another team which includes another higher order entity—the group of *s-bots* maintaining or decomposing the path.

- $(N, D) = (12, 240)$: Figure 6.4 shows a sequence of images taken during this trial. During the path formation phase, two chains are formed concurrently (Figure 6.4b), and it takes several rearrangements of the chains until at time $t = 335\,s$ a path is formed. This path consists of a chain of 8 *s-bots* (Figure 6.4c). Shortly thereafter, two *s-bots* get recruited and assemble with the prey (Figure 6.4d). During transport, most *s-bots* of the pulling structure lose sight of the path, which gradually dissolves, and the prey is moved in the wrong direction (Figures 6.4e and f). However, the path gets re-established by a new *s-bot* extending the chain in the direction to the prey (Figure 6.4g). As a consequence, the transport resumes and can be completed (Figure 6.4h). This is an example of a situation in which teamwork among higher-order entities (such as teams or groups) requires a participating entity to adapt its configuration to unexpected environmental circumstances.

- $(N, D) = (7, 150)$: At time $t = 32\,\mathrm{s}$ a path between nest and prey is already established. At time $t = 91\,s$, two *s-bots* have been recruited and are assembled with the prey. The five remaining *s-bots* are aggregated in the chain form-

ing the path. During the transport, chain members disaggregate once in the immediate vicinity of the prey, and follow the path back to the nest to rest. After some time, the very same *s-bots* resume activity, follow the path, and eventually two of them assemble with the pulling structure and participate in transport. This is an example of how the composition of teams can adapt to changes in the workload of the underlying subtasks.

- $(N, D) = (7, 240)$: The *s-bot* group is too small to form a path. The state diagram shows a high flux between states `explore chain` and `join chain`. At some stages of the trial all *s-bots* are aggregated into chains. However, given that no prey is found, the chains always dissolve. At some stages only one *s-bot* is aggregated into chains. Thus, the system effectively restarts the search process and can form new chains into unexplored areas of the environment. The diagram suggests that the system retains this ability during the entire trial (i.e., for 25 minutes).

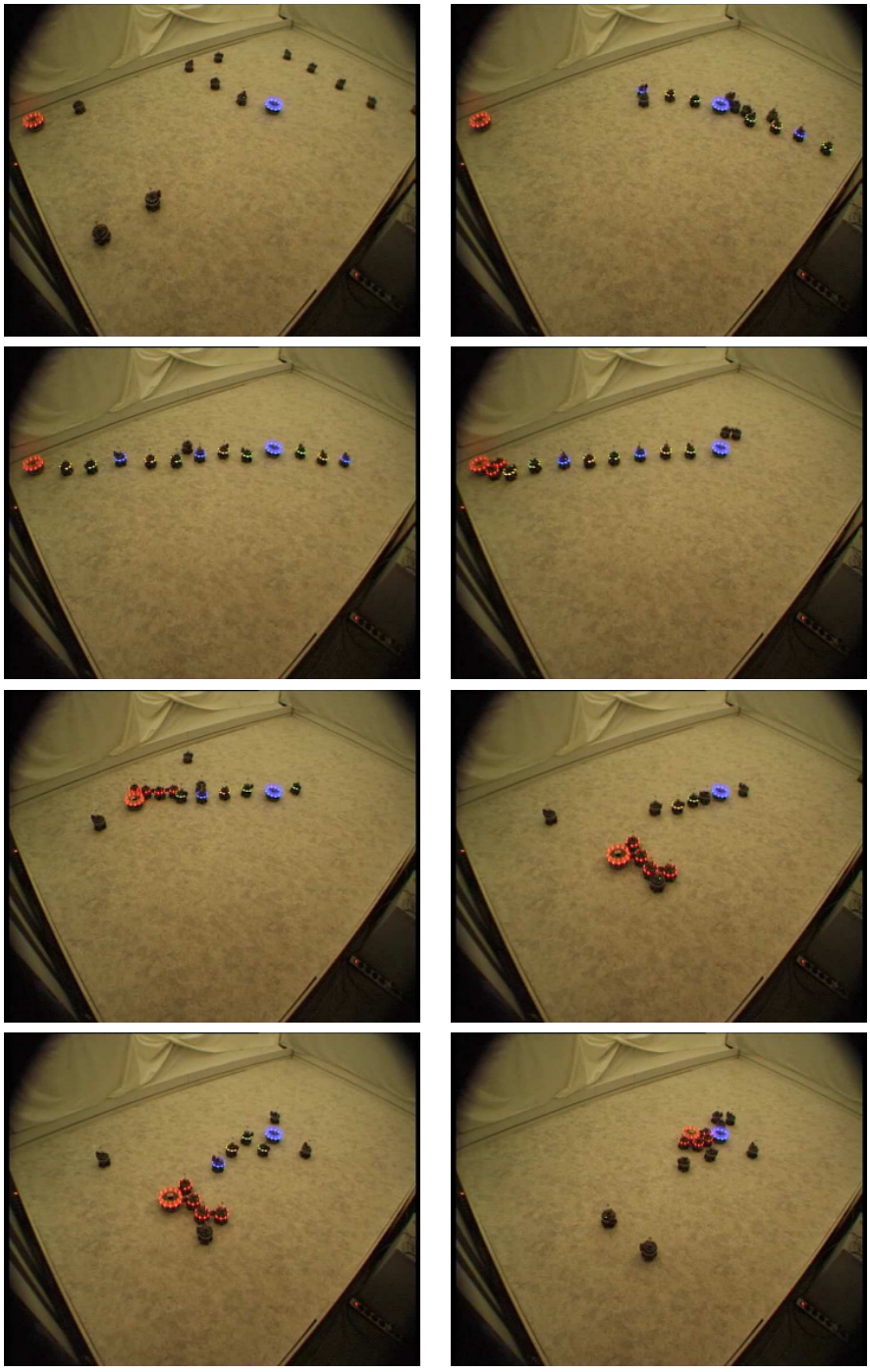
Figure 6.5 shows the number of distinct behavioural roles (i.e., states) an individual *s-bot* performed during the trials of the second set of experiments. In 75% of the cases, an *s-bot* performed either four or five of the seven roles. This suggests that the *s-bots* are indeed inter-changeable. The number of roles does not follow a binomial distribution. Only in 4% of the cases, an *s-bot* performed less than four behaviours during the trial. However, in 15.7% of the cases, an *s-bot* performed all seven behaviours.

Figure 6.6 shows the number of times an *s-bot* changes its behavioural role during the trials of our experiments. Six to ten changes in the behaviour have most frequently been observed. Note, however, that both mean and median number of changes are higher than this range of values (20.9 and 14.5, respectively).

6.3 Conclusions

6.3.1 Results

In this chapter, we have presented an experimental study in which a colony of autonomous robots has to solve a complex foraging task. The task requires a range of subtasks to be performed including (a) exploration of the environment, (b) formation of a path between a prey and a nest, (c) assembly of nest-mates to the prey, (d) self-assembly into pulling structures, and (e) group transport of the prey back to the nest. Due to the limited abilities of the robots, the accomplishment of the task requires the concurrent activity of at least i robots, where $i \in \{2, 3, 4, 5, 6, 8, 9, 10\}$ depends on the experimental setting. For $i > 2$, the accomplishment of the task requires division of labour, in other words, the robots need to perform different subtasks concurrently.



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 6.4: Sequence of images taken for the trial with group size $N = 12$ *s-bots* and distance $D = 240\,\text{cm}$ between the nest (blue cylindrical object) and the prey (red cylindrical object): (a) $t = 0\,\text{s}$, (b) $t = 140\,\text{s}$, (c) $t = 360\,\text{s}$, (d) $t = 480\,\text{s}$, (e) $t = 720\,\text{s}$, (f) $t = 780\,\text{s}$, (g) $t = 810\,\text{s}$, and (h) $t = 900\,\text{s}$.

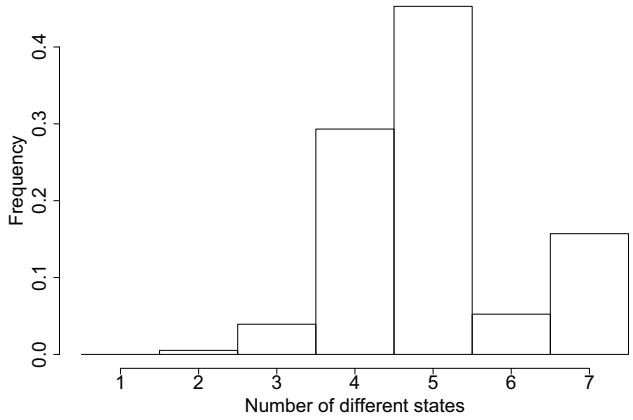


Figure 6.5: Number of distinct behavioural roles (i.e., states) an *s-bot* performed during a trial. Data from all *s-bots* and all trials of the second set of experiments.

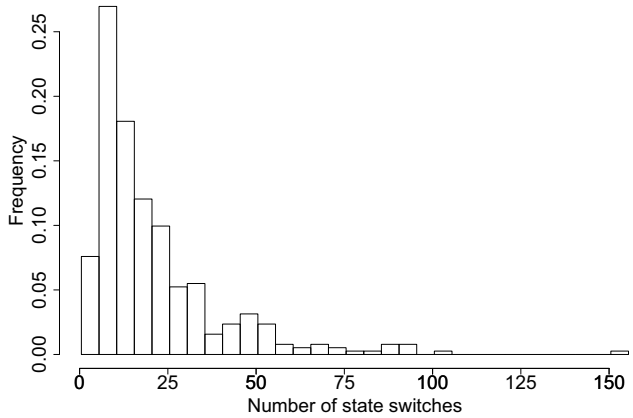


Figure 6.6: Number of times an *s-bot* changed its behavioural role (i.e., states) during a trial. Data from all *s-bots* and all trials of the second set of experiments.

The robots also have to perform subtasks that are organised sequentially. In general, we believe that the investigated problem provides a framework that captures the essence of a variety of problems that are addressed at the collective level in social insect colonies.

Inspired by the behaviour of the natural counterparts, we developed a relatively simple, decentralized control algorithm. Each robot executes a copy of this algorithm, thus all robots have an identical control. The robots do not require any explicit knowledge of the environment beyond their local perceptual range.

A series of experimental results from systematic trials with up to twelve physical robots confirm the efficacy of the system. In almost all of the trials where the group size is sufficient to accomplish the overall task, the group succeeded in retrieving the prey to the nest.

One of the mechanisms we identified to be crucial for the performance of the system is a robot's ability to recover from situations in which it is prevented from achieving its current objective. Such a recovery mechanism was applied in the behaviours path formation, self-assembly and (to some extent) group transport. For path formation, chains of visually connected robots that do not extend to a prey disaggregate with some probability and re-aggregate into other directions. For self-assembly, the recovery mechanism consists of a simple timeout after which the robot gives up assembling and moves back to the nest instead. For group transport, a recovery mechanism allows robots unable to perceive the path to interact with those robots that are able to perceive the path. Still in a few trials the task was not completed because of some unexpected behaviour during the transport phase. A recovery mechanism (allowing to suspend the transport behaviour) might have prevented stagnation in such circumstances.

It is worth noting that the assignment of individual roles to the robots was context-dependent, and thus changed both in space and in time. For example, a transporter robot (i.e., a robot assembled in a pulling structure with the prey) would behave as a "leader" or as a "follower", depending on whether it perceived the path towards the nest or not. An explorer robot (i.e., a robot moving along a chain of robots) would become a transporter robot if it encountered the prey and succeeded in assembling to it, however it could assimilate another role under other circumstances.

By self-organising, the colony displayed a dynamically changing hierarchy of teamwork in which collaboration took also place among high-order entities including groups and teams. The higher-order entities (including the entire system) proved surprisingly robust with respect to the inaccurate and sometimes malfunctioning behaviour of their component modules—parts of a robot such as the tracks, entire robots, and even groups of robots broke down or exhibited unexpected behaviour.

We believe that these experiments are among the most sophisticated examples

of self-organisation in robotics to date. The study confirms that complex forms of division of labour can indeed result from the interactions of robots that follow relatively simple and local rules. The study also demonstrates that teamwork requires neither individual recognition (the robots we use are inter-changeable) nor inter-individual differences (the robots we use are homogeneous in terms of "morphology" and "brain"), and as such might contribute to the ongoing debate on the role of such characteristics for the division of labour in social insects.

6.3.2 Transfer of Controller from Simulation to Robot

When designing the chain controller in simulation, we tried to model all characteristics of the robot as realistically as possible. However, the robot platform was still in a prototype phase, and some of the characteristics had to be assumed without knowing exactly if the assumptions would hold on the real robot. In this section we report on the problems we observed during the transfer of the controller from simulation to the real robot.

Four of the main problems we observed concern the camera:

i. Number of colours: In our simulation model, we assumed that a robot could distinguish between at least five colours. We used three colours for the cyclic directional pattern that leads to the directionality of a chain, one colour for the nest and one colour for the prey. However, when implementing the camera software, we recognized that only four colours can be reliably distinguished. We therefore redesigned the controller and used one of the chain colours for the nest. The robots are then not able to uniquely identify the nest.

ii. Length of a time step: In simulation we assumed a control time step to have a length of $100,\mathrm{ms}$. However, due to the processing speed of the robot and the camera software used we had to increase the length of a time step to $125,\mathrm{ms}$. This had implications on the possible maximum speed of a robot, which therefore had to be decreased accordingly.

iii. Erroneous perception: We observed that some robots have an erroneous perception with the camera for individual time steps, that is, that either they perceive a coloured object when there is none, or that they perceive a wrong colour. We therefore implemented a simple mechanism that filters out these erroneous perceptions. A colour perceived with the camera software is then only accepted in case it is perceived for eight consecutive time steps. Also this decreases the possible maximum speed of a robot because it decreases a robot's reaction time.

iv. Perceptual range: In simulation we assumed a perceptual range of $100,\mathrm{cm}$ for all colours perceived with the camera. On the real robot however, we observed

that the actual perceptual range differs for each colour and for each robot. In average the perceptual range is approximately 50, cm for the three chain colours, and approximately 80, cm for the prey colour red. Furthermore, due to the semi-spherical shape of the mirror the precision of a distance to a coloured object decreases quickly with its distance. We observed that a reliable distance estimate can only be made up to a distance of 30, cm. We set the distance between two chain members accordingly because it is important for a chain's stability that the distance between two chain members is roughly constant.

After observing these problems we redesigned the simulation model used so that they are more realistic. The simulation results presented in this thesis were conducted using the simulation model that takes the observed problems into account. However, we still observe some differences between simulation and real robot. This mainly concerns the more physical tasks of assembly and transport. The situations that occur during these tasks are far more complicated to simulate than for the chain controller as it mainly relies on visual interactions as opposed to the physical interactions required for assembly and transport.

Chapter 7

Vectorfield Controller

In this chapter we describe the vectorfield controller. We first give a general description of the behaviour in Section 7.1 and of the different variants used in Section 7.2. In order to integrate the different behaviours required to solve the overall task, we follow a behaviour based approach (Arkin, 1998) as it allows us to comfortably merge different approaches and sub-controllers into one structure. Each behaviour consists of a collection of motor schemas, that is, the low level control mechanisms. The motor schemas used are described in Section 7.3. Afterwards, the different behaviours are explained in Section 7.4, and the rules that trigger a transition from one behaviour to another are detailed in Section 7.5.

7.1 General Description

The controller that we designed and implemented to run our experiments consists of nine behaviours, each of which is designed to achieve a specific goal. The overall task can be split into the subtasks path formation, assembly and transport. The individual behaviours for the path formation subtask, which is the focus of this thesis, are implemented using the motor schema paradigm. For the assembly and transport subtasks the behaviours are based on the work of Groß et al. (2006a); Groß and Dorigo (2004) and Groß et al. (2006b), and rely on neural networks or simple hand written commands. The nine behaviours are detailed in Section 7.4.

The initial situation is the same as for the chain controller. The robots are randomly positioned and first have to search for the nest (`Search VF` behaviour). The first robot to find the nest stops moving and activates a colour pattern with its LEDs pointing towards the nest (see Figure 7.1). Another robot that perceives such a colour pattern uses the indicated direction to move away from the nest (`Explore VF` behaviour). It will end up joining the structure of LED-activated robots when it reaches the *border*, that is, when it perceives only one LED-activated robot at a dis-

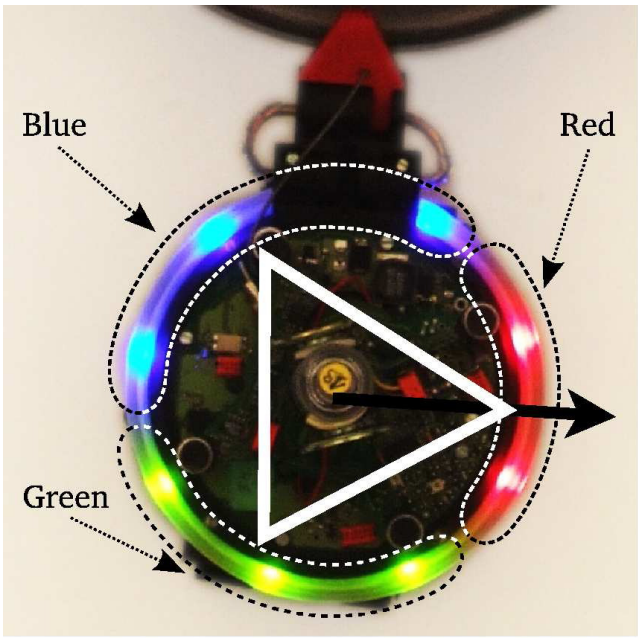


Figure 7.1: A robot activating its LEDs to indicate a direction as employed by the vectorfield controller.

tance greater than 30 cm. It activates its LEDs to point towards the LED-activated robot it perceives (`Join VF` behaviour). The resulting robot structure can be considered as a vectorfield globally leading to the nest. The structure of the vectorfield exhibits stronger branching than the chain controller, because the vectorfield controller employs a different rule, which allows a robot to join the structure at various positions, and not only at the tail.

The process of leaving the vectorfield is probabilistic. At each time step a robot leaves the vectorfield with probability P_{out}, but only if it is situated at the vectorfield's border. Similarly to the chains, the process of joining/leaving the vectorfield leads to a continuous exploration of the environment until the prey is found. If a robot of the vectorfield perceives the prey it sets its value of P_{out} to zero, so that the established path becomes stable. Again, there are two possibilities: If the prey is closer than 30 cm, the task of path formation is successfully accomplished, and if the prey is further away than 30 cm, then other robots can still join the vectorfield.

When a robot leaves the vectorfield it starts a random walk during which it does not react to the perception of the vectorfield (`Random Walk` behaviour). Due to the random walk it might reach a different branch, stay in the vicinity of the same branch, or lose contact with the vectorfield completely. The time it remains in this

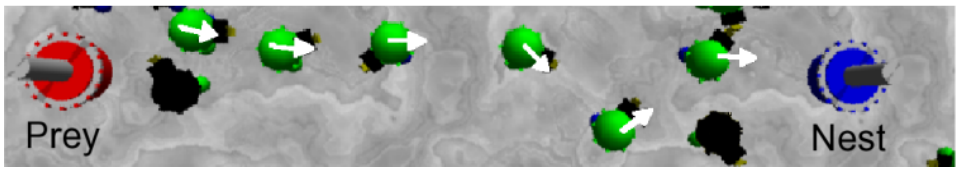


Figure 7.2: A vectorfield. The six robots with an arrow on their top represent a path between nest and prey. The arrows indicate the directions the robots are pointing to and that lead towards the nest.

state is determined by the probability P_{in}. When the robot enters the search state again, it continues the random walk until it perceives the vectorfield. This process is continued until the vectorfield encounters the prey and in this way forms a path. Figure 7.2 shows a sequence of robots forming such a path. The arrows on top of the robots represent the directions they are pointing to.

Once a path is formed, it is maintained and other robots arriving at the prey attempt to assemble to it (`Assemble` behaviour). If a robot that tries to assemble to the prey does not succeed within a certain time, it gives up, moves back to the nest and rests for a while (`Recovery-A` behaviour). When a sufficient number of robots has assembled to the prey, the prey transport starts. Robots assembled to the prey transport it by moving towards the closest perceived member of the vectorfield (`Transport Target` behaviour). When the prey pulling structure moves close to a vectorfield member, the latter leaves the vectorfield and moves back to the nest to rest for a while (`Recovery-P` behaviour). In this way the pulling structure of robots is guided from node to node of the vectorfield to eventually reach the nest. A robot leaving the vectorfield to rest at the nest emits a sound signal for a period of 30 s. A robot transporting the prey and perceiving this sound signal reacts to it by pausing the transport. Otherwise, if it continues moving towards the sound emitting robot, there is a risk that the robot is blocked from moving back to the nest. This situation can occur when the pulling structure of transporters touches the respective robot. No other robots react to the sound signal.

7.2 Variants

We implemented the same prey extension mechanism as used for the chain controller. When this mechanism is active, robots that perceive the red prey, but no chain, activate their red LEDs with a certain probability, in this way increasing the area in which the prey can be perceived by the vectorfield (`Extend Prey` behaviour). This mechanism potentially speeds up the path formation process because a second path,

starting from the prey, is formed in parallel to the chains.

As we use a behaviour based architecture the same module used for the chain controller can easily be integrated without any need for modifications.

7.3 Motor Schemas

The same motor schemas Adjust_distance, Avoid_collisions and Move_straight as used for the chain controller are used for the vectorfield controller as well. The only new motor schema used for the vectorfield controller is Follow_vectorfield:

- **Follow_vectorfield(***indicated_directions***)**: returns a vector $\overrightarrow{v_{FV}}$ that takes into account directions α_i as indicated by all perceived robots i in the vectorfield, and returns a vector that points in the opposite direction, in this way moving away from the nest:

$$\overrightarrow{v_{FV}} = -\sum_i \begin{pmatrix} cos(\alpha_i) \\ sin(\alpha_i) \end{pmatrix}.$$

The active motor schemas are summed up, weighting each one with a gain value g_i. The individual gain values are given in the next section and were found through trial and error. Once the active motor schemas have been summed up, the resulting vector $\overrightarrow{v_{RES}}$ has to be translated into movement of the two wheels. This is done by the following function:

$$\begin{pmatrix} lSpeed \\ rSpeed \end{pmatrix} = \begin{cases} \begin{pmatrix} cos(2 \cdot \alpha_{RES}) \\ 1 \end{pmatrix} & , 0 \leq \alpha_{RES} < \frac{\pi}{2} \\ \begin{pmatrix} cos(2 \cdot \alpha_{RES} - \pi) \\ -1 \end{pmatrix} & , \frac{\pi}{2} \leq \alpha_{RES} < \pi \\ \begin{pmatrix} -1 \\ -cos(2 \cdot \alpha_{RES}) \end{pmatrix} & , \pi \leq \alpha_{RES} < \frac{3 \cdot \pi}{2} \\ \begin{pmatrix} 1 \\ -cos(2 \cdot \alpha_{RES} - \pi) \end{pmatrix} & , \frac{3 \cdot \pi}{2} \leq \alpha_{RES} < 2 \cdot \pi \end{cases},$$

where $lSpeed$ and $rSpeed$ denote the normalized speed of left and right wheel, and α_{RES} is the desired direction of movement with respect to the current heading. The resulting speed of the wheels is independent from the length of the summed vector v_{RES}, and depends on the maximum allowed velocity v_{max}, which is set to $12.37 \frac{cm}{sec}$, the same maximum velocity as used for the chain controller.

7.4 Behaviours

The behaviours and the rules that trigger a transition from one behaviour to another are illustrated by the state diagram in Figure 7.3. Each state corresponds to a robot

behaviour, and arrows connecting states represent behaviour transitions. Using the aforementioned motor schemas as basic building blocks, the individual behaviours used for path formation were implemented as follows:

- **Search VF**: perform a random walk (until the vectorfield is perceived). LEDs are off. Active motor schemas: Move_straight($g_{MS} = 1$), Random($g_{RA} = 0.1$), Avoid_collisions($g_{AC} = 0.1$).
- **Explore VF**: follow the vectorfield away from the nest. LEDs are off. Active motor schemas: Follow_vectorfield($g_{FV} = 1$), Avoid_collisions($g_{AC} = 0.1$).
- **Join VF**: do not move. The LEDs are used to activate a pattern which indicates the direction towards the precedent robot in the vectorfield. Active motor schemas: none.
- **Random Walk**: perform a random walk (even if the vectorfield is perceived). LEDs are off. Active motor schemas: Move_straight($g_{MS} = 1$), Random($g_{RA} = 0.1$), Avoid_collisions($g_{AC} = 0.1$).
- **Recovery-P**: Move back to the nest and rest. Emit a sound signal for 30 s. LEDs are off. Active motor schemas: Move_perpendicular($g_{MP} = 1$), Adjust_distance($g_{AD} = 10$), Avoid_collisions($g_{AC} = 0.1$).
- **Extend Prey**: The LEDs are activated with red. Keep a constant distance of 60 cm to the closest red object perceived. This is either the prey or another robot in the prey extending structure. Active motor schemas: Adjust_distance($g_{AD} = 10$), Avoid_collisions($g_{AC} = 0.1$).

For assembly and transport the same behaviours as for the chain controller are used:

- **Assemble**: We use the same hand written control algorithm as used for the chain controller. The simulated robot approaches the object towards which it attempts a connection, tries to connect, and given the distance and the angle towards the object it has a given probability that the assembly is successful. If the attempt to connect was not successful it moves backwards, and tries to connect again.
- **Transport**: if a sound signal is perceived the robot rests. Otherwise, if a chain member is perceived, the robot orients its chassis towards the closest chain member, which indicates the direction to the nest, and starts pulling. A detailed description of the behaviour can be found in Groß et al. (2006b). If no chain member is perceived a robot that does not perceive the path does not move at all.

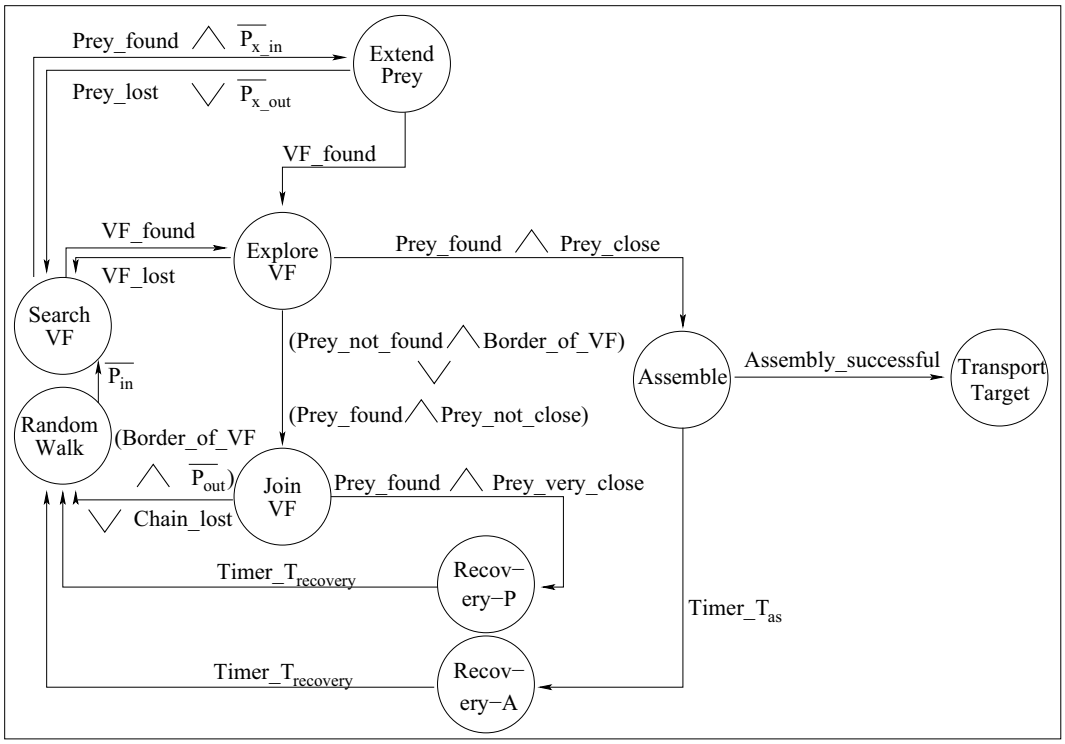


Figure 7.3: State diagram of the finite state machine that controls each robot. Circles represent states (i.e., behaviours). Edge labels specify conditions that trigger transitions between the corresponding states. The initial state is `Search VF`. $\overline{P_{in}}$, $\overline{P_{out}}$, $\overline{P_{x-in}}$ and $\overline{P_{x-out}}$ are boolean variables, which are set to *True* with the probabilities P_{in}, P_{out}, P_{x-in} and P_{x-out}, and *False* otherwise. The value of P_{in} (P_{out}) determines the rate at which robots join (leave) the vectorfield. The value of P_{x-in} (P_{x-out}) determines the rate at which robots join (leave) the prey extending structure.

Finally, a recovery behaviour, similar to the one used for the path formation, is used for the assembly as well. The only difference is that no sound signal is emitted.

- **Recovery-A**: Move back to the nest and rest. LEDs are off. Active motor schemas: Move_perpendicular($g_{MP} = 1$), Adjust_distance($g_{AD} = 10$), Avoid_collisions($g_{AC} = 0.1$).

7.5 Behaviour Transitions

The following set of conditions trigger behaviour-transitions:

- **Search VF → Explore VF**: if the vectorfield (this includes the nest) is perceived. Note that a robot in the `Search VF` state does not react to the perception of the prey, unless the prey extension mechanism is used.
- **Search VF → Extend Prey**: if the prey, but no chain member is perceived, the robot joins the prey extending structure with probability P_{x-in} per time step.
- **Explore VF → Search VF**: if the vectorfield is no longer perceived.
- **Explore VF → Join VF**: (i) if the prey is not perceived and only one vectorfield robot is perceived at a distance $> 30\,\text{cm}$, or (ii) if the prey is perceived at a distance $> 30\,\text{cm}$.
- **Explore VF → Assemble**: if the prey is detected at a distance $< 35\,\text{cm}$.
- **Join VF → Random Walk**: (i) if the robot is situated at the border of the vectorfield, it leaves the vectorfield with probability P_{out} per time step, or (ii) if the contact to the vectorfield is lost, that is, if there is no other vectorfield robot that points towards this robot within a threshold angle of 50^o for five consecutive time steps.
- **Join VF → Recovery-P**: if the prey is perceived at a very close distance (i.e., less than 5 cm), which only occurs if the prey is transported towards the vectorfield.
- **Random Walk → Search VF**: with probability P_{in} per time step.
- **Extend Prey → Search VF**: if no red object is perceived any more, or with probability P_{x-out} per time step.
- **Extend Prey → Explore VF**: if the vectorfield is perceived.

- **Recovery-P → Random Walk**: if $T_{recovery} = 30\,\text{s}$ has elapsed.
- **Assemble → Recovery-A**: if the robot does not succeed in connecting to an object within $T_{as} = 90\,\text{s}$.
- **Assemble → Transport-Target**: if the robot succeeds in connecting to an object.
- **Recovery-A → Random Walk**: if $T_{recovery} = 30\,s$ has elapsed.

Chapter 8

Vectorfield Controller: Experiments in Simulation

In this chapter we present the results obtained in the TwoDee simulation environment for the vectorfield controller, and also compare the performance with and without the prey extension mechanism. After describing the experimental setup in Section 8.1, we explain the method used for parameter selection and give an overview of the performance in Section 8.2, and then present the results with the selected parameters for the different tests performed in Section 8.3.

8.1 Experimental Setup

The simulation environment is the same as for our experiments with the chain controller in Chapter 5. We employ a bounded arena of size $5\,\mathrm{m} \times 5\,\mathrm{m}$. The task consists in forming a path between two locations in the environment, the nest and the prey, to assemble to the prey and to transport it back towards the nest. The nest is placed in the centre of the arena, and the prey is placed towards one of the corners. Obstacles are cubes with a side length of 0.5 m (i.e., one obstacle occupies 1% of the arena). An instance of the task is defined by the triplet (N, D, O), where:

- N is the robot group size,
- D is the distance between nest and prey (in meters),
- O is the number of obstacles in the environment.

The initial position and orientation of the robots, as well as the positions of the obstacles, are chosen randomly.

The primary performance measure is the *completion time*. We distinguish the three completion times for the three subtasks path formation (T_p), assembly (T_a),

and transport (T_t). For practical reasons, we allow a maximum completion time of 10,000 seconds. If this time does not suffice to establish a path, the trial is stopped and considered to be a failure. As a second performance measure we use the *success rate*, which we define as the ratio of successful trials. Again we distinguish three success rates, one for each subtask.

In the following sections we describe the method used for selecting the four parameters characterizing our controller. Afterwards, we present the results of an extensive post-evaluation of the selected parameter sets.

8.2 Parameter Selection

The overall behaviour of the vectorfield controller is a function of the two parameters P_{in} and P_{out}, as they determine the rate at which robots join and leave the vectorfield. Additionally, in the case where the prey extension mechanism is employed, the two parameters P_{x-in} and P_{x-out} determine probability per time step at which the robots join or leave the prey extending structures. To assess the general impact of these probabilities we have conducted a parameter study. For each probability we examined ten values defined by $0.001 * 2^x$, with $x \in \{0, 1, 2, 3, \ldots, 9\}$, resulting in 10 candidates in the range $[0.001, 0.512]$. We first discuss the parameter landscape and select a parameter set for the probabilities of P_{in} and P_{out} in Section 8.2.1 using the same racing algorithm as for the chain controller (Birattari et al., 2002; Birattari, 2005). Then, based on these two selected parameters, we discuss the parameter landscape and select a parameter set for the probabilities of P_{x-in} and P_{x-out} in Section 8.2.2.

8.2.1 Parameters P_{in} and P_{out}

Figure 8.1 shows surface plots of the success rate of the P_{in}/P_{out} parameter landscapes for a group of (a) 10 robots, (b) 20 robots, (c) 40 robots and (d) 80 robots. All trials were conducted in an environment without obstacles and a nest to prey distance of 3 meters.

Independently from the group size, the most successful parameter combinations are in the proximity of the line $P_{in} = P_{out}$. If $P_{in} \ll P_{out}$ only few robots get aggregated into the forcefield at all, and if $P_{in} \gg P_{out}$ a structure is formed quickly, but the exploration of the environment is limited because the robots leave the vectorfield very rarely and then join it fast at a nearby position.

The performance increases with the group size. While for a group size of $N = 10$ the success rate is very low throughout all tested parameters, for $N = 80$ there is a wide range of parameters with a success rate of 100%.

Table 8.1 shows the selected parameter combination and success rate as found through the racing algorithm. The success rates reached with the chain controller

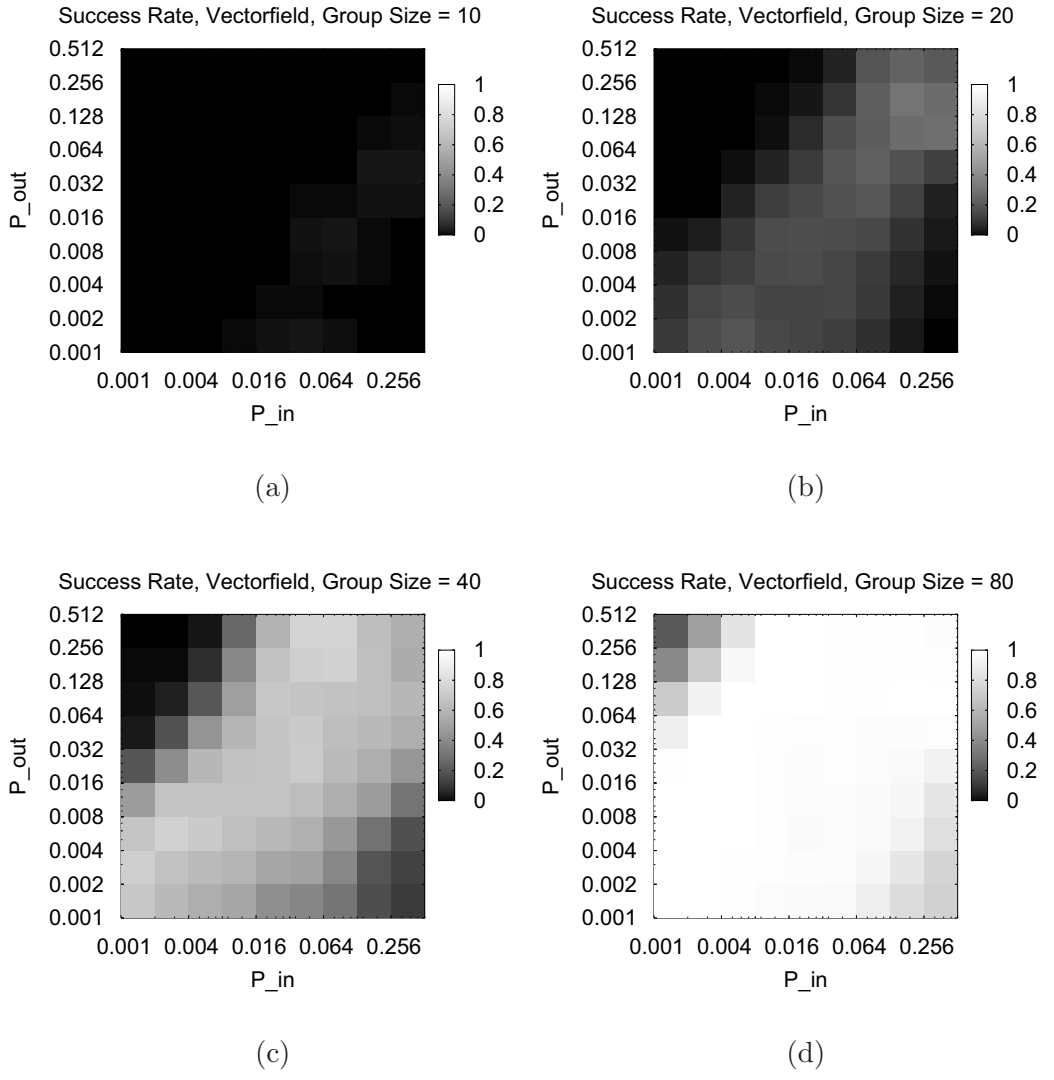


Figure 8.1: Surface plots of the success rates of the parameter landscapes when changing the two probability parameters P_{in} and P_{out} (100 observations per parameter combination), for the vectorfield controller and a group of (a) 10 robots, (b) 20 robots, (c) 40 robots and (d) 80 robots. All experiments were conducted in an environment without obstacles and a nest to prey distance of 3 meters. The axes of the parameters are plotted in logarithmic scale. The lighter the surface the higher is the success rate.

Table 8.1: The selected parameter sets for P_{in} and P_{out} based on the outcome of a racing algorithm on 27 experimental setups obtained considering all the possible combinations of values for N, D, and O, with $N \in \{10, 20, 40\}$, $D \in \{2, 2.5, 3\}$, and $O \in \{0, 10, 20\}$. Each setup was initialized in 100 different ways.

Controller	P_{in}	P_{out}	Success Rate	Median Completion Time
Vectorfield	0.064	0.016	48.1 %	10000 seconds

Table 8.2: The selected parameter set for P_{x-in} and P_{x-out} based on the outcome of a racing algorithm on 27 experimental setups obtained considering all the possible combinations of values for N, D, and O, with $N \in \{10, 20, 40\}$, $D \in \{2, 2.5, 3\}$, and $O \in \{0, 10, 20\}$. Each setup was initialized in 100 different ways. The parameters P_{in} and P_{out} are fixed according to Table 8.1.

Controller	P_{x-in}	P_{x-out}	Success Rate	Median Completion Time
Vectorfield	0.016	0.002	65.3 %	3102 seconds

is higher than for the vectorfield controller. This is mainly due to the problem mix which includes group sizes of 10, 20, or 40 robots. As will be explained in more detail later, the vectorfield reaches a high degree of efficiency only for larger group sizes.

8.2.2 Parameters P_{x-in} and P_{x-out}

Given the parameter selection for P_{in} and P_{out} we performed a similar study for the parameters P_{x-in} and P_{x-out}. While P_{in} and P_{out} represent the rate at which robots join and leave a chain, P_{x-in} and P_{x-out} determine the rate at which robots join and leave the prey extending structure.

Figure 8.2 shows surface plots of the success rate of the P_{x-in}/P_{x-out} parameter landscapes for a group of (a) 10 robots, (b) 20 robots, (c) 40 robots and (d) 80 robots. All trials were conducted in an environment without obstacles and a nest to prey distance of 3 meters.

The parameter landscapes are qualitatively similar to those observed with the chain controller. In particular, briefly repeating our findings from Section 5.2.2, what can be observed is that the most successful parameter combination of P_{x-in} and P_{x-out} have (i) a ratio of $\frac{P_{x-in}}{P_{x-out}}$ roughly in the range $[2, 16]$, (ii) values of P_{x-in} in the range $[0.016, 0.128]$, and (iii) values of $P_{x-out} \leq 0.032$.

The parameters found to be successful when used with the chain controller prove efficient with the vectorfield controller as well. The selected parameter set as found from the racing algorithm is specified in Table 8.2.

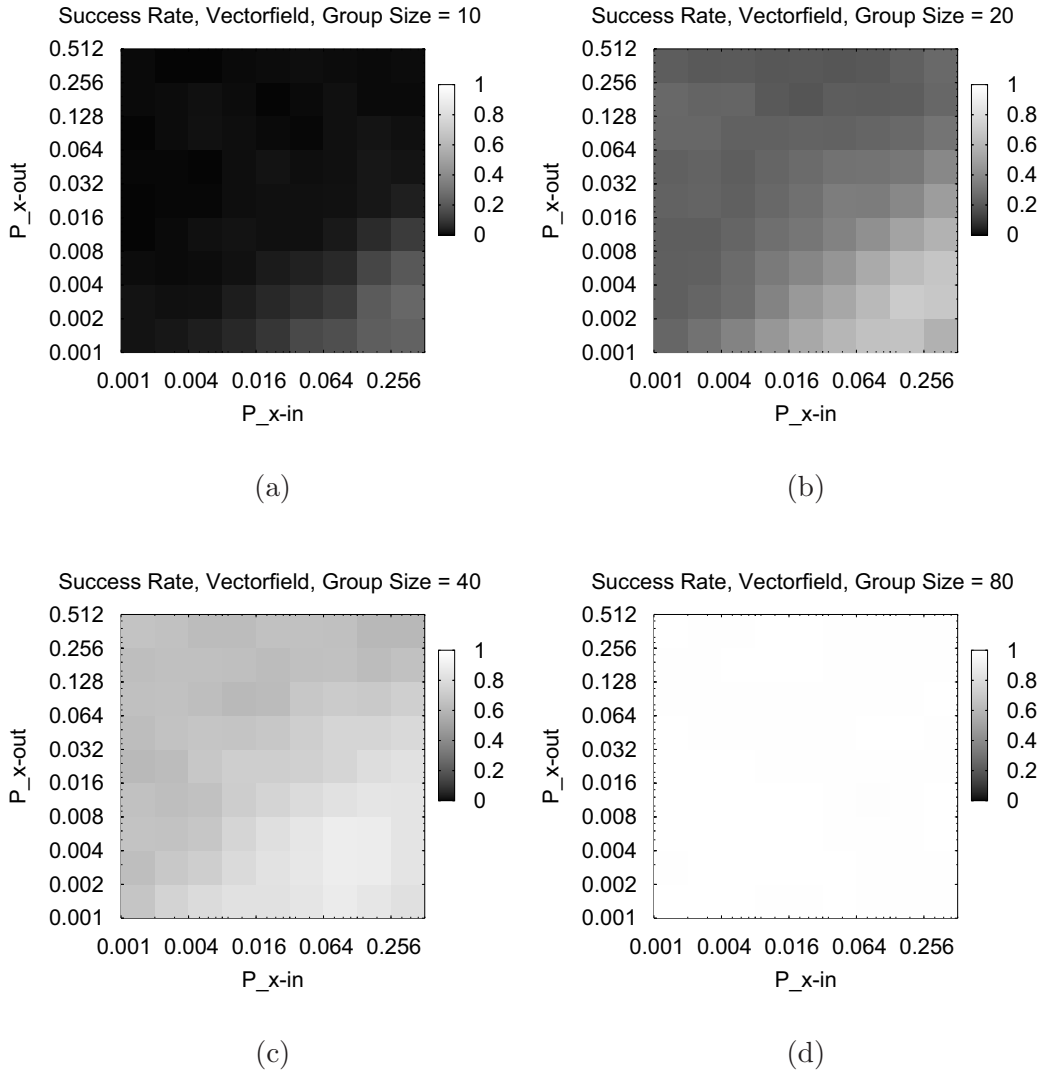


Figure 8.2: Surface plots of the success rates of the parameter landscapes when changing the two probability parameters P_{x-in} and P_{x-out} (100 observations per parameter combination), for the vectorfield controller and a group of (a) 10 robots, (b) 20 robots, (c) 40 robots and (d) 80 robots. All experiments were conducted in an environment without obstacles and a nest to prey distance of 3 meters. The axes of the parameters are plotted in logarithmic scale. The lighter the surface the higher is the success rate. The parameters P_{in} and P_{out} are fixed according to Table 8.1.

8.3 Performance Evaluation

In this section we report on the experiments performed with the vectorfield controller based on the parameters selected in the previous section. Following the same structure as in Section 5.3, we will first give an overview of the behaviour and discuss some general characteristics in Section 8.3.1, and then show the results of systematic experiments under various conditions:

- In a difficulty test we vary the distance D between the nest and the prey in the range $[1\,\mathrm{m}, 3\,\mathrm{m}]$ (Section 8.3.2).
- In a scalability test we vary the number of robots N in the range $[10, 200]$ (Section 8.3.3).
- In an obstacle test we vary the number of obstacles O in the range $[0, 30]$ and additionally test two predefined obstacle environments (Section 8.3.4).
- In a set of robustness tests we vary the noise of various sensors (Section 8.3.5).
- In a set of fault tolerance tests we vary the fraction of robots that suffer from individual failure by disabling various sensors or actuators (Section 8.3.6).

Afterwards, in Section 8.4 we draw some conclusions.

8.3.1 General Evaluation

In this section we discuss the general performance of the vectorfield controller and the prey extension mechanism. We describe first the general behaviour in Section 8.3.1.1. Then we discuss the branching behaviour of the vectorfield in Section 8.3.1.2, and finally the environment exploration in Section 8.3.1.3.

8.3.1.1 Behaviour

A sequence of snapshots from typical simulation trials of the vectorfield controller is displayed in Figure 8.3 when the prey extension mechanism is not employed, and in Figure 8.4 when it is employed.[1]

In both trials we use a group size of 20 robots, a prey to nest distance of 2 m, and an environment without obstacles. Let us start describing the behaviour by summarizing the differences between chains and vectorfield. There are three main differences between the chains and the vectorfield.

First, the nature of the signal in the structure is different. In the case of chains a direction can only be deduced when seeing at least two members of the structure,

[1] A selection of movies can be found at `http://iridia.ulb.ac.be/supp/IridiaSupp2008-014`.

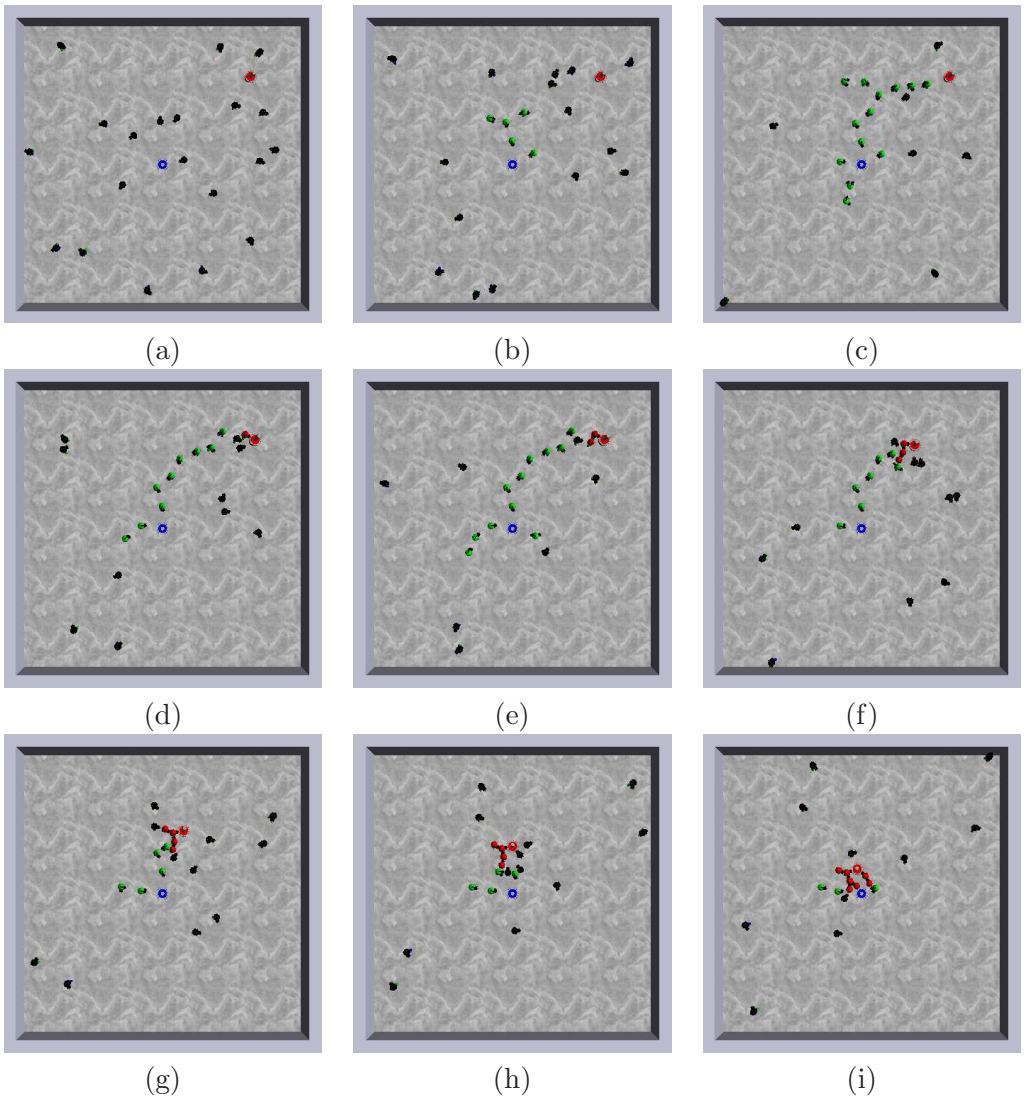


Figure 8.3: Sequence of images taken for a simulation trial with group size $N = 20$ *s-bots* and distance $D = 2\,\mathrm{m}$ between the nest (blue cylindrical object in the centre) and the prey (red cylindrical object on the top right), when using the vectorfield controller: (a) $t = 0\,\mathrm{s}$, (b) $t = 57\,\mathrm{s}$, (c) $t = 185\,\mathrm{s}$, (d) $t = 323\,\mathrm{s}$, (e) $t = 372\,\mathrm{s}$, (f) $t = 552\,\mathrm{s}$, (g) $t = 593\,\mathrm{s}$, (h) $t = 620\,\mathrm{s}$, and (i) $t = 644\,\mathrm{s}$.

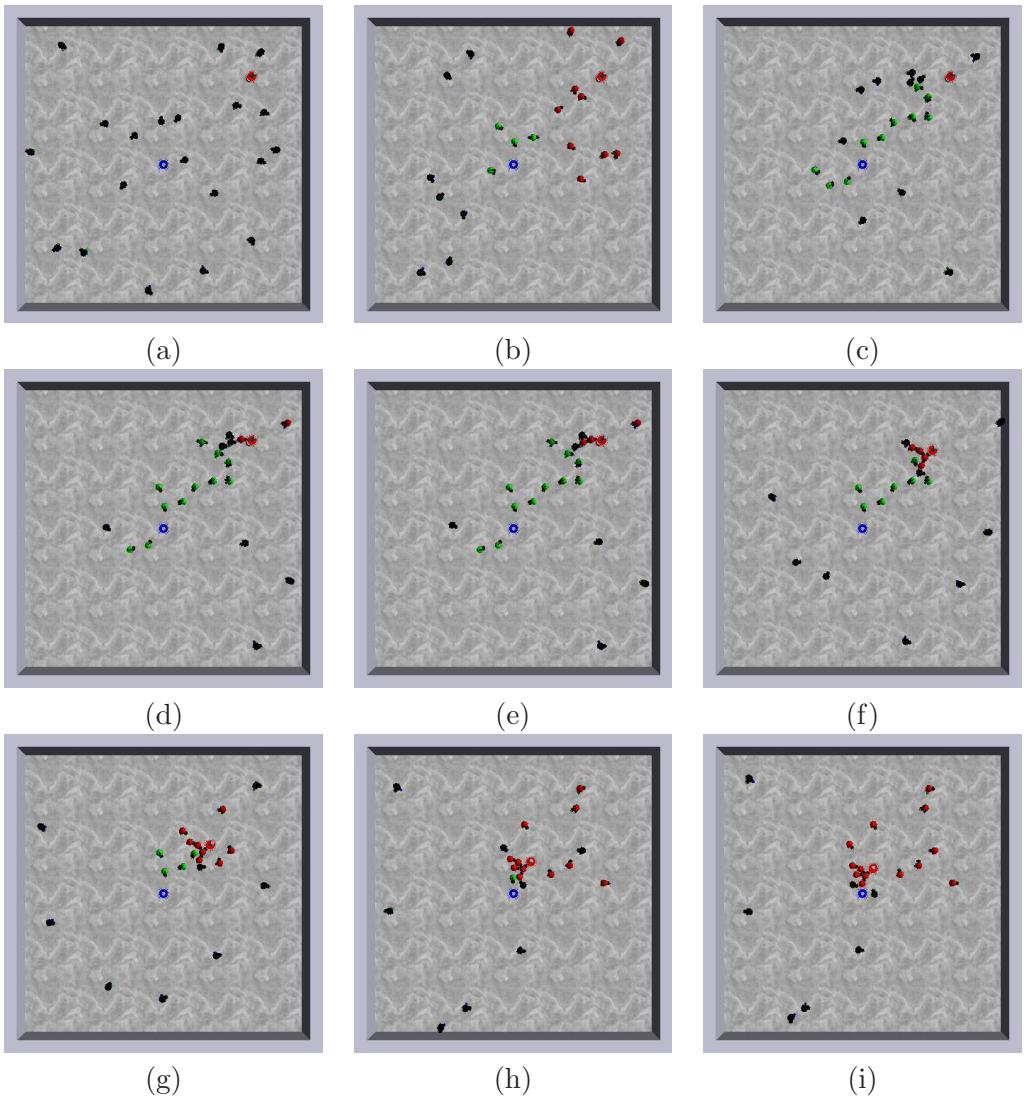


Figure 8.4: Sequence of images taken for a simulation trial with group size $N = 20$ *s-bots* and distance $D = 2\,\text{m}$ between the nest (blue cylindrical object in the centre) and the prey (red cylindrical object on the top right), when using the vectorfield controller with the prey extension mechanism: (a) $t = 0\,\text{s}$, (b) $t = 68\,\text{s}$, (c) $t = 248\,\text{s}$, (d) $t = 318\,\text{s}$, (e) $t = 331\,\text{s}$, (f) $t = 367\,\text{s}$, (g) $t = 407\,\text{s}$, (h) $t = 439\,\text{s}$, and (i) $t = 454\,\text{s}$.

whereas in a vectorfield each member explicitly broadcasts a direction. Explorers moving along a chain follow the direction given by the order of two members of the chain. Explorers moving along the vectorfield follow the direction indicated by the vectorfield members perceived.

This leads us to the second difference, which consists in the process of joining the path forming structure. For the chains this process is probabilistic, and an explorer usually joins a chain at its tail. For the vectorfield however, it is deterministic as robots immediately join the vectorfield when they reach its border. This in general leads to a higher degree of branching of the vectorfield structure.

Finally, the rule employed for leaving the vectorfield leads to a higher degree of randomness because the robot performs a random walk and might lose sight of the structure, having to start the search from scratch. When leaving a chain, a robot tries to stay in the vicinity of the chain while moving back to the nest to then follow another chain. Because of this lower degree of randomness, we expect the chains to perform better when there is a low density of robots.

8.3.1.2 Branching

Branching of the vectorfield was already briefly mentioned in the previous section. The differences between chain and vectorfield lead to a higher degree of branching as is also shown in Figure 8.5.

The number of branches directly connected to the nest is roughly constant throughout all group sizes, and in general higher than for chains (see Figure 5.13). This is the case because a robot searching for the nest or the vectorfield immediately starts a branch when it perceives the nest. When employing the chain controller a robot that finds the nest first explores its vicinity to see if there are any other chains. Therefore, there are more branches directly at the nest.

For small group sizes $N \leq 20$ there are hardly any sub-branches, that is, splits within one branch are very rare. For larger group sizes however, sub-branching becomes more pronounced.

8.3.1.3 Exploration

We performed the same exploration test as for the chain controller. Initially, the robots are randomly positioned and the environment is completely unexplored. A position in the environment is defined to be explored if it is within a radius of 40 cm of an explorer or vectorfield robot. Figure 8.6 shows the percentage of the explored arena for the vectorfield controller.

Compared to the chain controller (see Figures 5.14, 5.15 and 5.16) the exploration rate is low especially for small group sizes. Indeed, if the density of robots is low there is continuously a high fraction of the robots in the search state performing a

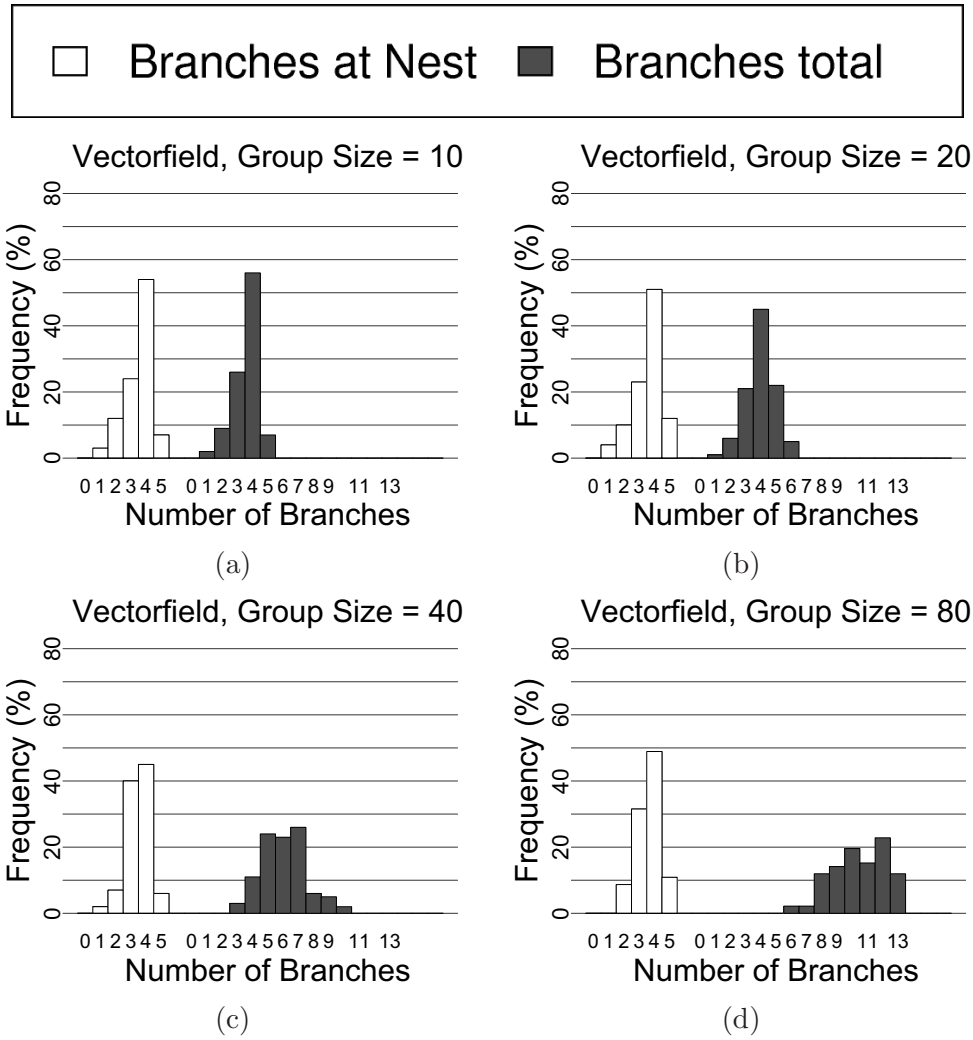


Figure 8.5: The histograms show the frequency of the number of branches formed for the vectorfield controller and a group size of (a) 10, (b) 20, (c) 40 and (d) 80 robots in 100 evaluations. The results are collected at $t = 3600\,\text{sec}$ in an obstacle free environment with no prey. The white bars on the left of each histogram indicate the number of branches formed directly from the nest. The grey bars on the right also include splits in branches that lead to different sub-branches.

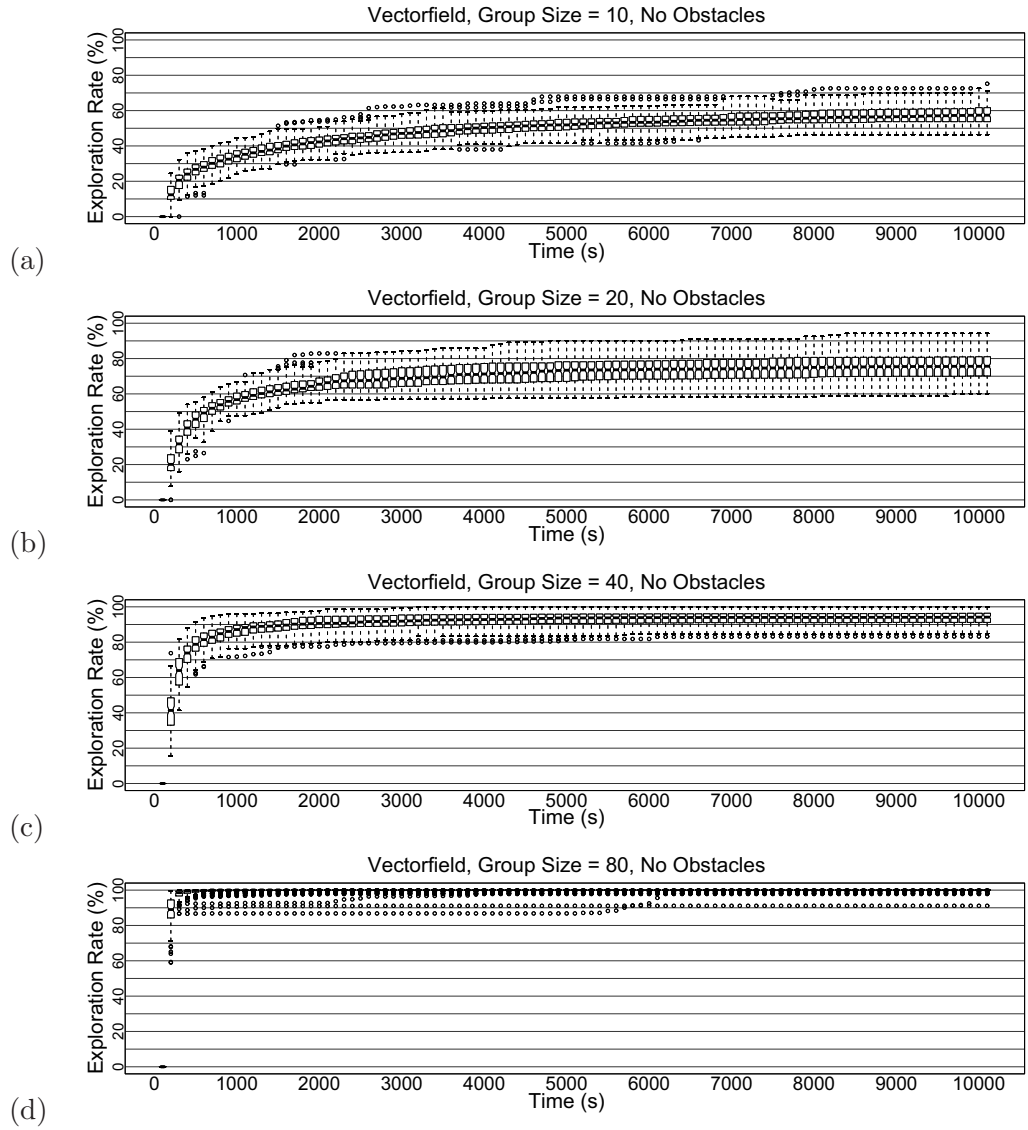


Figure 8.6: The box-and-whisker-plots (Becker et al., 1988) show 100 evaluations of the exploration rate for a group of (a) 10 robots, (b) 20 robots, (c) 40 robots and (d) 80 robots using the vectorfield controller in an obstacle free environment. Given that no prey is present in the arena the results do not rely on the prey extension mechanism. Boxes represent the inter-quartile range of the data, while the horizontal bars inside the boxes mark the median values. The whiskers extend to the most extreme data points within 1.5 of the inter-quartile range from the box. The empty circles mark the outliers.

random walk. This is due to the fact that when a robot leaves the vectorfield it performs a random walk. When the density of robots is low, the search can take a considerable amount of time. However, when there is a high density of robots, the random walk will possibly lead to another branch of the vectorfield quickly. And as can be seen in Figure 8.6c and d, for larger group sizes the performance of the vectorfield is comparable to that of chains, and the initial exploration is even faster.

8.3.2 Difficulty Test

In the difficulty we test the ability of the vectorfield controller to cope with changes in the difficulty of the task, that is, when changing the nest to prey distance D. For a group size of $N = 20$ Figure 8.7 shows the completion time, and Figure 8.8 shows the normalized completion time, that is, the completion time divided by the prey distance ($\frac{T}{D}$), for the subtasks (a) path formation, (b) assembly, and (c) transport. Table 8.3 displays the success rates.

As already expected, the results reveal that for this group size the vectorfield controller performs rather poorly when compared to the chains. Without prey extension mechanism the success rate to form a path drops below 50% for distances $D > 2\,\mathrm{m}$, while it stays constantly high for the chain controller. The use of the prey extension mechanism improves the performance, but nevertheless it remains below the one of the chains. The reason for this is mainly that due to the higher degree of randomness the vectorfield controller requires a higher density of robots to work

Table 8.3: Success rates for the difficulty test of the vectorfield controller with and without prey extension mechanism, compared to the aligning chain controller for a group size of $N = 20$. The three values represent percentages of the success rate from 100 trials for path formation, assembly and transport in this order.

D	Vectorfield	Aligning	Vectorfield-X	Aligning-X
1.0	100/100/99	100/100/100	100/98/97	100/100/100
1.2	100/100/100	100/100/100	100/100/100	100/100/100
1.4	97/92/91	100/100/100	100/97/97	100/100/100
1.6	91/82/80	100/100/98	95/89/88	100/100/100
1.8	85/73/73	100/100/100	94/85/85	100/100/99
2.0	60/50/48	100/100/99	86/82/79	100/100/99
2.2	49/42/42	99/99/98	75/62/61	100/100/98
2.4	57/44/43	100/100/100	64/57/53	100/100/98
2.6	37/27/27	100/100/95	53/44/42	100/100/95
2.8	38/26/26	100/98/90	52/42/40	100/100/96
3.0	17/15/15	100/100/95	41/31/31	100/100/94

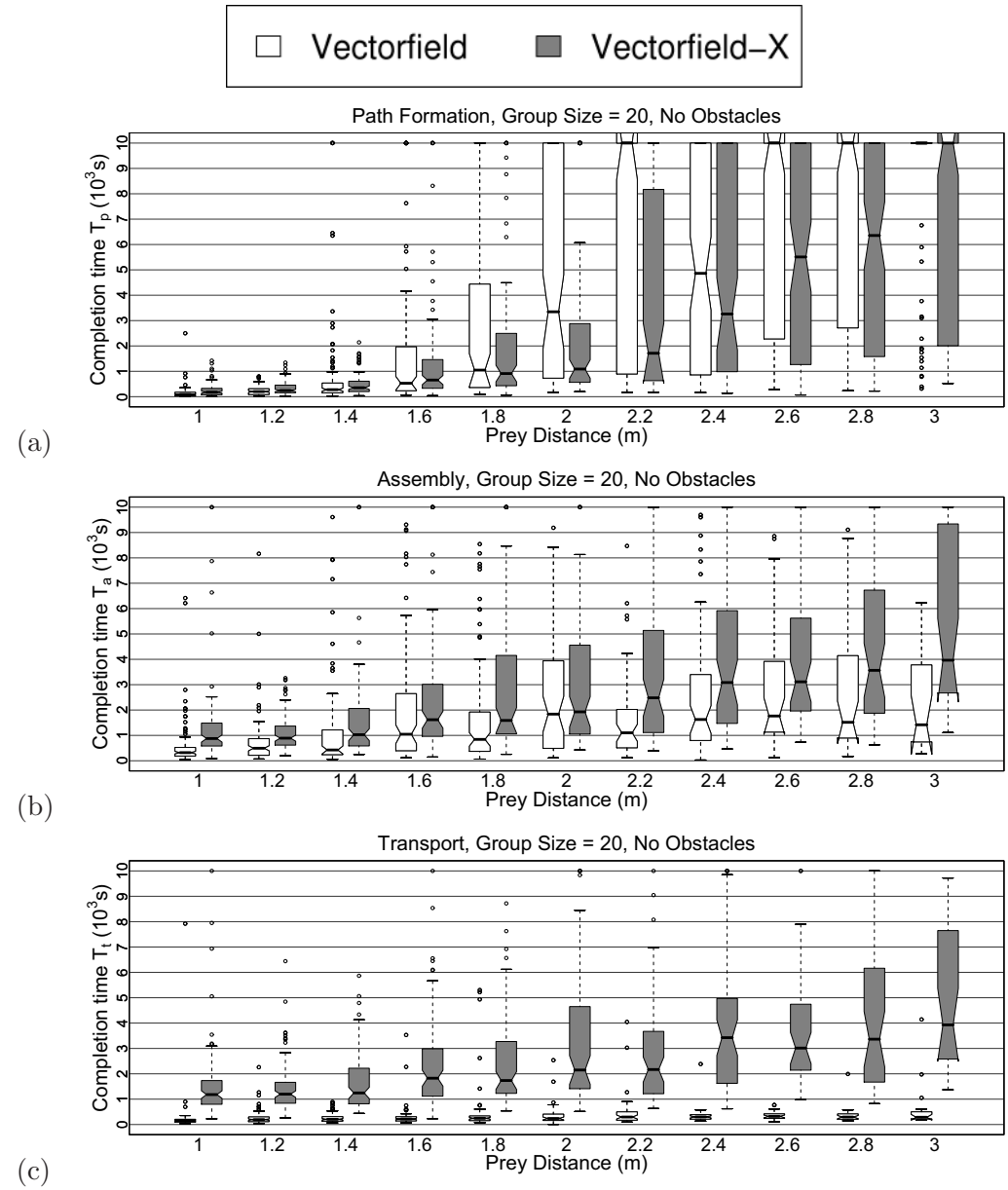


Figure 8.7: The box-and-whisker-plots (Becker et al., 1988) show 100 evaluations per box of the completion time when changing the nest to prey distance, for a group of (a) 10 robots, (b) 20 robots, (c) 40 robots and (d) 80 robots using the vectorfield controller with and without prey extension mechanism in an environment without obstacles. See the caption of Figure 8.6 for an explanation of the box-and-whisker plots.

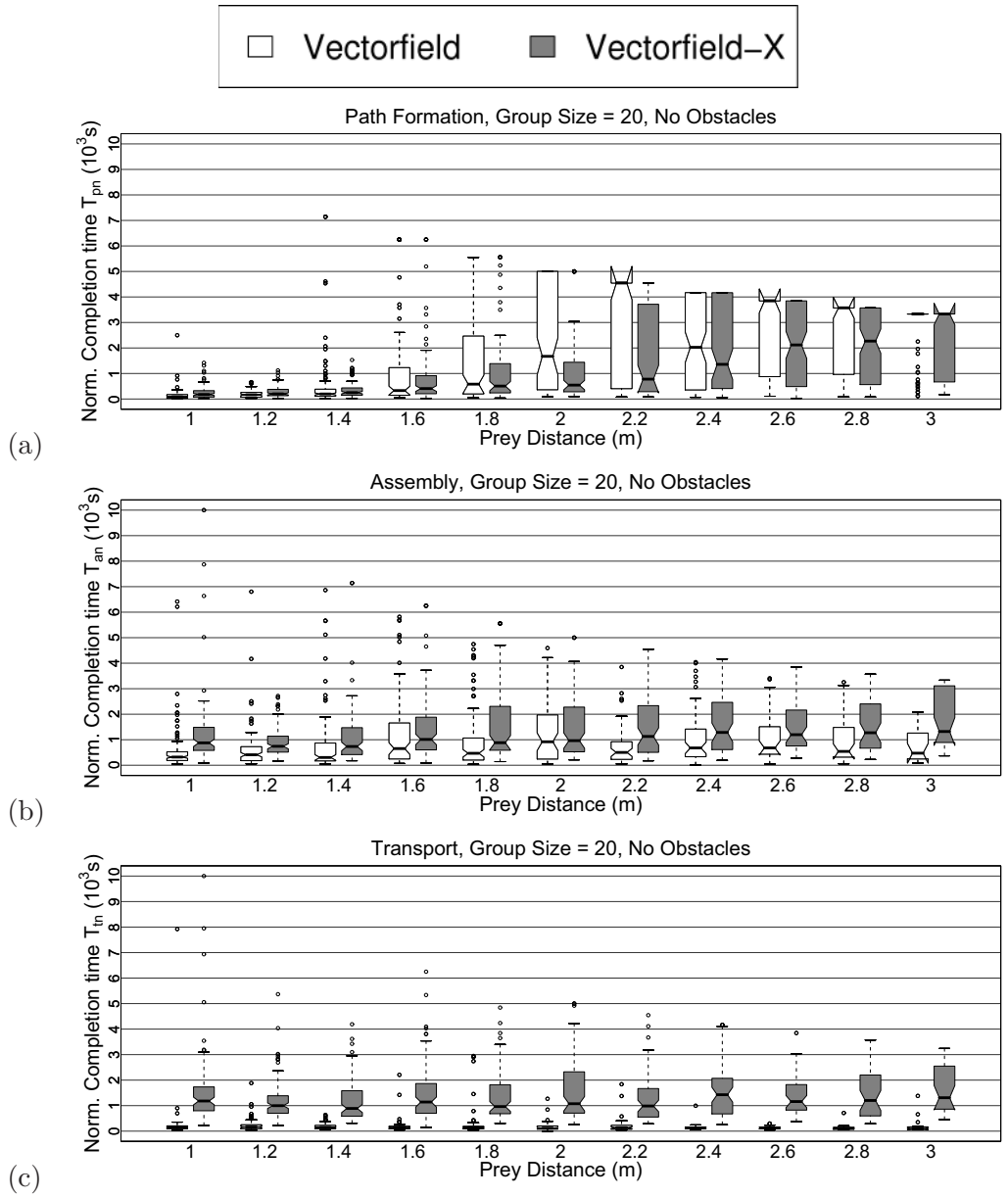


Figure 8.8: The box-and-whisker-plots (Becker et al., 1988) show the results taken from 100 trials of the normalized completion time (i.e. the completion time divided by the distance $\frac{T}{D}$) for subtasks (a) path formation, (b) assembly, and (c) transport when changing the nest to prey distance for a group 20 robots using the different chain strategies with and without prey extension mechanism in an environment without obstacles. Note that subtask assembly is only taken into account in case a path has been formed, and respectively subtask transport is only taken into account in case the assembly was successful. See the caption of Figure 8.6 for an explanation of the box-and-whisker plots.

efficiently. This will be investigated in more detail in the following section where scalability is tested.

Also for subtask assembly the performance of the vectorfield controller remains below the one of chains. The reason is that compared the chains, the recruitment process for the vectorfield controller leads the explorers less directly to the prey once a path is formed. A robot following a chain usually reaches its tail, and in case the chain connects nest and prey, the prey. Due to the higher degree of branching, when following the vectorfield structure, a robot may follow a different branch or start a new one itself, even if the prey has been found already.

For subtask transport the performance of the vectorfield controller is better. Even though the overall success rate usually remains below the one of the chain controller, the relative success rate, that is, the fraction of trials with a successful transport given a successful assembly, is higher. Furthermore, the transport is faster than is the case for chains. This has mainly two causes, both of which are related to the less explicit recruitment of robots to the prey. First, the size of the assembled structure of robots transporting the prey is smaller. This is beneficial for the transport because in a large pulling structure there are usually some robots that do not perceive the goal direction and therefore disturb the transport. Second, once that two robots are connected to the prey, they can transport it with less robots arriving and trying to assemble to the prey which in effect blocks the transport. For the chains, this leads to a higher success rate for subtask assembly, but it makes the transport more difficult and in general less successful and slower.

For subtasks assembly and transport the performance decreases when using the prey extension mechanism. This has the same reason as for the chains. The prey extending structure remains after a path has been formed and in this way distracts the vectorfield. Additionally, given that there is a higher degree of branching for the vectorfield, using the prey extending structure increases the probability to have multiple branches that form a path between nest and prey. This distracts the transport as the robots in the prey pulling structure have multiple goal directions towards which they try to transport the prey.

8.3.3 Scalability Test

In a scalability test we investigate the performance of the vectorfield controller when changing the group size N. We use groups of up to $N = 200$ robots, keep the nest to prey distance constant at a distance of $D = 3\,\mathrm{m}$, and use an obstacle free environment. A summary of the results is given in Table 8.4 for the achieved success rates and the median completion times for path formation, in Figure 5.19 for the completion time of subtasks (a) path formation, (b) assembly, and (c) transport, and in Figure 5.20 for the overall effort, that is, the product of completion time and robot group size. This measure is a good indicator of scalability and can be used to

investigate super-linearity in the system. In the following we give a summary of the results for each subtask:

- **Path formation:** The performance of the vectorfield is comparably bad for small group sizes. While the aligning chain strategy achieves success rates of close to 100% already for a group size of $N = 10$, the vectorfield achieves a success rate of more than 50% only for group sizes $N > 30$. However, the bigger the group size the smaller is the gap between vectorfield and aligning chains, and from a group size of $N = 60$ the vectorfield outperforms the aligning chains. When used without prey extension mechanism, the overall effort of the vectorfield controller decreases up to a group size of roughly 100 robots, and then stays roughly constant. The same could be observed for the chain strategies, but the value at which the overall effort of the vectorfield controller stabilizes is lower.

  While the higher degree of randomness in the vectorfield controller appears to be a disadvantage for small densities of robots, it apparently turns into an advantage for larger groups. In fact, it allows the robots to move more freely after they have left the vectorfield, in this way allowing for a more homogeneous dispersion of the robots in the environment. For increasing group sizes chains tend to become overcrowded with robots moving along them. This increases the amount of physical interactions and makes it difficult for the robots to move efficiently. Furthermore, the process of joining the vectorfield structure is simpler than joining a chain, which allows for a faster formation of a structure and a path.

  The prey extension mechanism has a beneficial effect for small group sizes, but leads to a decrease in performance for larger group sizes. This can be observed for both vectorfield and chains. For small group sizes it speeds up the path formation process as it leads to the formation of a path from both nest and prey, and in this way makes a more efficient use of the resources. For large group sizes it can lead to confusion in the path formation as prey extending structure is formed very quickly and the environment is filled with robots that activate their LEDs in the colour of the prey. The decrease in performance for large group sizes is more pronounced for the chain controller than for the vectorfield controller. The reason for this is that the vectorfield controller is formed more quickly, and therefore leaves less time to the prey extension structure to be built up.

- **Assembly:** As could already be observed in the difficulty test of the previous section, the success rate of assembly for the vectorfield controller is below the one of the chain controller due to a less explicit recruitment mechanism. For larger group sizes the gap between the two controllers becomes smaller, and

Table 8.4: Success rates for the scalability test of the vectorfield controller compared to the aligning chain controller with and without prey extension mechanism for selected group sizes N. The three values in the first row of each box represent percentages of the success rate from 100 trials for path formation, assembly and transport in this order, and the value in the second row is the median completion time for path formation T_p.

N	Vectorfield	Aligning	Vectorfield-X	Aligning-X
10	0/0/0 T_p =10000	89/34/9 T_p =3531	1/0/0 T_p =10000	96/50/19 T_p =1520
12	4/0/0 T_p =10000	93/93/75 T_p =2057	5/2/2 T_p =10000	97/96/80 T_p =1405.5
14	6/3/3 T_p =10000	93/91/81 T_p =1820.5	15/10/9 T_p =10000	95/95/84 T_p =866.5
16	16/11/11 T_p =10000	100/99/93 T_p =1619	22/14/12 T_p =10000	100/99/94 T_p =970
18	18/13/12 T_p =10000	100/100/91 T_p =946	35/24/22 T_p =10000	100/100/92 T_p =847
20	17/15/15 T_p =10000	100/100/95 T_p =1540.5	41/31/31 T_p =10000	100/100/94 T_p =768
25	36/22/22 T_p =10000	100/99/93 T_p =1069	54/41/38 T_p =5548	100/100/93 T_p =576
30	50/37/36 T_p =9919.5	100/100/94 T_p =757.5	71/63/62 T_p =1461.5	100/100/99 T_p =609
40	66/58/57 T_p =1301.5	100/100/95 T_p =479	81/63/62 T_p =770.5	100/100/100 T_p =613.5
50	84/69/68 T_p =434	100/100/94 T_p =366.5	89/71/68 T_p =459.5	100/100/97 T_p =737
60	100/84/82 T_p =187.5	100/100/100 T_p =212	99/86/83 T_p =296	100/100/100 T_p =772
80	100/93/89 T_p =92	100/100/97 T_p =167	100/98/80 T_p =224	100/100/99 T_p =969.5
100	100/100/95 T_p =58	100/100/100 T_p =128	100/100/77 T_p =161.5	100/100/95 T_p =987.5
140	100/100/94 T_p =39.5	100/100/99 T_p =110	100/100/74 T_p =142.5	100/100/96 T_p =680
200	100/100/100 T_p =28	100/100/100 T_p =72.5	100/100/92 T_p =135.5	100/100/98 T_p =332.5

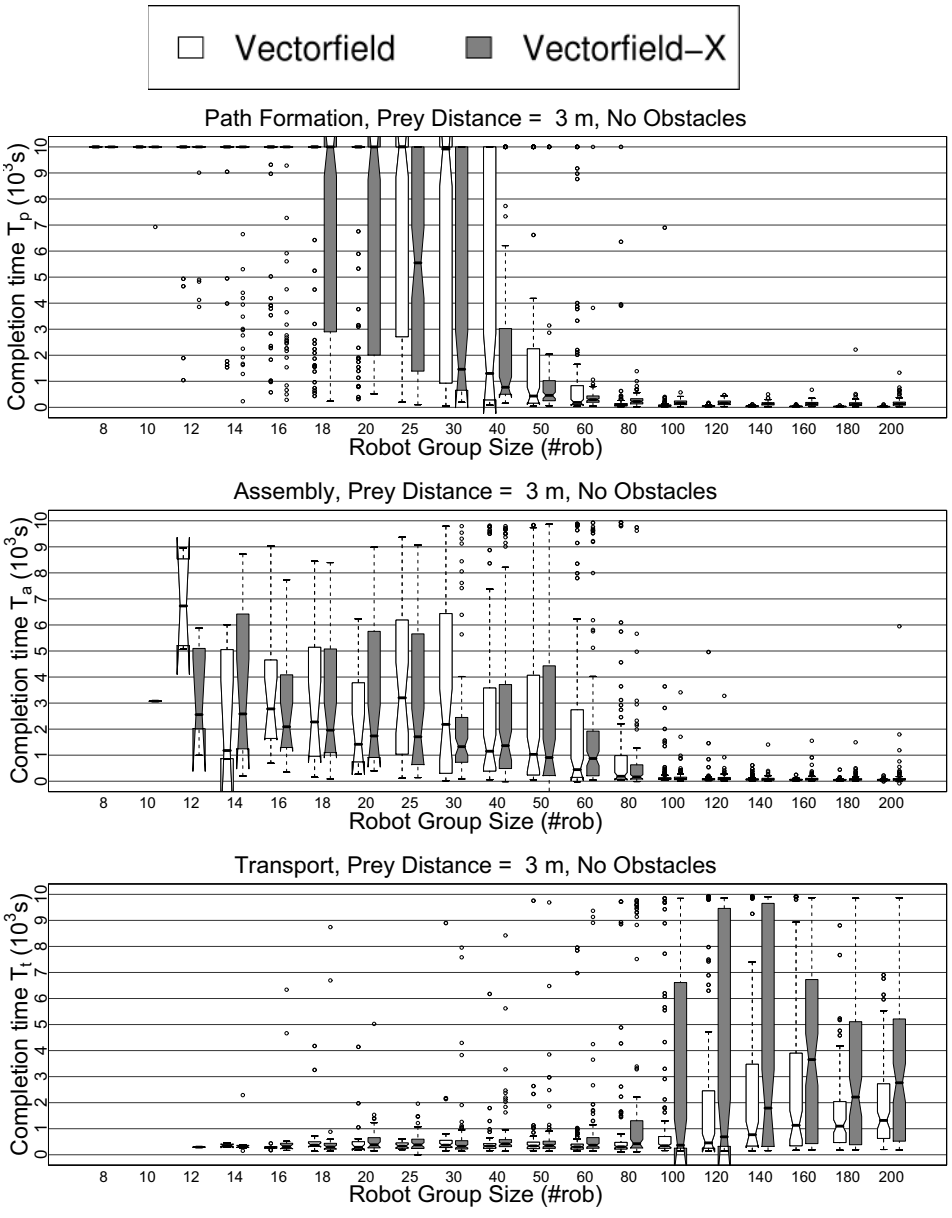


Figure 8.9: The box-and-whisker-plots (Becker et al., 1988) show the results taken from 100 trials of the completion times for subtasks (a) path formation, (b) assembly, and (c) transport when changing the robot group size using vectorfield controller with and without prey extension mechanism in an environment without obstacles. Note that the subtask assembly is only taken into account in case a path has been formed, and respectively the subtask transport is only taken into account in case the assembly was successful. See the caption of Figure 8.6 for an explanation of the box-and-whisker plots.

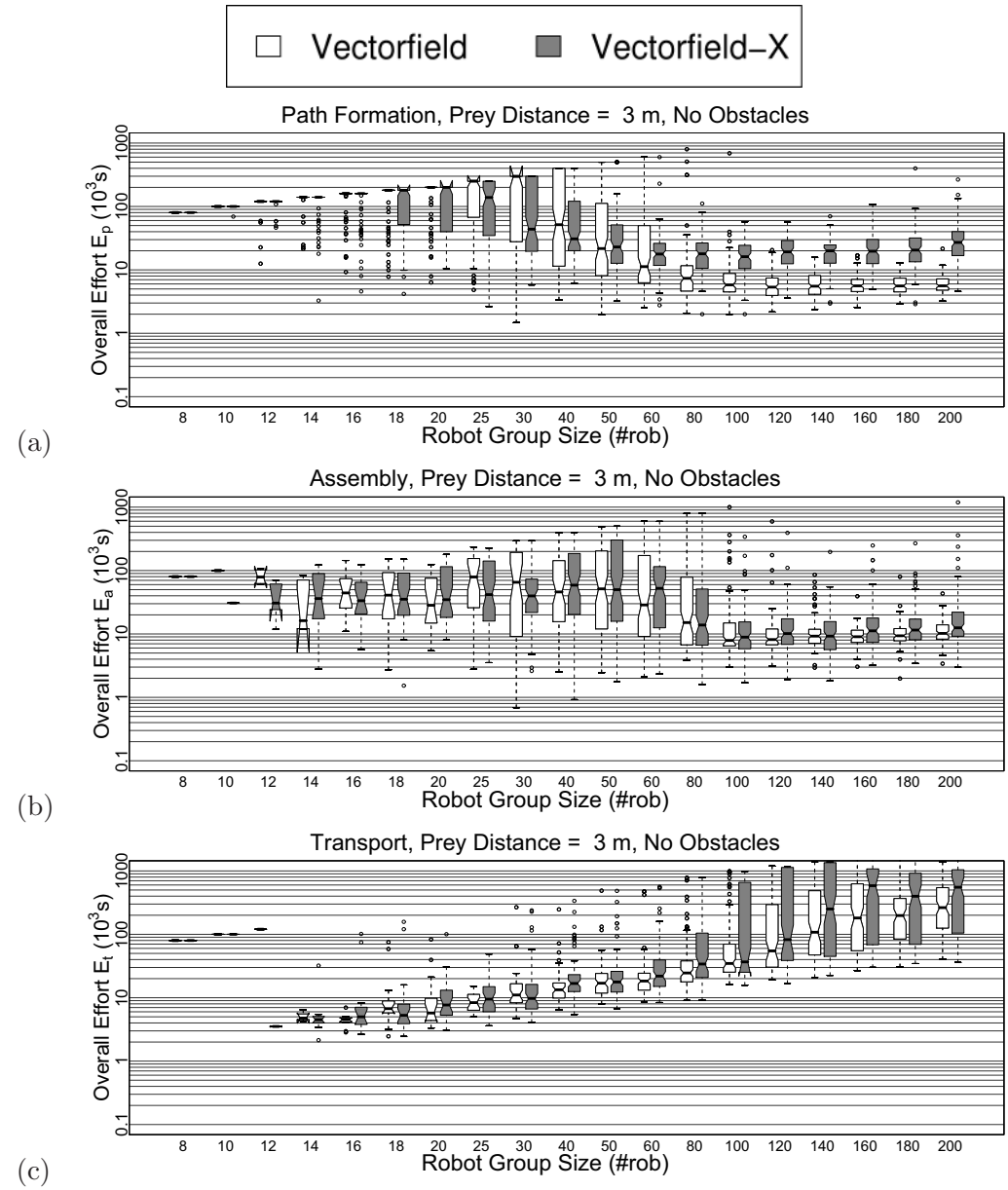


Figure 8.10: The box-and-whisker-plots (Becker et al., 1988) show the results taken from 100 trials of the overall efforts for subtasks (a) path formation, (b) assembly, and (c) transport when changing the robot group size using the vectorfield controller with and without prey extension mechanism in an environment without obstacles. Note that the subtask assembly is only taken into account in case a path has been formed, and respectively the subtask transport is only taken into account in case the assembly was successful. See the caption of Figure 8.6 for an explanation of the box-and-whisker plots.

from a group size of $N = 100$ disappears for both success rate and completion time.

- **Transport:** In general, the completion time remains roughly constant for $N < 60$ and then increases for larger group sizes as there are more robots trying to assemble to the prey and thereby disturb the transport. The same is true for the chain controller, where the completion time starts to increase from a group size of $N = 20$. Given the less explicit recruitment mechanism of the vectorfield, the completion time is less affected by larger group sizes and remains roughly constant up to a group size of $N = 80$. Up to a group size of $N = 120$ the vectorfield is faster than the chain controllers when the prey extension mechanism is not used. As mentioned in the previous section, when using the prey extension mechanism the performance drops for larger group as there are often multiple paths formed to the prey, and the prey pulling structure of transporter robots has multiple goal directions.

8.3.4 Obstacle Test

To assess the capability of the vectorfield controller to cope with obstacles, we tested its performance in three types of obstacle environments, as shown in Figure 5.21: (a) the R-arena with a random configuration of obstacle cubes, (b) the X-arena with four corridors, one of which leads to the prey, and (c) the U-arena, where the prey is positioned behind a U-shaped obstacle. The prey distance is 3 m for all cases except for the U-arena, where it is placed behind a long corridor at a distance of 2.12 m.

Table 8.5 shows the success rates and median completion times for subtask path formation for six selected obstacle environments. Figures 8.11 and 8.12 show the results of the individual subtasks for the obstacle test for group sizes of (a) 40 and (b) 80 robots. For group sizes of $N < 40$ the performance of the vectorfield controller is poor already for an environment with no obstacles. Therefore, we chose not to display the results for $N < 40$. In the following we give a summary of the results for each subtask:

- **Path formation:** As was the case for chains, for vectorfield there is a performance drop when there are obstacles in the environment. In general, as already observed in the previous tests, the vectorfield controller is outperformed by the aligning chain controller for $N = 40$, and outperforms the aligning chain for $N = 80$.

  The X-arena is the only one where the performance of the vectorfield controller is comparable to the one of the aligning chains for $N = 40$. The reason for this lies in the higher degree of branching observed for the vectorfield. In the X-arena, there are four corridors, and respectively four paths that can

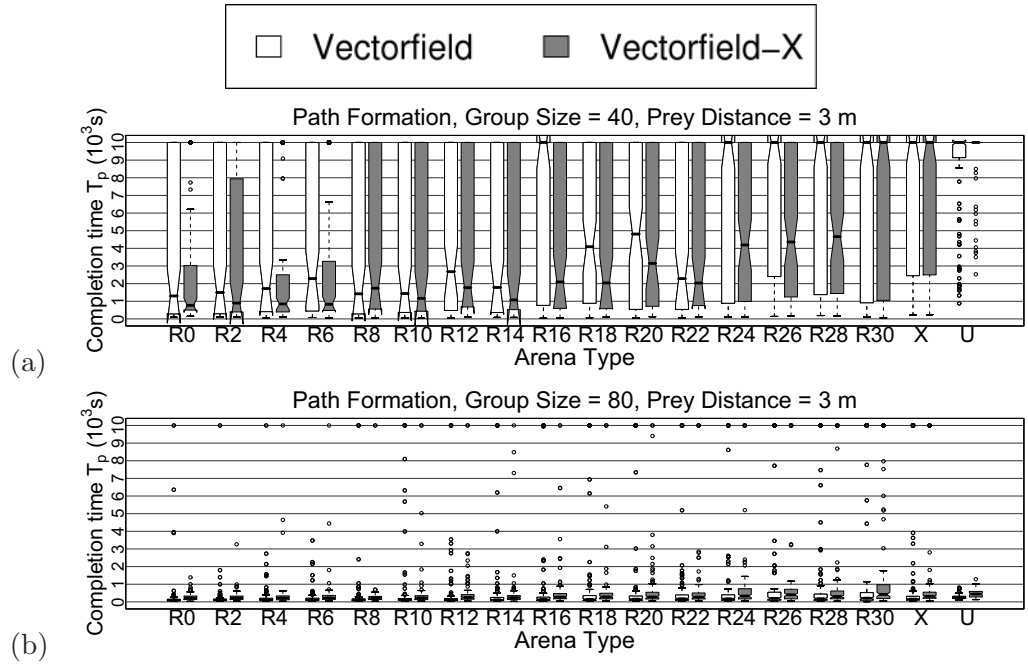


Figure 8.11: The box-and-whisker-plots show 100 evaluations per box of the completion time for subtask path formation when changing the number and configuration of obstacles in the environment for a nest to prey distance of 3 meters and a group of (a) 40 robots and (b) 80 robots using the vectorfield controller with and without prey extension mechanism. For arenas of type R the number of obstacles is indicated. See the caption of Figure 8.6 for an explanation of the box-and-whisker plots.

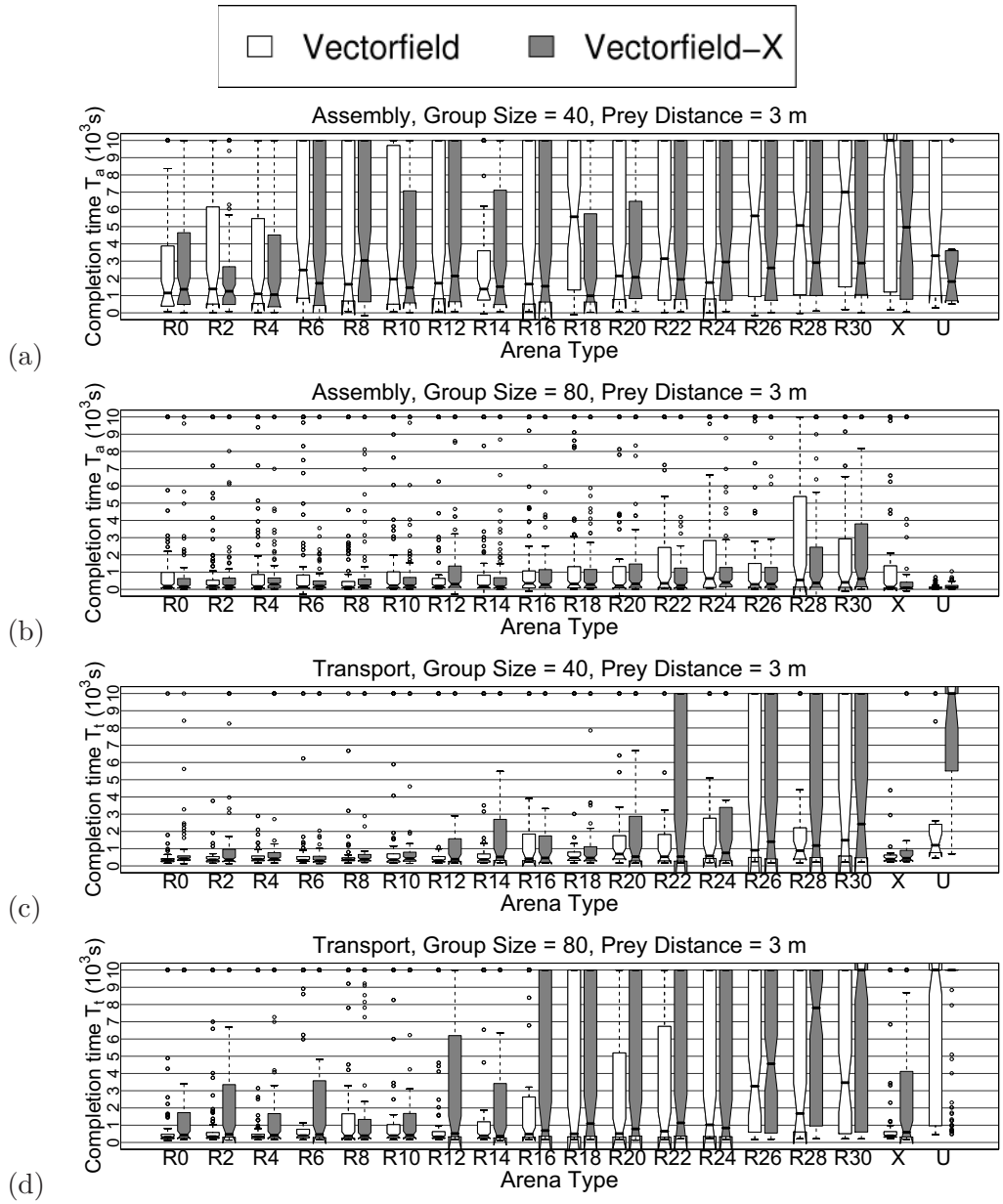


Figure 8.12: The box-and-whisker-plots show the results from 100 trials of the completion time for subtasks assembly and transport when changing the number and configuration of obstacles in the environment for a nest to prey distance of 3 meters and a group of (a) 40 robots and (b) 80 robots using the vectorfield with and without prey extension mechanism. For arenas of type R the number of obstacles is indicated. Note that a trial for assembly (transport) is only taken into account in case subtask path formation (assembly) is successful. See the caption of Figure 8.6 for an explanation of the box-and-whisker plots.

Table 8.5: Success rates for selected setups of the obstacle test of the vectorfield controller compared to the aligning chain controller. The three values in the first row of each box represent percentages of the success rate from 100 trials for path formation, assembly and transport in this order, and the value in the second row is the median completion time for path formation T_p.

Arena	Vectorfield	Aligning	Vectorfield-X	Aligning-X
$N = 40$				
R0	66/58/57 T_p =1301.5	100/100/95 T_p =479	81/63/62 T_p =770.5	100/100/100 T_p =613.5
R10	67/50/46 T_p =1437	98/96/73 T_p =615.5	71/55/50 T_p =1162.5	100/100/91 T_p =761
R20	61/39/36 T_p =4808.5	92/84/57 T_p =766	61/46/37 T_p =3150.5	97/94/71 T_p =947.5
R30	45/25/18 T_p =10000	80/50/23 T_p =1411.5	48/29/18 T_p =10000	85/75/39 T_p =1632
X	38/18/18 T_p =10000	42/35/6 T_p =10000	40/22/19 T_p =10000	40/38/13 T_p =10000
U	27/15/13 T_p =10000	100/100/9 T_p =913	24/12/3 T_p =10000	100/100/39 T_p =1298.5
$N = 80$				
R0	100/93/89 T_p =92	100/100/97 T_p =167	100/98/80 T_p =224	100/100/99 T_p =969.5
R10	97/87/76 T_p =106	100/100/82 T_p =215.5	98/93/75 T_p =225	100/100/81 T_p =711.5
R20	96/84/65 T_p =151.5	100/99/68 T_p =256	98/84/59 T_p =279	100/97/72 T_p =948.5
R30	95/81/49 T_p =209.5	92/81/43 T_p =648	93/75/34 T_p =424.5	99/88/38 T_p =893
X	93/78/68 T_p =157.5	78/69/22 T_p =282.5	94/82/66 T_p =312	85/80/32 T_p =940.5
U	100/100/49 T_p =254	100/100/2 T_p =513.5	100/100/24 T_p =446	100/99/11 T_p =1545.5

be followed. While for $N = 40$ there are usually only two chains formed simultaneously, there are in most cases four branches, and many more sub-branches for the vectorfield. Therefore, there is a higher probability that all corridors are explored by the vectorfield controller, which explains its good performance when compared to the other arenas.

The vectorfield's performance for $N = 40$ is particularly poor for the U-arena.

At least 22 robots are required to form a path. Due to the particular shape of the arena there are only two possible paths that can be followed. Given the very low probability of a homogeneous split of the robots for each path, the aligning chain reaches a success rate of 100% already for group size $N = 40$. However, for the vectorfield controller the success rate is just 27% without, and 24% with prey extension mechanism. The reason for this lower success rate is mainly given by the random, non-aligning structure of the vectorfield. In case a path is formed, it usually consists of 30 or more robots that connect nest and prey. If more than 30 robots are allocated to the same of the two possible paths, they are very likely to form a path connecting nest and prey. For a group size of $N = 40$ there is a low probability for this to happen, which is represented by the low success rate. However, for $N = 80$, there are at least 40 robots following each path. Hence the success rate of 100%.

- **Assembly:** There are two main observations: First, the assembly performance decreases in the presence of obstacles. The degree of performance drop is similar as for chains and is due to obstacles making the access to the prey more difficult. Second, the vectorfield generally performs worse for subtask assembly than for chains. This could already be observed in the previous tests and is due to the less explicit recruitment of robots to the prey once a path is formed.

- **Transport:** As for the previous two subtasks, also the transport performance is in general diminished in the presence of obstacles in the environment. In general, the vectorfield reaches a better performance than the chains. The reasons for this are the same as stated in the previous sections. Compared to the aligning chains, the transport performance of the vectorfield is particularly strong in the X- and U-arenas. These arenas pose difficulties to the transport subtask due to the increased length of the path (4.2 m for the X-arena, and approximately 7.8 m for the U-arena), and due to the very steep turns around corners. The latter difficulty poses a bigger problem to the aligning (and moving) chains, as the aligning mechanism leads the chains to be closer to a corner, which can block the transport. This is not the case for the vectorfield (and the static chains), where the aligning mechanism is not used. Therefore, the distance to corners is usually higher, and the performance of the transport subtask for these two arena types is higher.

8.3.5 Robustness Tests

In order to test the robustness of the vectorfield controller with respect to noisy information, we conducted a series of tests in which we vary the noise of the various sensors. The noise is calculated at each time step as a uniformly random value

within the range $[-noise_{max}; noise_{max}]$, and is added to the considered sensor value. Repeating the same tests performed with the chain controller in Section 5.3.5, we varied the noise of the direction to objects[2] perceived by the camera (Section 8.3.5.1), the distance to objects perceived by the camera (Section 8.3.5.2), and the proximity sensor (Section 8.3.5.3). In addition, we perform one test in which we vary the noise of the direction-to-nest vector indicated by other robots in the vectorfield, and which is perceived by the camera (Section 8.3.5.4).[3] In all robustness tests we employ group sizes of $N \geq 40$, as for smaller group sizes the performance of the vectorfield controller is poor already for the case that no additional noise is injected in the controller.

8.3.5.1 Camera Direction

We test the robustness to the noise of the camera direction by adding maximum noise levels in the range $[0^o, 180^o]$. Table 5.6 shows the achieved success rates for the different strategies and subtasks, and the median completion times for subtask path formation T_p.

Among the robustness tests performed for the chain controller, the camera direction was shown to be the most sensitive sensory information. Similarly, the performance of the vectorfield drops significantly for increasing noise levels as well. In fact, the vectorfield becomes unstable for noise levels above approximately 50^o due to a threshold angle related to the connection between two neighbouring robots. Robots that are aggregated in the vectorfield structure indicate the direction towards their predecessor. If a robot does not perceive its predecessor at a direction that is within a threshold of 50^o with respect to the indicated direction for a given number of consecutive time steps, it leaves the vectorfield because it assumes that the vectorfield structure broke up, in this way triggering all the robots behind to leave the vectorfield as well. Therefore, if the noise of the direction perceived using the camera grows beyond this threshold, robots in the vectorfield falsely assume that the structure broke up and leave, which leads to the instability of the structure. Surprisingly, as can be seen in the results, for $N = 80$ the success rate to form a path is still at 87% for the maximum level of noise. However, a path formed during a successful trial with such a high level of noise is not stable and will disintegrate at some point.

When the prey extension mechanism is used the success rate for path formation drops even stronger. The instability of the vectorfield structure leads the majority of the robots to join the prey extending structure, which makes path formation im-

[2] An object can be another robot, the nest or the prey. Obstacles can not be perceived with the camera.

[3] As mentioned in Section 3.2.2, the values used in all experiments for these four noise levels are 18^o, 10 cm 0.2, and 36^o. In the robustness tests we manipulate one value at a time.

Table 8.6: Success rates of the vectorfield controller compared to the aligning chain controller for the robustness test on the perception of direction using the camera. The three values in the first row of each box represent percentages of the success rate from 100 trials for path formation, assembly and transport in this order, and the value in the second row is the median completion time for path formation T_p.

Noise	Vectorfield	Aligning	Vectorfield-X	Aligning-X
$N = 40$				
0^o	68/59/58 T_p =3416.5	100/100/93 T_p =371.5	83/63/62 T_p =881	100/100/98 T_p =504
36^o	61/0/0 T_p =3671	98/98/91 T_p =732.5	67/0/0 T_p =2222.5	99/99/97 T_p =1052
72^o	17/0/0 T_p =10000	88/73/22 T_p =3077	2/0/0 T_p =10000	99/95/41 T_p =1909.5
108^o	4/0/0 T_p =10000	33/13/0 T_p =10000	0/0/0 T_p =10000	50/21/0 T_p =9782
144^o	1/0/0 T_p =10000	10/0/0 T_p =10000	0/0/0 T_p =10000	33/1/0 T_p =10000
180^o	0/0/0 T_p =10000	35/0/0 T_p =10000	0/0/0 T_p =10000	0/0/0 T_p =10000
$N = 80$				
0^o	100/94/89 T_p =119	100/100/96 T_p =184	100/98/80 T_p =195	100/100/97 T_p =994.5
36^o	100/6/1 T_p =182	100/100/98 T_p =217	99/0/0 T_p =1130	99/99/98 T_p =1082.5
72^o	100/0/0 T_p =512	100/97/24 T_p =738	1/0/0 T_p =10000	99/97/29 T_p =1431.5
108^o	99/0/0 T_p =893	76/34/0 T_p =3571.5	0/0/0 T_p =10000	91/69/0 T_p =3108.5
144^o	96/0/0 T_p =1865	58/3/0 T_p =6468.5	0/0/0 T_p =10000	54/3/0 T_p =9790.5
180^o	87/0/0 T_p =2488	85/0/0 T_p =2720	0/0/0 T_p =10000	0/0/0 T_p =10000

possible. As already explained for the chains, the connectivity of the prey extending structure is based on simpler rules. Therefore, its stability is far less disturbed by the injection of noise than is the case for chains and vectorfield.

The drop of the success rate for subtask assembly is more pronounced for the vectorfield than for the aligning chains. This is mainly due to the fact that the stability of the vectorfield structure is disrupted. Even if a path is formed, it is

usually destroyed quickly, and there is in most cases not enough time for robots to get recruited to assemble to the prey. The same is true for subtask transport, even though the results are not very meaningful due to the lack of trials with a successful assembly. If assembly is successful, the transport is made very difficult for increasing levels of noise as the vectorfield structure is instable, and therefore a goal direction for the transporters is missing.

8.3.5.2 Camera Distance

In addition to the direction towards an object, the camera informs a robot about the distance. This distance information is used for navigating along the vectorfield, for assembly, and for transport. Table 8.7 displays the achieved success rates for the different subtasks and the median completion times for path formation comparing the vectorfield to the aligning chains.

For what concerns path formation, the vectorfield controller is in general less affected by the presence of camera distance noise than the aligning chains. The same could be observed for static chains when compared to aligning and moving chains. The path forming structures of vectorfield and static chains do not move, and are therefore not broken up once formed. For the aligning and moving chains, neighbouring chain robots adjust their distance with respect to each other. Falsely assuming that the neighbour is close may lead the robot to move away from the neighbour, and in case it gets out of sight, the chain breaks up.

For $N = 40$ the success rate to form a path is even higher for the vectorfield controller when high levels of noise are injected than at the default level of noise (10 cm). A similar effect was observed for the static chains. The reason is that due to noise the distance between neighbouring robots in the vectorfield structure can be increased.

For high levels of noise, the prey extension mechanism leads to a higher degree of sensitivity. For the aligning chains this is mainly caused by the instability of the chains due to which the majority of the robots joins the prey extending structure. The stability of the vectorfield structure is much less affected by the presence of noise. The reason that the performance drops is the mechanism that lets a robot leave the vectorfield in case it perceives the prey (or a robot that activates its LEDs in the same colour as the prey) perceived at a close distance. This is often and especially for large group sizes falsely the case if the prey extension mechanism is used. Therefore, for large group sizes the performance decrease is stronger for high levels of noise when the prey extension mechanism is used.

The success of subtask assembly, while also diminishing with increasing levels of noise, is less affected than when used with the aligning chains. This can be explained with the lower degree of stability for the aligning chains that tend to break up more easily when a high level of noise is injected. Therefore, a robot does often not have

Table 8.7: Success rates of the vectorfield controller compared to the aligning chain controller for the robustness test on the perception of distance using the camera. The three values in the first row of each box represent percentages of the success rate from 100 trials for path formation, assembly and transport in this order, and the value in the second row is the median completion time for path formation T_p.

Noise	Vectorfield	Aligning	Vectorfield-X	Aligning-X
$N = 40$				
0 cm	76/62/60 T_p =1255.5	100/100/87 T_p =325	89/80/77 T_p =722.5	100/100/97 T_p =643
10 cm	66/58/57 T_p =1301.5	100/100/95 T_p =479	81/63/62 T_p =770.5	100/100/95 T_p =613.5
20 cm	69/56/2 T_p =1806	100/98/90 T_p =835.5	79/60/0 T_p =1290	100/100/98 T_p =445.5
30 cm	75/59/1 T_p =1577.5	99/20/4 T_p =1623	86/38/0 T_p =1194	100/24/7 T_p =1449.5
40 cm	80/17/0 T_p =1563	94/0/0 T_p =3202.5	75/0/0 T_p =2895.5	36/0/0 T_p =10000
50 cm	88/0/0 T_p =1807	83/0/0 T_p =4495.5	33/0/0 T_p =10000	4/0/0 T_p =10000
$N = 80$				
0 cm	99/98/93 T_p =133	100/100/96 T_p =151	99/96/82 T_p =210	100/100/97 T_p =998
10 cm	100/93/89 T_p =92	100/100/97 T_p =167	100/98/80 T_p =224	100/100/99 T_p =969.5
20 cm	100/99/5 T_p =77	100/100/79 T_p =152	100/97/3 T_p =354.5	100/100/97 T_p =500
30 cm	100/98/0 T_p =103	100/59/16 T_p =165.5	100/67/0 T_p =550	100/20/1 T_p =1483
40 cm	100/58/0 T_p =110	100/2/0 T_p =394.5	10/0/0 T_p =10000	41/0/0 T_p =10000
50 cm	100/1/0 T_p =246	100/0/0 T_p =488.5	1/0/0 T_p =10000	2/0/0 T_p =10000

enough time to assemble.

While the vectorfield showed a higher degree of robustness for subtasks path formation and assembly, for subtask transport it appears to be more sensitive than the aligning chain controller. From a noise level of 20 cm the transport success rate drops to near 0. The reason is that a higher degree of noise also leads to a higher degree of branching, and therefore to a higher probability that multiple branches

Table 8.8: Success rates of the vectorfield controller compared to the aligning chain controller for the robustness test on the perception of obstacles using the proximity sensors. The three values represent percentages of the success rate from 100 trials for path formation, assembly and transport in this order.

Noise	Vectorfield	Aligning	Vectorfield-X	Aligning-X
$N = 40$				
0%	63/41/38	100/100/95	83/65/59	100/100/96
20%	57/40/37	100/100/95	81/64/60	100/100/96
40%	62/48/45	100/100/93	80/69/66	100/100/96
60%	64/50/48	100/100/88	78/63/58	100/100/99
80%	61/49/47	100/100/90	85/72/71	100/100/95
100%	59/44/42	100/100/92	84/63/61	100/100/96
$N = 80$				
0%	99/91/85	100/100/98	100/99/87	100/100/98
20%	100/96/84	100/100/99	99/95/78	100/100/96
40%	100/94/80	100/100/97	100/95/77	100/100/97
60%	99/92/78	100/100/99	98/94/84	99/98/97
80%	100/91/79	100/100/95	100/94/75	100/100/99
100%	99/90/77	100/100/96	99/95/78	98/92/78

lead to the prey. In this case, once that assembly is finished, the robots may perceive different goal directions and therefore do not succeed to move the prey.

8.3.5.3 Proximity Sensors

In this test we investigate the capability to cope with a noisy perception of obstacles when using the proximity sensors. The level of noise injected is normalized by the maximum activation of a proximity sensor, obtained when a robot is placed directly next to a wall, an obstacle, or another robot. We tested six different ratios of noise in the range $[0, 1]$. The achieved success rates for the different strategies and subtasks are reported in Table 8.8.

Similarly as found for the chain controller, the performance of the vectorfield controller is very robust with respect to noise of the proximity sensors. Even though there are more collisions and the behaviour appears more random with a high degree of noise, the success rates of none of the three subtasks diminishes significantly.

8.3.5.4 Vectorfield Indicated Direction

In a final robustness test, we investigate the sensitivity with respect to the perception of the direction indicated by robots in the vectorfield. We test for six different

Table 8.9: Success rates of the vectorfield controller for the robustness test on the perception of the direction indicated by the vectorfield for the vectorfield controller. The three values in the first row of each box represent percentages of the success rate from 100 trials for path formation, assembly and transport in this order, and the value in the second row is the median completion time for path formation T_p.

Noise	Vectorfield	Vectorfield-X
$N = 40$		
0^o	69/52/50 T_p =1428	80/66/64 T_p =789
36^o	58/41/38 T_p =3074	76/55/49 T_p =754.5
72^o	61/41/39 T_p =4066	79/65/60 T_p =922
108^o	85/52/50 T_p =2084	95/87/84 T_p =977.5
144^o	74/3/1 T_p =3920	97/3/2 T_p =2173.5
180^o	7/0/0 T_p =10000	30/0/0 T_p =10000
$N = 80$		
0^o	98/92/80 T_p =114	100/96/85 T_p =222
36^o	99/90/84 T_p =104.5	100/94/79 T_p =211.5
72^o	100/93/84 T_p =116	100/100/86 T_p =256.5
108^o	100/92/84 T_p =153	99/97/89 T_p =547.5
144^o	100/14/8 T_p =223.5	92/1/0 T_p =2316
180^o	100/0/0 T_p =448.5	26/0/0 T_p =10000

maximum noise levels in the range $[0^o, 180^o]$. Table 8.9 shows the achieved success rates for the different subtasks, and the median completion time T_p for subtask path formation.

The direction indicated by robots in the vectorfield serves two purposes. First, robots moving along the vectorfield follow the opposite direction of the one indicated. In this way they move away from the nest and towards the border of the vectorfield.

Second, a robot in the vectorfield regularly checks the directions towards which the neighbouring vectorfield robots are pointing. The robot can only leave the vectorfield in case it perceives no other robot pointing towards it, as otherwise it considers itself to be situated at the border of the vectorfield and—by leaving—it would risk to break up the vectorfield structure. For increasing levels of noise in the perception of the indicated direction, the risk increases that a robot does not recognise a neighbour pointing towards it. Therefore, the stability of the vectorfield structure is decreased.

Surprisingly, the success rate for path formation does not decrease even for a very high level of noise. In fact, for $N = 40$ it even increases up to a noise level of 144^o. However, even if the success rate increases, the median completion time increases as well. This means that when noise is injected, the problem of forming a path is solved more reliably, but more slowly as well. The reason for this is that, when the structure breaks up, the robots rearrange and form a new structure. In this way time is lost, but—in case a break up does not occur too frequently—it gives the system the possibility for a fresh start, increasing the probability to search previously unexplored areas of the arena.

The performance is similarly affected when using prey extension mechanism. However, for $N = 80$ and the highest level of noise, the performance decreases to 26%, which is even less than the 30% reached for a smaller group size of $N = 40$. The high level of noise leads to a very unstable vectorfield structure. When the prey extension mechanism is used the majority of the robots joins it. This becomes more pronounced for a bigger group size, as then the prey extending structure spreads all over the environment.

For subtasks assembly and transport, the indicated direction by vectorfield robots is of no importance. However, for high levels of noise, a formed path has a very short lifetime and therefore there is often no time for other robots to be recruited and get near the prey in order to assemble to it. If they manage to do this, the transporters often lose sight of the vectorfield structure due to its instability and therefore lack a goal direction.

8.3.6 Fault Tolerance Tests

In four fault tolerance tests we investigate the ability of the vectorfield controller to cope with individual failure by deactivating a sensor or actuator for varying fraction of the robot group in the range $[0\%, 100\%]$. As it was done for the chain controller in Section 5.3.6, we disabled the camera (Section 8.3.6.1), the LEDs (Section 8.3.6.2), the proximity sensors (Section 8.3.6.3), or the tracks (Section 8.3.6.4).

Table 8.10: Success rates of the vectorfield controller compared to the aligning chain controller for the fault tolerance test on disabled cameras. The three values represent percentages of the success rate from 100 trials for path formation, assembly and transport in this order.

Fraction	Vectorfield	Aligning	Vectorfield-X	Aligning-X
$N = 40$				
0%	66/58/57	100/100/95	81/63/62	100/100/100
20%	63/58/58	100/100/91	81/73/72	100/100/93
40%	52/48/47	100/100/90	62/60/58	100/100/98
60%	22/17/17	99/99/95	45/40/40	100/100/95
80%	0/0/0	76/0/0	2/0/0	93/0/0
100%	0/0/0	0/0/0	0/0/0	0/0/0
$N = 80$				
0%	100/93/89	100/100/97	100/98/80	100/100/99
20%	100/91/88	100/100/95	100/97/79	100/100/98
40%	94/89/86	100/100/100	99/94/79	100/100/99
60%	74/63/59	100/100/97	91/85/77	100/100/92
80%	25/15/12	97/97/91	53/40/29	99/99/94
100%	0/0/0	0/0/0	0/0/0	0/0/0

8.3.6.1 Camera

The only behaviours of both vectorfield and chain controller that do not depend on the camera are obstacle avoidance and random walk. Respectively, a robot with a disabled camera can not contribute to the solution of any of the three subtasks. Instead, it performs a random walk and can be considered as a mobile obstacle. Our results, as shown in Table 8.10, confirm this.

Disabling the camera of a robot leads to a similar performance as removing it entirely from the arena. Therefore, the tolerance to failure depends on whether the group size is large enough to accomplish the task without the erroneous robots. The chain controller reaches a higher fault tolerance than the vectorfield controller because in general it requires less robots to solve the task. The prey extension mechanism is beneficial for reaching a higher degree of fault tolerance as well, because it further increases the efficiency for small group sizes.

8.3.6.2 LEDs

If a robot's LEDs are deactivated, it can not signal its internal state to the other robots. It can still recognise other robots that signal their state, the nest and the prey, and can therefore assemble to and transport the prey, and in this way par-

Table 8.11: Success rates of the vectorfield controller compared to the aligning chain controller for the fault tolerance test on disabled LEDs. The three values represent percentages of the success rate from 100 trials for path formation, assembly and transport in this order.

Fraction	Vectorfield	Aligning	Vectorfield-X	Aligning-X
$N = 40$				
0%	66/58/57	100/100/95	81/63/62	100/100/100
20%	61/45/44	100/100/87	80/61/59	100/100/93
40%	55/36/35	100/99/79	64/61/55	100/100/94
60%	26/13/13	98/86/49	44/35/30	100/100/93
80%	0/0/0	9/3/2	4/0/0	35/20/15
100%	0/0/0	0/0/0	0/0/0	0/0/0
$N = 80$				
0%	100/93/89	100/100/97	100/98/80	100/100/99
20%	100/92/88	100/100/84	100/95/85	100/100/99
40%	94/87/77	100/100/64	95/89/83	100/100/91
60%	72/53/46	100/96/15	82/67/62	100/100/65
80%	38/17/8	79/22/0	46/27/11	96/58/6
100%	0/0/0	0/0/0	0/0/0	0/0/0

ticipate in the accomplishment of these two subtasks. However, the fault tolerance test on disabled LEDs with the chain controller (see Section 5.3.6.2) revealed that by disabling the LEDs of a robot its behaviour becomes even more disruptive for the whole system than by disabling the camera. The reason is that there is an overcrowding of the robots without LEDs along chains which can cause the chains to break up. Furthermore, when a path is formed, all erroneous robots try to assemble to the prey at the same time and disturb each other from connecting to the prey.

As can be seen in Table 8.11, the success rate of the vectorfield decreases as well when disabling the LEDs. As in the previous test, the vectorfield is less fault tolerant than the chains as it requires a higher density of robots. However, for the vectorfield the performance decrease is comparable to the one observed when disabling the camera, which means that the effect is less disruptive than for the chains. The reason for this is that the vectorfield structure does not suffer from a loss of stability, as due to the higher degree of randomness the behaviour has less or no overcrowding of the erroneous robots, and as due to the fact that a robot in the vectorfield does not move it is more stable by default.

Table 8.12: Success rates of the vectorfield controller compared to the aligning chain controller for the fault tolerance test on disabled proximity sensors. The three values represent percentages of the success rate from 100 trials for path formation, assembly and transport in this order.

Fraction	Vectorfield	Aligning	Vectorfield-X	Aligning-X
$N = 40$				
0%	66/58/57	100/100/95	81/63/62	100/100/100
20%	49/41/40	100/100/93	75/64/63	100/100/98
40%	45/42/39	100/100/95	69/65/62	100/100/94
60%	24/24/22	100/99/94	54/52/47	100/100/95
80%	18/15/12	96/96/89	48/47/41	99/99/95
100%	3/3/3	73/71/54	20/20/19	98/97/73
$N = 80$				
0%	100/93/89	100/100/97	100/98/80	100/100/99
20%	100/93/82	100/100/98	99/92/77	100/100/97
40%	98/94/89	100/100/99	100/96/84	100/100/100
60%	94/91/81	100/100/96	99/96/88	100/100/99
80%	85/78/68	100/100/91	97/91/82	100/100/98
100%	92/85/64	99/98/72	97/94/90	100/100/97

8.3.6.3 Proximity Sensors

A robot without proximity sensors is not able to detect other robots (that have not activated their LEDs), obstacles and walls. Therefore, when making a random walk it might get stuck in corners without recognising this. However, the robustness test showed that the performance stays comparably high even for high levels of noise. Also the fault tolerance test of the chain controller (see Section 5.3.6.3) showed that the task can still be accomplished very reliably if the proximity sensors are deactivated.

The results of our fault tolerance test for the vectorfield, as summarized in Table 8.12, show a stronger decrease in performance than for the chains. The reason for this is that more robots tend to get stuck at corners than is the case for the chains. Robots leaving a chain move back to the nest and then follow another chain. They therefore usually do not lose sight of the chain and do not get lost once they found the nest. Robots leaving the vectorfield structure perform a random walk for some time and therefore might get out of sight of the vectorfield and risk to get stuck.

This is particularly problematic when the density of robots is not very high, as for $N = 40$. For $N = 80$ the density is sufficiently high so that robots get stuck less

Table 8.13: Success rates of the vectorfield controller compared to the aligning chain controller for the fault tolerance test on disabled tracks. The three values represent percentages of the success rate from 100 trials for path formation, assembly and transport in this order.

Fraction	Vectorfield	Aligning	Vectorfield-X	Aligning-X
$N = 40$				
0%	66/58/57	100/100/95	81/63/62	100/100/100
20%	58/35/35	99/99/95	76/53/49	100/100/97
40%	43/25/23	97/95/89	70/52/51	99/99/94
60%	29/10/10	83/78/68	45/21/18	96/94/84
80%	4/0/0	50/0/0	20/0/0	73/0/0
100%	0/0/0	0/0/0	0/0/0	0/0/0
$N = 80$				
0%	100/93/89	100/100/97	100/98/80	100/100/99
20%	96/82/73	100/100/96	99/93/79	100/100/98
40%	92/66/62	100/100/92	97/90/81	100/100/99
60%	66/32/30	99/98/86	86/64/58	99/97/88
80%	48/13/12	85/79/60	59/22/19	82/67/53
100%	1/0/0	6/0/0	0/0/0	5/0/0

frequently and the performance stays high even if the proximity sensors of all robots are deactivated.

8.3.6.4 Tracks

When a robot's tracks are disabled it becomes immobile. It can still perceive other robots and use its LEDs to signal its internal state, and in case it is by chance situated at the right position, it can even join the path forming structure. But as it cannot move at all, it hardly participates in the accomplishment of any of the subtasks and can be considered as an immobile obstacle.

Not surprisingly, the results from Table 8.13 show that the performance decreases for growing fractions of robots with disabled tracks. As was the case for the previous tests, the vectorfield exhibits a lower degree of fault tolerance than the chains as it requires a higher density of robots to work efficiently.

8.4 Conclusions

In this chapter we analysed the performance of the vectorfield controller on the task of forming a path between nest and prey, assembling to the prey, and transporting

it back to the nest, and compared it to the chain controller. As it was done for the chain controller, we also tested the prey extension module in combination with the vectorfield controller.

As opposed to the chain controller, we tested the vectorfield controller only in simulation, and not on the real robot, which is due to three reasons: (i) A deficit of robots. To work efficiently, the vectorfield requires a high density of robots, and the number of available robots was not sufficiently high to perform meaningful experiments. (ii) A deficit of processing power. The software used to identify directions indicated by the vectorfield increases the length of a time step fourfold, which reduces the maximum speed of the robots and increases the length of an experiment.[4] (iii) A deficit of battery power. For the chain controller, a fully loaded battery can be used for up to approximately 45 minutes, which is very few for the vectorfield controller when considering the low number of available robots and the reduced speed of an experiment.

In general, we could observe that, due to a more uniform dispersal of robots in the structure, the vectorfield controller requires a higher density of robots than the chain controller to work efficiently. For low robot densities, the success rate remains below the one of chains. However, if the density of robots is high, the vectorfield controller outperforms the chain controller, in general leading to a faster formation of a path between nest and prey.

Similarly to what was observed for the chain controller, the benefit of the prey extension mechanism on path formation depends on the resources, that is, the size of the robot group, given to solve the task. If the resources are scarce, the prey extension mechanism has a beneficial effect on the overall performance, in general speeding up the path formation process. However, when the robot group is sufficiently large, the prey extension mechanism has a negative impact because the arena gets quickly covered with the prey extending structure, and in this way misleads the vectorfield structures towards regions which are only covered by robots that activate their LEDs in the colour of the prey, instead of the prey itself.

For subtasks assembly and transport we used the same control modules as were previously tested with the chain controller. Nevertheless, we found differences in the overall behaviour when they were used in combination with the vectorfield controller. The performance of subtask assembly is in general worse with the vectorfield controller than with the chain controller. The success rate is lower and the completion time is higher. The reason for this can be found in the less explicit recruitment of the vectorfield once a path between nest and prey is formed. A robot following a chain usually reaches its tail, and in case the chain connects nest and prey, it is led to the prey. Due to the higher degree of branching of the vectorfield, when following

[4]To allow for a better comparison, we set the maximum speed of a robot with the vectorfield controller in simulation to the same as used for the chain controller.

the vectorfield structure, a robot may follow a different branch or start a new one itself, even if the prey has been found already.

The situation is different for subtask transport. The performance of the vectorfield controller is better when compared to the chain controller. Even though the overall success rate usually remains below the one of the chain controller, the relative success rate, that is, the fraction of trials with a successful transport given a successful assembly, is higher. Furthermore, the transport is faster than is the case for chains. This has mainly two causes, both of which are related to the less explicit recruitment of robots to the prey. First, the size of the assembled structure of robots transporting the prey is smaller. This is beneficial for the transport because in a large pulling structure there are usually some robots that do not perceive the goal direction and therefore disturb the transport. Second, once two robots are connected to the prey, they can transport it with fewer robots arriving and trying to assemble to the prey which in effect blocks the transport. For the chains, this leads to a higher success rate for subtask assembly, but it makes the transport more difficult and in general less successful and slower.

In a series of robustness tests, we showed that the vectorfield controller can cope with a noisy perception of sensory data. Repeating the same tests performed for the chain controller, we varied the noise of the direction at which objects are perceived using the camera, the distance at which objects are perceived using the camera, and of the proximity sensors. The influence of noise in the sensory data has a similar impact as observed for the chains. As it was the case for chains, the performance drop is most pronounced for a noisy perception of the direction to other objects, and the performance is in general very robust to a noisy perception of the proximity sensors. Additionally, we performed one test in which we varied the noise of the direction-to-nest vector indicated by other robots in the vectorfield, and which is perceived by the camera. The performance of the vectorfield remains high for subtask path formation even when a high amount of noise is injected. However, the noise affects the stability of the vectorfield, so that a high level of noise leads to a high probability that a path once formed gets destroyed. Consequently, this leads to a decrease in the success rates of subtasks assembly and transport.

Finally, in a series of fault tolerance tests we analysed the ability of the vectorfield controller to cope with individual failure by deactivating the camera, the LEDs, the proximity sensors, or the tracks of a given fraction of robots in the group. In general, the level of fault tolerance was below the one observed for the chain controller. The reason for this is that the vectorfield controller requires a high density of robots to work efficiently, while the chains reach a high performance already for comparably small robot densities.

Chapter 9

Conclusions

In this last chapter, we provide a summary of the work done and indicate some directions for further research. In particular, in Section 9.1 we describe the task; in Section 9.2, we summarize the control approaches proposed, and in Sections 9.3 and 9.4, we present the experiments performed and an outline of the results achieved. Finally, in Section 9.5 we discuss possible directions for future work.

9.1 The Task

In this thesis, we studied a complex retrieval task in which a group of robots has to (i) form a path between two objects (referred to as nest and prey), (ii) assemble to the prey, and (iii) transport it back to the nest. The difficulty of the task is determined by the given constraints. First, the robots have to cooperate in order to form a path because a robot's perceptual range is small when compared to the distance between nest and prey. Second, the robots have no explicit knowledge about the location of nest or prey. Third, communication among robots is limited to a small set of simple signals that are locally broadcast. Fourth, the prey is too heavy to be transported by a single robot and therefore requires the cooperative effort of multiple robots to be moved.

9.2 Control Approaches

The focus and the main original contribution of this thesis lie in the solution of the path formation subtask. When designing our control algorithms we emphasized the cooperation and collectivity of the robot group and relied on principles such as simplicity of rules, homogeneity and distributedness of control, and locality of communication and information. We developed two novel navigation algorithms:

chains and vectorfield. Both algorithms lead the robots to form paths by building up visually connected structures of robots that start from the nest.

In addition to the two control algorithms, we developed a mechanism—referred to as prey extension—that results in a parallel formation of a second visually connected structure of robots that starts from the prey. The prey extension mechanism can be combined with either of the two approaches, and aims at extending the perception of the prey.

We developed three variants of the chain algorithm, differing from each other by the degree of motion allowed to the robots in the path forming structure. In the simplest case, the robots are not allowed to move at all. In the second case, members of a chain align themselves with respect to each other. In the last case, the chains move while keeping a visual connection to the nest.

For subtasks assembly and transport the controllers we used were based on the work of Groß et al. (2006a); Groß and Dorigo (2004) and Groß et al. (2006b), and modified in order to integrate them with our navigation algorithms.

9.3 Experiments

The chain controller was tested in simulation as well as on the real robot. For the vectorfield controller our experiments were only performed in simulation. The reasons why we decided not to perform experiments on the real robot are all related to limited resources. We did not have enough robots, and the robot computing power and battery life were less than what would have been needed to run the experiments.

In simulation, we first performed a study of the various parameters of the controllers, and selected the most successful combination of parameters for detailed evaluation. The controllers were tested under a wide range of experimental conditions, varying the distance between nest and prey, the number of robots used, and the obstacle configurations in the environment. Furthermore, we performed robustness tests in which we varied the noise added to several sensors, and fault tolerance tests in which we disabled sensors or actuators of a given fraction of robots. For the chain controller, we performed experiments in which we varied the distance between nest and prey, and the robot group size.

9.4 Results

One of the main differences (see Table 9.1) between the two proposed algorithms is that in the case of chains the created structures are linear, while in the case of vectorfield the structures are branched. This leads to a more uniform dispersal of the robots controlled by the vectorfield algorithm, that therefore requires a higher density of robots to work efficiently. This is also shown by the experimental results,

Table 9.1: Comparison of chain and vectorfield controllers.

Chain Controller	**Vectorfield Controller**
Linear structure	Branched structure → more uniform dispersal
Good for low robot density	Good for high robot density → better scalability
Faster assembly	Faster transport

where the vectorfield performs worse than the chains for low robot densities. However, when the robot density is high, the chains can get overcrowded because there are many robots following the same linear path. Due to the more uniform dispersal such an overcrowding does not occur when the vectorfield controller is used, which therefore outperforms the chains for high robot densities. Both controllers exhibit very good scalability characteristics. In our experimental conditions, the efficiency of the system in path formation increases with the group size for up to 100 robots and then remains roughly constant. For assembly and transport the scalability characteristics are worse than for path formation. In the case of assembly, large group sizes lead to disturbances because multiple robots try to assemble at the same time. If many robots are assembled to the prey and attempt to transport it, some of the transporters may not be able to perceive the goal direction (i.e., the path formed by chain or vectorfield), in this way making the transport more difficult.

Even though we used the same assembly and transport control modules for both navigation strategies, they reach different levels of success. Assembly is in general faster when used with the chain controller. This is due to the more explicit recruitment process. A robot moving along a chain usually reaches its tail, while a robot moving along a vectorfield has a higher probability of starting a new branch. Therefore, for the chains there are more robots assembling to the transporting structure, which in turn influences the transporting performance. As explained in the previous paragraph, a large number of robots in the transporting structure is disadvantageous. Therefore, and because there are less robots arriving to assemble to the prey, the vectorfield outperforms the chains in the transport subtask.

For growing distances between nest and prey, the completion time for assembly and transport increases linearly. The completion time to form a path grows more than linearly because the area that has to be explored grows quadratically.

We tested our controllers in three different obstacle environments, two of which are predefined, and one where the obstacles are randomly positioned. Even though the performance drops, both controllers are able to form a path also if the environment contains obstacles. The performance drop is mainly due to three reasons: (i) The increased difficulty of navigation, (ii) the possibility that nest and prey are hidden behind obstacles, and (iii) the possibility of an increased length of a path in case the straight path is blocked by an obstacle. For subtask assembly, the performance

drops because a connection to the prey can be restricted from several directions. For subtask transport, obstacles pose a particular problem as robots trying to move the prey can get stuck at an obstacle.

In a series of robustness tests we showed that both controllers are able to cope with noisy conditions. We tested different noise levels of the direction and distance perceived with the camera, and of the proximity sensor. Most of the information that is required for the controllers is provided by the camera. The proximity sensors are used only to avoid collisions. Therefore, the system is in general more sensitive to a noisy perception of the information provided by the camera, and in particular to the direction information. The higher the level of noise, the less stable becomes the path forming structure, and the higher the probability for it to break up. For the vectorfield controller we performed one additional robustness test, in which we varied the noise in the perception of directions indicated by robots in the vectorfield. Also in this case high levels of noise lead to a less stable vectorfield structure that can be broken up.

In a series of fault tolerance tests we analysed the ability of the controllers to cope with individual failures by deactivating either the camera, the LEDs, the proximity sensors, or the tracks of a given fraction of robots in the group. In general, as also observed for the robustness test, a large group size also leads to a higher degree of fault tolerance. Furthermore, the chain controller in general results in a higher degree of fault tolerance than the vectorfield, because the vectorfield requires a higher density of robot to work efficiently. Among the four tests performed, the proximity sensor led to the lowest degree of performance decrease. By disabling the camera a robot acts like a mobile obstacle and can not contribute to any of the three subtasks. By disabling the LEDs the effect on the behaviour is even more disruptive. A high degree of erroneous robots then has a disruptive effect on existing chains. In the case of disabled tracks a robot acts like an immobile obstacle. However, if by chance it happens to be positioned at the right place, it can in principle contribute to form a path, but not to any of the other two subtasks.

The prey extension mechanism has a good impact on the overall performance in case the resources (i.e., robots) are scarce, because in this case the resources are used more efficiently. However, for high densities of robots the prey extension mechanism was found to be less efficient, as in this case the arena gets quickly covered with the prey extending structure. Therefore, the chains take longer until they find the real prey instead of a robot activating its LEDs in the colour of the prey. In the robustness tests we observed that the prey extension mechanism results in a less tolerant behaviour to noise. The prey extending structure relies on simpler rules than the chain or vectorfield structure. Therefore, its stability remains high even for high levels of sensor noise. If a chain or vectorfield structure breaks up, the robots get quickly assimilated by the prey extending structure, in this way not leaving a

sufficient amount of robots to form a path.

Among the three variants of chains, we found that the two dynamic strategies outperform the static one. This is due to three reasons. First, the static chains are not aligned and therefore cover shorter distances from the nest. Second, the two dynamic chain strategies allow some motion to the chain members, which leads to an exploration of the arena even when chains are already formed. Third and last, the static chains have a higher risk to create loops in the form of a successive order of three chain members.

In addition to the experiments performed in simulation, we tested the chain controller on a real robot platform: the *s-bot*. We believe that these experiments are among the most sophisticated examples of self-organisation in robotics to date. The study confirms that complex forms of division of labour can indeed result from the interactions of robots that follow relatively simple and local rules. The study also demonstrates that teamwork requires neither individual recognition (the robots we use are inter-changeable) nor inter-individual differences (the robots we use are homogeneous in terms of "morphology" and "brain"), and as such might contribute to the ongoing debate on the role of such characteristics for the division of labour in social insects.

Summarizing our results, we can say that the proposed navigation algorithms have complementary characteristics. For low densities of robots, the chains outperform the vectorfield, and the prey extension mechanism improves the performance. However, for high robot densities the opposite is the case: The vectorfield controller outperforms the chains, and the prey extension mechanism has a negative impact on the performance.

9.5 Future Work

Swarm robotics is one of the most interesting and fascinating branches of robotics nowadays. Ever cheaper, more powerful, and smaller hardware components are opening new possibilities for the development of platforms that can be used to study swarm robotics. While for the moment swarm robotics is mainly used in research, the advantages of the swarm robotics approach will certainly lead to future applications. For instance, in space exploration the use of a swarm of robots instead of just one or a few robots has the advantage that the robot group will be able to cope better with the failure of an individual robot. Also, as the swarm robotics approach has a high degree of robustness, a swarm of robots could cope better with unknown and unexpected conditions that are found on unexplored planets. Other applications that could make use of robot swarms are rescue missions, where a swarm of robots could replace humans. The robots could then autonomously explore dangerous sites and in this way decrease the risk for humans.

For the topic of swarm robotics navigation, the future will bring the possibility of new communication channels. In our work, given the limitations of the available hardware, we use LEDs and a camera as the only way to communicate. In this way there are only very few signals that can be transmitted and perceived, and the signalling range is comparably short ($< 1\,\mathrm{m}$). While this is compatible with the emphasis of the swarm robotics approach to make use of simple and local communication, it limits the possibilities of our approach. The principle of simplicity does not strictly require the number of signals to be limited to a small set, and the principle of locality does not limit the communication range to a certain value. In the future, wireless communication channels will allow for interesting applications in which the robots can act in a larger physical range, and in which the use of a wider range of signals can lead to a more sophisticated outcome.

More concretely concerning the control mechanisms presented in this thesis, we see possible directions for future work in the use of adaptive controllers. We observed that for a given experimental setup there appears to be an optimal number of s-bots for each subtask. For instance, and in particular for the real robot we observed that if too many robots get engaged in transport there are usually some of the robots in the transporting structure that do not perceive the direction towards which they should move (i.e. the path). Therefore, a more advanced task allocation mechanism could be used to prevent too many robots to engage in transport. Furthermore, adaptivity of the controller could be beneficial in environments in which multiple prey objects are dynamically added to and removed from the arena. In the parameter study we showed that the set of parameters employed influences the overall behaviour of the robot swarm. On the one hand, a given parameter set leads to the formation of one path at a time which is beneficial for an environment with one prey that is far away. On the other hand, another parameter leads to the parallel formation of multiple paths, which would be more efficient in environments with multiple prey objects. By identifying the density of prey objects in the environment the robots could adapt their parameters in order to optimize the behaviour of the swarm for the given environmental conditions. We have already conducted preliminary tests in environments with multiple prey objects, and observed that both control mechanisms are able to form multiple concurrently formed paths to the different preys. In particular, we observed that due to its more branched structure the vectorfield controller is very useful in such environments.

Finally, another interesting direction for future work would be the investigation of our algorithms within the framework of failure modes and effects analysis (FMEA) Dailey (2004), which is a procedure to analyse a system with respect to robustness and fault tolerance. Failure causes are any errors or defects in process or design, and effects analysis refers to studying the consequences of such failures. We already test our controllers for robustness and fault tolerance. However, in order to

allow such systems to be applied in real world tasks a more systematic study—such as offered by FMEA—would be required.

Bibliography

Y. Aiyama, M. Hara, T. Yabuki, J. Ota, and T. Arai. Cooperative transportation by two four-legged robots with implicit communication. *Robot. Auton. Syst.*, 29 (1):13–19, 1999.

C. Anderson and N. R. Franks. Teams in animal societies. *Behav. Ecol.*, 12(5): 534–540, 2001.

C. Anderson and N. R. Franks. Teamwork in ants, robots and humans. *Adv. Stud. Behav.*, 33:1–48, 2004a.

C. Anderson and N. R. Franks. Teamwork in ants, robots and humans. *Adv. Stud. Behav.*, 33:1–48, 2004b.

C. Anderson and E. McMillan. Of ants and men: Self-organized teams in human and insect organizations. *Emergence*, 5(2):29–41, 2003.

C. Anderson and D. W. McShea. Intermediate-level parts in insect societies: Adaptive structures that ants build away from the nest. *Insectes Soc.*, 48(4):291–301, 2001.

C. Anderson and F. L. W. Ratnieks. Task partitioning in insect societies: Novel situations. *Insectes Soc.*, 47(2):198–199, 2000.

C. Anderson, G. Theraulaz, and J.-L. Deneubourg. Self-assemblages in insect societies. *Insect. Soc.*, 49(2):99–110, 2002.

R. C. Arkin. Motor schema-based mobile robot navigation. *International Journal of Robotics Research*, 8(4):92–112, 1989.

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

R.C. Arkin. *Behavior-Based Robotics*. MIT Press, Cambridge, MA, 1998.

K. Balakrishnan, O. Bousquet, and V. Honavar. Spatial learning and localization in rodents: a computation model of the hippocampus and its implications for mobile robots. *Adaptive Behavior*, 7(2):173–216, 1999.

T. Balch and R. Arkin. Communication in reactive multiagent robotic systems. *Autonomous Robots*, 1(1):27–52, 1994.

M. Batalin and G. S. Sukhatme. Spreading out: A local approach to multi-robot coverage. In *Proceedings of the Sixth International Symposium on Distributed Autonomous Robotic Systems*, pages 373–382. Springer Verlag, Berlin, Germany, Jun 2002.

R.A. Becker, J.M. Chambers, and A.R. Wilks. *The New S Language. A Programming Environment for Data Analysis and Graphics*. Chapman and Hall, London, UK, 1988.

R.E. Bellman. *Dynamic Programming*. Princeton University Press, Princeton, NJ, 1957.

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

A. Billard, A.J. Ijspeert, and A. Martinoli. A multi-robot system for adaptive exploration of a fast changing environment: probabilistic modeling and experimental study. *Connection Science*, 11(3/4):357–377, 2000.

M. Birattari. *The Problem of Tuning Metaheuristics as Seen from a Machine Learning Perspective*, volume 292 of *DISKI*. Infix, AKA/IOS Press, Berlin, Germany, 2005.

M. Birattari. *The problem of tuning metaheuristics as seen from a machine learning perspective*. PhD thesis, Université Libre de Bruxelles, Brussels, Belgium, 2004a.

M. Birattari. On the estimation of the expected performance of a metaheuristic on a class of instances. How many instances, how many runs? Technical Report TR/IRIDIA/2004-01, IRIDIA, Université Libre de Bruxelles, Brussels, Belgium, 2004b.

M. Birattari, T. Stützle, L. Paquete, and K. Varrentrapp. A racing algorithm for configuring metaheuristics. In W. B. Langdon et al., editors, *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 11–18. Morgan Kaufmann Publishers, San Francisco, CA, USA, 2002.

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

V. Braitenberg. *Vehicles*. MIT Press, Cambridge, MA, 1984.

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

A. Campo, S. Nouyan, M. Birattari, R. Groß, , and M. Dorigo. Negotiation of goal direction for cooperative transport. In *Proceedings of the 18th Belgium-Netherlands Conference on Artificial Intelligence*, pages 365–366. University of Namur, Namur, Belgium, 2006a.

A. Campo, S. Nouyan, M. Birattari, R. Groß, and M. Dorigo. Negotiation of goal direction for cooperative transport. In M. Dorigo et al., editors, *Ant Colony Optimization and Swarm Intelligence: 5th International Workshop, ANTS 2006*, volume 4150 of *Lecture Notes in Computer Science*, pages 191–202. Springer Verlag, Berlin, Germany, 2006b.

A. Castano, A. Behar, and P. M. Will. The Conro modules for reconfigurable robots. *IEEE/ASME Trans. Mechatron.*, 7(4):403–409, 2002.

A. L. Christensen. Efficient neuro-evolution of hole-avoidance and phototaxis for a swarm-bot. Technical Report TR/IRIDIA/2005-14, Université Libre de Bruxelles, Belgium, Octobre 2005. DEA Thesis.

A. L. Christensen and M. Dorigo. Evolving an integrated phototaxis and hole-avoidance behavior for a swarm-bot. In *Artificial Life X: Proceedings of the Tenth International Conference on the Simulation and Synthesis of Living Systems*, pages 248–254, Cambridge, MA, 2006. MIT Press, Cambridge, MA.

A. L. Christensen, R. O'Grady, and M. Dorigo. Morphology control in a self-assembling multi-robot system. *IEEE Robotics & Automation Magazine*, 14(4): 18–25, 2007.

A. L. Christensen, R. O'Grady, M. Birattari, and M. Dorigo. Fault detection in autonomous robots based on fault injection and learning. *Autonomous Robots*, 24 (1):49–67, 2008.

W.W. Cohen. Adaptive mapping and navigation by teams of simple robots. *Robotics and Autonomous Systems*, 18:411–434, 1996.

T. S. Collett. Landmark learning and guidance in insects. *Philosophical Transactions of the Royal Society of London*, B(337):295–303, 1992.

T. S. Collett, E. Dillmann, A. Giger, and R. Wehner. Visual landmarks and route following in desert ants. *Journal of Computational Physiology*, A 158:835–851, 1986.

T. S. Collett, S. N. Fry, and R. Wehner. Sequence learning by honeybees. *Journal of Computational Physiology*, A 172:693–706, 1993.

K.W. Dailey. *The FMEA Handbook*. DW Publishing, 2004.

J.-L. Deneubourg, S. Aron, S.Goss, and J.-M. Pasteels. The self-organizing exploratory pattern of the argentine ant. *J. Insect Behavior*, 3:159–168, 1990.

J. Desai, C.-C. Wang, M. Žefran, and V. Kumar. Motion planning for multiple mobile manipulators. In *Proc. of the 1996 IEEE Int. Conf. on Robotics and Automation*, volume 3, pages 2073–2078. IEEE Computer Society Press, Los Alamitos, CA, 1996.

B. R. Donald, J. Jennings, and D. Rus. Information invariants for distributed manipulation. *Int. J. Robot. Res.*, 16(5):673–702, 1997.

M. Dorigo and M. Birattari. Swarm intelligence. *Scholarpedia*, 2(9):1462, 2007.

M. Dorigo and E. Şahin. Swarm robotics — special issue editorial. *Autonomous Robots*, 17(2–3):111–113, 2004.

M. Dorigo, E. Tuci, R. Groß, V. Trianni, T. H. Labella, S. Nouyan, C. Ampatzis, J.-L. Deneubourg, G. Baldassarre, S. Nolfi, F. Mondada, D. Floreano, and L. M. Gambardella. The SWARM-BOTS project. In E. Sahin and W.M. Spears, editors, *Swarm Robotics: SAB 2004 International Workshop*, volume 3342 of *Lecture Notes in Computer Science*, pages 31–33. Springer Verlag, Berlin, Germany, 2004.

M. Dorigo, E. Tuci, V. Trianni, R. Groß, S. Nouyan, C. Ampatzis, T. H. Labella, R. O'Grady, M. Bonani, and F. Mondada. SWARM-BOT: Design and implementation of colonies of self-assembling robots. In G. Y. Yen and D. B. Fogel, editors, *Computational Intelligence: Principles and Practice*, chapter 6, pages 103–135. IEEE Computational Intelligence Society, New York, 2006.

A. Drogoul and J. Ferber. From tom thumb to the dockers: Some experiments with foraging robots. In *From Animals to Animats 2. Proceedings of the Second International Conference on Simulation of Adaptive Behavior (SAB92)*, pages 451–459. MIT Press, Cambridge, MA, 1992.

F. C. Dyer. Bees acquire route-based memories but not cognitive maps in a familiar landscape. *Animal Behaviour*, 41:239–246, 1991.

D. Filliat and J.-A. Meyer. Map-based navigation in mobile robots - I. a review of localization strategies. *J. of Cognitive Systems Research*, 4:243–282, 2003.

N. R. Franks. Teams in social insects: Group retrieval of prey by army ants (*Eciton burchelli, Hymenoptera: Formicidae*). *Behav. Ecol. Sociobiol.*, 18(6):425–429, 1986.

M. O. Franz and H. A. Mallot. Biomimetic robot navigation. *Robotics and Autonomous Systems*, 30:133–153, 2000.

M. O. Franz, B. Schölkopf, H. A. Mallot, and H. H. Bülthoff. Where did i take that snapshot? Scene-based homing by image matching. *Byological Cybernetics*, 79: 191–202, 1998.

T. Fukuda and T. Ueyama. *Cellular Robotics and Micro Robotic Systems*. World Scientific Publishing, London, UK, 1994.

T. Fukuda, S. Nakagawa, Y. Kawauchi, and M. Buss. Self organizing robots based on cell structures—CEBOT. In *Proc. of the 1988 IEEE/RSJ Int. Workshop on Intelligent Robots and Systems*, pages 145–150. IEEE Computer Society Press, Los Alamitos, CA, 1988.

T. Fukuda, T. Ueyama, and K. Sekiyama. Distributed intelligent systems in cellular robotics. In S. G. Tzafestas and H. B. Verbruggen, editors, *Artificial Intelligence in Industrial Decision Making, Control and Automation*, pages 225–246. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1995.

S. Garnier, J. Gautrais, and G. Theraulaz. The biological principles of swarm intelligence. *Swarm Intelligence*, 1(1):3–31, 2007.

P. Gaussier and S. Zrehen. Perac: A neural architecture to control artificial animals. *Robotics and Autonomous Systems*, 16:291–320, 1995.

B. P. Gerkey and M. J. Matarić. Sold!: Auction methods for multirobot coordination. *IEEE Trans. Robot. Automat.*, 18(5):758–768, 2002.

S. Gillner and H. A. Mallot. Navigation and acquisition of spatial knowledge in a virtual maze. *Journal of Cognitive Neuroscience*, 10(4):445–463, 1998.

S. Goss and J.-L. Deneubourg. Harvesting by a group of robots. In *Proc. of the 1st Europ. Conf. on Artificial Life*, pages 195–204. MIT Press, Cambridge, MA, 1992.

P. P. Grassé. La reconstruction du nid et les coordinations interindividuelles chez *Bellicositermes natalensis* et *Cubitermes sp.* La théorie de la stigmergie: Essai d'interprétation du comportement des termites constructeurs. *Insectes Sociaux*, 6:41–81, 1959.

R. Groß and M. Dorigo. Group transport of an object to a target that only some group members may sense. In *Parallel Problem Solving from Nature – 8th Int. Conf. (PPSN VIII)*, volume 3242 of *LNCS*, pages 852–861. Springer Verlag, Berlin, Germany, 2004.

R. Groß, M. Bonani, F. Mondada, and M. Dorigo. Autonomous self-assembly in swarm-bots. *IEEE Trans. Robot.*, 22(6):1115–1130, 2006a.

R. Groß, F. Mondada, and M. Dorigo. Transport of an object by six pre-attached robots interacting via physical links. In *Proc. of the 2006 IEEE Int. Conf. on Robotics and Automation*, pages 1317–1323. IEEE Computer Society Press, Los Alamitos, CA, 2006b.

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

R. Groß, S. Nouyan, M. Bonani, F. Mondada, and M. Dorigo. Division of labour in self-organised groups. In *From Animals to Animats 10. Proceedings of the Tenth International Conference on Simulation of Adaptive Behavior (SAB08)*, pages 426–436. Springer Verlag, Berlin, Germany, 2008.

B. Hölldobler and E. O. Wilson. *The Ants*. Harvard University Press, Cambridge, MA, 1990.

N. Jakobi. Evolutionary robotics and the radical envelope of noise hypothesis. *Adaptive Behavior*, 6:325–368, 1997.

N. Jakobi, P. Husbands, and I. Harvey. Noise and the reality gap: The use of simulation in evolutionary robotics. In F. Morán, A. Moreno, J. J. Merelo, and P. Chacón, editors, *Advances in Artificial Life. Proceedings of the 3rd European Conference on Artificial Life (ECAL 1995)*, volume 929 of *Lecture Notes in Artificial Intelligence*, pages 704–720. Springer Verlag, Berlin, Germany, 1995.

R. L. Jeanne. The evolution of the organization of work in social insects. *Monit. Zool. Ital.*, 20:119–133, 1986.

M. W. Jørgensen, E. H. Østergaard, and H. H. Lund. Modular ATRON: Modules for a self-reconfigurable robot. In *Proc. of the 2004 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, volume 2, pages 2068–2073. IEEE Computer Society Press, Los Alamitos, CA, 2004.

S. Koenig, B. Szymanski, and Y. Liu. Efficient and inefficient ant coverage methods. *Ann. Math. Artif. Intell.*, 31:41–76, 2001.

K. Kosuge and T. Oosumi. Decentralized control of multiple robots handling an object. In *Proc. of the 1996 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, volume 1, pages 318–323. IEEE Computer Society Press, Los Alamitos, CA, 1996.

C. R. Kube and E. Bonabeau. Cooperative transport by ants and robots. *Robot. Auton. Syst.*, 30(1–2):85–101, 2000.

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

C. R. Kube and H. Zhang. Task modelling in collective robotics. *Auton. Robots*, 4 (1):53–72, 1997.

B. J. Kuipers and Y.-T. Byun. A robust, qualitative method for robot spatial learning. In *Proc. 7th National Conf. on Artificial Intelligence (AAAI-88)*, pages 774–779. Morgan Kaufman, Los Altos, CA, 1988.

D. Kurabayashi, J. Ota, T. Arai, and E. Yoshida. Cooperative sweeping by multiple mobile robots. In *Proc. of the 1996 IEEE Int. Conf. on Robotics and Automation*, pages 1744–1749. IEEE Computer Society Press, Los Alamitos, CA, 1996.

R. Kurazume and N. Shigemi. Cooperative positioning with multiple robots. In *Proc. of the 1994 IEEE Int. Conf. on Robotics and Automation*, pages 1250–1257. IEEE Computer Society Press, Los Alamitos, CA, 2001.

Tod Levitt and Daryl Lawton. Qualitative navigation for mobile robots. *Artificial Intelligence*, 44:305–360, 1990.

Q. Li, M. De Rosa, and D. Rus. Distributed algorithms for guiding navigation across a sensor network. In *9th International Conference on Mobile Computing and Networking*, pages 313–325. ACM Press, New York, NY, 2003.

I. Lieblich and M. A. Arbib. Multiple representations of space underlying behavior. *Behavioral and Brain Sciences*, 5:627–659, 1982.

H. H. Lund and B. Webb. A robot attracted to the cricket *Gryllus bimaculatus*. In P. Husbands and I. Harvey, editors, *Proceedings of the Fourth European Conference on Artificial Life*, pages 246–255, MIT Press, Cambridge, MA, 1997.

H. A. Mallot, H. Bülthoff, P. Georg, B. Schölkopf, and K. Yasuhara. View-based cognitiv map learning by an autonomous robot. In F. Fogelman-Soulié and P. Gallinari, editors, *Proceedings of ICANN'95—International Conference on Artificial Neural Networks*, volume II, EC2, pages 381–386. Springer Verlag, Berlin, Germany, 1995.

M. Mamei and F. Zambonelli. Physical deployment of digital pheromones through RFID technology. In *Proc. of the 4th Int. Conf. on Autonomous Agents and Multi-Agent Systems, AAMAS 2005*, pages 1353–1360. IEEE Press, Piscataway, NJ, 2005.

M. T. Mason. Mechanics and planning of manipulator pushing operations. *Int. J. Robot Res.*, 5(3):53–71, 1986.

M. Matarić. Navigating with a rat brain: A neurobiologically-inspired model for robot spatial representation. In J. A. Meyer and S. W. Wilson, editors, *From Animals to Animats. Proceedings of the First International Conference on Simulation of Adaptive Behavior (SAB90)*, pages 169–175, MIT Press, Cambridge, MA, 1990.

M. J. Matarić, M. Nilsson, and K. T. Simsarian. Cooperative multi-robot box-pushing. In *Proc. of the 1995 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, volume 3, pages 556–561. IEEE Computer Society Press, Los Alamitos, CA, 1995.

J.-A. Meyer and D. Filliat. Map-based navigation in mobile robots - II. a review of map-learning and path-planning strategies. *J. of Cognitive Systems Research*, 4: 283–317, 2003.

O. Miglino, H. H. Lund, and S. Nolfi. Evolving mobile robots in simulated and real environments. *Artificial Life*, 4:417–434, 1995.

N. Miyata, J. Ota, T. Arai, and H. Asama. Cooperative transport by multiple mobile robots in unknown static environments associated with real-time task assignment. *IEEE Trans. Robot. Automat.*, 18(5):769–780, 2002.

M. W. Moffett. Ant foraging. *Natl. Geogr. Res. Explor.*, 8(2):220–231, 1992.

R. Möller, D. Lambrinos, R. Pfeifer, T. Labhart, and R. Wehner. Modeling and navigation with an autonomous agent. In R. Pfeifer, B. Blumberg, J. A. Meyer, and S. W. Wilson, editors, *From Animals to Animats 5. Proceedings of the Fifth International Conference on Simulation of Adaptive Behavior (SAB98)*, pages 185–194, MIT Press, Cambridge, MA, 1998.

F. Mondada, L. M. Gambardella, D. Floreano, S. Nolfi, J.-L. Deneubourg, and M. Dorigo. The cooperation of swarm-bots: Physical interactions in collective robotics. *IEEE Robot. Automat. Mag.*, 12(2):21–28, 2005.

S. Murata, E. Yoshida, A. Kamimura, H. Kurokawa, K. Tomita, and S. Kokaji. M-TRAN: Self-reconfigurable modular robotic system. *IEEE/ASME Trans. Mechatron.*, 7(4):431–441, 2002.

R. C. Nelson. Visual homing using an associative memory. *Biological Cybernetics*, 65:281–291, 1991.

S. Nouyan. Agent-based approach to dynamic task allocation. In M. Dorigo, G. Di Caro, and M. Sampels, editors, *Ant Colony Optimization and Swarm Intelligence: 3rd International Workshop, ANTS 2002*, volume 2463 of *Lecture Notes in Computer Science*, pages 28–39. Springer Verlag, Berlin, Germany, 2002.

S. Nouyan. Path formation and goal search in swarm robotics. Technical Report TR/IRIDIA/2004-14, Université Libre de Bruxelles, Belgium, September 2004. DEA Thesis.

S. Nouyan and M. Dorigo. Chain based path formation in swarms of robots. In M. Dorigo et al., editors, *Ant Colony Optimization and Swarm Intelligence: 5th International Workshop, ANTS 2006*, volume 4150 of *Lecture Notes in Computer Science*, pages 120–131. Springer Verlag, Berlin, Germany, 2006.

S. Nouyan, R. Ghizzioli, M. Birattari, and M. Dorigo. An insect-based algorithm for the dynamic task allocation problem. *Künstliche Intelligenz*, 4(5):25–31, 2005.

S. Nouyan, R. Groß, M. Bonani, F. Mondada, and M. Dorigo. Group transport along a robot chain in a self-organised robot colony. In *Proc. of the* 9^{th} *Int. Conf. on Intelligent Autonomous Systems*, pages 433–442. IOS Press, Amsterdam, The Netherlands, 2006.

S. Nouyan, R. Groß, M. Bonani, F. Mondada, and M. Dorigo. Teamwork in self-organised robot colonies. *IEEE Transactions on Evolutionary Computation*, accepted for publication.

S. Nouyan, A. Campo, and M. Dorigo. Path formation in a robot swarm. *Swarm Intelligence*, 2(1), in press.

K.J. O'Hara and T.R. Balch. Pervasive sensor-less networks for cooperative multi-robot tasks. In *Proceedings of the Seventh International Symposium on Distributed Autonomous Robotic Systems*, pages 305–314. Springer Verlag, Tokyo, Japan, 2004.

G. F. Oster and E. O. Wilson. *Caste and ecology in the social insects*. Princeton Univ. Press, Princeton, NJ, 1978.

L. E. Parker. Adaptive heterogeneous multi-robot teams. *Neurocomputing*, 28(1-3): 75–92, 1999.

D. Payton, M. Daily, R. Estowski, M. Howard, and C. Lee. Pheromone robotics. *Autonomous Robots*, 11:319–324, 2001.

D. Payton, R. Estkowski, and M. Howard. Pheromone robotics and the logic of virtual pheromones. In E. Şahin et al., editor, *Swarm Robotics: SAB 2004 International Workshop*, volume 3342 of *Lecture Notes in Computer Science*, pages 45–57. Springer Verlag, Berlin, Germany, 2004.

F. L. W. Ratnieks and C. Anderson. Task partitioning in insect societies. *Insectes Soc.*, 46(2):95–108, 1999.

M. Recce and K. D. Harris. Memory of places: A navigational model in support of Marr's theory of hippocampal function. *Hippocampus*, 6:735–748, 1996.

I. Rekleitis, G. Dudek, and E. Milios. Multi-robot exploration of an unknown environment, efficiently reducing the odometry error. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1340–1345. Morgan Kaufman, Los Altos, CA, 1997.

I. Rekleitis, G. Dudek, and E. Milios. Accurate mapping of an unknown world and online landmark positioning. In *Proc. Vision Interface (VI)*, pages 455–461. IEEE press, Piscataway, NJ, 1998.

I. Rekleitis, G. Dudek, and E. Milios. Collaborative exploration for the construction of visual maps. In *Proc. of the 2001 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, volume 3, pages 1269–1274. IEEE press, Piscataway, NJ, 2001.

M. Rubenstein, K. Payne, P. Will, and W.-M. Shen. Docking among independent and autonomous CONRO self-reconfigurable robots. In *Proc. of the 2004 IEEE Int. Conf. on Robotics and Automation*, volume 3, pages 2877–2882. IEEE Computer Society Press, Los Alamitos, CA, 2004.

D. Rus, Z. Butler, K. Kotay, and M. Vona. Self-reconfiguring robots. *Commun. ACM*, 45(3):39–45, 2002.

A. B. Sendova-Franks and N. R. Franks. Self-assembly, self-organization and division of labour. *Phil. Trans. R. Soc. B*, 354(1388):1395–1405, 1999.

K. Singh and K. Fujimura. Map making by cooperating mobile robots. In *Proc. of the 1993 IEEE Int. Conf. on Robotics and Automation*, pages 254–259. IEEE Computer Society Press, Los Alamitos, CA, 1993.

T. G. Sugar and V. Kumar. Control of cooperating mobile manipulators. *IEEE Trans. Robot. Automat.*, 18(1):94–103, 2002.

H. Sugie, Y. Inagaki, S. Ono, H. Aisu, and T. Unemi. Placing objects with multiple mobile robots—mutual help using intention inference. In *Proc. of the 1995 IEEE Int. Conf. on Robotics and Automation*, volume 2, pages 2181–2186. IEEE Computer Society Press, Los Alamitos, CA, 1995.

J. Svennebring and S. Koenig. Building terrain covering ant robots: a feasibility study. *Autonomous Robots*, 116(3):193–221, 2004.

E. A. Tibbetts. Visual signals of individual identity in the wasp *Polistes fuscatus*. *Proc. R. Soc. Lond. B*, 269:1423–1428, 2002.

O. Trullier, S. Wiener, A. Berthoz, and J. Meyer. Biologically-based artificial navigation systems: Review and prospects. *Progress in Neurobiology*, 51:483–544, 1997.

E. Tuci, R. Groß, V. Trianni, M. Bonani, F. Mondada, and M. Dorigo. Cooperation through self-assembling in multi-robot systems. *ACM Trans. on Autonomous and Adaptive Systems*, 1(2):115–150, 2006.

Z.-D. Wang, E. Nakano, and T. Takahashi. Solving function distribution and behavior design problem for cooperative object handling by multiple mobile robots. *IEEE Trans. Syst., Man, Cybern. A*, 33(5):537–549, 2003.

B. Webb. Using robots to model animals: A cricket test. *Robotics and Autonomous Systems*, 16:117–134, 1995.

B. Werger and M. Matarić. Robotic food chains: Externalization of state and program for minimal-agent foraging. In *From Animals to Animats 4. Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior (SAB96)*, pages 625–634. MIT Press, Cambridge, MA, 1996.

S. Werner, B. Krieg-Brückner, H. A. Mallot, K. Schweizer, and C. Freksa. Spatial cognition: The role of landmark, route, and survey knowledge in human and robot navigation. In M. Jarke, K. Pasedach, and K. Pohl, editors, *Informatik 97*, pages 41–50. Springer Verlag, Berlin, Germany, 1997.

G. M. Whitesides and B. Grzybowski. Self-assembly at all scales. *Science*, 295 (5564):2418–2421, 2002.

E. O. Wilson. *Sociobiology*. Harvard Univ. Press, Cambridge, 1975.

E. O. Wilson. *The Insect Societies*. Harvard University Press, Cambridge, MA, 1971.

S. Yamada and J. Saito. Adaptive action selection without explicit communication for multirobot box-pushing. *IEEE Trans. Syst., Man, Cybern. C*, 31(3):398–404, 2001.

B. Yamauchi. Frontier-based exploration using multiple robots. In *Proc. 2nd Int. Conf. Autonomous Agents*, pages 47–53, 1998.

B. Yamauchi, A. Schultz, and W Adams. Integrating exploration and localization for mobile robots. *Adaptive Behaviour*, 7(2):217–229, 1999.

M. Yim, Y. Zhang, and D. Duff. Modular robots. *IEEE Spectr.*, 39(2):30–34, 2002a.

M. Yim, Y. Zhang, K. Roufas, D. Duff, and C. Eldershaw. Connecting and disconnecting for chain self-reconfiguration with PolyBot. *IEEE/ASME Trans. Mechatron.*, 7(4):442–451, 2002b.

R. Zlot, A. Stentz, M. Bernadine Dias, and S. Thayer. Multi-robot exploration controlled by a market economy. In *Proc. of the 2002 IEEE Int. Conf. on Robotics and Automation*, volume 3, pages 3016–3023. IEEE Computer Society Press, Los Alamitos, CA, 2002.

V. Zykov, E. Mytilinaios, B. Adams, and H. Lipson. Self-reproducing machines. *Nature*, 435(7039):163–164, 2005.