CHAPTER 6

SWARM-BOT:
DESIGN AND IMPLEMENTATION OF COLONIES OF SELF-ASSEMBLING ROBOTS

Marco Dorigo, Elio Tuci, Vito Trianni, Roderich Groβ, Shervin Nouyan, Christos Ampatzis, Thomas Halva Labella, Rehan O'Grady, Michael Bonani, and Francesco Mondada

I. INTRODUCTION

Recently, there has been a growing interest in multi-robot systems. This interest is motivated by the fact that inherent parallelism and redundancy make multi-robot systems more robust than single robot systems. Furthermore, a multi-robot system can be versatile enough to generate the different structures and functions required to undertake missions in unknown environmental conditions. Research in autonomous multi-robot systems often focuses on mechanisms to enhance the efficiency of the group through some form of cooperation among the individual agents. In the SWARM-BOTS project, the results of which are summarized in this paper, we investigated an innovative cooperation mechanism based on the *self-assembly* capabilities of the *s-bot*, a robot that we designed and built.

Self-assembly is the mechanism whereby a group of mobile robots autonomously form physical connections with one another. Self-assembly can allow a group of agents to cope with environmental conditions which prevent them from carrying out their task individually. For example, robots designed for all-terrain navigation could self-assemble when navigating in particularly rough terrains to reduce the risk of toppling over, or to reduce the risk of falling into holes larger than the body of a single robot. When required to transport an object, a group of self-assembled robots might be capable of pushing/pulling an object which, due to its characteristics (e.g., mass, size, shape), cannot be transported by a single robot.

Despite the relevance of self-assembly in the domain of multi-robot systems, progress in the design of control policies for self-assembling robots has been slow and fraught with difficulties. Excluding the work presented here, there are no other examples of self-assembling robots in which more than two autonomous mobile units managed to approach and connect with each other.

Self-assembly is likely to play a key role in multi robot systems of the future. Despite this, there is a lack of efficient self-assembling systems in the literature. The reasons for this are twofold. Firstly, surprisingly little attention has been devoted to this research field. Secondly, the mechanism of self-assembly is intrinsically complex, and existing systems have not succeeded in overcoming this complexity. All self-assembling systems to date require a high level of hardware precision both for the creation of the assembled structure and in the operation of the aggregate structure once assembled (see Tuci et al., 2006). Our goal in the SWARM-BOTS project was to address this discrepancy and bring self-assembly to the forefront of multi-robot research. To do so, we needed to build a robust, reliable, and scalable self-assembling system: a system composed of multiple mobile autonomous robots capable of using self-assembly and self-organization to adapt to different environmental conditions.

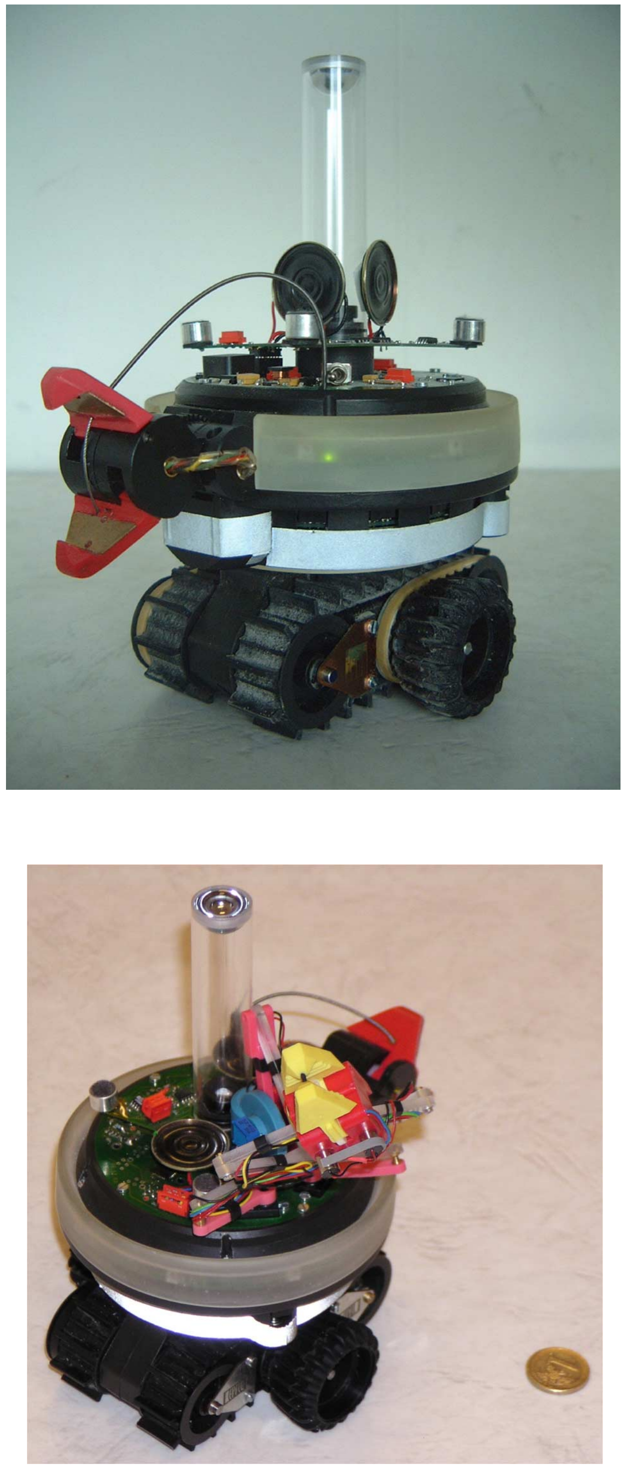
The project focused on the design and implementation of 35 small robots, called *s-bots* (see Fig. 1). These robots are equipped with a number of sensors and actuators, basic communication devices, and on-board computational capabilities. Additionally, these robots are endowed with physical connection mechanisms that allow them to attach to (and detach from) each other to form collective physical structures. We call these collective physical structures *swarm-bots*: a *swarm-bot* is an aggregate of *s-bots* that has the potential to exhibit capabilities that go beyond those of a single *s-bot* (see Mondada et al., 2004, 2005a,b). A *swarm-bot* forms as the result of self-organizing rules followed by each individual *s-bot* rather than via a global template and is expected to move as a whole and reconfigure when needed. For example, it might have to adopt a different shape in order to go through a tunnel or overcome an obstacle.

The scientific challenge of the SWARM-BOTS project was the design and realization of the hardware and software for such a robotic system. In this chapter, our aim is to introduce the theory underlying our research agenda and to present some of our results. In Section II, we illustrate the general principles and methodological choices that guided our research. In Section III, we give a brief description of the robot hardware, and of the experimental methodology employed to develop the *s-bots* controllers. In Section IV, we first introduce the experimental scenario that we have chosen as a test-bed for our approach. We go on to present the results of several experiments in which *s-bots* autonomously perform tasks related to this scenario. Subsections in Section IV describe (1) a set of cooperative navigation strategies that enable a *swarm-bot* to move efficiently as a result of the individual actions of the connected *s-bots*, (2) empirical studies in which the capability of an *s-bot* to assemble to objects and to other *s-bots* (self-assembly) was investigated in the context of object retrieval. Here, self-assembly and coordinated action of several robots is required when the object to be retrieved cannot be transported by a single robot, due to its size, shape and/or mass, and (3) empirical studies in which we investigate the capability of a group of *s-bots* to establish a path between two distant locations in the environment. The path is formed through the creation of a chain of visually linked *s-bots*. The chain of *s-bots* begins close to the starting location and terminates close to the goal location. The chapter closes with sections dedicated to general discussion and conclusions.

II. GENERAL PRINCIPLES FOR THE DESIGN OF THE S-BOTS

Swarm robotics is a rapidly expanding field of collective robotics that studies robotic systems composed of *swarms* of robots tightly interacting and cooperating to reach their goals (Dorigo and Sahin, 2004). Swarm robotics has its theoretical roots in recent studies of social insect societies, such as those of ants and bees. Social insects are highly successful at performing group-level tasks even when there is noise in the environment, when information processing errors occur, when there is no global information available, and even when some individuals make errors in performing the task. In keeping with the social insect metaphor, swarm robotics emphasizes principles of decentralization of control, limited communication between robots mainly through stigmergy, and "genetic relatedness." This latter characteristic is "imported" into robotics through the use of *homogeneous systems*. [Note that swarm robotic systems should consist of relatively few homogeneous groups of robots, and the number of robots in each group should be large. Studies that are concerned with highly heterogeneous robot groups, no matter how large the group size, are considered to be less "swarm robotic" (see

Figure 1. (a) Above: The *s-bot*. (b) Below: The *s-bot* close to a one Euro coin.



Dorigo and Sahin, 2004).] That is, groups of robots in which all the agents share the same physical and control structure (see Bonabeau et al., 1999).

The SWARM-BOTS project approach to the design and realization of self-assembling and self-organizing robotic systems is highly innovative. We designed and implemented a robotic system comprised of many autonomous robots with the unique ability to attach to (and detach from) one another. Our choices were motivated by the desire to ensure that our robotic system – the *swarm-bot* – would be robust and versatile. We also wanted a system capable of operating on rough terrain.

Conceptually, the *swarm-bot* lies somewhere between a traditional monolithic robot and a colony of cooperating robots. This is the key innovation of the SWARM-BOTS project. A *swarm-bot* can be considered as a single complex robot composed of many detachable parts (the individual *s-bots*). As in conventional cooperative robotics, these parts are capable of autonomous movement and control. The *s-bots* use their autonomy to act independently when they are not attached to each other, to self-assemble into a *swarm-bot* when necessary, and to implement autonomous reconfiguration and shape-changing activities when in *swarm-bot* configuration. A *swarm-bot*, once assembled, is not limited to a single configuration, but can change its shape according to its needs (as dictated by the demands of the task or environment).

Our approach to controller design was also innovative. We rigorously applied swarm intelligence principles throughout. In particular we only used distributed control mechanisms and our controllers only made use of locally available information. Our adherence to these principles is further reflected by our use of evolutionary computation techniques and behavior-based control architectures.

The behavior-based approach implies the design of controllers through the identification of a set of tasks and a set of behaviors (Brooks, 1991). Each behavior corresponds to a particular task or subtask the robot must perform. A behavior is a set of simple actions triggered by a particular set of sensory inputs. If, after an evaluation of the state of the robot's sensors, it turns out that there are two or more behaviors potentially capable of solving the task the robot is currently facing, an arbitration mechanism is used to choose which behavior to employ. This approach is particularly applicable in cases where it is possible to decompose each agent's task into a set of sub-tasks and a set of corresponding behaviors.

However, in swarm robotic applications, it is often very difficult to map the goals of the system into a clearly defined set of tasks and behaviors for each agent. In these cases evolutionary computation techniques provide a solution, since their use does not require an explicit analysis of the robotic task (Dorigo et al., 2004). Evolutionary computation automates the design process of robot control policies through the use of mechanisms inspired by natural selection (see Nolfi and Floreano, 2000, for details). Artificial evolution bypasses the problem of decomposition at two levels. It is no longer necessary to determine the mechanisms that lead to the emergent global behavior, nor are we faced with the challenge of implementing low-level behaviors on the *s-bots*. Artificial evolution relies on the evaluation of the system as a whole. The human designer only has to specify the desired global behavior – the individual agent behaviors that will create this emergent global behavior are then found by a process of trial and error using evolutionary algorithms.

The evolutionary approach is most commonly used to synthesize artificial neural networks (ANNs). These ANNs are then used to control the robots. ANNs are distributed computational systems whose structural and functional properties loosely resemble the brain of natural organisms (see Haykin, 1999). To develop the *s-bots'* controllers with evolutionary computation techniques, we used a 3D dynamics simulator called

Swarmbot3d, which provides realistic simulations of dynamics and collisions of rigid bodies in 3 dimensions. **Swarmbot3d** implements *s-bot* models with the functionality available on the real *s-bots*.

III. S-BOTS AND SWARM-BOTS

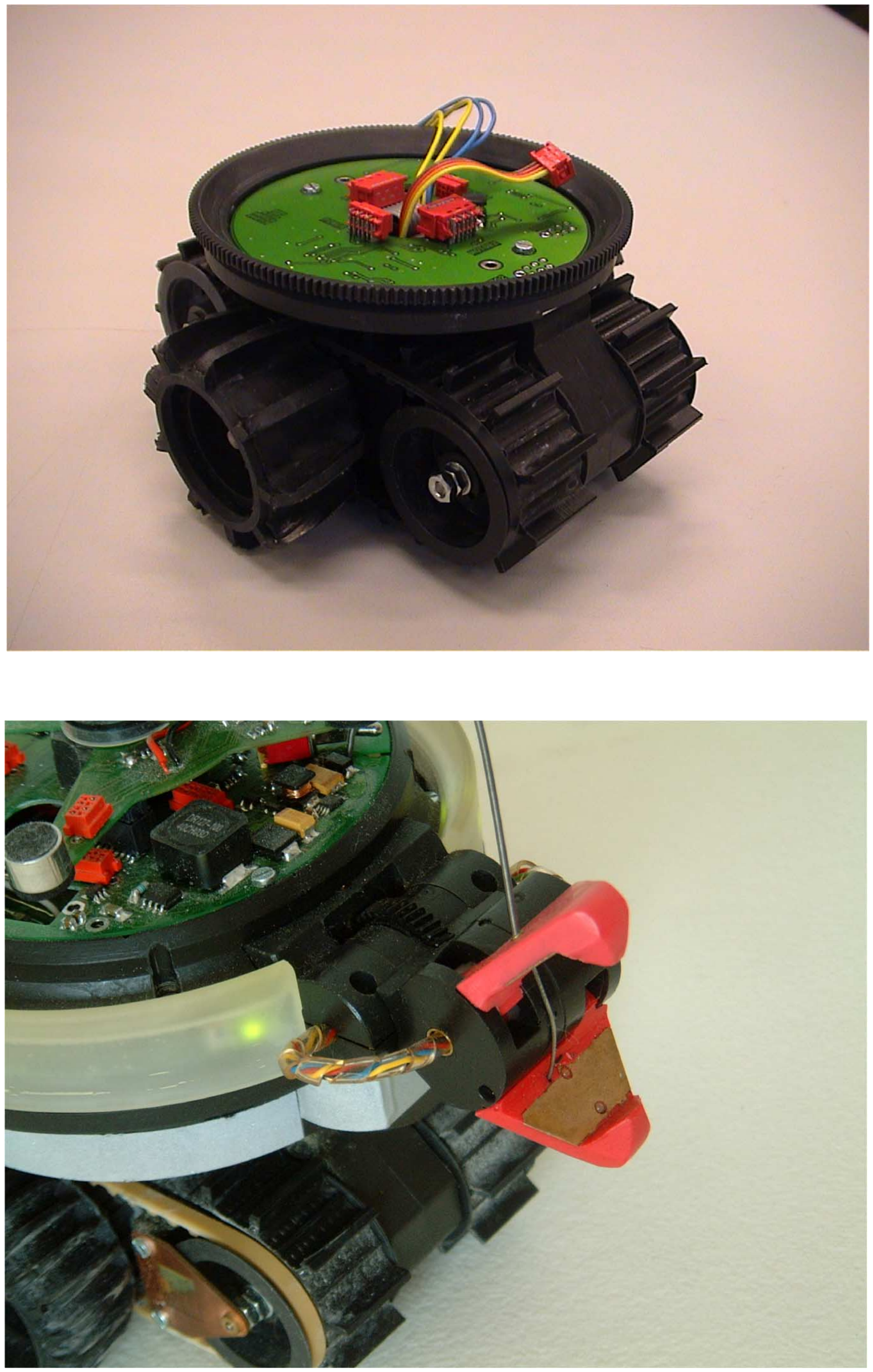
S-bots are the constituent components of a *swarm-bot*. Each *s-bot* (see Fig. 1) is a fully autonomous mobile robot capable of performing simple tasks such as autonomous navigation, perception of the environment and grasping of objects. In addition to these features, one *s-bot* can communicate with other *s-bots* and physically connect to them, thus forming a *swarm-bot*. A *swarm-bot* can perform tasks that are impossible for a single *s-bot*. Such tasks can include exploration, navigation, and transportation of heavy objects on rough terrain. The *s-bots'* individual and collective responses are triggered by several sensor modalities and accomplished through the exploitation of multiple actuators mounted on each agent.

The *s-bot's* innovative navigation system makes use of both tracks and wheels (see Fig. 2a). One motor controls the wheel and track for a single side of the *s-bot*. The combination of the left- and right-side motors provides a differential drive system. This differential drive system allows efficient rotation on the spot due to the large diameter of the wheels. It also gives the traction system a shape close to that of the cylindrical main body (turret), thus avoiding the typical rectangular shape of simple tracks and improving the *s-bot's* mobility. The *s-bot's* traction system can rotate with respect to the main body – i.e., the robot's turret – by means of a motorized joint. The turret holds the rigid gripper: a device which allows the *s-bots* to establish physical interconnections, thus enabling self-assembly into a *swarm-bot* configuration (see Fig. 2b). Such a gripper has a very large acceptance area allowing it to realize a secure grasp at any angle and, if necessary, to support the full weight of or lift another *s-bot* (see Fig. 3).

S-bots have a wide range of sensory systems, used both for the perception of the surrounding environment and for proprioception. Infrared proximity sensors are distributed around the rotating turret, and can be used for detection of obstacles and of other *s-bots*. Four proximity sensors are placed under the chassis, and can be used for perceiving holes or the terrain's roughness. Additionally, an *s-bot* has eight light sensors, two temperature/humidity sensors, a 3-axes accelerometer, and incremental encoders on each degree of freedom. Each *s-bot* is also equipped with audio and video devices to detect and communicate with other*s-bots*: an omni-directional camera, colored LEDs around the *s-bot's* turret, microphones and loudspeakers. Eight groups of three colored LEDs each – red, green, and blue – are mounted around the *s-bot's* turret. The color emitted by a robot's LEDs can be detected by other *s-bots* by using the omni-directional camera, enabling a form of local communication. The omni-directional camera allows the *s-bots* to grab panoramic views of its surroundings.

Proprioceptive sensors provide the *s-bot* with information about internal efforts, physical connections, and reactions at connection points with other *s-bots*. These include torque sensors on all joints as well as a traction sensor to measure the pulling/pushing forces exerted on the *s-bot's* turret. The traction sensor is placed at the junction between the turret and the chassis. This sensor measures the direction (i.e., the angle with respect to the chassis orientation) and the intensity of the traction force (henceforth called "traction") that the turret exerts on the chassis. The traction perceived by one robot can be caused either by the force applied by the robot itself while pulling/pushing an object grasped through the gripper element, or by the mismatch of its movement with respect to the movement of other robots connected to it, or by both the previous circumstances at the same time. The turret of an *s-bot* physically integrates, through a vector summation, the forces that are applied to it by another *s-bot*, as well as the force the *s-bot*

Figure 2. (a) Above: The traction system of an *s-bot*. (b) Below: The *s-bot*'s rigid gripper.



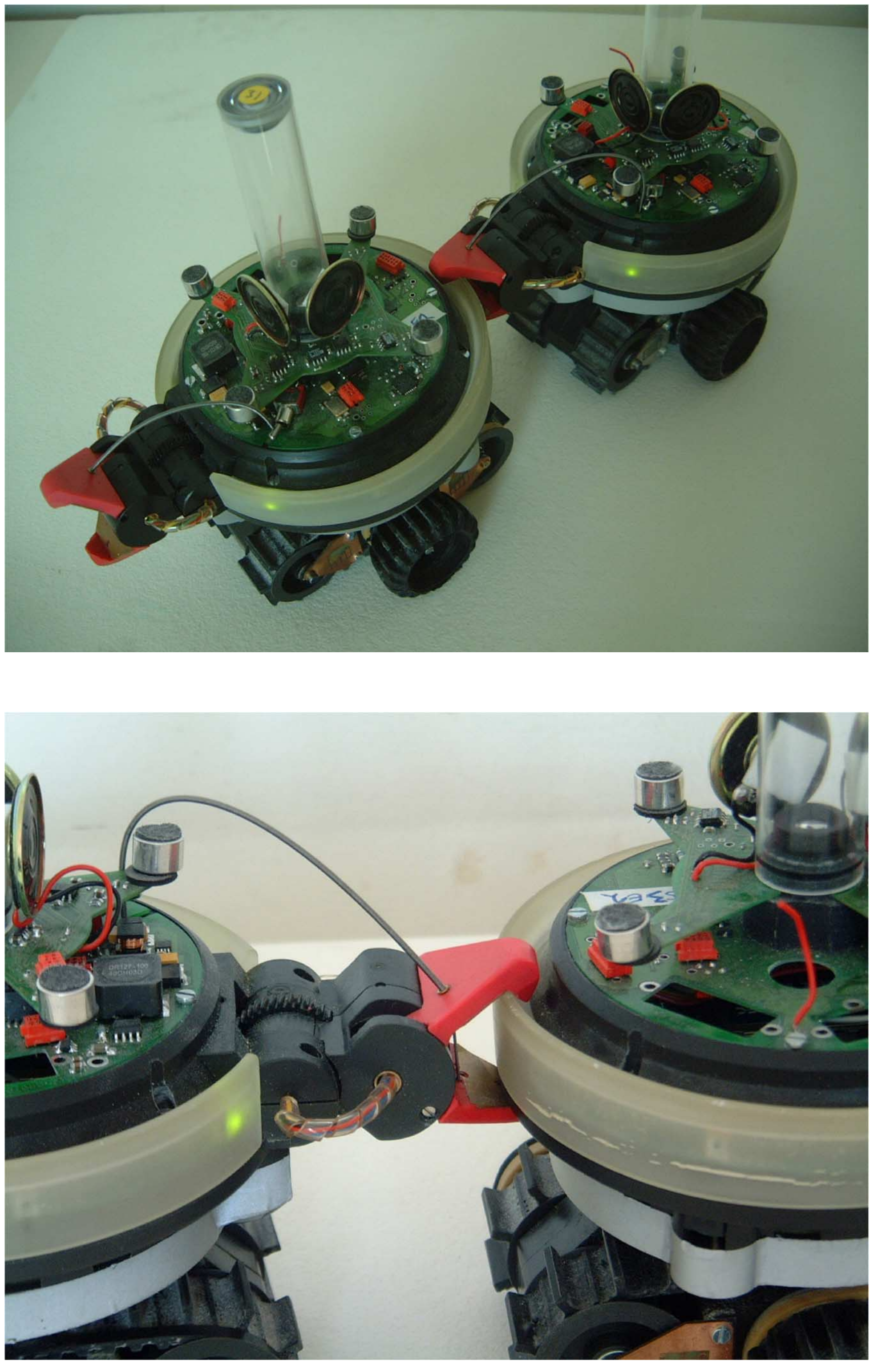


Figure 3. (a) Above: Two connected *s-bots*. (b) Below: Detailed view of a connection between two *s-bots*.

itself applies to an object grasped. The traction sensor plays an important role in the context of coordinated movement of a group of physically connected *s-bots* – i.e., a *swarm-bot*. In particular, it can be employed to provide an *s-bot* with an indication of the average direction toward which the *swarm-bot* is trying to move. The traction sensor measures the mismatch between the direction in which the *s-bot's* own chassis is trying to move and the direction in which the whole group is trying to move (see Baldassarre et al., 2004a, for details).

IV. THE DESIGN OF THE S-BOT'S CONTROL STRUCTURES

To structure our research, we came up with the following scenario. The scenario consists of a series of tasks to be performed by our robotic system, in order to achieve the following goals: object retrieval and transport (Fig. 4).

> A swarm of up to 35 *s-bots* must find and transport a heavy object from its initial location to a goal location. There are several possible paths from the initial to the goal location and these paths may have different lengths and may require avoiding obstacles (i.e., walls and holes). The weight of the object is such that its transportation requires the coordinate effort of at least n *s-bots*, where $n > 1$ is a parameter.

This scenario acted as yardstick against which we could measure the performance of our system on an ongoing basis. In addition, it directed our controller development. We focused on the design of controllers corresponding to the individual and collective behaviors required by the subtasks of the scenario.

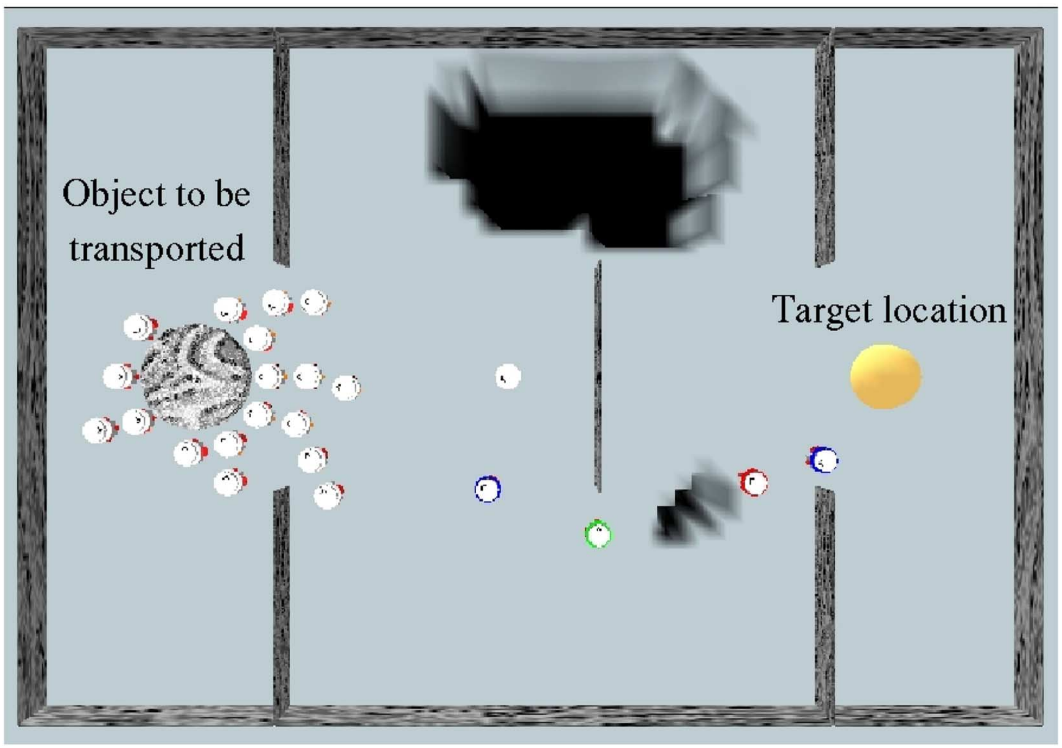
- **Coordinated motion and hole avoidance**. This is the capacity of a group of s-*bots* assembled in a swarm-bot to coordinate their actions to choose a common direction of motion. Such coordination is essential for efficient motion of the *swarm-bot* as a whole. Moreover, if the environment presents holes in which the *swarm-bot* risks remaining trapped, an avoidance action should be cooperatively performed by the *s-bots*, letting the *swarm-bot* take a safer direction of motion.
- **Self-assembly**. This is the capacity of a group of s-*bots* to autonomously connect to and disconnect from each other using their rigid grippers.
- **Cooperative transport**. This is the capacity of a group of *s-bots* to transport a heavy object from its initial location to a target location.
- **Path formation**. This is the capacity of a group of *s-bots* to establish a path between two distant locations in their environment. The path is formed through the creation of a chain of visually linked *s-bots*. The chain of *s-bots* must begin close to the starting location and terminate close to the goal location.

These subtasks are addressed in each of the following subsections, respectively.

Coordinated Motion and Hole Avoidance

The most basic ability for a mobile robotic system is navigating in the environment while avoiding hazards that would hinder its motion. When *s-bots* are physically assembled into a *swarm-bot*, they first need to coordinate their actions in order to choose a common direction of motion. They then have to collectively avoid hazardous areas of the environment, for example holes in which the *swarm-bot* risks getting trapped. We chose

Figure 4. Example of a possible integration scenario: A swarm of up to 35 *s-bots* must transport a heavy object from an initial to a goal location. On the right is the yellow goal location. On the left is the grey object to be transported, surrounded by *s-bots*. A few colored *s-bots* mark a path connecting the grey object to the goal location. Also visible are two types of obstacles: walls and a hole.



the study of hole avoidance because individual *s-bots* suffer severe limitations in the perception of holes. Obstacle avoidance, in contrast, is more easily performed by a single *s-bot* using its infrared proximity sensors. The position of the ground sensors prevents the detection of holes that are sidelong with respect to the direction of motion. The *swarm-bot* can overcome the limitations of single *s-bots* and perform hole avoidance by exploiting its larger physical structure and making use of cooperation among its constituent *s-bots*. Due to the complex dynamics that characterize the movement of *s-bots* connected in a *swarm-bot*, it is difficult to handcraft efficient controllers. For this reason, we make use of artificial evolution to synthesize neural controllers for the *s-bot* (see also Trianni and Dorigo, 2006). In the following, we detail the experimental setup and the results we obtained.

Experimental Setup

In this section we present experiments on the evolution of coordinated motion and hole avoidance behaviors. Additionally, we describe the development of communication modalities appropriate to the task. For the latter purpose, we compare three different approaches for communication among the *s-bots*. In the first setup – referred to as *Direct Interactions* setup (*DI*) – the *s-bots* communicate only through pulling/pushing forces that they exert on each other, as perceived by their traction sensor. The second and third setup also make use of sound signals for communication. In the second setup – referred to as *Direct Communication* setup (*DC*) – the *s-bots* emit a tone as a handcrafted reflex action to the perception of a hole. In the third setup – referred to as *Evolved Communication* setup (*EC*) – the signaling behavior is not defined a *priori*, but it is evolved.

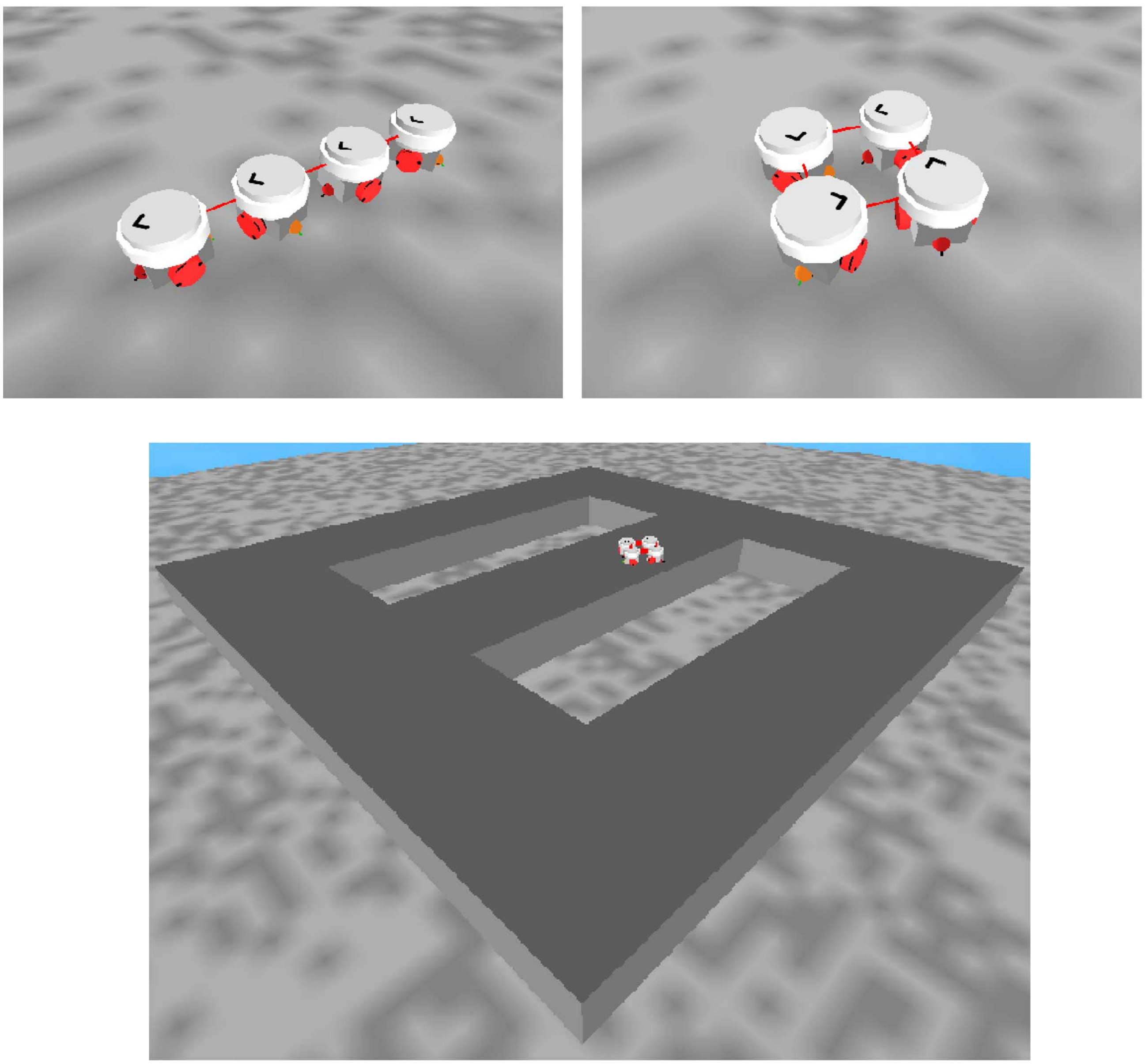
Our goal is to evolve coordinated motion and hole avoidance behaviors for a group of four *s-bots*. To this end, we test the *swarm-bot* in environments both with and without holes (see Fig. 5). The evolutionary process is performed using the **Swarmbot3d** simulator. The best evolved controllers are subsequently downloaded and tested on the real robots. The *s-bots* are controlled by identical feed-forward neural networks. In the basic *DI* setup, the traction and the ground sensors are used as inputs, while the two outputs directly control the left and the right wheel. In the *DC* and *EC* setups, additional binary inputs encode the information perceived by the microphones. The activation of the loudspeaker has been handcrafted in the *DC* setup in such a way that an *s-bot* emits a sound signal whenever one of its ground sensors detects the presence of a hole. Here, evolution is responsible for shaping the correct reaction to the perceived signals. By contrast, in the *EC* setup the sound signal is controlled by an additional output added to the neural network. Therefore, in this setup, evolution is responsible for shaping not only the response to the emission of a signal, but also the signaling behavior – i.e., the complete communication paradigm.

The weights of the synaptic connections are genetically encoded parameters. The simple evolutionary algorithm used in these experiments exploits a fitness function that rewards straight and fast motion of the *s-bots*, and penalizes those groups of *s-bots* that do not coordinate their movements or that spend too much time in the vicinity of a hole. This last component is computed simply looking at the activation of the traction and the ground sensors. Additionally, if the behavior results in the *swarm-bot* falling into a hole, the corresponding genotype is penalized. The fitness assigned to a genotype is the average performance measured over different trials. We have defined three different conditions for the evolution of coordinated motion and hole avoidance (see Fig. 5). During evolution, the *swarm-bot* is initialized to one of these different conditions for 4 trials, thus obtaining 12 trials in total per genotype (for more details, see Trianni and Dorigo, 2006).

Results

For all setups – *DI*, *DC* and *EC* – the evolutionary experiments were replicated 10 times, so that 30 evolutionary runs have been performed in total. All evolutionary runs were successful, each achieving a high level of performance. Observing the behavior produced by the evolved controllers, we note that the coordination phase is largely consistent between the controllers evolved in the three different setups. At the beginning of the trial, the *s-bots* move in the direction they are initially positioned. Within a few simulation cycles, the physical connections transform this disordered motion into traction forces that are exploited to coordinate the group. When an *s-bot* feels a traction

Figure 5. Experimental conditions in which the *swarm-bot* is evolved. In conditions “a” and “b” (above left and right), a *swarm-bot* is initialized on a flat terrain and has to perform coordinated motion. The *swarm-bot’s* shape is either a line or a square. In condition “c” (below), a square *swarm-bot* is positioned in an arena with open borders and holes.



force, it rotates its chassis in order to reduce this force. Once the chassis of all the *s-bots* are oriented in the same direction, the traction forces disappear and the coordinated motion of the *swarm-bot* starts (see also Baldassarre et al., 2004b; Trianni and Dorigo, 2006).

In contrast, the hole avoidance behavior is significantly influenced by the communication abilities of the *s-bots*. Therefore, various differences can be observed between the three experimental setups. In the *DI* setup, *s-bots* can rely only on direct interactions in the form of traction forces in order to communicate the presence of a hole. As a consequence, the avoidance action of the group can be triggered only if the force produced by the *s-bot* that first perceives the hole is high enough. A faster reaction to the detection of a hole seems to be achieved in the *DC* and *EC* setup, in which *s-bots* have the possibility to exploit direct communications in the form of sound signals. In the *DC* setup, the perception of the signal generally activates the rotation on the spot of the chassis of all the *s-bots* except for the one that perceives the hole. The latter tries to move away from the arena border and, in doing so, it does not encounter much resistance from the others. It continues to move away from the hole until it ends up not detecting the hole any more. At this point, the signaling ceases and the group reorganizes, moving in a new direction. The situation is more complex for the *EC* setup. In fact, evolution produced a variety of different responses to the sound signal, all well adapted to the hole-avoidance task. In this case too, sound is interpreted as an alarm signal that speeds up the response of the *swarm-bot* to the perception of a hole. However, the evolved communication and behavioral strategies are characterized by various mechanisms for the inhibition of sound signaling. These mechanisms help in reducing the influence that a continuous tone has on the *s-bots* that perceive it, thus contributing to achieve a fast and reliable reaction to the perception of the hole.

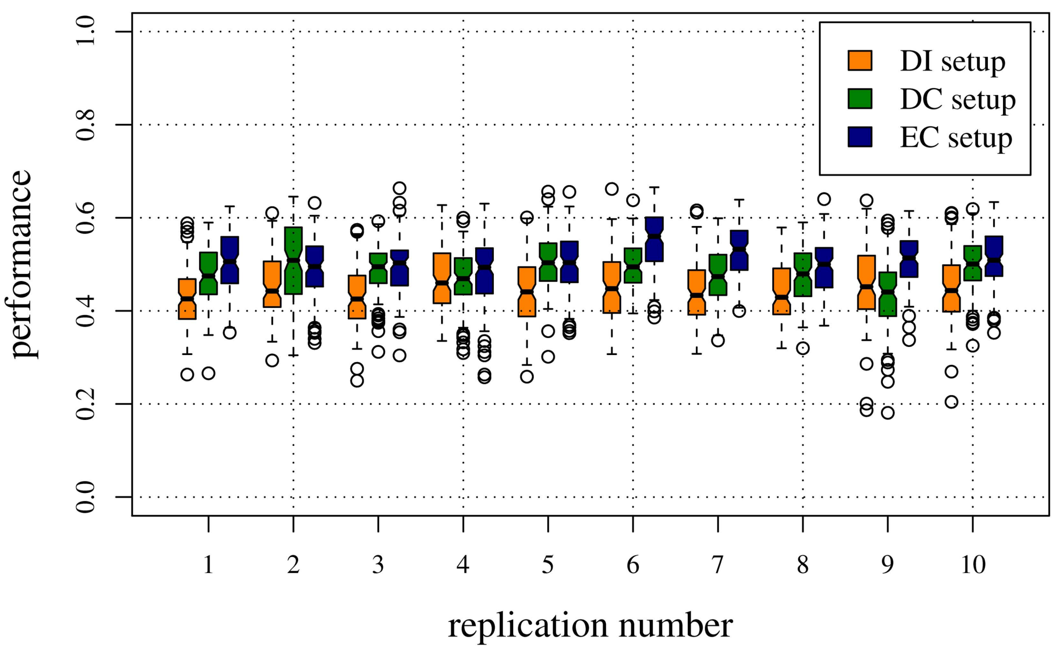
The qualitative analysis suggests that the use of direct communication results in a faster reaction to the detection of a hole and therefore in a more efficient avoidance behavior. In addition, the evolved communication strategy appears more adaptive than the handcrafted solution. This intuition is confirmed by a statistical analysis, performed by re-evaluating all the best genotypes synthesized in the different evolutionary runs (see Fig. 6). This analysis revealed with 99% confidence that the *EC* setup is indeed better than both the *DC* and the *DI* setups. Moreover, the *DC* setup also outperforms the *DI* setup, confirming that direct communication is beneficial for the hole-avoidance task (see Trianni and Dorigo, 2006, for more details).

The controllers evolved in simulation prove robust enough to be tested on real robots. We chose to test a single controller per setup in order to compare the performance between simulation and reality. To do this, we used a performance metric that corresponds to the distance covered by the *swarm-bot* along its trajectory. This measure allowed us to compare the results obtained in simulation and on the real robots – it was not possible to use the original fitness function due to the high levels of noise present in the real *s-bot* sensors.

The *swarm-bot* was put in a small square arena, its side measuring 180 cm. This arena was actually built, making the comparison between simulation and reality possible (see Fig. 7a). On the basis of the obtained results, a controller has been chosen to represent each setup: the controllers evolved in the 9th, 6th, and 10th evolutionary runs, respectively, for the *DI*, *DC* and *EC* setup. Each selected controller was downloaded onto the real *s-bots* and evaluated in 30 trials, always starting with a different random initialization. The obtained data were used to compute the performance of the system.

Qualitatively, the behavior produced by the evolved controllers tested on the physical *s-bots* is very good and closely corresponds to that observed in simulation (see Fig. 7). *S-bots* coordinate more slowly in reality than in simulation, taking a few seconds to

Figure 6. Post-evaluation analysis of the best controller produced by all evolutionary runs of the three different setups. Boxes represent the inter-quartile range of the data, while the horizontal lines inside the boxes mark the median values. The whiskers extend to the most extreme data points within the inter-quartile range from the box. The empty circles mark the outliers. Note that the *EC* setup tends to outperform the other two setups. This is confirmed by the statistical analysis performed on these data.



agree in a common direction of motion. Hole avoidance is also performed with the same modalities observed in simulation. From a quantitative point of view, it is possible to recognize some differences between simulation and reality, as shown in Fig. 8. We compare the performance recorded in 100 trials in simulation with that obtained from the 30 trials performed on the real *s-bots*. Generally, we observe a decrease in the maximum performance, mainly due to a slower coordination among the *s-bots*. This means that physical *s-bots* start moving in coordination later than the simulated ones, both at the beginning of a trial and after the perception of a hole. This influences the performance, as the *swarm-bot* cannot cover high distances until coordination among the *s-bots* is achieved. Looking at Fig. 8, we notice that the performance of the *DI* controller is better

Figure 7. (a) Above: The square arena used for the comparison between simulation and physical *s-bots*. (b) Below: A physical *swarm-bot* performing hole avoidance. Note how physical connections among the *s-bots* can serve as support when a robot is suspended out of the arena, still allowing the whole system to work. Even in the difficult situation above, the *swarm-bot* still manages to avoid falling.

.

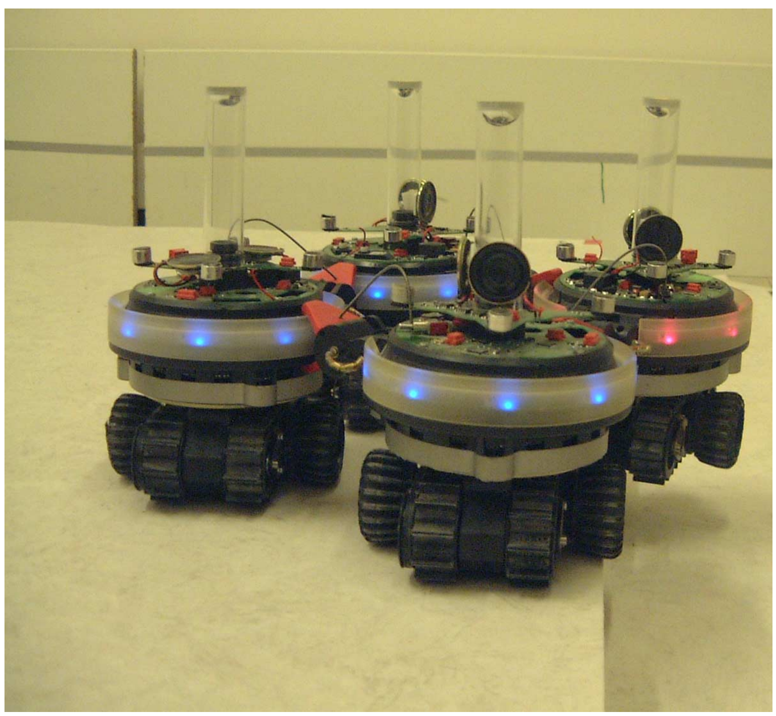
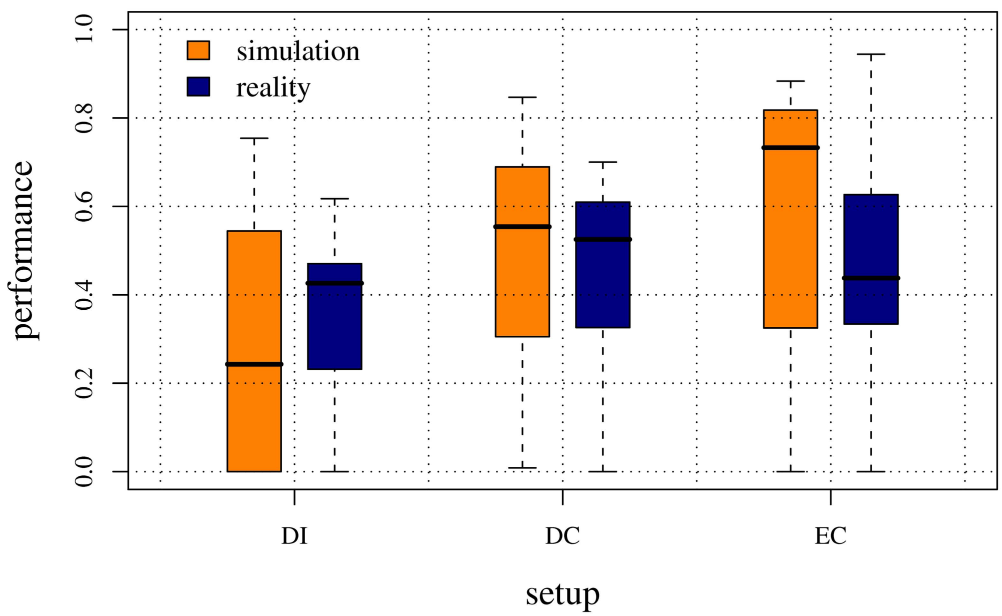


Figure 8. Comparison of the performance produced in the different settings by the selected controllers tested both in simulation and reality. For an explanation of the plot, see Fig. 6.



in reality, as a consequence of the high friction provided by the tracks of the real *s-bots*, which enhances the effect of the direct interactions among the *s-bots*. For the *DC* controller, the performance difference between simulation and reality is minimal. In this case, we observed some communication failure, whose negative effects were compensated by the higher force transmitted from one *s-bot* to the other due to the high friction of the treels system. Finally, the best controller of the *EC* setup does not perform as well on the real *s-bots* as it does in simulation. *S-bots* are always able to coordinate and to perform coordinated motion and hole avoidance. However, we observe here that *s-bots* are slower in avoiding holes due mainly to some failures in the communication system, which is necessary to trigger and support the avoidance action. For this reason, quantitatively the performance decreases. However, the behavior is by in large good and corresponds closely to that observed in simulation from a qualitative point of view.

Self-Assembly

Probably the most characteristic capacity of the *swarm-bot* system is that it can self-assemble, that is, move from a situation characterized by the independent activity of a number $n > 1$ of *s-bots* to a situation in which these n *s-bots* physically connect to each other to form a *swarm-bot*. We used similar evolutionary computation techniques to those described above in the coordinated motion experiments. To develop controllers capable of letting *s-bots* self-assemble we used an artificial neural network whose connection weights were evolved using an evolutionary algorithm (Groß and Dorigo, 2004; Groß et al., 2006a). Below, we start by describing the control structure we designed to allow the *s-bots* to self-assemble. We go on to illustrate the results of an experiment in which we tested the performance of the resulting self-assembly behavior.

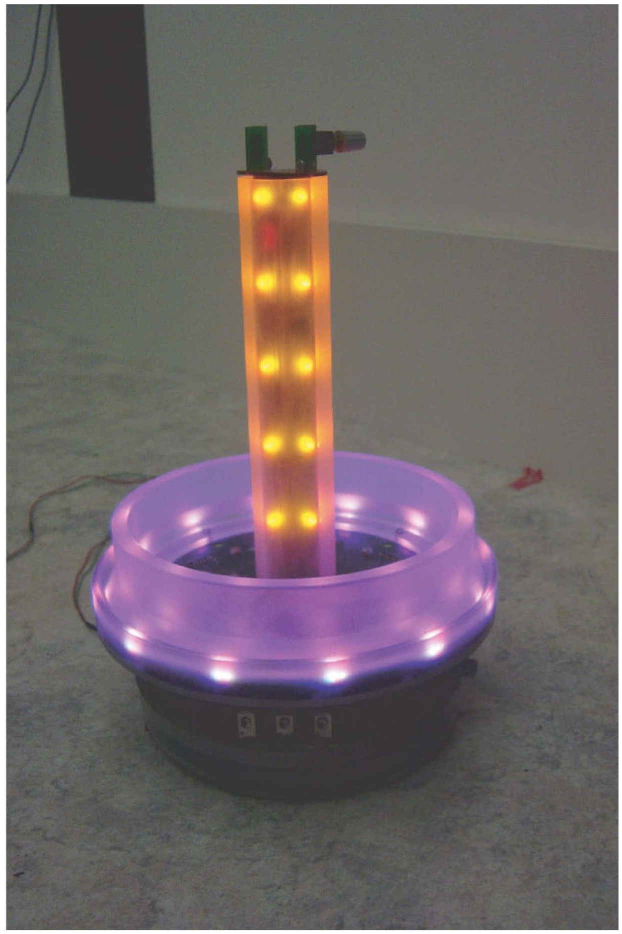
The Control Structure

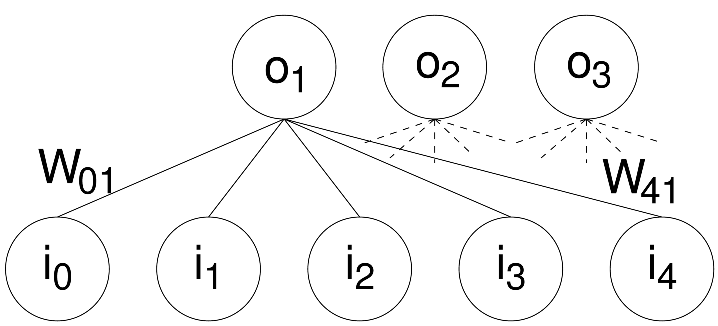
Algorithm I – The assembly module

```
1  activate color ring in blue
2  repeat
3    (i1, i2) ← feature extraction (camera)
4    (i3, i4) ← sensor readings (proximity)
5    (o1,o2, o3) ← neural network (i1, i2, i3, i4)
6
7    if (o3 > 0.5) ^ (grasping requirements fulfilled)
8        then
9            close gripper
10           if (successfully connected)
11               then
12                   activate color ring in red
13                   halt until timeout reached
14               else
15                   open gripper
16           fi
17   fi
18   apply (o1,o2) to traction system
19 until timeout reached
```

Algorithm I, called "assembly module," controls the self-assembly behavior of the *s-bots*. It implements a set of mechanisms designed to allow an *s-bot* to connect to another *s-bot* or to the *s-toy* (see Fig. 9a). In a group of disconnected *s-bots*, the process of self-assembly is triggered by the perception of colors. In fact, the assembly module allows an *s-bot* to move towards the nearest red object and avoid collisions with blue objects. If an *s-bot* manages to successfully connect to a red object, it changes the color ring from blue to red. In so doing, it becomes itself an object with which other *s-bots* seek to establish a connection. At the heart of the assembly module is a feed-forward artificial neural network – a single-layer perceptron – along with some hand-designed code to pre-process sensory input and to make sure that the output of the network is correctly "interpreted" by the *s-bots'* actuators. The parameters of the neural network – i.e., the connection weights – have been determined in simulation by using evolutionary algorithms (Groß and Dorigo, 2004). As illustrated in Fig. 9b, the neural network of the assembly module has four input nodes i_1, i_2, i_3, and i_4, a bias i_0, three output nodes o_1, o_2, and o_3, and 15 connection weights (w_{ij}). At each cycle, the network takes as input the *s-bot's* sensor readings. The input neuron i_1 and i_2 are set by extracting and preprocessing data from the *s-bot's* vision system (Algorithm I, line 3). In particular, the feature extraction algorithm first checks whether any red or blue colored object is perceived within a limited perceptual range bounded to the left and right side of the *s-bot's* heading. Subsequently, the algorithm assigns a value to the input $i_1 \in \{0,1\}$ and $i_2 \in \{0,1\}$ according to the rules detailed in (Groß et al., 2006a). The input variable $i_3 \in [0,1]$ and $i_4 \in [0,1]$ are set by taking the reading of the front-left-side and front-right-side proximity sensors (Algorithm I, line 4).

Figure 9. (a) Above: The *s-toy*. It can be used either as an object to be retrieved or as a landmark. The overall weight of the *s-toy* can easily be changed in the range of 700 to 3000 grams. The *s-toy* has the same external ring as the *s-bots*, so that *swarm-bots* can connect to it. Its ring can change color in the same way as in the *s-bots* (red, green, blue, and various combinations). The central turret (which can be removed) has two different color LEDs (green and red). The external diameter is 20 cm, the height 30 is cm. The *s-toy* can also emit sounds that might be used by *s-bots* for localization. (b) Lower-left: A graphical representation of the feed-forward two-layer artificial neural network of the assembly module. i_1, i_2, i_3, and i_4 are the nodes which receive input from the *s-bots* sensors. i_0 is the bias term. o_1, o_2, and o_3 are the output nodes. (c) Lower-right: The equations used to compute the network output values.





$$\forall j \in \{1,2,3\}:$$
$$o_j = \frac{1}{1+e^{-x_j}},$$
$$x_j = \sum_{k=0}^{4} \omega_{kj} i_k$$

The network has three outputs $o_1 \in [0,1]$, $o_2 \in [0,1]$, and $o_3 \in [0,1]$. The output neuron o_1 and o_2 set the angular speed of the left and right *s-bot's* wheels. The values of the speed vector (o_1, o_2) are linearly scaled within the range defined by the *s-bot* speed limits. The output neuron o_3 is used to control the status of the gripper. In particular, the gripper is closed (a) if the output neuron $o_3 > 0.5$, (b) if a red object is detected by the camera, and (c) if the gripper optical light barrier detects an object between the lower and the upper teeth of the gripper. While closing the teeth, the gripper is slightly moved up and down several times to facilitate a tight connection. Failures of the grasping procedure can be detected by monitoring the aperture of the grasping device. In case of failure the gripper is opened again and the assembly procedure restarts from the beginning. If a red object is successfully gripped, then the *s-bot* sets the colour of its ring to red and stops.

Experimental Results

We tested the effectiveness of the assembly module algorithm in a task that requires six *s-bots* to connect directly, or indirectly via a chain of *s-bots*, to the *s-toy*. The experiment was repeated on three types of terrain: standard flat terrain, moderately rough terrain, and very rough terrain.

At the beginning of each trial, the *s-bots* were placed in arbitrary positions inside a circle of radius 70 cm around the *s-toy*. To encourage interactions among the *s-bots*, we limited their initial position to a 90 degrees segment of the circle. The same density could be obtained by putting a swarm of 24 *s-bots* inside a full circle of the same radius. The *s-bots* were positioned in a way that ensured a minimum distance of 20cm between the centers of any two objects. This allowed all *s-bots* to turn on the spot with no collision of their gripper elements. Fig. 10 shows the initial and the final configuration of one of the trials. Fig. 11a shows a bar-plot of the 34 trials performed. The gray value of each bar indicates the number of *s-bots* that could successfully connect within the time frame. The height of the bar represents the number of elapsed seconds until the last *s-bot* connected. In total, an *s-bot* succeeded in establishing a connection 199 times. In only five cases an *s-bot* failed. In 30 out of 34 trials, all seven objects (i.e., the six *s-bots* and the *s-toy*) were physically connected; on average this took 96.4 seconds.

To the best of our knowledge, this is the first study in which six autonomous robots manage to successfully connect to an object and/or to each other. Moreover, the procedure was shown to be scalable, as it works for increasing numbers of *s-bots* (experiments with up to 16 physical *s-bots* were run successfully (Groß et al., 2006a)), and robust, as it can control self-assembling *s-bots* moving on both flat and rough terrain. Fig. 11b summarizes the self-assembly results obtained for the three types of terrain that we considered. Overall, the performance of the algorithm, which was developed for flat terrain conditions, is not affected by the fact that the *s-bots* move on a terrain of moderate roughness. In fact, for both the flat and moderately rough terrains, a single *s-bot* connected in 100 percent of the cases, while in the case of six *s-bots*, the connection rate was 98%. Even when moving on the very rough terrain, a single *s-bot* connected to the *s-toy* in 95% of the cases, while when part of a group of size six, a single *s-bot* connected still in 91% of the cases.

Cooperative Transport

Cooperative transport refers to the capability of a group of *s-bots* to transport a heavy object from its initial location to a target zone. Our goal was the design and implementation of control algorithms that allow a group of *s-bots* to perform a task that requires them to pull and/or push cooperatively a heavy object – the *s-toy*, see Fig. 9a – towards

Figure 10. (a) Above: Six *s-bots* at the start of a self-assembly experiment on flat terrain. (b) Below: Two self-assembled *swarm-bots*, each comprised of three *s-bots*, connected to the *s-toy*.

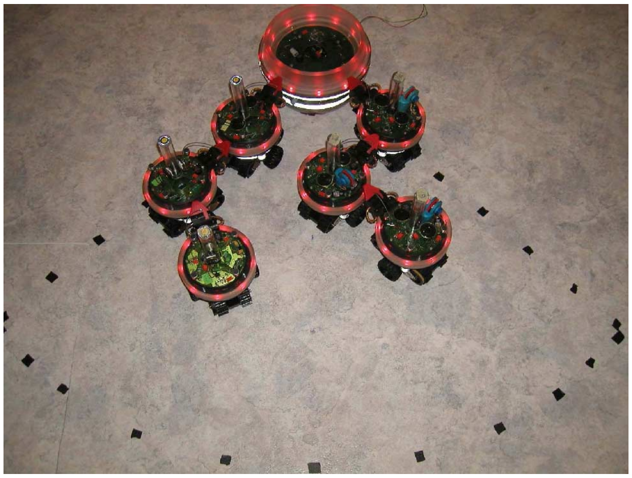


Figure 11. (a) Above: Self-assembly of six *s-bots* with an *s-toy* (34 repetitions). (b) Below: Self-assembly on different types of terrain: percentage of successful connections (from the left box to the right box: 160, 204, 40, 120, 40, and 120 observations). The percentage of connected *s-bots* is 91% for the most challenging setup (6 *s-bots*, very rough terrain).

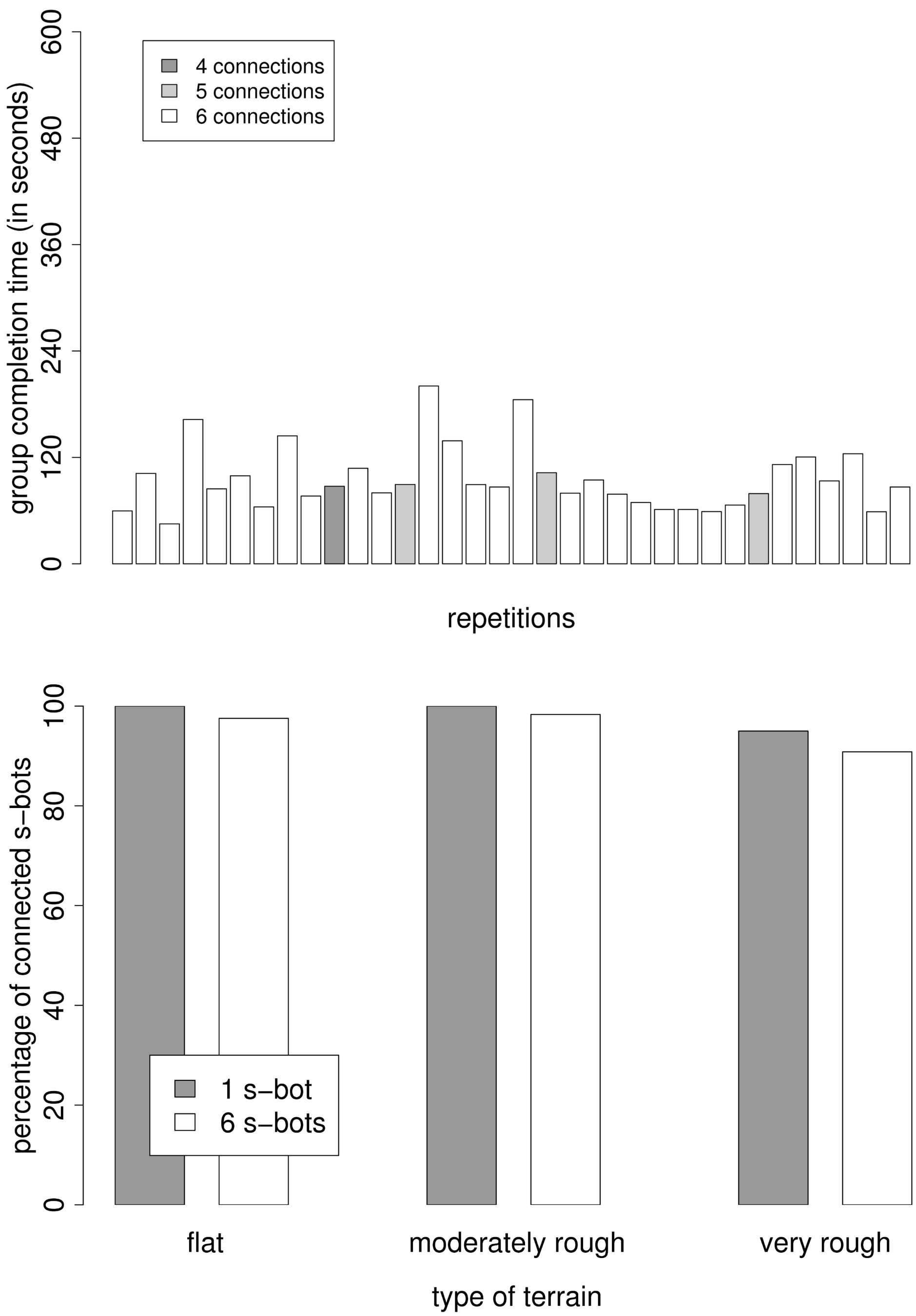
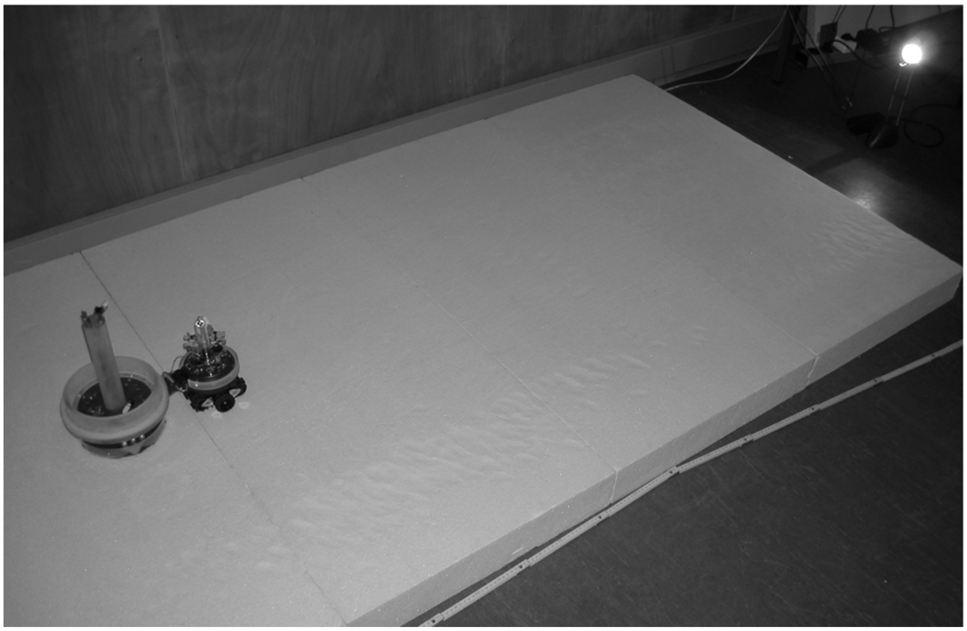


Figure 12. Experimental setup: (a) Above: *s-toy* with one *s-bot* attached (bottom-left) and light beacon (top-right); (b) Below: Example of spatial arrangement of two *s-bots* and the *s-toy*.



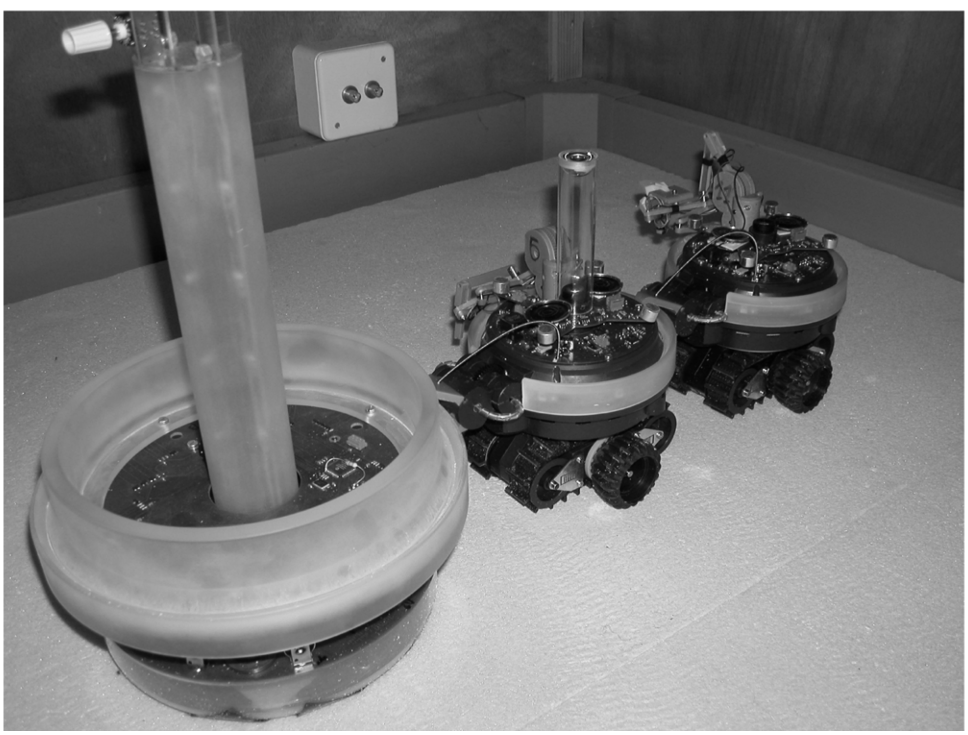


Table 1. Friction coefficients for terrains T_0 and T_1.

	s-toy	*s-bot* (lateral)	*s-bot* (longitudinal)
Terrain T_0	0.46	0.57	0.58
Terrain T_1	0.41	1.30	1.80

a target location indicated by a light beacon. Cooperation is required as the *s-toy* is too heavy to be transported efficiently by a single *s-bot*. In our experimental setup, the *s-toy* has a mass of 813 g and is initially put in a fixed location, at a distance of 250 cm from the light beacon. At the start of the experiment, the *s-bots* are attached to the *s-toy*, either directly or via other *s-bots* if they are in a *swarm-bot* formation (see Fig. 12). Their task is to transport the *s-toy* as close as possible to the target location within a fixed time period of 15 seconds. To test the effectiveness of our multi-robot system we tested groups of 1 to 3 *s-bots* attached to the *s-toy* with different spatial arrangements (see Groß et al., 2006c, for more details).

In our experiments, we examine the performance of the system on two different types of terrains (here referred to as T_0 and T_1, respectively). Both terrains are flat, the friction coefficients are listed in Table 1. The force necessary to move the *s-toy* on Terrain T_0 is similar to the force required on Terrain T_1. For Terrain T_0 the magnitude of friction between the tracks and the ground is moderate. For Terrain T_1 there is so much friction between the tracks and the ground that if a strong lateral force is applied to the *s-bot* it will either topple down or it will be displaced by a sequence of irregular movements. Terrain T_1 is a very difficult test-bed, since a group of *s-bots* connected to each other and/or the *s-toy* might easily get stuck, if they do not coordinate their movements properly.

The Control Structure

Algorithm II – The transport module

```
repeat
  α ← compute target direction (camera)
  if (stagnation)
     then
        execute recovery move
     else
        if (risk of stagnation)
           then
              hard alignment (α)
           else
              soft alignment (α) and forward motion
        fi
fi
until timeout reached
```

Algorithm II describes the transport module which allows a connected *s-bot* (1) to align its chassis towards the light beacon indicating the target-zone, and (2) to apply pushing/pulling forces in order to move the *s-toy* towards the target.

During the transport, the *s-bot* monitors the magnitude of the torque acting on its traction system and on the turret. If the torque reading values exceed a certain threshold, there is *stagnation*. In this case, a short recovery move is performed to prevent the hardware from being damaged.

The transport module uses the camera vision system to detect the direction of the light source with respect to the *s-bot's* heading. [Note: At the time we carried out the first experiments, the *s-bot* camera device driver was not yet available. Instead, the proximity sensors were used (in ambient light mode) to detect the target direction.] By adjusting the orientation of the chassis with respect to the *s-bot's* heading (i.e., the orientation of the turret) the controller sets the direction of motion. The realignment of the chassis is supported by the motion of the traction system. We implemented two different types of realignment referred to as "hard" and "soft" alignment. The hard alignment makes the *s-bot* turn on the spot. The soft alignment makes the *s-bot* turn while moving forward with maximum speed. The hard alignment is executed if there is risk of stagnation. This is the case, for instance, if the angular mismatch between the current and the desired orientation of the chassis exceeds a certain threshold.

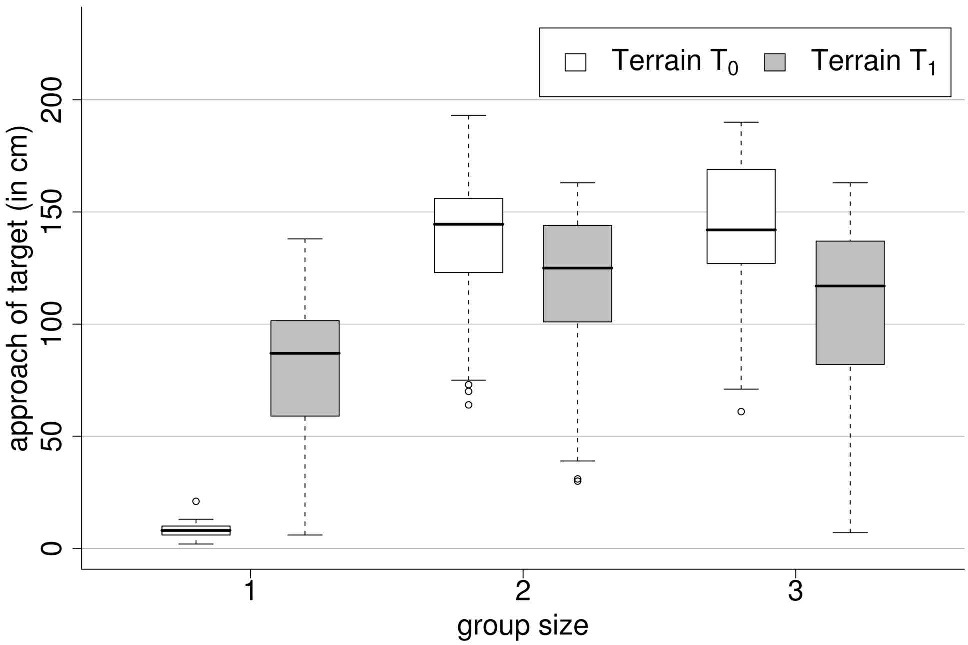
Experimental Results

To assess the performance of the physical *s-bots* on Terrain T_0 and T_1, in total more than 500 trials have been performed. The performance metric we used is the distance by which the *s-toy* approaches the light beacon within the time period of 15 seconds, that is, the difference between the initial and the final distance between the *s-toy* and the light beacon.

The distance an *s-bot* can cover on Terrain A or B during the time frame of 15 seconds is about 232 cm. On Terrain T_0, an *s-bot* attached to an *s-toy* can pull it for about 8 cm by moving backwards, while a chain of two *s-bots* can pull the *s-toy* for about 210 cm. Since a group cannot transport the *s-toy* faster than a single *s-bot* can move, two *s-bots* are sufficient for reaching almost optimal performance in this case (more than 91% of the maximum speed of a single *s-bot*). Fig. 13 plots the distance (in cm) by which the *s-toy* approached the light beacon. The white boxes refer to the transport performance of groups of 1 to 3 *s-bots* on Terrain T_0. In all trials, one *s-bot* alone was nearly incapable of moving the *s-toy*. On the contrary, two and three *s-bots* have transported the *s-toy* during each of the 90 trials for more than 60 cm. The average distance (in cm) the *s-toy* was moved by a group of 1, 2, and 3 *s-bots* is respectively 8.1, 135.9, and 143.0. This is respectively 3.5%, 58.6%, and 61.6% of the upper bound (i.e., 232 cm). The gray boxes in Fig. 13 refer to the transport performance of groups of 1 to 3 *s-bots* on Terrain T_1. Due to the better grip the traction system has on Terrain T_1, a single *s-bot* itself is already capable of transporting the *s-toy*. Nevertheless, for the group sizes 2 and 3 the system performs significantly better on Terrain T_0 (Mann-Whitney test, 0.05 significance level) – even though the magnitude of the force necessary to move the *s-toy* is slightly bigger than for Terrain T_1 (see Table 1). The average distance (in cm) the *s-toy* was moved by a group of 1, 2, and 3 *s-bots* is respectively 78.5, 117.3, and 107.9. This is respectively 33.9%, 50.6%, and 46.5% of the upper bound.

As discussed previously, the task can be solved near optimally by two *s-bots*. For Terrain T_0, the performance for group size 3 is better, but not significantly better, than the performance for group size 1 or 2. On the contrary, for Terrain T_1, the performance is best for group size 2 (Mann-Whitney test, 0.05 significance level). Moreover, we recognized that the spatial arrangement of the *s-bots* affects the performance of the group (Groß et al., 2006c). In those arrangements of three *s-bots* that are symmetric with respect to the light beacon, the *lowest* transport distance observed over all trials on terrain T_0 (T_1) was still 67% (54%) of the upper bound.

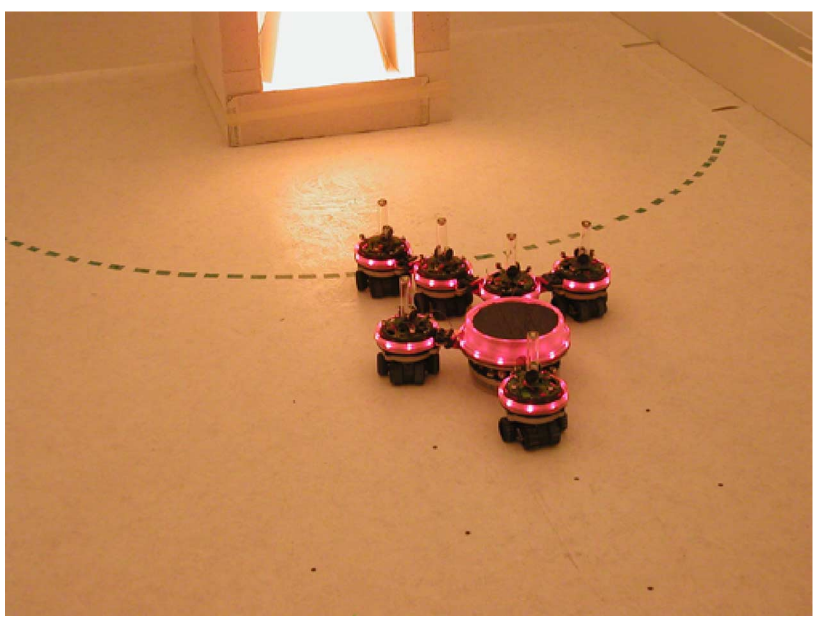
Figure 13. Transport performance of 1 to 3 physical *s-bots* on terrain T_0 and T_1. Observations per box (from the left to the right): 42, 75, 90, 120, 105, and 105.



We also studied the situation in which some robots of a group are able to locate the transport target, while the others, called *blind s-bots*, are not (Groß et al., 2006b; Groß and Dorigo, 2004). To enable a blind *s-bot* to contribute to the group's performance, it used sensors to perceive whether or not it was moving, and to detect the traction forces acting between its turret and its chassis. For group sizes ranging from 2 to 16, it was shown, in simulation, that blind *s-bots* make an essential contribution to the group's performance. This capability was validated on the real system, with groups of 2 to 6 *s-bots* (see Fig. 14a). The same controllers also proved successful at transporting the *s-toy* over various types of rough terrain. Furthermore, the controllers enabled the *swarm-bot* to navigate over a terrain with holes in it.

Finally, we carried out an experiment having the *s-bots* start separately, from random positions in the environment. The *s-bots* had to assemble with the *s-toy* and with each other, prior to transportation (see Fig.14b). The *s-toy* required the cooperative effort of four *s-bots* to be moved. The number of *swarm-bots* involved in the transport, their global shape or size and their internal structure were not pre-determined, but resulted from a self-organized process in which the *s-bots* autonomously grasped each other and/or the *s-toy*. Apart from a few cases, in which not all *s-bots* correctly assembled, the transport speed was more than half the maximum speed of a single *s-bot* without any load (Tuci et al., 2006; Groß et al., 2006c).

Figure 14. Transport of the *s-toy* by (a) (above) six manually connected *s-bots*, four of which are not capable of locating the target location, (b) (below) six *s-bots*, four of which have formed a *swarm-bot*.



Exploration and Path Formation

Environment exploration, navigation, and path formation are a prerequisite for the accomplishment of a wide range of tasks. In the context of the scenario as shown in Fig. 4, the robots have to form a path between the object to be transported and the goal location. In designing our controllers, we avoid complex navigation strategies, as they do not naturally scale with respect to the number of robots, and require careful engineering of the controller in order to deal with the difficulties related to dynamic environments and multiple robots.

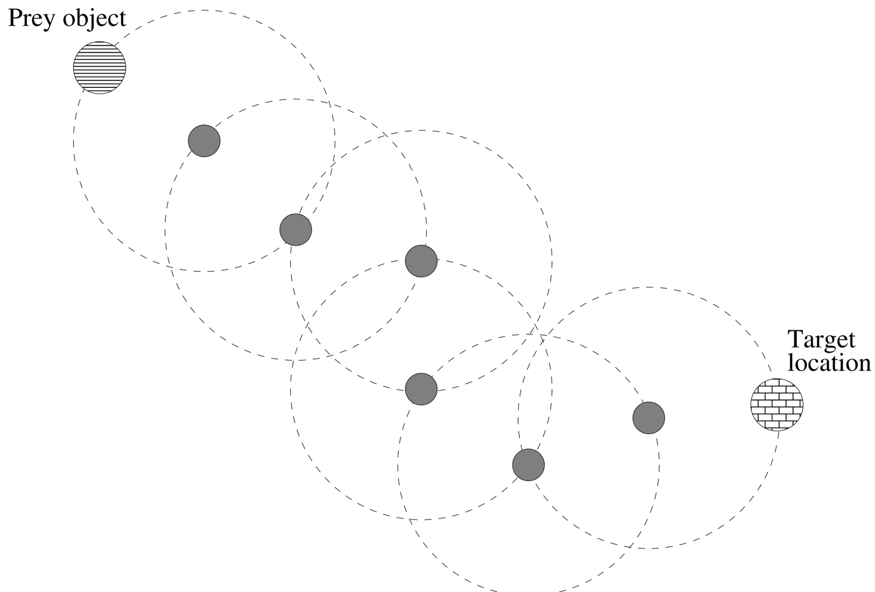
The approach we have followed in our research is inspired by the path formation behavior of ants. Ants deposit pheromones on the ground while walking and this gives raise to paths shared at the colony level. As our *s-bots* cannot deposit pheromones, they build visual paths as follows. They start from the target location identified by a blue *s-toy* (see Fig. 9a) and randomly explore the space around it. When they reach a certain distance (given by a parameter) from the blue *s-toy* they become beacons of the forming visual path. This means they stop moving and turn on their light. Other *s-bots* continue the random search around the beacon and can become beacons themselves extending in this way the visual path (see Fig. 15a). The direction of growth of the visual path is therefore random and is not guaranteed to reach the object to be retrieved. However, visual paths under formation have some probability of dissolving (given by another parameter of the procedure) and therefore unsuccessful searches (that is, incomplete visual paths that do not reach the object to be retrieved) can restart until a complete visual path is constructed. Once this stochastic procedure finds a visual path connecting the target location to the object to be retrieved, the visual path can be exploited by the other *s-bots* to reach the *s-toy* and then to retrieve it (see Fig. 15b). [Note: Video recordings of these experiments can be found at http://www.swarm-bots.org/chain-formation.html.] The main advantage of this exploration strategy is that it relies on local information and simple rules and does not require the *s-bots* to create a map-like representation of the world (more details can be found in Nouyan et al., 2006; Nouyan and Dorigo, 2006).

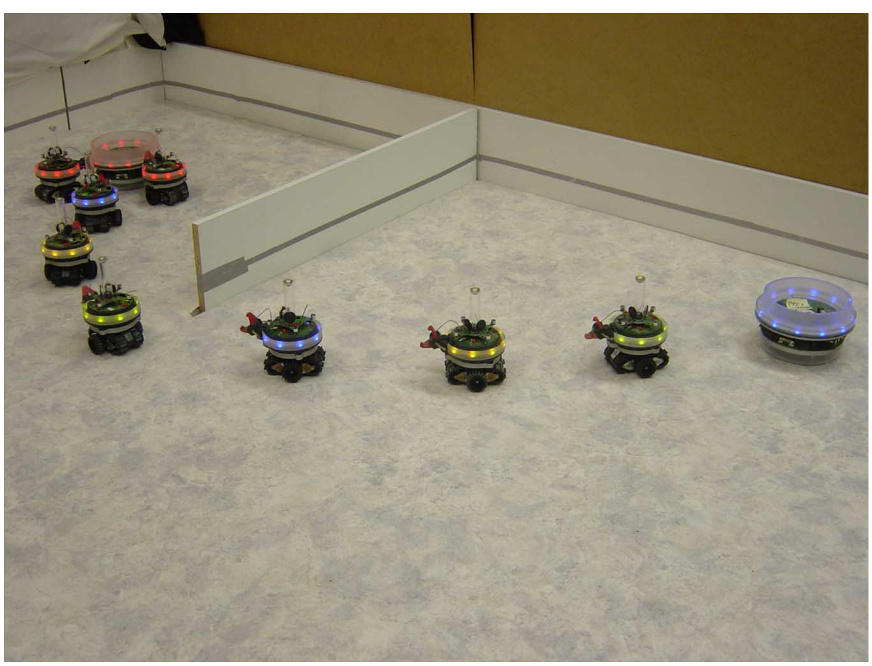
To implement the exploration and path formation strategy we have employed a behavior-based approach. We have shown that by varying parameters of the *s-bots* controller it is possible to generate a variety of exploration strategies. Different strategies are better adapted to specific environments. In particular, we have implemented two strategies. In the first one, we have static visual paths: the *s-bots* beacons do not move. In the second setup, the *s-bots* that form a visual path move in a coordinated way without breaking the path. The controllers developed in simulation have been ported successfully onto the real *s-bots*. The time required to build a chain is a function of the complexity of the environment and in particular depends on the presence, or absence, of obstacles.

V. DISCUSSION

When given the task of building a robotic system, the main decisions to be taken by the researchers concern the architecture of the hardware and of the control system. In this chapter, we have presented the results of the SWARM-BOTS project. The SWARM-BOTS project fundamentally focused on the evaluation of two particular choices in robotic system design. For the hardware, we chose to implement a system comprised of many autonomous robots with a unique ability to attach to (and detach from) one another so as to form bigger, physically connected structures. To control this system,

Figure 15. (a) (Above) Depiction of a chain of six *s-bots* (small grey circles) visually connecting a target location (big circle on the right) to a prey (big circle on the left). The dashed circles represent the visual range of the *s-bots*. (b) (Below) A chain of six *s-bots* connect the prey (represented by the top-left *s-toy* to the target location (represented by the right *s-toy*).





we chose to use only distributed controllers that could only make use of locally available information. These choices were motivated by the desire to make our robotic system – the *swarm-bot* – robust and versatile [Note: By saying that a robot is versatile we mean to say that it is capable of dynamically changing shape and control functionality depending on the situation it faces], as well as allowing it to navigate on rough terrain.

Our research falls between the domains of collective robotics and self-reconfigurable robotics. It is loosely bio-inspired, in the sense that many of our choices and techniques have as inspiration some natural process or biological observation. However, we do not try to replicate faithfully any natural system: we rather take inspiration from natural processes and let these principles guide our engineering choices.

As in collective robotics, we are concerned with the performance of groups of cooperating robots. Our work differs from collective robotics, however, in that we are interested in the study of self-assembling structures and in their exploitation for the solution of problems for which cooperation through physical connection is a necessity.

As in self-reconfigurable robotics, we study robotic structures (i.e., *swarm-bots*) that can change their shape as a function of the task they are performing. Our work differs from self-reconfigurable robotics, however, in that the units composing our self-reconfigurable robot are autonomous units that can perform tasks independently of each other or in cooperation, as required by the particular task considered.

Concerning the expected impact of our research work, it is worth noting that the SWARM-BOTS project aimed at demonstrating that it is possible to build and control a self-assembling and self-organizing multi-robot system rather than to address any specific application. However, if we make an imaginative leap into the future, several potential applications of a mature *swarm-bot* technology can be conceived. A few examples are listed below.

- Navigation in highly constrained and unstructured environments. The *swarm-bots* can form groups to move in complex environments. The shapes of these groups can vary on-line to adapt to the constraints imposed by the environment and to the requirements of the ongoing task performance.
- Formation of bridges, buttresses, and other civil structures in times of emergency. Major flood events can easily tear down bridges, and bringing help to the affected population is often a time critical task. Hundreds of *swarm-bots* could assemble in order to build emergency structures to allow the access of the rescue teams.
- Transportation of objects on rough terrain. Traditional vehicles cannot cope with very rough terrain. Moreover, their size and configuration is fixed. In contrast *swarm-bots* could, for example, be capable of self-organization into moving carpets to transport objects of various sizes and dimensions in an efficient way.
- Performing inspections and repairs in constrained environments such as pipelines, nuclear reactors and sewage systems, is another possible application.
- *Swarm-bots* could help human teams in search and rescue activities to help save human lives in dangerous environments (see Fig. 16). As a proof-of-concept demo, we let nineteen *s-bots* self-assemble to a prone child and pull her for a couple of meters. A video-recording of the experiment is available at http://www.swarm-bots.org/pulling_a_child.html.

Figure 16. A simulated rescue operation carried out by a group of *s-bots*.

More generally, this type of system can be used for the self-organizing unfolding of structures without any human intervention, from space stations or satellites to scaffolds.

VI. CONCLUSIONS

In this chapter we have illustrated the most important features of a novel robotics concept, called a *swarm-bot*. A *swarm-bot* is a self-organizing, self-assembling artifact composed of a variable number of autonomous units, called *s-bots*. As illustrated in Section III, each *s-bot* is a fully autonomous robot capable of displacement, sensing and acting based on local information. Moreover, the self-assembling ability of the *s-bots* enables a group of them to execute tasks that are beyond the capabilities of the single *s-bot*.

The hardware developed over the course of the SWARM-BOTS project proved versatile and robust to failure. This is attributable to the fact that the system is made up of many autonomous entities that can self-assemble into a single body and disband any time the union is no longer required. Previous robotic systems composed of units capable of reconfiguring themselves are much less versatile. In these other systems the individual units have little or no mobility, very limited sensing capabilities, and are often centrally controlled (see Castano et al., 2000; Fukuda and Ueyama, 1994; Brown et al., 2002; Murata et al., 2002; Yim et al., 2000).

The *s-bots'* controllers developed over the course of the SWARM-BOTS project allowed the *s-bots* to perform a wide repertoire of individual and collective behaviors. We ran experiments in which all the components described in Section IV, coordinated motion, self-assembly, cooperative transport, and path formation were executed by a group of up to 12 *s-bots* in a single integrated experiment (video recordings of these experiments are available on-line at http://www.swarm-bots.org/scenario_12sbots.html). These experiments were very successful and make our work the current state-of-the-art in swarm robotics. The controllers proved robust enough to deal with environmental changes, and their functionality scaled well when increasing the number of participating robots.

The work carried out by the SWARM-BOTS project has revealed yet unexplored research topics, hypotheses and conjectures which need further investigation. Thus, ongoing work is taking place in more or less all the research areas illustrated in Section IV. We are also pursuing ongoing research on the following topics not mentioned yet, but that are of particular importance in swarm robotics:

Adaptive task allocation. Task allocation and division of labor are two areas of research in collective and swarm robotics. Previous studies have shown that small groups of robots might perform a collective task at least as well as a larger group (Schneider-Fontán and Mataric, 1996). However, inherent inefficiency of large robot groups can be avoided if such large groups are equipped with an adaptive task allocation mechanism which distributes the resources of the group based on the nature of the task and the diversity among the individuals of the group. In our research we are interested in designing an adaptive task allocation mechanism which allocates a sufficient number of *s-bots* to each task, in order to improve the efficiency of the entire group. In particular, we have been working on a mechanism which adaptively tunes the number of active robots in a foraging task: that is, searching for objects and retrieving them to a nest location (Labella et al., 2004, 2006).

Functional self-assembly. Self-assembly only becomes truly meaningful in an autonomous robotic system if it is used as a means to achieve a specific goal. The term *functional self-assembly* was coined to describe this goal-driven self-assembly (Trianni et al., 2004). More precisely, a group of robots is said to exhibit functional self-assembly if the robots can choose to self-assemble in response to the demands of their task and environment. Functional self-assembly is used as an adaptive response mechanism by a group of autonomous robots when faced with a contingency which prevents the robots from carrying out their tasks individually. The mechanism of functional self-assembly can be decomposed into three sub mechanisms, each of which is highly complex in its own right. An individual *s-bot* must (a) decide whether or not the environmental contingencies require self-assembly, (b) coordinate its movements to connect to and/or facilitate the connection of other *s-bots*, and (c) coordinate its movements once connections are established. Our research work aims at developing a behavioral repertoire for a robot so that it can initiate either an individual or a collective response based on the demands of its task and environment (O'Grady et al., 2005).

Integration over time of perceptual input. In order for the *s-bots* to be able to perform tasks of higher complexity, it might be necessary to complement the functional and structural characteristics of their controllers with memory and learning mechanisms. Such controllers can allow the *s-bot* to integrate over time sensorial cues and to exploit its perceptual stream to make individual and collective choices. We make use of "leaky-integrator" neurons as basic building blocks for building non-reactive neuro-structures (Tuci et al., 2005).

ACKNOWLEDGMENTS

The authors wish to thank the many people who contributed in many different ways to the success of the presented research. In particular, André Guignard and Dario Floreano for the realization of the *s-bot* hardware used in the experiments; Luca Maria Gambardella, Giovanni Pettinaro and Ivo Kwee for contributions in the realization of the simulation environment; Stefano Nolfi, Gianluca Baldassarre and Jean-Louis Deneubourg for the many discussions concerning ways to implement distributed control; Alexandre Campo, Anders Christensen, and Stéphane Magnenat for help in some technical aspects of control software development; Mauro Birattari for help in the design of the experiments. This work was supported by the "SWARM-BOTS" project, funded by the Future and Emerging Technologies programme (IST-FET) of the European Commission, under grant IST-2000-31010. The information provided 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 publication. Marco Dorigo acknowledges support from the Belgian FNRS, of which he is a Senior Research Associate, and from the "ANTS" project, an "Action de Recherche Concerté" funded by the Scientific Research Directorate of the French Community of Belgium.

REFERENCES

Baldassarre, G., Parisi, D., Nolfi, S., 2004a. Coordination and behaviour integration in cooperating simulated robots. In: Schaal, S., Ijspeert, A., Billard, A., Vijayakamur, S., Hallam, J., Meyer, J.-A. (Eds.), From Animals to Animats 8. Proceedings of the Eighth International Conference on Simulation of Adaptive Behavior (SAB04). MIT Press, Cambridge, MA, pp. 385-394.

Baldassarre, G., Parisi, D., Nolfi, S., 2004b. Coordination and behaviour integration in cooperating simulated robots. In: Schaal, S., Ijspeert, A., Billard, A., Vijayakamur, S., Hallam, J., Meyer, J.-A. (Eds.), From Animals to Animats 8. Proceedings of the Eighth International Conference on Simulation of Adaptive Behavior (SAB04). MIT Press, Cambridge, MA, pp. 385-394.

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

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

Brown, H., Weghe, M.V., Bereton, C., Khosla, P., 2002. Millibot trains for enhanced mobility. IEEE/ASME Transactions on Mechatronics 7 (2), 452-461.

Castano, A., Shen, W.-M., Will, P., 2000. CONTRO: Towards deployable robots with inter-robots metamorphic capabilities. Autonomous Robots 8 (3), 309-324.

Dorigo, M., Sahin, E., 2004. Swarm robotics – special issue editorial. Autonomous Robots 17 (2-3), 111-113.
Dorigo, M., Trianni, V., Sahin, E., Groß, R., Labella, T.H., Baldassarre, G., Nolfi, S., Deneubourg, J.-L., Mondada, F., Floreano, D., Gambardella, L.M., 2004. Evolving self-organizing behaviors for a swarm-bot. Autonomous Robots 17 (2-3), 223-245.
Fukuda, T., Ueyama, T., 1994. Cellular Robotics and Micro Robotic Systems. World Scientific Publishing, London, UK
Groß, R., Bonani, M., Mondada, F., Dorigo, M., 2006a. Autonomous self-assembly in swarm-bots. IEEE Transactions on Robotics, in press.
Groß, R., Dorigo, M., 2004. Group transport of an object to a target that only some group members may sense. In: Yao, X., Burke, E., Lozano, J., Smith, J., Merelo-Guervós, J., Bullinaria, J., Rowe, J., Tiòo, P., Kabán, A., Schwefel, H.-P. (Eds.), The 8th International Conference on Parallel Problem Solving From Nature (PPSN 2004). Vol. 3242 of Lecture Notes in Computer Science. Springer Verlag, Berlin, Germany, pp. 852-861.
Groß, R., Mondada, F., Dorigo, M., 2006b. 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. IEEE Computer Society Press, Los Alamitos, CA, pp. 1317-1323.
Groß, R., Tuci, E., Dorigo, M., Bonani, M., Mondada, F., 2006c. Object transport by modular robots that self-assemble. In: Proc. of the 2006 IEEE Int. Conf. on Robotics and Automation. IEEE Computer Society Press, Los Alamitos, CA, pp. 2558-2564.
Haykin, S., 1999. Neural Networks: A Comprehensive Foundation. ISBN 0-13-273350-1. Prentice Hall.
Labella, T.H., Dorigo, M., Deneubourg, J.-L., 2004. Efficiency and task allocation in prey retrieval. In: Ijspeert, A., Murata, M., Wakamiya, N. (Eds.), Biologically Inspired Approaches to Advanced Information Technlogy: First International Workshop, BioADIT 2004. Vol. 3141 of Lecture Notes in Computer Science. Springer Verlag, Berlin, Germany, pp. 274-289.
Labella, T.H., Dorigo, M., Deneubourg, J.-L., 2006. Division of labour in a group of robots inspired by ants' foraging behaviour. ACM Transactions on Autonomous and Adaptive Systems, in press.
Mondada, F., Bonani, M., Guignard, A., Magnenat, S., Studer, C., Floreano, D., 2005a. Superlinear physical performances in a SWARM-BOT. In: Capcarrere, M., A.A. Freitas, P.B., Johnston, C., Timmis, J. (Eds.), Proceedings of the 8th European Conference in Artificial Life (ECAL05). No. 3630 in LNAI. Springer Verlag, Berlin, Germany, pp. 282-291.
Mondada, F., Gambardella, L.M., Floreano, D., Nolfi, S., Deneubourg, J.-L., Dorigo, M., June 2005b. The cooperation of swarm-bots: Physical interactions in collective robotics. IEEE Robotics & Automation Magazine 12 (2), 21-28.
Mondada, F., Pettinaro, G.C., Guignard, A., Kwee, I.V., Floreano, D., Deneubourg, J.-L., Nolfi, S., Gambardella, L.M., Dorigo, M., 2004. SWARM-BOT: A new distributed robotic concept. Autonomous Robots 17 (2-3), 193-221.
Murata, S., Yoshida, E., Kamimura, A., Kurokawa, H., Tomita, K., Kokaji, S., 2002. M-TRAN: Self-reconfigurable modular robotic system. IEEE/ASME Transactions on Mechatronics 7 (4), 431-441.
Nolfi, S., Floreano D., 2000. Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines. MIT Press/Bradford Books, Cambridge, MA.

Nouyan, S., Dorigo, M., 2006. Swarm robotic path formation through chains. In: Dorigo, M., Gambardella, L.M., Birattari, M., Martinoli, A., Poli, R., Stützle, T. (Eds.), Ant Colony Optimization and Swarm Intelligence – Proceedings of ANTS 2006 – Fifth International Workshop. Vol. 4150 of Lecture Notes in Computer Science. Springer Verlag, Berlin, Germany, in press.

Nouyan, S., Groß, R., Bonani, M., Mondada, F., Dorigo, M., 2006. Group transport along a robot chain in a self-organised robot colony. In: Arai, T., Pfeifer, R., Balch, T., and Yokoi, H. (Eds.), Proceedings of the 9th Int. Conf. on Intelligent Autonomous Systems. IOS Press, Amsterdam, The Netherlands, pp. 433-442.

O'Grady, R., Groß, R., Mondada, F., Bonani, M., Dorigo, M., 2005. Self-assembly on demand in a group of physical autonomous robots navigating rough terrain. In: Capcarrère, M., A.A. Freitas, P.B., Johnston, C., Timmis, J. (Eds.), Proceedings of the 8th European Conference in Artificial Life (ECAL05). No. 3630 in LNAI. Springer Verlag, Berlin, Germany, pp. 272-281.

Schneider-Fontán, M., Mataric, M.J., 1996. A study of territoriality: The role of critical mass in adaptive task division. In: Maes, P., Matariæ, M.J., Meyer, J.-A., Pollack, J., Wilson, S.W. (Eds.), From Animals to Animats 4, Fourth International Conference on Simulation of Adaptive Behavior (SAB-96). MIT Press/Bradford Books, Cambridge, MA, pp. 553-561.

Trianni, V., Dorigo, M., 2004. Evolving functional self-assembling in a swarm of autonomous robots. In: Schaal, S., Ijspeert, A., Billard, A., Vijayakamur, S., Hallam, J., Meyer, J.-A. (Eds.), From Animals to Animats 8. Proceedings of the Eight International Conference on Simulation of Adaptive Behavior (SAB04). MIT Press, Cambridge, MA, pp. 405-414.

Tuci, E., Ampatzis, C., Dorigo, M., 2005. Evolving neural mechanisms for an iterated discrimination task: A robot based model. In: Capcarrère, M., A.A. Freitas, P.B., Johnston, C., Timmis, J. (Eds.), Proceedings of the 8th European Conference in Artificial Life (ECAL05). No. 3630 in LNAI. Springer Verlag, Berlin, Germany, pp. 231-240.

Tuci, E., Groß, R., Trianni, V., Mondada, F., Bonani, M., Dorigo, M., 2006. Cooperation through self-assembly in multi-robot systems. ACM Transactions on Autonomous and Adaptive Systems, in press.

Yim, M., Duff, D.G., Roufas, K.D., 2000. PolyBot: A modular reconfigurable robot. In: Proceedings of the 2000 IEEE/RAS International Conference on Robotics and Automation. Vol. 1. IEEE Robotics and Automation Society, Piscataway, NJ, pp. 514-520.