Teamwork in Self-Organized Robot Colonies

Shervin Nouyan, Roderich Groß, *Member, IEEE*, Michael Bonani,
Francesco Mondada, *Member, IEEE*, and Marco Dorigo, *Fellow, IEEE*

***Abstract*— Swarm robotics draws inspiration from decentralized self-organizing biological systems in general and from the collective behavior of social insects in particular. In social insect colonies, many tasks are performed by higher order group or team entities, whose task-solving capacities transcend those of the individual participants. In this paper, we investigate the emergence of such higher order entities. We report on an experimental study in which a team of physical robots performs a foraging task. The robots are "identical" in hardware and control. They make little use of memory and take actions purely on the basis of local information.**

Our study advances the current state of the art in swarm robotics with respect to the number of real-world robots engaging in teamwork (up to 12 robots in the most challenging experiment). To the best of our knowledge, in this paper we present the first self-organized system of robots that displays a dynamical hierarchy of teamwork (with cooperation also occurring among higher order entities). Our study shows that teamwork requires neither individual recognition nor differences between individuals. This result might also contribute to the ongoing debate on the role of these characteristics in the division of labor in social insects.

***Index Terms*— Cooperation, division of labor, foraging, group transport, hierarchy, path formation, self-assembly, self-organization, swarm robotics, teamwork.**

I. Introduction

BENEFITING from the development of ever cheaper and smaller components, the study of multirobot 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 centralized management of the robots' activities and on excessive information exchange rapidly reach their limits. Furthermore, such approaches are less tolerant to individual failure.

To overcome these problems, researchers in swarm robotics draw inspiration from decentralized self-organizing biological systems in general and from the collective behavior of social insects in particular [27]. Swarm robotics systems are then typically composed of robots that, at the individual level, have relatively limited task-solving abilities and limited knowledge about their environment. Still, the overall system can exhibit complex behavior. This is realized in a bottom-up fashion: complexity arises from the numerous interactions among the robots and between the robots and their environment. The general paradigm is often referred to as *swarm intelligence* [14], [19], [25].

Presently, little is known about how to design swarm intelligence systems. Thus, it is not surprising that the complexity exhibited in current implementations does not come close either to the complexity of biological systems, or to the complexity of systems that men built following the more traditional top-down approach.

In this paper, we investigate 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 invaluable 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 organization of individuals when performing a task. In an insect colony, various organizational levels can be observed. Behaviors both at the individual level as well as at the colony level have been extensively studied [38]. "However, between these two extremes, numerous functional adaptive units, or 'parts' exist" [5, p. 291]. These *intermediate-level* parts comprise *groups* and *teams*.

Teamwork is widely observed in vertebrates. Here, *individual recognition* is believed to be an important factor [81]. Fewer examples of teamwork are known in invertebrates. Oster and Wilson ([63]; reviewed in [3]) argue that members of social insect colonies cannot form teams as a consequence of their low grade of discrimination: social insects can discriminate "nestmates from aliens, [and] members of one caste as opposed to another" [63, p. 151], however, "there is very

Manuscript received January 20, 2008; revised September 17, 2008; accepted November 13, 2008. Current version published August 14, 2009. This work was supported by the Sixth Framework Programme of the European Community in the form of the IST FET project "SWARM-BOTS" (Grant No. IST-2000-31010) and of a Marie Curie Intra-European Fellowship (Contract No. MEIF-CT-2006-040312), and by the Scientific Research Directorate of the French Community of Belgium in the form of the "ANTS" project, an "Action de Recherche Concertée." It reflects only the authors' views. The European Community is not liable for any use that may be made of the information. M. Dorigo acknowledges support from the Belgian F.R.S.–FNRS, of which he is a research director.

S. Nouyan was with the Institut de Recherches Interdisciplinaires et de Développements en Intelligence Artificielle (IRIDIA), CoDE, Université Libre de Bruxelles, Brussels 1050, Belgium. He is currently with Audi, Ingolstadt 85045, Germany (e-mail: snouyan@iridia.ulb.ac.be).

R. Groß, M. Bonani, and F. Mondada are with the Laboratoire de Systèmes Robotiques (LSRO), Ecole Polytechnique Fédérale de Lausanne (EPFL), 1015 Lausanne, Switzerland (e-mail: roderich.gross@ieee.org; michael.bonani@epfl.ch; francesco.mondada@epfl.ch).

M. Dorigo is with the Institut de Recherches Interdisciplinaires et de Développements en Intelligence Artificielle (IRIDIA), CoDE, Université Libre de Bruxelles, 1050 Brussels, Belgium (e-mail: mdorigo@ulb.ac.be).

The first two authors of this paper contributed in the same way to the research published and they both deserve to be first author.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TEVC.2008.2011746

1051-8215/$26.00 © 2009 IEEE

little evidence that social insects can recognize each other as individuals (but see Tibbetts [76])" [3, p. 6]. In contrast, in the recent literature [3], [21], [38], biologists suggest that teams are indeed formed in social insects and that they do not require individual recognition. Another aspect that is the subject of ongoing debate is whether inter-individual differences (e.g., members of different castes) are fundamentally required for teamwork [2], [13], [38].

One of the merits of studying robotic systems is that the individual morphology and behavior are system variables that are controlled. Therefore, we can investigate whether tasks that require a complex division of labor fundamentally require individual recognition or differences between individuals.

In this paper, we illustrate the methods and results of a series of experimental works in which a set of "identical" robots is required to perform a complex cooperative task. At the beginning of a trial, the robots are 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:

1) C_1: the prey requires concurrent, physical handling by multiple robots to be moved;
2) C_2: each robot's perceptual range is small when compared to the distance between the nest and the prey. Moreover, perception is unreliable;
3) C_3: no robot has any (explicit) knowledge about the environment beyond its perceptual range;
4) 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 labor 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). Moreover, they can not communicate (directly) with other robots near the nest (constraint C_4). 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, along which the transport is guided toward the nest.

In the following we use the terms groups and teams as defined by Anderson and Franks [2]. In particular, a *group* is a set of individuals that tackles a group task; a *team* is a set of individuals that tackles 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" (p. 535). Furthermore, a *partitioned task* is "a task that is split into two or more subtasks that are organized sequentially (Jeanne [41]; reviewed in Ratnieks and Anderson [69]; Anderson and Ratnieks [6])" [3, p. 4]. Anderson and Franks [2], [3], and Anderson and McMillan [4] 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.

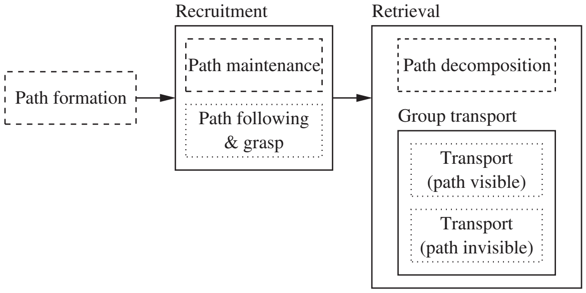


Fig. 1. Illustration of the division of labor to accomplish the foraging task under constraints C_1, C_2, C_3, and C_4 (for details, see text). The overall task is a partitioned task. It splits into three (sub-) tasks—*path formation*, *recruitment* and *retrieval*—that are organized sequentially (indicated by arrows). *Path formation* requires multiple robots to explore collectively the environment and establish a path between the nest and the prey. *Recruitment* requires some robots to maintain the established path (*path maintenance*) and other robots to follow the path from the nest to the prey before either grasping the prey directly or grasping other robots that are already attached to the prey (*path following & grasp*). *Retrieval* requires those robots that are gripping the prey and/or other robots, to transport the prey along the path back to the nest (*group transport*). The path gradually dissolves as the transport proceeds (*path decomposition*). *Group transport* involves multiple robots, each of which can or cannot perceive the path depending on the particular situation [*transport* (*path visible*) and *transport* (*path invisible*)]. Individual tasks, group tasks, and team tasks are framed respectively by dotted, dashed and solid lines.

Fig. 1 illustrates the division of labor present in our system. The overall task can be considered a partitioned task comprising three subtasks that are organized sequentially: 1) *path formation* requires robots to explore the environment and form a path in between the nest and the prey; 2) *recruitment* requires robots to follow the path from the nest to the prey and then grasp either the prey directly or other robots already attached to the prey; 3) *retrieval* requires robots to transport the prey along the path to the nest. *Path formation* itself is a group task, because only a group of robots can establish a path. Similarly, *path maintenance* and the stepwise *path decomposition* are group tasks. *Recruitment* is a team task, because it requires two different sub-tasks to be performed concurrently—*path maintenance* and *path following & grasp*, where the latter is an individual task.[1] *Retrieval* 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 toward the nest. *Group transport* is a team task, as 1) multiple robots are required to transport the prey, and 2) the transporting robots that are (temporarily) unable to perceive the path (and thus the transport direction) need to perform distinct actions to avoid the group transport being ineffective.

The remainder of this paper is organized as follows. In Section II, we overview related work. In Sections III and IV, we detail the robot's hardware and control. In Section V, we present the experimental setup and the results. Finally, in Section VI, we discuss the results and conclude the paper.

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

II. Related Work

The task tackled in this paper involves three broad challenges: 1) navigation, that is, the formation, maintenance and stepwise decomposition of a path as well as path following; 2) self-assembly, in other words, the formation of physically connected (pulling) structures; and 3) group transport, that is, the cooperative pushing and pulling of the prey toward the nest. The problems of navigation, self-assembly, and group transport have been extensively studied, each in its own right. In the following, we give an overview of the related work in these areas.

A. Navigation

As mentioned in the introduction, we are interested in approaching the problem for large groups of robots following the swarm robotics principles, where the cooperation and the collectivity of a robot group are emphasized.

These characteristics can be observed in social insects, such as ants, bees, or termites. When foraging, ants of many species lay trails of pheromone, a chemical substance that attracts other ants. Deneubourg *et al.* [16] showed that pheromone trails help ants to find the shortest path between a nest and a food source. When approaching the problem of controlling swarms of robots, researchers often take inspiration from social insects, and in the case of path formation sometimes directly refer to the term pheromone [51], [66], [67], or to ants [75].

All these approaches employ distributed control mechanisms, and mostly rely on simple strategies and on local information. We can roughly distinguish between two categories of distributed multiagent path planning.

1) **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 usually has very limited perception 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 make them ideally suited for large-scale experiments. O'Hara and Balch [62], for instance, use a sensor network with up to 156 sensor nodes that compute the shortest path using the distributed Bellman–Ford algorithm [12], and test the impact of different configurations of sensor placement. Li *et al.* [50] 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 [11] 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 nine nodes.

   In the simplest case, network nodes do not have any sensory capabilities and are used as landmarks or as a medium for indirect, so-called *stigmergic*, communication [29]. An example is *radio frequency identification* (RFID) based devices. Mamei and Zambonelli [51], [52] use passively powered RFID tags in an office environment to mark fixed locations, such as desks, and to identify objects that may be moved around, such as books. In their experiments, robots can manipulate the RFID tags and leave a trail which enables other robots to find particular objects. In addition to their low production cost, such devices in general 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.

2) **The robots serve as landmarks or beacons themselves.** This is the case for our approach, as we will explain in more detail in Section IV. When designing our controllers, we took inspiration from Goss and Deneubourg [28], who have studied robot chains in the context of foraging. In their approach, every robot in a chain emits a signal indicating its position in the chain. A similar system was implemented by Drogoul and Ferber [20]. Both works were carried out in simulation, and in both works robots in a chain structure need to discriminate between as many signals as there are robots in the chain. In contrast, in our approach robots need to discriminate only between a constant number of signals independent of the length of the chain.

   Werger and Matarić [79] use physical robots to form a chain in a prey retrieval task. The robots in the chain rely on physical contacts: adjacent robots touch each other regularly in order to maintain the chain.

   Ichikawa and Hara [39], [40] and Payton *et al.* [66], [67] study robot networks which can be used to represent a path as well. Initially, the robots are gathered in one place. Then, they gradually expand the area covered by their network while maintaining nearest-neighbor communications. Ichikawa and Hara consider a system of up to 40 robots of which one broadcasts locally and periodically a signal. The other robots move at random. When a robot does not perceive any signal anymore, it retreats to reestablish the contact it has lost. It then becomes itself a static beacon, thereby expanding the network. On the other hand, Payton *et al.* consider a gas expansion model that leads to a uniform distribution. A group of robots first spreads in the environment using simple attraction/repulsion mechanisms. Then the robots communicate three different sorts of pheromone to select the shortest of the many different possible paths. Payton *et al.* also consider another strategy to form the network, which is referred to as guided growth. One robot is selected to be the leader and the other robots follow the leader. In this way the robot structure stretches to form a line. The leading robot can for instance be designated by the user.

B. Self-Assembly

Following Whitesides and Grzybowski [80], self-assembly can be defined as a process by which pre-existing discrete components organize into patterns or structures without human intervention.

Self-assembly is widely observed in social insects [7], [72]. Via self-assembly, ants, bees, and wasps can organize into functional units at an intermediate level between the individual and the colony. Anderson *et al.* [7] 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) defense, 2) pulling structures, 3) thermoregulation, 4) colony survival under inclement conditions, and 5) ease of passage when crossing an obstacle" (p. 99). Anderson *et al.* [7] claim that in almost all of the observed instances, the function could not be achieved without self-assembly.

Self-reconfigurable robots [71], [83] hold the potential to self-assemble and thus to mimic the complex behavior of social insects. In current implementations [42], [58], [71], [83], 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 prearranged at known positions [84]. Some instances of less constrained self-assembly are reported (for an overview see [34]). For example, Fukuda *et al.* [22], [24] demonstrate self-assembly among robotic cells of the cellular robotic system (CEBOT) [23]. In the experiment, a moving cell, which was controlled by a finite-state automaton, approached a static cell and connected to it. Rubenstein *et al.* [70] demonstrate the ability of two modular robots to self-assemble. Each robot consisted of a chain of two linearly-linked configurable robot (CONRO) modules [15]. 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 also been demonstrated with the swarm-bot system [57]. Experiments were conducted on different terrains and with up to 16 physical robots [30].

C. Group Transport

Group transport can be defined as the "conveyance of a burden by two or more individuals" [56, p. 227]. In the biological literature, group transport is almost exclusively reported in the context of ants. In fact, Moffett [56, p. 220] 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. Therefore, the control typically requires a sensory feedback mechanism. Most studies consider two robots pushing a wide box simultaneously from a single side [18], [26], [53], [64], [74]. To coordinate the robots' actions, robots are specifically arranged [18], [26], [53], [64], control is synchronized [53], relative positions are known [18], [64], explicit communication is used [53], [64], and/or individual tasks are generated by a designated leader agent [26].

Only a few studies consider more than two robots, pushing a box simultaneously [45], [46], [48], [82]. In these cases, the control is homogeneous and decentralized; the robots make no use of explicit communication.

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 assembled with the object [1], [17], [43], [54], [73], [78]. In some studies, the robots are also capable of self-assembling [31], [32], [35], [37], [77]. 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 the robots of the group. The object is transported as each robot follows the given trajectory by making use of dead-reckoning [17]. In other systems, the manipulation is planned in real time by an external workstation, which communicates with the robots [54]. 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* [73], [78]. However, in many leader–follower systems explicit communication is not required [1], [36], [43].

III. Hardware

We use a robotic system called swarm-bot lying at the intersection between collective and self-reconfigurable robotics [57]. The system is composed of basic robotics units, called *s-bots*,[2] which are fully autonomous and mobile, and capable of autonomously connecting to each other.

Fig. 2(a) shows the physical implementation of the *s-bot*. The robot has a total height of 19 cm, a diameter of 12 cm, and weighs approximately 700 g. In the following, we briefly overview the actuators and sensors most relevant to this paper. For a more comprehensive description of the *s-bot*'s hardware see [57].

The *s-bot* has five degrees of freedom (DOF) all of which are rotational:

1) two DOF for the differential *treels* system—a combination of tracks and two external wheels;
2) one DOF to rotate the *s-bot*'s upper part (called the *turret*) with respect to the lower part (called the *chassis*);
3) one DOF for the grasping mechanism of the gripper (in what we define to be the *s-bot*'s front);
4) one DOF for elevating the arm to which the gripper is attached (e.g., to lift another *s-bot*).

These DOF are actuated by DC motors equipped with an incremental encoder and controlled in torque, position, or speed by a PID controller.

For the purpose of communication, the *s-bot* is equipped with eight RGB LEDs distributed around the module, and two loudspeakers.

[2] 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).

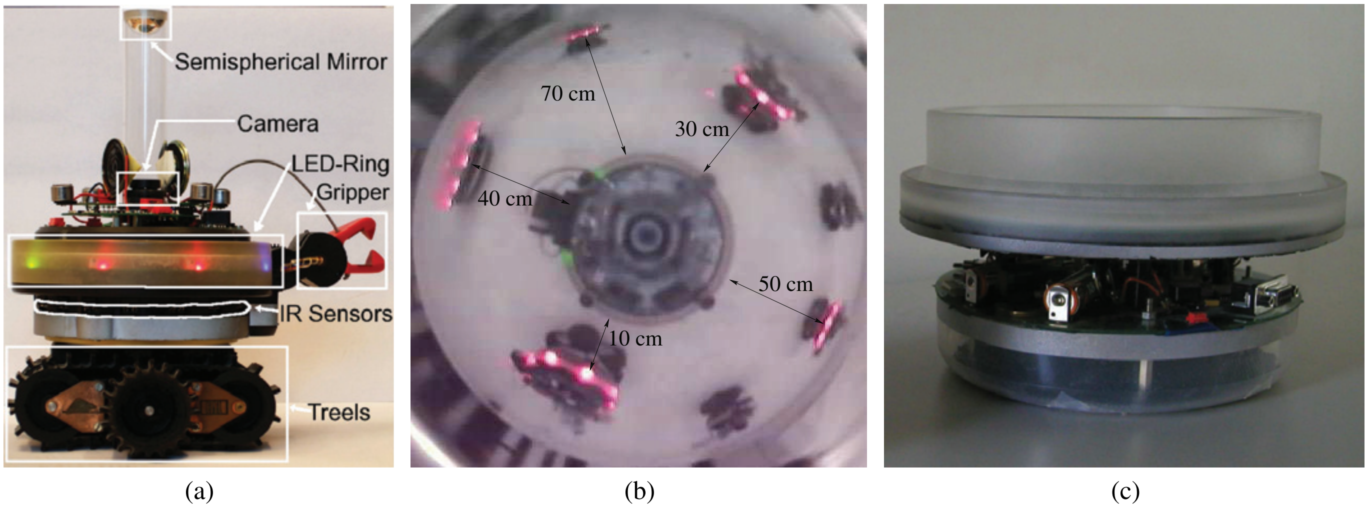


Fig. 2. Overview of the hardware. (a) *s-bot* robot. (b) Image taken by the omnidirectional camera of the *s-bot*. It shows seven other objects (an *s-toy* and six *s-bots* at various distances), five of which have activated their LEDs in red. (c) One of the two *s-toys*, which is used either as a nest or as a prey.

An *s-bot* can connect with another by grasping the connection ring with the gripper, and it can receive connections on more than two-thirds of its perimeter.

The *s-bot* is equipped with a variety of sensors:

1) four proximity sensors fixed underneath (ground sensors);
2) fifteen proximity sensors distributed around the turret;
3) two optical barriers integrated in the gripper;
4) one force sensor between the turret and the chassis (2-D traction sensor);
5) one torque sensor on the elevation arm of the gripper;
6) three-axes inclinometer;
7) eight light sensors distributed around the module;
8) four omnidirectional microphones;
9) one VGA camera directed toward a spherical mirror to provide an omnidirectional view.

Furthermore, proprioceptive sensors provide internal motor information such as torque, position, and speed.

The chassis of each *s-bot* can be rotated in any horizontal direction. This allows *s-bots*, once assembled within a modular robot (and not necessarily aligned with one another), to move in a common direction. The 2-D traction sensor, mounted between the *s-bot*'s turret and the chassis, measures the mismatch between the direction in which the chassis is trying to move and the direction in which the modular robot as a whole is trying to move.

A snapshot taken from an *s-bot*'s camera is shown in Fig. 2(b). The software used to detect colored objects allows the recognition of the red colored prey for distances up to 70–90 cm, and of objects colored blue, green, or yellow for distances up to 35–60 cm (depending on the light conditions and on which *s-bot* is used). Due to the spherical shape of the mirror the camera is directed to, only distances to close objects (up to 30 cm) can be approximated with good precision.

Fig. 2(c) shows the *s-toy*, an object that we use either as nest or as prey (depending on its color). It has a diameter of 20 cm and, like the *s-bot*, is equipped with RGB LEDs. The nest is immobile. The prey weighs 800 g and requires the cooperative effort of two or more *s-bots* to be moved.

IV. Controller

The controller consists of a collection of behaviors, each of which is designed to achieve a specific goal. Control policies for navigation, self-assembly, and group transport have been implemented and tested independently of each other by different researchers not necessarily using the same design approach. The individual behaviors are implemented using either the motor schema paradigm, neural networks, or simple hand-written commands. In order to integrate the different behaviors, we follow a behavior-based approach [10]. The behaviors and the rules that trigger a transition from one behavior to another are illustrated by the state diagram in Fig. 3.

The *s-bots* are initially located at random positions. If an *s-bot* does neither perceive a chain nor the nest, it performs a random walk (state search chain). An *s-bot* that finds a chain or the nest follows the perimeter of the encountered structure in a clockwise sense (state explore chain). The nest can be considered as the root of all chains. When the *s-bot* reaches the tail of a chain, it will join the chain with probability P_{in} per time step (state join chain). *S-bots* that are part of a chain do not 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. Therefore, 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

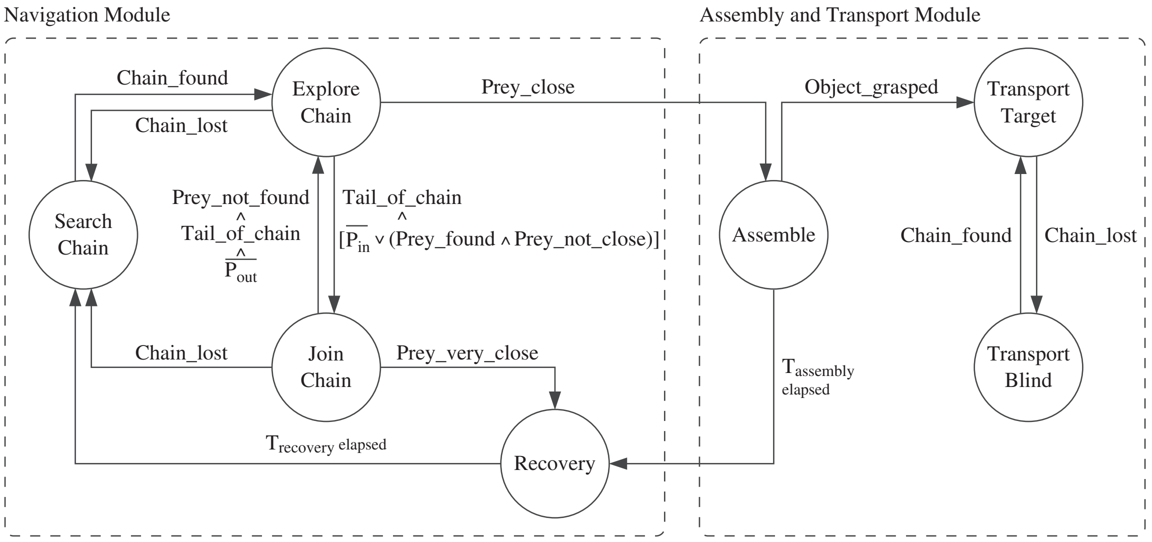


Fig. 3. State diagram of the finite state machine that controls each *s-bot*. Circles represent states (i.e., behaviors). Edge labels specify conditions that trigger transitions between the corresponding states. The initial state is search chain. $\overline{P_{in}}$ ($\overline{P_{out}}$) is a boolean variable, which is set to *True* with probability P_{in} (P_{out}), and *False* otherwise. The value of P_{in} (P_{out}) determines the rate at which *s-bots* join (leave) a chain.

a path is formed, it is maintained and in this way automatically recruits other *s-bots* to assemble to the prey (state assemble). If a robot that tries to assemble to the prey does not succeed within a certain time period, it moves back to the nest and rests for a while (state recovery). When a sufficient number of *s-bots* has assembled to the prey, the transport effectively starts. Robots assembled to the prey transport it by moving toward the closest perceived member of a chain (state transport target). In the event that some *s-bots* cannot perceive the path, they use their force sensors to estimate the direction of transport (state transport blind). When the prey reaches the tail of the path, the corresponding *s-bot* leaves the path and moves back to the nest to rest for a while (state recovery). In this way the transporting structure of *s-bots* is guided from node to node of the dissolving path to eventually reach the nest. An *s-bot* leaving the path to rest at the nest emits a sound signal for a period of 30 s. Transporting *s-bots* respond to this signal by temporarily suspending the transport. This prevents the transporting *s-bots* from colliding with the *s-bot* leaving the tail of the path as it gives the latter sufficient time to move away. No other *s-bot* reacts to the sound signal.[3] The behaviors can be grouped into two main modules: 1) the *navigation module* for the exploration of the environment to form a path between nest and prey, the recruitment of *s-bots* to the prey, and the guidance of the transporters back to the nest; and 2) the *assembly and transport module* for the formation of pulling structures and the group transport of the prey along the path to the nest.

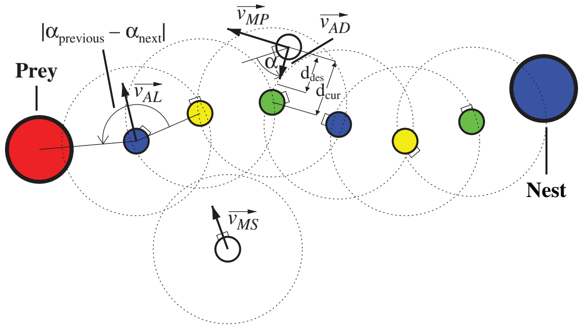

$|\alpha_{previous} - \alpha_{next}|$
Prey
$\overrightarrow{v_{AL}}$
$\overrightarrow{v_{MP}}$
$\overrightarrow{v_{AD}}$
α
d_{des}
d_{cur}
Nest
$\overrightarrow{v_{MS}}$

Fig. 4. Concept of cyclic directional patterns (CDP) chains. The small colored circles represent *s-bots* that have formed a CDP chain that connects the nest to the prey. Three colors are sufficient to give a directionality to the chain. The large dotted circles surrounding the *s-bots* indicate their perceptual range. The small uncolored circle above the chain represents an explorer *s-bot* following the chain toward the prey. The small uncolored circle below the chain represents an *s-bot* performing a random walk. Vector $\overrightarrow{v_{AL}}$, here indicated for one chain member only, represents the motor schema that leads to an overall alignment of the chain. Vectors $\overrightarrow{v_{MP}}$ and $\overrightarrow{v_{AD}}$, here indicated for the explorer *s-bot*, represent the motor schemas that lead to a tangential trajectory along the chain. Vector $\overrightarrow{v_{MS}}$ represents the motor schema that leads to forward motion.

A. Navigation Module

The navigation module lets *s-bots* organize into directional chains that regulate the exploration, recruitment, and guidance processes. The directionality in the chains relies on the concept of *cyclic directional patterns* (Fig. 4). Each *s-bot* emits one out of three signals (using its LED colors) depending on its position in the chain. By taking into account the sequence of the signals in the chain, an *s-bot* can determine the direction toward the nest, or toward the prey. The prey and the nest can be recognized by their color. The nest is blue, and the prey is red.

1) Motor Schemas: In the navigation module, the behaviors are realized following the motor schema paradigm [8], [9]. 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 behavior. Each motor

[3] In principle, sound can be considered to be a global signal. However, as the sound signal is used only to avoid (local) physical collisions, it plays the role of a local signal. It would have been more elegant to use a purely local signaling method instead of sound, for example, to assign a fifth color to the LEDs of *s-bots* that leave the path to rest at the nest, and to let the transporting *s-bots* respond to this color. Unfortunately, we could not use five different colors, as our vision software cannot discriminate between more than four colors reliably. Note that using a purely local signaling method could be of advantage, for example, if multiple objects were retrieved at a same time.

schema outputs a vector representing the desired direction of motion. For each behavior, 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.[4] Common to all behaviors is a motor schema for collision avoidance.

The following motor schemas are used (see also Fig. 4).

1) **Adjust_Distance**(α, d_{cur}, d_{des}): returns a vector $\overrightarrow{v_{AD}}$ that points toward an object at angle α if the current distance to the object d_{cur} is larger than the desired distance d_{des}, and in the opposite direction otherwise. The length of the returned vector is proportional to the value of $|d_{cur} - d_{des}|$. In order to avoid an oscillating behavior, the vector is set to zero if $|d_{cur} - d_{des}| < 5$ cm.

2) **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.

3) **Avoid_Collisions**($p_0, p_1, \ldots, p_{14}$): returns a vector $\overrightarrow{v_{AC}}$ that takes into account the activations of the turret's proximity sensors ($p_0, p_1, \ldots, p_{14}$) that are above a threshold. The direction of the vector is opposite to the direction of the sensor with maximum activation, and its length is proportional to the difference between the activation and the threshold.

4) **Move_Straight**: returns a unit vector $\overrightarrow{v_{MS}}$ that points forward.

5) **Align**($\alpha_{previous}$, α_{next}): returns a vector $\overrightarrow{v_{AL}}$ that leads to the alignment between the previous and the next chain neighbor which are perceived at angles $\alpha_{previous}$ and α_{next} (in degrees). The length of the vector is proportional to the value of $180° - |\alpha_{previous} - \alpha_{next}|$. In order to avoid an oscillating behavior, the vector is set to zero if $|\alpha_{previous} - \alpha_{next}| > 170°$ (with 180° representing perfect alignment).

2) Behaviors: The navigation module comprises four behaviors.

1) **Search Chain**: perform a random walk by moving straight until an obstacle is detected ahead. Then turn on the spot for a random angle. LEDs are off. Active motor schemas: move_straight, avoid_collisions.

2) **Explore Chain**: cycle at a constant distance around the nearest chain member in a clockwise sense (repeat this action once the next chain member becomes the nearest chain member). Thereby, the explorer moves along the chain. In case an *s-bot* 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 to start (probabilistically) a new chain by itself. LEDs are off. Active motor schemas: move_perpendicular, adjust_distance, avoid_collisions.

3) **Join Chain**: activate LEDs with the appropriate color, which depends on the color of the previous chain neighbor. Align with the two closest neighbors in the chain (see Fig. 4). (This behavior improves the overall length of the chain. A side effect of this is that loops within the chain are avoided.) Furthermore, adjust the distance with respect to the previous neighbor to roughly 27 cm (to guarantee a visual connection of the chain neighbors). Active motor schemas: adjust_distance, align, avoid_collisions.

4) **Recovery**: move back to the nest and rest. Emit a sound signal for 30 s.[5] LEDs are off. Active motor schemas: move_perpendicular, adjust_distance, avoid_collisions.

3) Behavior Transitions: The following set of conditions trigger behavior transitions.

1) **Search Chain** → **Explore Chain**: if a chain member is perceived. Note that the nest is perceived as chain member, and that *s-bots* in state search chain do not respond to the perception of the prey.

2) **Explore Chain** → **Search Chain**: if no chain member is perceived any more.

3) **Explore Chain** → **Join Chain**: 1) if the prey is not perceived and the tail of a chain is reached (i.e., only one chain member is perceived), the *s-bot* joins the chain with probability P_{in} per time step, or 2) if the prey is perceived at a distance ≥ 30 cm and the tail of a chain is reached.

4) **Explore Chain** → **Assemble**: if the prey is detected at a distance < 30 cm.

5) **Join Chain** → **Search Chain**: if the previous chain neighbor is no longer perceived.

6) **Join Chain** → **Explore Chain**: if situated at the tail of a chain and if the prey is not perceived, the *s-bot* leaves the chain with probability P_{out} per time step.

7) **Join Chain** → **Recovery**: if the prey is perceived at a very close distance (i.e., less than 5 cm), which only occurs if the prey is transported toward the chain member.

8) **Recovery** → **Search Chain**: if $T_{recovery} = 30$ s has elapsed.

The two probabilistic parameters P_{in} and P_{out} have a significant effect on the overall behavior of the *s-bot* group. This concerns in particular the number and length of the formed chains and the dynamics of the process that governs the formation and decomposition of chains. For instance, low values for P_{in} result in a rather patient behavior; in most cases a single chain is formed slowly. For P_{in} close to 1, 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. Using computer simulations we evaluated the performance of the navigation module for different combinations of P_{in} and P_{out} [59]–[61]. Following this, we have chosen $P_{in} = 0.14$ and $P_{out} = 0.007$. On average, these parameter values fit well the environmental circumstances. A more elaborate strategy would be to let the robots themselves adapt their parameters, for example, see [49], [55], [65].

[4] The time step has a length of approximately 0.120 s. This value is not constant because it depends on the time required for image processing.

[5] No sound signal is emitted if state recovery was triggered from state assemble. In a strict sense, thus, two distinct recovery states exist.

B. Assembly and Transport Module

The assembly and transport module lets *s-bots* organize into pulling structures that move the prey along a path established by other *s-bots* back to the nest. The *s-bots* can exert forces either directly on the prey or indirectly via other *s-bots* they are physically connected with. In the following the behaviors and behavioral transitions are detailed.

1) Behaviors: The assembly and transport module comprises three behaviors.

1) **Assemble**: activate a feed-forward artificial neural network—a single-layer perceptron—that takes input from the camera as well as sensor readings from the left-front and right-front proximity sensors (p_0 and p_{14}). Use the network's output to control the speed and status of the tracks and the connection mechanism. LEDs are off. The network is trained to let the *s-bot* approach and grasp nearby objects that have activated their LEDs in red. (Initially, the prey is the only object with red LEDs. Upon connection, an *s-bot* activates its own LEDs in red. Therefore, it becomes itself an object with which to establish a connection.) The connection strength parameters of the neural network controller have been synthesized by an evolutionary algorithm [33]. A detailed description of the behavior can be found in [30].
2) **Transport Target**: activate LEDs with color red. If a sound signal is perceived, rest. Otherwise, orient the chassis toward the closest chain member, which indicates the direction to the nest, and start pulling. A detailed description of the behavior can be found in [37], [77].
3) **Transport Blind**: activate LEDs with color red. If a sound signal is perceived, rest. Otherwise, activate a simple recurrent neural network that is fed with input from the force and torque sensors, and use the network's output to control the speed of the tracks and the desired orientation of the chassis. The force sensor indicates the mismatch between the *s-bot*'s own direction of motion and the motion of other *s-bots* it is connected with; moreover, it is influenced by the prey that is transported. The torque sensors (of the treels and of the turret) give an estimate on whether stagnation is present. The connection strength parameters of the neural network controller have been synthesized by an evolutionary algorithm. A detailed description of the behavior can be found in [36].

2) Behavior Transitions: The following set of conditions trigger behavior transitions:

1) **Assemble** → **Recovery**: if $T_{assembly} = 90\,\mathrm{s}$ has elapsed;
2) **Assemble** → **Transport Target**: if connected to an object;
3) **Transport Target** → **Transport Blind**: if no chain member is perceived;
4) **Transport Blind** → **Transport Target**: if a chain member is perceived.

TABLE I
NUMBER OF *s-bots* REQUIRED TO ACCOMPLISH SUB-TASKS *path formation* (N_p), *recruitment* (N_r) AND *retrieval* (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_r	2	3	4	5	6	8	9	10
N_t	2	3	4	5	6	8	9	10

V. EXPERIMENTS

In this section we first describe the experimental setup and then present and discuss the obtained results.

A. Experimental Setup

The experiments take place in a bounded arena of size 500 cm × 300 cm. The nest is positioned in the center of the arena. The prey is put at distance D away from the nest toward one of the four corners. N *s-bots* are positioned on a grid composed of 60 points, which are 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 centimeters. 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 bounds for the number of *s-bots*, we assume the *s-bots* to be organized in a single chain, which is perfectly linear and directed toward the prey. The lower bound values can be calculated from the distances between the individual nodes forming the path from the nest and the distance of the last chain member from the prey (all distances are measured from center to center). The distance between neighboring nodes forming the path is programed to be constant. The actual distances vary slightly due to imprecision in the *s-bots*' perception. The average distance observed between neighboring 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.[6] The lower bound values so computed are given in Table I. For the accomplishment of the overall foraging task, two additional *s-bots* are required to transport the prey.

During experimentation, the *s-bots* are fully autonomous. The only exception to this is when an *s-bot* topples over. To

[6] The average distance between the first chain member and the nest is slightly larger than expected. The reason for this is that the radius of the nest (as well as of the prey) is about 4.0 cm bigger than the radius of the *s-bot*.

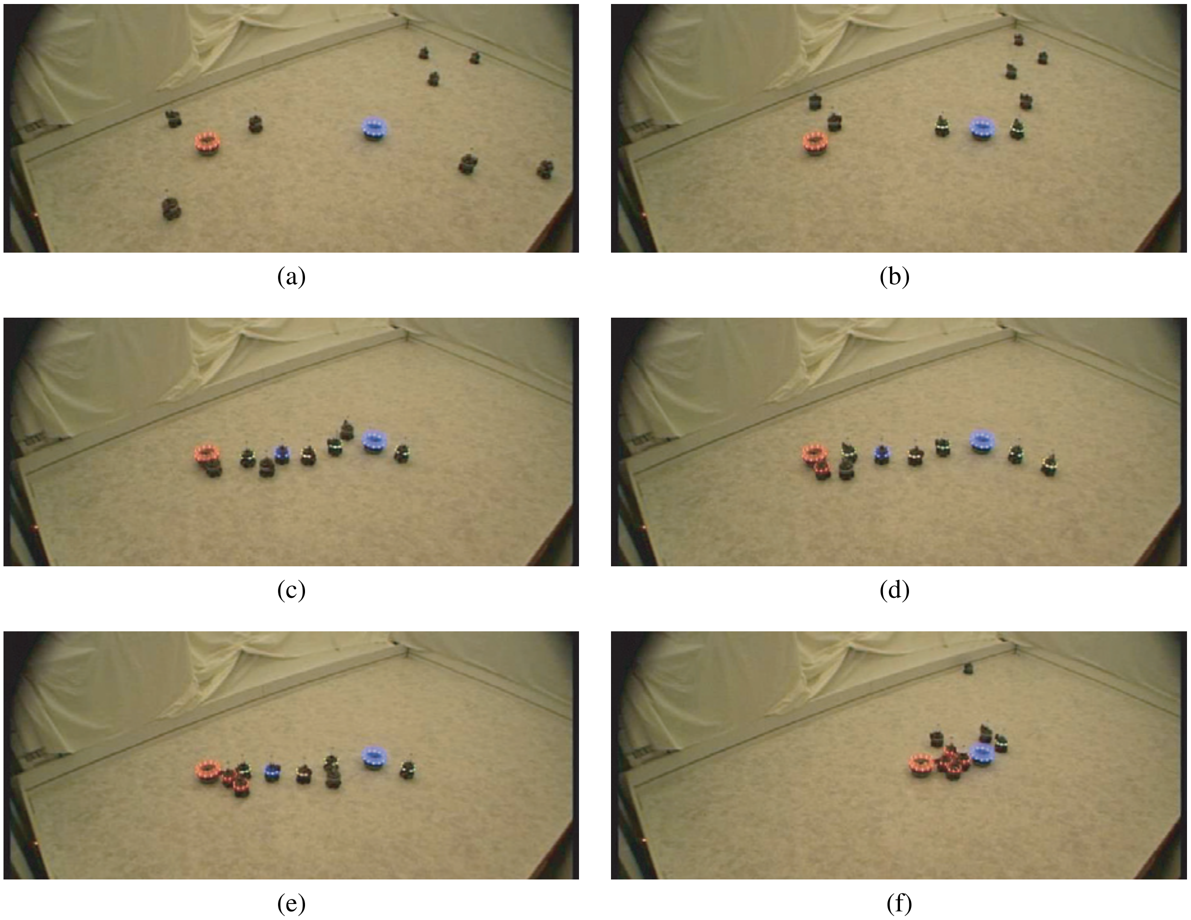


Fig. 5. Sequence of images taken for a trial with group size $N = 8$ *s-bots* and distance $D = 120\,\text{cm}$ between the nest (blue cylindrical object) and the prey (red cylindrical object): (a) $t = 0\,\text{s}$, (b) $t = 43\,\text{s}$, (c) $t = 101\,\text{s}$, (d) $t = 146\,\text{s}$, (e) $t = 181\,\text{s}$, and (f) $t = 307\,\text{s}$.

protect its hardware from being damaged (e.g., the camera mirror), we then remove the *s-bot* manually from the arena, and do not replace it until the end of the trial.

B. Results

According to the task description in Fig. 1, we can distinguish three levels of success, which are satisfied respectively, if:

1) *path formation* is completed (success level 1); in other words, a path connecting nest and prey has been formed and can be traversed in both directions;
2) *recruitment* is completed (success level 2); in other words, two or more *s-bots* have been recruited and are physically assembled with the prey so that the transport can start;
3) *retrieval* is completed (success level 3); in other words, the prey, or an *s-bot* transporting it, is in physical contact with the nest.

Variables T_p, T_r, and T_t denote the completion times (in s) for sub-tasks *path formation*, *recruitment*, and *retrieval*.

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 sub-task *retrieval* as an *s-bot* incorrectly assumed that it was part of the transport structure.

Fig. 5 shows a series of images taken during trial 8 of the setup $(N, D) = (8, 120)$. Within the first 96 s, four *s-bots* find the nest [Fig. 5(a) and (b)], and establish a path to the prey [Fig. 5(c)]. At time $t = 144\,\text{s}$, one of the four remaining *s-bots* is successfully recruited, and thus has gripped the prey and activated its LEDs in red [Fig. 5(d)]. This *s-bot* alone is not strong enough to pull the prey. However, shortly after, another *s-bot* becomes part of the pulling structure [Fig. 5(e)]. 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 $t = 307\,\text{s}$ [Fig. 5(f)].

Table II lists the measured completion times T_p, T_r, and T_t for the different sub-tasks. In the trials with setups $(N, D) = (2, 30)$ and $(4, 60)$ sub-task *path formation* is accomplished faster than any other sub-task. 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 I). Most time was spent for sub-task *recruitment*, on average 211.9 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

TABLE II

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_r 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 RETRIEVE THE PREY TO THE NEST. ALL RESULTS ARE GIVEN IN SECONDS. IF NO VALUE IS GIVEN, THE RESPECTIVE SUB-TASK WAS NOT SUCCESSFULLY COMPLETED

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	60	53	152
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	319
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	39		
7	0	394	14	408	29	106	25	160	69	86	86	241
8	0	414	25	439	28	65	82	175	96	82	129	307
9	0	132	23	155	48	119	28	195	72	154	49	275
10	0	86	114	200	19	151	41	211	114	42	91	247
Mean	0	211.9	32.6	244.5	24.5	133.3	51.8	209.6	92.0	85.1	90.6	274.6
	(0%)	(86.7%)	(13.3%)	(100%)	(11.7%)	(63.6%)	(24.7%)	(100%)	(34.2%)	(32.9%)	(33.0%)	(100%)
Std. Dev.	0	127.5	29.2	118.8	10.8	72.8	39.8	89.3	51.5	57.7	46.3	61.7

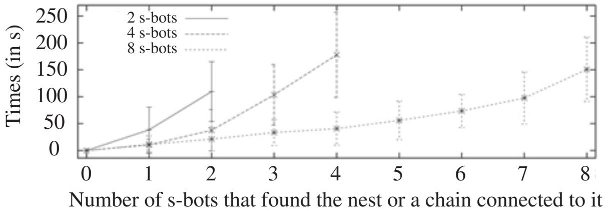


Fig. 6. Time until the nth *s-bot* finds either the nest or a chain (which provides a path to the nest) for the first set of experiments with $(N, D) = (2, 30)$, $(4, 60)$, and $(8, 120)$.

time until two out of two, or three out of four *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 eight *s-bots*. Only 85.1 s was spent on average for sub-task *recruitment*, which is significantly less than the times observed for group sizes two or four, respectively (two-sided Mann–Whitney, $p = 0.05$ before Bonferroni correction). Also, the time until a sufficient number of *s-bots* have found the nest drops from 109 s (103 s) for group size two (four) to 56 s for group size eight (see Fig. 6). We believe that there are two reasons for this. First, the higher the number of robots, the higher the degree of redundancy: in the system with eight *s-bots*, five of them are candidates for recruitment (while only two are required), whereas in the system with two *s-bots* (four *s-bots*) only two (three) are candidates for recruitment (see Table I). Second, 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.

TABLE III

OVERALL LEVEL OF SUCCESS ACHIEVED FOR SETUPS (N, D) IN THE SECOND SET OF EXPERIMENTS (AS DEFINED AT THE BEGINNING OF SECTION V-B): NO SUCCESS (0), SUB-TASK *path formation* ACCOMPLISHED (1), SUB-TASK *recruitment* ACCOMPLISHED (2), AND SUB-TASK *retrieval* 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. GRAY LEVELS OF CELLS REPRESENT THE BEST ACHIEVABLE LEVEL OF SUCCESS (SEE TABLE I): WHITE DENOTES NO SUCCESS, LIGHT GRAY DENOTES SUCCESS LEVEL 1, AND DARK GRAY 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	2
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	2	2
240	(0)	(0)	(0)	(0)	(0)	0	0	0	3	3

The time spent during retrieval T_t grows approximately linearly with the distance between nest and prey: 32.6, 51.8, and 90.6 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.

2) Second Set of Experiments: We examined the system under a wide range of group sizes (N) and prey distances (D). We conducted 70 trials, one for each different setup (for details see Section V-A).[7]

Table III gives an overview of the level of success reached. In 46 out of the 70 setups, a path can in principle be formed (see Table I). In 44 out of the corresponding 46 trials, the *s-bots* succeeded in forming a path. Only in 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

[7] The choice of running 70 trials was dictated by the limited amount of experimental time available.

TABLE IV

COMPLETION TIMES (IN S) OF SUB-TASKS *path formation* (T_p), *recruitment* (T_r), AND *retrieval* (T_t) IN SETUPS (N, D) OF THE SECOND SET OF EXPERIMENTS. IF NO VALUE IS GIVEN, THE RESPECTIVE SUB-TASK 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	493	160	88	97	65
150				662	337	486	32	379	511	78
180					317	1975	902	562	222	1649
210						2135	988		2370	810
240									827	335

T_r: time required for *recruitment*

D / N	1	2	3	4	5	6	7	8	10	12
60			286	62	67	104	257	135	75	61
90				59	272	69	159	168	64	41
120					181	193	281	458	35	94
150							69	635	314	72
180								594	97	787
210									176	170
240									229	133

T_t: time required for *retrieval*

D / N	1	2	3	4	5	6	7	8	10	12
60			17	121	246	276	33	47	19	183
90				41	23	158	80	20	538	129
120					56	80		245	144	
150							201		123	165
180								63	146	170
210										
240									258	447

explorers, thereby making it impossible to form a path. For setup $(N, D) = (8, 240)$ a path requires all eight *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 $t = 2204$ 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 $30.5 + 4 \cdot 27 + 38.5 = 177$ cm ($30.5 + 5 \cdot 27 + 38.5 = 204$ cm), which is 3 cm (6 cm) less than the distance that needs to be covered, and therefore is still within the range of perceptual error of the camera.

In 33 out of the 46 setups, also sub-tasks *recruitment* and *retrieval* can in principle be accomplished by the given number of *s-bots*. In 27 out of these 33 setups, the *s-bot* group was able to do so, and therefore the entire task was completed. In six setups, however, although a path was formed and two or more *s-bots* were recruited (and thus gripping the prey), the retrieval back to the nest was not successful. For setups $(N, D) = (7, 120)$, $(8, 150)$, $(10, 210)$, and $(12, 120)$, the gripper of one of the *s-bots* in the pulling structure opened during the transport phase (e.g., this happened when the corresponding *s-bot* stopped due to empty batteries). In this way, the transport was blocked. 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. For setup $(N, D) = (12, 210)$ the trial was stopped when seven *s-bots* were part of a linear structure that pulled the prey. In such a long structure most members cannot perceive the path, and thus the prey could not be moved any more.

TABLE V

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			317	247	327	395	301	194	113	245
90				123	394	255	271	223	616	178
120					411	766		791	276	
150							302		948	315
180								1219	465	2606
210										
240									1314	915

TABLE VI

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. IF NO VALUE IS GIVEN, SUB-TASK *path formation* WAS NOT SUCCESSFULLY COMPLETED

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									9	8

TABLE VII

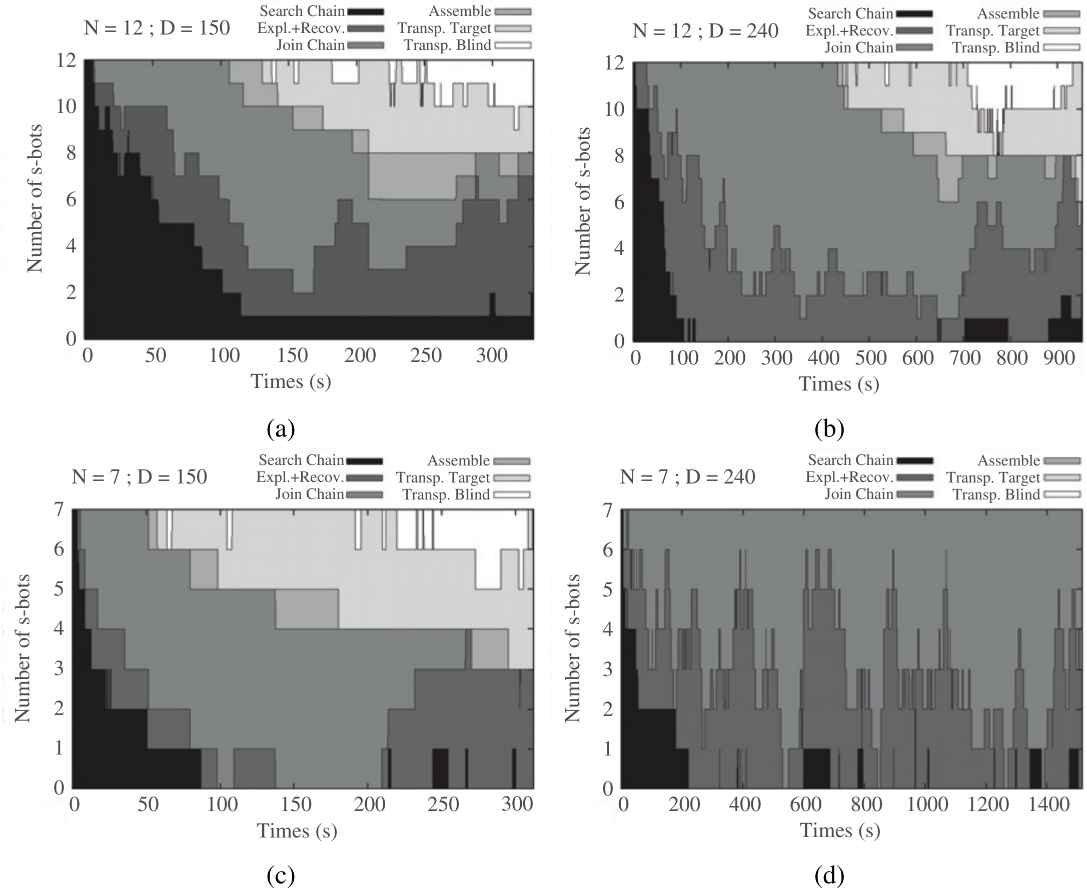
NUMBER OF *s-bots* THAT ARE PART OF THE TRANSPORT GROUP WHEN THE PREY REACHES THE NEST IN SETUPS (N, D) OF THE SECOND SET OF EXPERIMENTS. IF NO VALUE IS GIVEN, THE OVERALL TASK WAS NOT SUCCESSFULLY COMPLETED

D / N	1	2	3	4	5	6	7	8	10	12
60			2	4	4	5	2	3	3	3
90				2	2	3	2	2	9	4
120					2	3		3	5	
150							4		4	4
180								2	3	2
210										
240									3	4

Table IV lists the completion times for each of the three sub-tasks *path formation*, *recruitment*, and *retrieval*. Table V lists the overall completion time.

In the following, we examine the cooperation within our robot colonies in more detail. Tables VI and VII detail the sizes of the two main collaborating elements—the path-forming *s-bots* and the transporting *s-bots*. In particular, Table VI details the number of *s-bots* that formed the initial path connecting nest and prey. Table VII details the number of *s-bots* that were part of the transport group when the prey reached the nest.

Fig. 7 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



(a) (b) (c) (d)

Fig. 7. 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 gray levels indicate the number of *s-bots* in states search chain, explore chain plus recovery, join chain, assemble, transport target, and transport blind.

form a path, and thus also too small to accomplish the task. In the following, the four setups are discussed in more detail.

1) $(N, D) = (12, 150)$: 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-organize into an additional chain. Recall that the information that the path has formed does not spread within 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 $t = 134$ s, the first *s-bot* is recruited and assembled with the prey, joined by a second *s-bot* 16 s later. While the prey is transported toward the nest, the chain gradually dissolves. During 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 Fig. 7(a), 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. The team structure is illustrated in Fig. 1. This structure was observed in all but two trials in which the *s-bots* operated in the *retrieval* phase. In the remaining two trials all transporting *s-bots* always perceived the path, and therefore there was no division of labor within the transport group.

2) $(N, D) = (12, 240)$: Fig. 8 shows a sequence of images taken during this trial. During the path-formation phase, two chains are formed concurrently [Fig. 8(b)], 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 eight *s-bots* [Fig. 8(c)]. Shortly thereafter, two *s-bots* get recruited and assemble with the prey [Fig. 8(d)]. During retrieval, the *s-bots* of the pulling structure lose sight of the path [see the number of *s-bots* that are in state transport blind in Fig. 7(b)], which gradually dissolves, and the prey is moved in the wrong direction [Fig. 8(e) and (f)]. However, the path gets re-established by an *s-bot* extending the chain in the direction to the prey [Fig. 8(g)]. As a consequence, the transport resumes

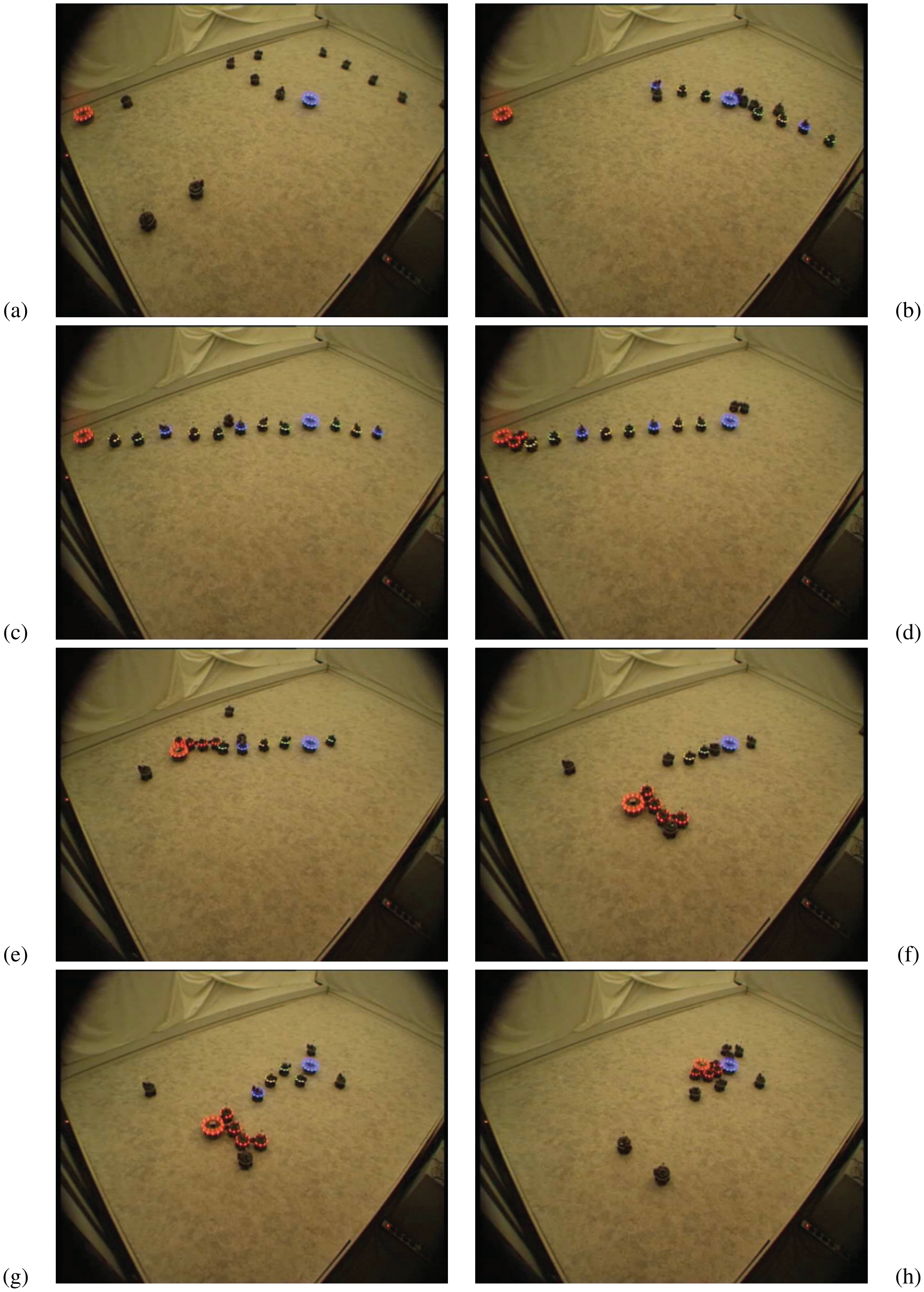


Fig. 8. Sequence of images taken for the trial with group size $N = 12$ *s-bots* and distance $D = 240\,\text{cm}$ between the nest (blue cylindrical object) and the prey (red cylindrical object): (a) $t = 0\,\text{s}$, (b) $t = 140\,\text{s}$, (c) $t = 348\,\text{s}$, (d) $t = 480\,\text{s}$, (e) $t = 708\,\text{s}$, (f) $t = 770\,\text{s}$, (g) $t = 810\,\text{s}$, and (h) $t = 915\,\text{s}$.

and can be completed [Fig. 8(h)]. 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. Similar abilities, involving several *s-bots* helping to reestablish a broken path, were observed in the trials of setups $(N, D) = (7, 120)$ and $(8, 120)$.

3) $(N, D) = (7, 150)$: At time $t = 32\,\text{s}$, a path between nest and prey is already established. At time $t = 187\,\text{s}$, three *s-bots* have been recruited and are assembled with the prey. The four remaining *s-bots* are aggregated in the

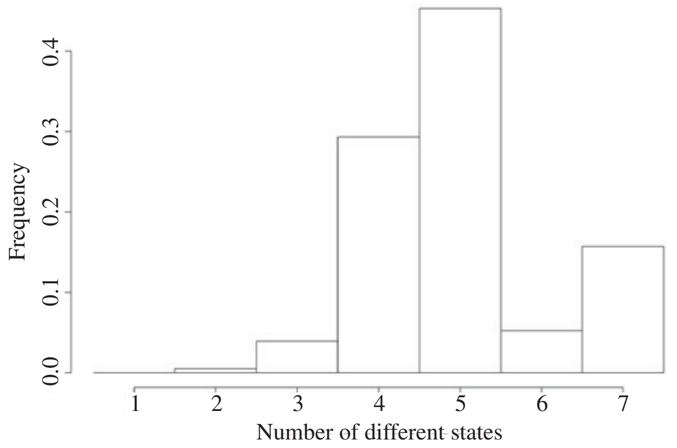


Fig. 9. Number of distinct behavioral roles (i.e., states) an *s-bot* performed during a trial (see Fig. 3). Data from all *s-bots* and all trials of the second set of experiments.

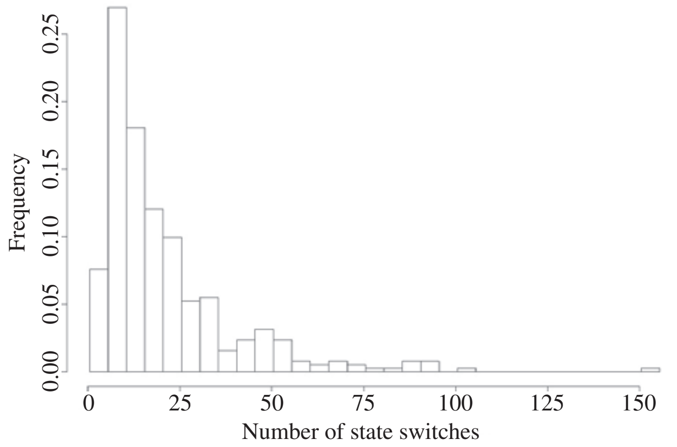


Fig. 10. Number of times an *s-bot* changed its behavioral role (i.e., states) during a trial. Data from all *s-bots* and all trials of the second set of experiments.

chain, forming the path [see Fig. 7(c)]. 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, and 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 sub-tasks. Situations in which collaborating entities exchange some of their members during task performance were observed in many trials. For example, by looking at Tables VI and VII, one can identify five setups—$(N, D) = (4, 60), (6, 120), (7, 150), (10, 90)$, and $(10, 240)$—in which members must have exchanged between different groups. In these five setups, the overall group size (N) is smaller than the size of the group that formed the initial path plus the size of the group of transporters upon task completion.

4) $(N, D) = (7, 240)$: The *s-bot* group is too small to form a path. The state diagram [Fig. 7(d)] shows a high flux between states explore chain (with recovery) 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).

Fig. 9 shows the number of distinct behavioral roles (i.e., states) individual *s-bots* 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 interchangeable. Only in 4% of the cases an *s-bot* performed less than four behaviors during the trial. In 15.7% of the cases, an *s-bot* performed all seven behaviors.

Fig. 10 shows the number of times an *s-bot* changes its behavioral role during the trials of our experiments. The most frequently observed number of changes in behavior belongs to the 6–10 changes range. Note, however, that both the mean and the median number of changes are higher than this range of values (20.9 and 14.5, respectively).

VI. Discussion

In this paper, we have presented an experimental study in which a homogeneous colony of autonomous robots has to solve a complex foraging task. The task requires a range of sub-tasks to be performed including 1) exploration of the environment, 2) formation of a path between a prey and a nest, 3) recruitment of nest mates to the prey, 4) self-assembly into pulling structures, and 5) group transport of the prey back to the nest.

Due to the limited abilities of the robots, the accomplishment of the task requires:

1) concurrent activity of at least i robots;
2) for $i > 2$, division of labor (i.e., the robots need to perform different sub-tasks concurrently),

where $i \in \{2, 3, 4, 5, 6, 8, 9, 10\}$ depends on the experimental setting. We believe that the task is one of the most complex tasks addressed by a swarm robotic system to date. Similar foraging tasks are reported in the literature (e.g., [9], [44], [45], [68], [77], [79], [82]). However, they typically do not require any complex division of labor. For example, in some research the object to be moved can be transported by a single robot. In other cases, the robots can rely on some form of global communication or perception (e.g., a strong light source indicating the goal location), again obviating the need for division of labor. In general, the problem we investigated provides a framework that captures the essence of a variety of problems that are addressed at the collective level in social insect colonies.

We developed a decentralized control algorithm. Each robot executes a copy of this algorithm, and thus all robots have an identical control. The robots do not require any explicit knowledge of the environment beyond their local perceptual range. They make little use of memory and follow relatively simple rules. Overall, one might believe that the members of our robot colony are subjected to similar constraints as individuals of social insect colonies.

A series of experimental results from systematic trials with up to 12 physical robots confirm the efficacy of the system.[8] 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 behaviors 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 behavior during the transport phase. A recovery mechanism allowing suspension of the transport behavior (e.g., see [47]) might have prevented stagnation in such circumstances.

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 toward 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 take another role under other circumstances. As the assignment of individual roles was context-dependent, it changed frequently. Still, the assignment followed static rules (deterministic and stochastic ones), and as such the system was not adaptive (as opposed, for example, to systems presented in [49], [55], [65]).

Our study shows that a colony of robots, by self-organizing, can display a dynamically changing hierarchy of teamwork (with cooperation occurring also among higher order entities such as groups and teams). The higher order entities (including the entire system) proved surprisingly robust with respect to the inaccurate and sometimes malfunctioning behavior of their component modules—parts of a robot such as the tracks, entire robots, and even groups of robots broke down or exhibited unexpected behavior. In this respect, it is worth noting that we have conducted our experiments using a physical robotic system. We believe that by using computer simulations instead, one would have hardly observed the response of the system to the range of circumstances described above. Moreover, by using a physical system we can be certain that our robots have not only addressed the high level coordination problems (e.g., to organize into a hierarchy of teamwork), but the low level coordination problems (e.g., grasping and manipulating objects) as well. A challenge that remains is to design robotic systems that can cope with more realistic natural environments.

We believe that our experiments are among the most sophisticated examples of self-organization in robotics to date. The study shows that complex forms of division of labor can indeed result from the interactions of individuals that follow relatively simple and local rules. The study also shows that teamwork requires neither individual recognition (the robots we use are interchangeable) nor differences between individuals (the robots we use are homogeneous in terms of "morphology" and "brain"). This result might also contribute to the ongoing debate on the role of these characteristics in the division of labor in social insects.

Acknowledgment

The authors thank Prof. N. R. Franks for stimulating discussions that helped in the preparation of the manuscript. In addition, the authors thank all the members of the SWARMBOTS and SWARMANOID projects for their support.

References

[1] Y. Aiyama, M. Hara, T. Yabuki, J. Ota, and T. Arai, "Cooperative transportation by two four-legged robots with implicit communication," *Robot. Auton. Syst.*, vol. 29, no. 1, pp. 13–19, Oct. 1999.

[2] C. Anderson and N. R. Franks, "Teams in animal societies," *Behav. Ecol.*, vol. 12, no. 5, pp. 534–540, Sep. 2001.

[3] C. Anderson and N. R. Franks, "Teamwork in animals, robots and humans," *Adv. Stud. Behav.*, vol. 33, pp. 1–48, Dec. 2003.

[4] C. Anderson and E. McMillan, "Of ants and men: Self-organized teams in human and insect organizations," *Emergence: Complexity Organization*, vol. 5, no. 2, pp. 29–41, Jun. 2003.

[5] C. Anderson and D. W. McShea, "Intermediate-level parts in insect societies: Adaptive structures that ants build away from the nest," *Insect. Soc.*, vol. 48, no. 4, pp. 291–301, Dec. 2001.

[6] C. Anderson and F. L. W. Ratnieks, "Task partitioning in insect societies: Novel situations," *Insect. Soc.*, vol. 47, no. 2, pp. 198–199, May 2000.

[7] C. Anderson, G. Theraulaz, and J.-L. Deneubourg, "Self-assemblages in insect societies," *Insect. Soc.*, vol. 49, no. 2, pp. 99–110, May 2002.

[8] R. C. Arkin, "Motor schema-based mobile robot navigation," *Int. J. Robot. Res.*, vol. 8, no. 4, pp. 92–112, Aug. 1989.

[9] R. C. Arkin, "Cooperation without communication: Multiagent schema-based robot navigation," *J. Robotic Syst.*, vol. 9, no. 3, pp. 351–364, May 1992.

[10] R. C. Arkin, *Behavior-Based Robotics*. Cambridge, MA: MIT Press, 1998.

[11] M. A. Batalin and G. S. Sukhatme, "Spreading out: A local approach to multi-robot coverage," in *Proc. 6th Int. Symp. Distributed Autonomous Robotic Syst.*. Berlin, Germany: Springer-Verlag, 2002, pp. 373–382.

[12] R. Bellman, *Dynamic Programming*. Princeton, NJ: Princeton Univ. Press, 1957.

[13] S. N. Beshers and J. H. Fewell, "Models of division of labor in social insects," *Annu. Rev. Entomol.*, vol. 46, pp. 413–440, Jan. 2001.

[14] E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm Intell.: From Natural to Artificial Syst.*. New York: Oxford Univ. Press, 1999.

[15] A. Castano, A. Behar, and P. M. Will, "The Conro modules for reconfigurable robots," *IEEE/ASME Trans. Mechatronics*, vol. 7, no. 4, pp. 403–409, Dec. 2002.

[16] J.-L. Deneubourg, S. Aron, S. Goss, and J.-M. Pasteels, "The self-organizing exploratory pattern of the Argentine ant," *J. Insect Behav.*, vol. 3, no. 2, pp. 159–168, Mar. 1990.

[17] J. Desai, C.-C. Wang, M. Žefran, and V. Kumar, "Motion planning for multiple mobile manipulators," in *Proc. 1996 IEEE Int. Conf. Robotics Automat.*, vol. 3. New York: IEEE, 1996, pp. 2073–2078.

[18] B. R. Donald, J. Jennings, and D. Rus, "Information invariants for distributed manipulation," *Int. J. Robot. Res.*, vol. 16, no. 5, pp. 673–702, Oct. 1997.

[19] M. Dorigo and M. Birattari, "Swarm intelligence," *Scholarpedia*, vol. 2, no. 9, p. 1462, 2007.

[8] A movie clip, which shows the retrieval of the prey by a colony of 12 robots, is available as Online supplementary material at http://ieeexplore.ieee.org. Further multimedia material is available at http://iridia.ulb.ac.be/supp/IridiaSupp2008-015.

[20] A. Drogoul and J. Ferber, "From tom thumb to the dockers: Some experiments with foraging robots," in *Proc. 2nd Int. Conf. Simulation Adaptive Behavior From Animals to Animats*. Cambridge, MA: MIT Press, 1992, pp. 451–459.

[21] N. R. Franks, "Teams in social insects: Group retrieval of prey by army ants (*Eciton burchelli*, Hymenoptera: Formicidae)," *Behav. Ecol. Sociobiol.*, vol. 18, no. 6, pp. 425–429, May 1986.

[22] T. Fukuda, S. Nakagawa, Y. Kawauchi, and M. Buss, "Self organizing robots based on cell structures-CEBOT," in *Proc. 1988 IEEE Int. Workshop Intell. Robots*. New York: IEEE, Nov. 1988, pp. 145–150.

[23] T. Fukuda and T. Ueyama, *Cellular Robotics and Micro Robotic Systems*. London, U.K.: World Scientific, 1994.

[24] T. Fukuda, T. Ueyama, and K. Sekiyama, "Distributed intelligent systems in cellular robotics," *Artificial Intell. Ind. Decision Making, Control and Automation*, Dordrecht, Netherlands: Kluwer, 1995, pp. 225–246.

[25] S. Garnier, J. Gautrais, and G. Theraulaz, "The biological principles of swarm intelligence," *Swarm Intell.*, vol. 1, no. 1, pp. 3–31, Jun. 2007.

[26] B. P. Gerkey and M. J. Matarić, "Sold!: Auction methods for multirobot coordination," *IEEE Trans. Robot. Autom.*, vol. 18, no. 5, pp. 758–768, Oct. 2002.

[27] D. M. Gordon, "Control without hierarchy," *Nature*, vol. 446, no. 7132, p. 143, Mar. 2007.

[28] S. Goss and J.-L. Deneubourg, "Harvesting by a group of robots," in *Proc. 1st Eur. Conf. Artificial Life*. Cambridge, MA: MIT Press, 1992, pp. 195–204.

[29] 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 (in French)," *Insect. Soc.*, vol. 6, no. 1, pp. 41–80, Mar. 1959.

[30] R. Groß, M. Bonani, F. Mondada, and M. Dorigo, "Autonomous self-assembly in swarm-bots," *IEEE Trans. Robot.*, vol. 22, no. 6, pp. 1115–1130, Dec. 2006.

[31] R. Groß and M. Dorigo, "Cooperative transport of objects of different shapes and sizes," in *Proc. 4th Int. Workshop Ant Colony Optimization Swarm Intell. (ANTS'04)*, LNCS vol. 3172. Berlin, Germany: Springer-Verlag, 2004, pp. 106–117.

[32] R. Groß and M. Dorigo, "Evolving a cooperative transport behavior for two simple robots," in *Proc. 6th Int. Conf. Artificial Evol., Evol. Artificielle*, LNCS vol. 2936. Berlin, Germany: Springer-Verlag, 2004, pp. 305–317.

[33] R. Groß and M. Dorigo, "Group transport of an object to a target that only some group members may sense," in *Proc. 8th Int. Conf. Parallel Problem Solving From Nature*, LNCS vol. 3242. Berlin, Germany: Springer-Verlag, 2004, pp. 852–861.

[34] R. Groß and M. Dorigo, "Self-assembly at the macroscopic scale," *Proc. IEEE*, vol. 96, no. 9, pp. 1490–1508, Sep. 2008.

[35] R. Groß and M. Dorigo, "Evolution of solitary and group transport behaviors for autonomous robots capable of self-assembling," *Adapt. Behav.*, vol. 16, no. 5, pp. 285–305, Oct. 2008.

[36] R. Groß, F. Mondada, and M. Dorigo, "Transport of an object by six pre-attached robots interacting via physical links," in *Proc. 2006 IEEE Int. Conf. Robotics Automat.*, New York: IEEE, May 2006, pp. 1317–1323.

[37] R. Groß, E. Tuci, M. Dorigo, M. Bonani, and F. Mondada, "Object transport by modular robots that self-assemble," in *Proc. 2006 IEEE Int. Conf. Robotics Automat.*, New York: IEEE, May 2006, pp. 2558–2564.

[38] B. Hölldobler and E. O. Wilson, *The Ants*. Cambridge, MA: Harvard Univ. Press, 1990.

[39] S. Ichikawa and F. Hara, "An experimental realization of cooperative behavior of multi-robot system," in *Proc. 2nd Int. Symp. Distributed Autonomous Robotic Syst.*, Berlin, Germany: Springer-Verlag, 1994, pp. 224–234.

[40] S. Ichikawa and F. Hara, "Experimental characteristics of multiple-robots behaviors in communication network expansion and object-fetching," in *Proc. 3rd Int. Symp. Distributed Autonomous Robotic Syst.*, Berlin, Germany: Springer-Verlag, 1996, pp. 183–194.

[41] R. L. Jeanne, "The evolution of the organization of work in social insects," *Monit. Zool. Ital. Suppl.*, vol. 20, no. 2, pp. 119–133, 1986.

[42] M. W. Jørgensen, E. H. Østergaard, and H. H. Lund, "Modular ATRON: Modules for a self-reconfigurable robot," in *Proc. 2004 IEEE/RSJ Int. Conf. Intell. Robots Syst.*, vol. 2. New York, Oct. 2004, pp. 2068–2073.

[43] K. Kosuge and T. Oosumi, "Decentralized control of multiple robots handling an object," in *Proc. 1996 IEEE/RSJ Int. Conf. Intell. Robots Syst.*, vol. 1. New York: IEEE, 1996, pp. 318–323.

[44] M. J. B. Krieger, J.-B. Billeter, and L. Keller, "Ant-like task allocation and recruitment in cooperative robots," *Nature*, vol. 406, no. 6799, pp. 992–995, Aug. 2000.

[45] C. R. Kube and E. Bonabeau, "Cooperative transport by ants and robots," *Robot. Auton. Syst.*, vol. 30, no. 1–2, pp. 85–101, Jan. 2000.

[46] C. R. Kube and H. Zhang, "Collective robotics: From social insects to robots," *Adapt. Behav.*, vol. 2, no. 2, pp. 189–218, Sep. 1993.

[47] C. R. Kube and H. Zhang, "Stagnation recovery behaviours for collective robotics," in *Proc. 1994 IEEE/RSJ/GI Int. Conf. Intell. Robots Syst.*, vol. 3. New York: IEEE, 1994, pp. 1883–1890.

[48] C. R. Kube and H. Zhang, "Task modelling in collective robotics," *Auton. Robot.*, vol. 4, no. 1, pp. 53–72, Mar. 1997.

[49] T. H. Labella, M. Dorigo, and J.-L. Deneubourg, "Division of labour in a group of robots inspired by ants' foraging behaviour," *ACM Trans. Auton. Adapt. Syst.*, vol. 1, no. 1, pp. 4–25, Sep. 2006.

[50] Q. Li, M. De Rosa, and D. Rus, "Distributed algorithms for guiding navigation across a sensor network," in *Proc. 9th Annu. Int. Conf. Mobile Comput. Networking*. New York: ACM, 2003, pp. 313–325.

[51] M. Mamei and F. Zambonelli, "Physical deployment of digital pheromones through RFID technology," in *Proc. 2005 IEEE Swarm Intell. Symp.*, New York: IEEE, Jun. 2005, pp. 281–288.

[52] M. Mamei and F. Zambonelli, "Physical deployment of digital pheromones through RFID technology," in *Proc. 4th Int. Joint Conf. Autonomous Agents Multiagent Syst.*, New York: ACM, 2005, pp. 1353–1354.

[53] M. J. Matarić, M. Nilsson, and K. T. Simsarian, "Cooperative multi-robot box-pushing," in *Proc. 1995 IEEE/RSJ Int. Conf. Intell. Robots Syst.*, vol. 3. New York: IEEE, 1995, pp. 556–561.

[54] 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. Autom.*, vol. 18, no. 5, pp. 769–780, Oct. 2002.

[55] T. Mizuguchi and K. Sugawara, "Proportion regulation in task allocation systems," *IEICE Trans. Fundamentals Electron., Commun. Comput. Sci.*, vol. E98-A, no. 10, pp. 2745–2751, Oct. 2006.

[56] M. W. Moffett, "Ant foraging," *Natl. Geogr. Res. Explor.*, vol. 8, no. 2, pp. 220–231, 1992.

[57] 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. Autom. Mag.*, vol. 12, no. 2, pp. 21–28, Jun. 2005.

[58] S. Murata, E. Yoshida, A. Kamimura, H. Kurokawa, K. Tomita, and S. Kokaji, "M-TRAN: Self-reconfigurable modular robotic system," *IEEE/ASME Trans. Mechatronics*, vol. 7, no. 4, pp. 431–441, Dec. 2002.

[59] S. Nouyan, "Path formation and goal search in swarm robotics," DEA Thesis, IRIDIA, CoDE, Univ. Libre de Bruxelles, Brussels, Belgium, Tech. Rep. TR/IRIDIA/2004-14, 2004.

[60] S. Nouyan, A. Campo, and M. Dorigo, "Path formation in a robot swarm: Self-organized strategies to find your way home," *Swarm Intell.*, vol. 2, no. 1, pp. 1–23, Mar. 2008.

[61] S. Nouyan and M. Dorigo, "Chain based path formation in swarms of robots," in *Proc. 5th Int. Workshop Ant Colony Optimization Swarm Intell.*, LNCS vol. 4150. Berlin, Germany: Springer-Verlag, 2006, pp. 120–131.

[62] K. J. O'Hara and T. R. Balch, "Pervasive sensor-less networks for cooperative multi-robot tasks," in *Proc. 7th Int. Symp. Distributed Autonomous Robotic Syst.*, Berlin, Germany: Springer-Verlag, 2007, pp. 305–314.

[63] G. F. Oster and E. O. Wilson, *Caste and Ecology in the Social Insects*. Princeton, NJ: Princeton Univ. Press, 1979.

[64] L. E. Parker, "Adaptive heterogeneous multi-robot teams," *Neurocomput.*, vol. 28, no. 1–3, pp. 75–92, Oct. 1999.

[65] L. E. Parker, "Lifelong adaptation in heterogeneous multi-robot teams: Response to continual variation in individual robot performance," *Auton. Robot.*, vol. 8, no. 3, pp. 239–267, Jun. 2000.

[66] D. Payton, M. Daily, R. Estowski, M. Howard, and C. Lee, "Pheromone robotics," *Auton. Robot.*, vol. 11, no. 3, pp. 319–324, Nov. 2001.

[67] D. Payton, R. Estkowski, and M. Howard, "Pheromone robotics and the logic of virtual pheromones," in *Proc. 1st Int. Workshop Swarm Robotics at SAB 2004*, LNCS vol. 3342. Berlin, Germany: Springer-Verlag, 2005, pp. 45–57.

[68] A. Perez-Uribe, D. Floreano, and L. Keller, "Effects of group composition and level of selection in the evolution of cooperation in artificial ants," in *Proc. 7th Eur. Conf. Artificial Life*, LNCS vol. 2801. Berlin, Germany: Springer-Verlag, 2003, pp. 128–137.

[69] F. L. W. Ratnieks and C. Anderson, "Task partitioning in insect societies," *Insect. Soc.*, vol. 46, no. 2, pp. 95–108, May 1999.

[70] M. Rubenstein, K. Payne, P. Will, and W.-M. Shen, "Docking among independent and autonomous CONRO self-reconfigurable robots," in *Proc. 2004 IEEE Int. Conf. Robotics Automation*, vol. 3. New York: IEEE, Apr. 2004, pp. 2877–2882.

[71] D. Rus, Z. Butler, K. Kotay, and M. Vona, "Self-reconfiguring robots," *Commun. ACM*, vol. 45, no. 3, pp. 39–45, Mar. 2002.

[72] A. B. Sendova-Franks and N. R. Franks, "Self-assembly, self-organization and division of labour," *Philos. T. Roy. Soc. B*, vol. 354, no. 1388, pp. 1395–1405, Aug. 1999.

[73] T. G. Sugar and V. Kumar, "Control of cooperating mobile manipulators," *IEEE Trans. Robot. Autom.*, vol. 18, no. 1, pp. 94–103, Feb. 2002.

[74] H. Sugie, Y. Inagaki, S. Ono, H. Aisu, and T. Unemi, "Placing objects with multiple mobile robots-mutual help using intention inference," in *Proc. 1995 IEEE Int. Conf. Robotics Automation*, vol. 2. New York: IEEE, 1995, pp. 2181–2186.

[75] J. Svennebring and S. Koenig, "Building terrain covering ant robots: A feasibility study," *Auton. Robot.*, vol. 16, no. 3, pp. 313–332, May 2004.

[76] E. A. Tibbetts, "Visual signals of individual identity in the wasp *Polistes fuscatus*," *P. Roy. Soc. Lond. B Bio.*, vol. 269, no. 1499, pp. 1423–1428, Jul. 2002.

[77] E. Tuci, R. Groß, V. Trianni, F. Mondada, M. Bonani, and M. Dorigo, "Cooperation through self-assembly in multi-robot systems," *ACM Trans. Auton. Adapt. Syst.*, vol. 1, no. 2, pp. 115–150, Dec. 2006.

[78] 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*, vol. 33, no. 5, pp. 537–549, Sep. 2003.

[79] B. B. Werger and M. J. Matarić, "Robotic food chains: Externalization of state and program for minimal-agent foraging," in *Proc. 4th Int. Conf. Simulation Adaptive Behavior*, Cambridge, MA: MIT Press, 1996, pp. 625–634.

[80] G. M. Whitesides and B. Grzybowski, "Self-assembly at all scales," *Science*, vol. 295, no. 5564, pp. 2418–2421, Mar. 2002.

[81] E. O. Wilson, *Sociobiology: The New Synthesis*. Cambridge, MA: Harvard Univ. Press, 1975.

[82] S. Yamada and J. Saito, "Adaptive action selection without explicit communication for multirobot box-pushing," *IEEE Trans. Syst., Man, Cybern. C*, vol. 31, no. 3, pp. 398–404, Aug. 2001.

[83] M. Yim, Y. Zhang, and D. Duff, "Modular robots," *IEEE Spectr.*, vol. 39, no. 2, pp. 30–34, Feb. 2002.

[84] V. Zykov, E. Mytilinaios, B. Adams, and H. Lipson, "Self-reproducing machines," *Nature*, vol. 435, no. 7039, pp. 163–164, May 2005.

Shervin Nouyan received the Diplom-Ingenieur degree in electrical engineering in 2002 from the Technische Universität München, Munich, Germany, the Diplôme d'Études Approfondies degree in applied science in 2004 and the Docteur en Sciences de l'Ingénieur degree in 2008 from the Université Libre de Bruxelles, Brussels, Belgium.

He is currently working as an Engineer at Audi, Ingolstadt, Germany.

Roderich Groß (S'06–M'08) received the Diploma degree in computer science from Universität Dortmund, Dortmund, Germany, in 2001, the Diplôme d'Études Approfondies degree in applied science in 2003 and the Ph.D. degree in engineering sciences from Université Libre de Bruxelles, Brussels, Belgium, in 2007.

In 2005, he held a short-term JSPS Postdoctoral Research Fellowship at the Department of Control and System Engineering, Tokyo Institute of Technology, Japan. From 2006 to 2007, he was a Research Assistant with the School of Biological Sciences, University of Bristol, U.K. In 2007, he was a Marie Curie Fellow at Unilever R&D, Port Sunlight, U.K. He is currently a Marie Curie Fellow at the EPFL, Lausanne, Switzerland. His current research interests include self-assembling systems, evolutionary robotics, and swarm intelligence.

Dr. Groß is a member of the Editorial Board of the *International Journal of Bio-Inspired Computation* and has been on the Program Committee of many international conferences.

Michael Bonani received the M.Sc. degree in microengineering in 2003 from the Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland, where he is currently working toward the Ph.D. degree in microengineering.

In 2003 and 2004, he was a Research Associate with the SWARM-BOTS project. Since 2005, he has been developing an educational robot called e-puck at EPFL. His research interests include swarm robotics and educational robotics.

Francesco Mondada (S'93–A'95–M'08) received the M.Sc. degree in microengineering and the Ph.D. degree from the Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland.

While working toward the doctoral degree, he cofounded the company K-Team, of which he was both CEO and President for about 5 years. He is one of the three main developers of the Khepera robot, considered as a standard in bio-inspired robotics and used by more than 1000 universities and research centers worldwide. Fully back in research in 2000 and after a short period at the California Institute of Technology, Pasadena, he has been the main developer of the s-bot platform within the SWARM-BOTS project. He has published more than 60 papers in the field of bio-inspired robotics and system-level robot design. He is co-editor of several international conference proceedings. He is currently a Senior Researcher at the EPFL. His interests include the development of innovative mechatronic solutions for mobile and modular robots, the creation of know-how for future embedded applications, and making robot platforms more accessible for education, research, and industrial development.

Dr. Mondada was awarded the Swiss Latsis University prize for his contributions to bio-inspired robotics in 2005.

Marco Dorigo (S'92–M'93–SM'96–F'06) received the Laurea (Master of Technology) degree in industrial technologies engineering in 1986 and the Ph.D. degree in information and systems electronic engineering in 1992 from Politecnico di Milano, Milan, Italy, and the title of Agrégé de l'Enseignement Supérieur, from the Université Libre de Bruxelles, Brussels, Belgium, in 1995.

From 1992 to 1993, he was a Research Fellow at the International Computer Science Institute of Berkeley. In 1993 he was a NATO-CNR Fellow, and from 1994 to 1996 a Marie Curie Fellow. Since 1996, he has been a tenured Researcher of the F.R.S.-FNRS, the Belgian National Funds for Scientific Research, and a Research Director of IRIDIA, the artificial intelligence laboratory of the Université Libre de Bruxelles. He is the inventor of the ant colony optimization metaheuristic. His current research interests include swarm intelligence, swarm robotics, and metaheuristics for discrete optimization.

Dr. Dorigo is the Editor-in-Chief of *Swarm Intelligence*, and an Associate Editor or Member of the Editorial Board of many journals in computational intelligence and adaptive systems. He was awarded the Italian Prize for Artificial Intelligence in 1996, the Marie Curie Excellence Award in 2003, the Dr A. De Leeuw-Damry-Bourlart award in Applied Sciences in 2005, and the Cajastur International Prize for Soft Computing in 2007.