Université Libre de Bruxelles

Institut de Recherches Interdisciplinaires et de Développements en Intelligence Artificielle

Evolving Self-Organizing Behaviors for a *Swarm-bot*

Marco Dorigo, Vito Trianni, Erol Şahin, Roderich Groß, Thomas H. Labella, Gianluca Baldassarre, Stefano Nolfi, Jean-Louis Deneubourg, Francesco Mondada, Dario Floreano, Luca M. Gambardella

Technical Report No.

TR/IRIDIA/2003-11

June 2004

Published in *Autonomous Robots*,17(2-3):223–245, 2004

Evolving Self-Organizing Behaviors for a *Swarm-bot*

Marco Dorigo[†], Vito Trianni[†], Erol Şahin[°], Roderich Groß[†], Thomas H. Labella[†], Gianluca Baldassarre[§], Stefano Nolfi[§], Jean-Louis Deneubourg[‡], Francesco Mondada[⋆], Dario Floreano[⋆], Luca M. Gambardella[#]

[†] *IRIDIA - Université Libre de Bruxelles, Belgium*

[°] *KOVAN - Department of Computer Engineering, Middle East Technical University, Ankara, Turkey*

[§] *Institute of Cognitive Sciences and Technologies - CNR, Roma, Italy*

[‡] *CENOLI - Université Libre de Bruxelles, Belgium*

[⋆] *ASL - Swiss Federal Institute of Technology, Lausanne, Switzerland*

[#] *IDSIA, Manno-Lugano, Switzerland*

Abstract

In this paper, we introduce a self-assembling and self-organizing artifact, called a *swarm-bot*, composed of a swarm of *s-bots*, mobile robots with the ability to connect to and to disconnect from each other. We discuss the challenges involved in controlling a *swarm-bot* and address the problem of synthesizing controllers for the *swarm-bot* using artificial evolution. Specifically, we study aggregation and coordinated motion of the *swarm-bot* using a physics-based simulation of the system. Experiments, using a simplified simulation model of the *s-bots*, show that evolution can discover simple but effective controllers for both the aggregation and the coordinated motion of the *swarm-bot*. Analysis of the evolved controllers shows that they have properties of scalability, that is, they continue to be effective for larger group sizes, and of generality, that is, they produce similar behaviors for configurations different from those they were originally evolved for. The portability of the evolved controllers to real *s-bots* is tested using a detailed simulation model which has been validated against the real *s-bots* in a companion paper in this same special issue.

Keywords: Swarm robotics, swarm intelligence, swarm-bot, evolutionary robotics.

1 Introduction

Swarm robotics is an emergent field of collective robotics that studies robotic systems composed of *swarms* of robots tightly interacting and cooperating to reach their goal. Based on the social insect metaphor [6], swarm robotics emphasizes aspects such as decentralization of the control, limited communication abilities among robots, use of local information, emergence of global behavior and robustness. In a swarm robotic system, although each single robot of the swarm is a fully autonomous robot, the swarm as a whole can solve problems that the single robot cannot cope with because of physical constraints or limited capabilities. This paper

Technical Report IRIDIA/TR/2003-11 of IRIDIA, Université Libre de Bruxelles, Brussels, Belgium. June 2004.

addresses the problem of synthesizing controllers for a robotic swarm. In particular, we discuss the challenges faced, and report some of the results obtained up to now, within the SWARM-BOTS project.[1]

The aim of the SWARM-BOTS project is the development of a new robotic system, called a *swarm-bot* [24, 18]. A *swarm-bot* is defined as an artifact composed of a swarm of *s-bots*, mobile robots with the ability to connect to and to disconnect from each other. A companion paper [17], accepted for publication in this same special issue, discusses the hardware and simulation realization of our swarm robotic system.[2] *S-bots* have simple sensors and motors and limited computational capabilities. Their physical links are used to assemble into a *swarm-bot* able to solve problems that cannot be solved by a single *s-bot*. In the *swarm-bot* form, the *s-bots* are attached to each other and, when needed, become a single robotic system that can move and reconfigure. For example, the *swarm-bot* might have to take different shapes in order to go through a narrow passage or overcome an obstacle. Physical connections between *s-bots* are essential for solving many collective tasks. *S-bots* can form pulling chains to retrieve a heavy object. Also, during navigation on rough terrain, physical links can serve as support if the *swarm-bot* has to pass over a hole larger than a single *s-bot*, or when it has to pass through a steep concave region. However, for tasks such as searching for a goal location or tracing an optimal path to a goal, a swarm of unconnected *s-bots* can be more efficient.

In this paper, we focus on providing the *s-bots* with two basic abilities that are of fundamental importance in many cooperative tasks: *aggregation* and *coordinated motion*. Aggregation is of particular interest since it stands as a prerequisite for other forms of cooperation. For instance, in order to assemble into a *swarm-bot*, *s-bots* should first be able to aggregate. Therefore, the aggregation ability can be considered as the precondition for other tasks that the *swarm-bot* is expected to be able to carry out. Coordinated motion represents another basic ability for a *swarm-bot* formed by connected *s-bots* that, being independent in their control, must coordinate their actions to choose a common direction of motion. This coordination ability is essential for an efficient motion of the *swarm-bot* as a whole. Aggregation and coordinated motion are the main focus of the experiments presented in this paper,[3] which is structured as follows.

We first address, in Section 2, the general problem of synthesizing the control system of the *s-bots* using artificial evolution. Then, in Section 3 we describe our experimental methodology. In Section 4 and 5, we present the results obtained evolving simple neural networks for the aggregation task and for the coordinated motion task. These evolved controllers are tested in a very detailed and realistic simulation of the *swarm-bot*, and the results of these tests are presented in Section 6. Finally, Section 7 describes some related works and Section 8 concludes the paper.

2 Challenges

In the previous section, we introduced the *swarm-bot* and some of the tasks that it should be able to perform. Even though this was only a rough description, it suggests that controlling such a system is a challenging problem. Distributedness, robustness, embodiment, locality of sensing, dynamic interactions between *s-bots* are aspects that have to be taken into account when developing a control system

[1]A project funded by the Future and Emerging Technologies Programme (IST-FET) of the European Community, under grant IST-2000-31010.

[2]Details regarding the hardware and simulation of the *swarm-bot* can also be found in the project web-site (`http://www.swarm-bots.org`).

[3]Note that all experiments described in this paper have been carried out in physics-based simulations because the robots were under construction at the time of writing.

for such an artifact. Is it possible to find some basic principles to be followed when facing this challenge?

A possible answer is suggested by the notion of *self-organization* [7]. Self-organization explains how a system can move from a disordered to an ordered state exploiting only local interactions among its components, without any reference to the system as a whole. When in a disordered state, a system's behavior is deeply influenced by the result of random actions of its components. At the change of the value of some parameters, a self-organizing process can be initiated, which exploits two basic mechanisms: *positive* and *negative feedback*. Positive feedback consists in the amplification of some properties of the system that emerge from the random interactions between the individual components: it can be seen as a snowball effect that strengthen exponentially in time these properties. On the contrary, negative feedback serves as a regulatory mechanism, and it is often a result of the amplification itself, that exhausts the resources of the system. Negative and positive feedback cooperate in maintaining a system in a stable state, making it robust against external influences [7].

A form of self-organization of particular interest for our work is *self-assembling*, the self-organized creation of structures. Self-assembling occurs in a wide range of natural systems ranging from chemistry to biology, and it characterizes the behavior of many social insects (for a review, see [1]). Self-organization and self-assembling are fundamental to the SWARM-BOTS project. In fact, *s-bots*, exploiting only local information, should be able to self-organize, self-assemble and coordinate their activities. Thus, understanding the mechanisms that drive the emergence of self-organization is of fundamental importance. If we are able to reproduce the mechanisms observed in self-organizing systems, then we can use them to efficiently control our artificial swarms.

However, designing a self-organizing control system for the *swarm-bot* is not a trivial task. From an engineering perspective, the design problem is generally decomposed into two different phases: (i) the behavior of the system should be described as the result of interactions among individual behaviors, and (ii) the individual behaviors must be encoded into controllers. Both phases are complex because they attempt to decompose a process (the global behavior or the individual one) that emerges from a dynamical interaction among its sub-components (interactions among individuals or between individual actions and the environment).

Nolfi and Floreano [19] claim that, since the individual behavior is the emergent result of the interaction between agent and environment, it is difficult to predict which behavior results from a given set of rules, and which are the rules behind an observed behavior. Similar difficulties are present in the decomposition of the organized behavior of the whole system into interactions among individual behaviors of the system components. Here, the understanding of the mechanisms that lead to the emergence of self-organization must take into account the dynamic interactions among individual components of the system and between these components and the environment. Thus, it is difficult to predict, given a set of individual behaviors, which behavior at the system level will emerge, and it is also difficult to decompose the emergence of a desired global behavior in simple interactions among individuals. The decomposition from the global to the individual behaviors could be simplified by taking inspiration from natural systems, such as insect societies [6]. However, it is not always beneficial to take inspiration from natural processes, because they may differ from the artificial systems in many important aspects (e.g., the physical embodiment, the type of possible interactions between individuals, and so forth), or because there are no natural systems that can be compared to the artificial one.

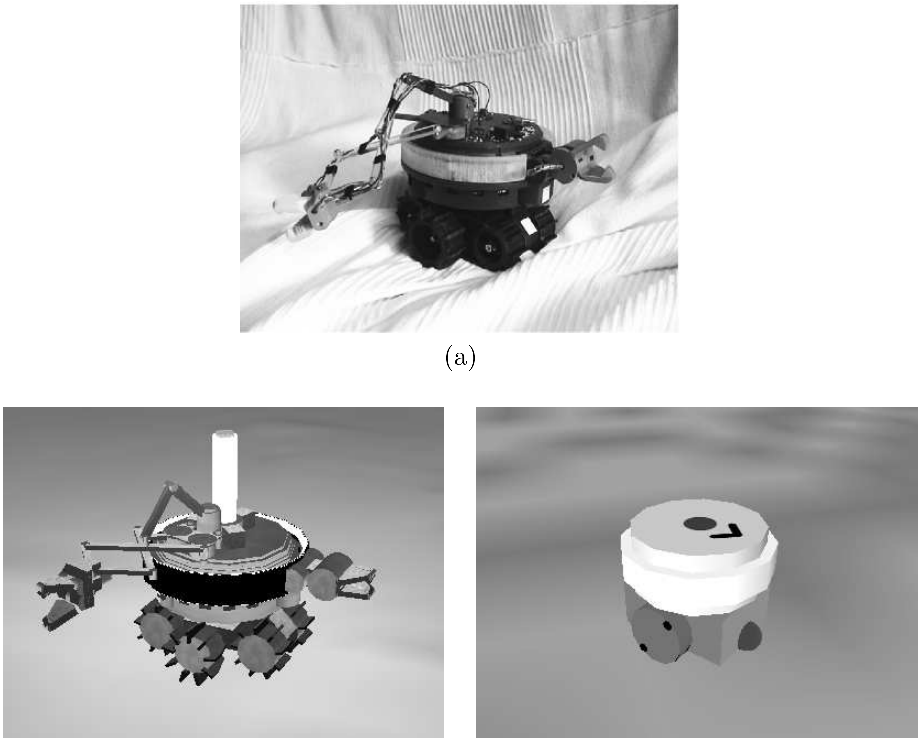
Our working hypothesis is that these problems can be efficiently solved using artificial evolution [19]. Evolution bypasses the problem of decomposition at both the level of finding the mechanisms that lead to the emergent global behavior and at the level of implementing those mechanisms in a controller for the *s-bots*. In fact, it relies on the evaluation of the system as a whole, that is, on the emergence of the desired global behavior starting from the definition of the individual ones. For example, in Section 4 we show how the aggregation problem can be solved by very simple evolved strategies, without the need of decomposition at any level. Moreover, evolution can exploit the richness of possible solutions offered by the dynamic agent-environment interactions [19]. In a multi-agent system such as the *swarm-bot*, these dynamic aspects are enriched not only by the presence of multiple agents, but also by the possible presence of physical links between the agents. Generally, these aspects are difficult to be exploited by manual design. On the contrary, the evolutionary process can take advantage of these dynamic properties of the system to synthesize efficient controllers. Section 5 describes an experimental setup which exemplifies this situation: in this case, physical connections between *s-bots* and their dynamic interactions become the main elements responsible for the efficiency of the evolved behaviors.

3 Experimental Methodology

In this section, we describe the methodology we follow for the design of the *s-bot* controller. As mentioned before, our approach consists in using artificial evolution for this task. However, the use of artificial evolution has some drawbacks. As pointed out by Matarić and Cliff [16], many issues must be addressed when trying to develop controllers for real robots using the evolutionary approach. In particular, a key aspect is the often prohibitive time needed to evolve controllers on real hardware. To overcome this problem, simulations are often used for the evolution of complex behaviors, but rarely evolved controllers have been tested on real hardware. Also, an accurate modeling is needed to deploy simulators that well represent the physical system [13].

Taking into account these challenges, we follow a methodology that can be described as a 5-step process: (i) The real robot is defined along with its hardware details. (ii) A simulator is developed, which gives the possibility to model the real *s-bot* at different levels of detail. (iii) A simple model is chosen, that provides the required speed in order to run the evolutionary experiments in a reasonable amount of time. (iv) The evolved controllers are validated (i.e., tested) using a detailed simulation model, closely related to the hardware. (v) Successful controllers are downloaded and tested on real hardware. In this paper, we stop at the validation of the evolved controllers with the detailed simulations, since there were not enough real *s-bots* available for experimentation. However, as discussed in detail in the companion paper [17], the behaviors of the detailed simulation model and of the real *s-bot* described in [17] were experimentally compared and the results show that the detailed simulation model closely matches the reality, thus suggesting that closing the gap to the real implementation of the evolved controllers should not be too challenging.

The first step of the described methodology brings forward the description of the hardware, along with its features and limits. This description is used to develop the *s-bot* prototype and to design the simulation tool. As shown in Figure 1a, the *s-bot* is provided with a traction system that couples both wheels and tracks (called *treels*©), useful for navigation in moderately rough terrain. Above the traction system, a rotating turret holds many sensory systems and the two grippers for making connections with other robots. One gripper is fixed to the turret and provides a very strong connection mechanism, powerful enough to lift another *s-bot*. The second gripper is mounted on an extensible arm and can provide flexible connections among *s-bots* (more details can be found in [17]).



(a)

(b) (c)

Figure 1: The *s-bot*. (a) The first prototype of the *s-bot*. (b) A graphical representation of the detailed simulation model which closely reproduces the mechanical structure of the *s-bot*. (c) A graphical representation of the simplified model, in which the details of the *s-bot* unnecessary for our experiments are omitted. Both the detailed and the simple models are implemented using physics-based software libraries.

In parallel with the construction of the *s-bot* prototype, the simulation software *Swarmbot3D* has been designed, based on the SDK Vortex$^{\mathrm{TM}}$ toolkit (Critical Mass Labs, Canada), which provides realistic simulations of dynamics and collision of rigid bodies in 3D. The mechanical drawings of the hardware were used for the design of a detailed simulation model for the *s-bot*, shown in Figure 1b. It is possible to notice how all the mechanical parts of the robot where carefully replicated, the only difference being the caterpillar rubber band of the treels system.[4] A number of experiments were conducted to compare the behavior of the real and simulated *s-bots*, leading to a fine tuning of the different parameters influencing both the sensory system and the acting abilities. Noise is also modeled in the simulation to provide realistic behaviors (see [17] for a detailed description)

However, the high degree of precision of the detailed simulation model requires a large amount of computation making simulations too slow to be applicable in evolutionary experiments. Therefore, the simulation models for the *s-bot* were created at different levels of detail, ranging from a simplified model that leaves out many features of the *s-bot*, to a full-fledged realistic model. In the experiments presented in this paper, we used a simplified *s-bot* model[5] which leaves out most of the mechanical details, yet preserving features of the *s-bot* that are considered to be important for the experiments (see Figure 1c).

[4]Experiments showed that in many situations this feature was of minor importance.

[5]This simplified *s-bot* model is called "fast model" in the companion paper [17].

The traction system of the *s-bot* is modeled by four wheels: two lateral motorized wheels which model the external wheels of the real *s-bot*, and two spherical, passive wheels placed in the front and in the back and which serve as support. The four wheels are connected to the chassis, which underpins the rotating turret, modeled as a cylinder. The turret holds a virtual gripper, which is modeled by dynamically creating a joint between two *s-bots* when needed, its position being represented by an arrow painted on the turret. The flexible gripper is not used in the experiments presented in this paper, so it is not included in the simple model. In order to speed up the simulation, spherical collision models are used for all the wheels and for the chassis, as they require less computations, even if they are graphically rendered with different geometries. The sensory systems were simulated either by using a sampling technique [19] or by set of equations. Details will be given in the following sections.

This simplified *s-bot* model provides the required simulation speed in order to run evolutionary experiments within a reasonable time. In the subsequent Sections 4 and 5 we describe the experimental setup and the results obtained for the evolution of self-organizing behaviors for the *swarm-bot*.

4 Evolving Aggregation Behaviors

The evolution of scalable aggregation behaviors is the main focus of the experiments presented in this section. In the following, we first describe the experimental setup, then we analyze the obtained results and we discuss the scalability of the evolved strategies.

4.1 Experimental Setup

The simplified simulation model of the *s-bot* described in Section 3 was used for the evolutionary experiments presented here. In this case, the rotational degree of freedom of the turret with respect to the chassis was not used. Also the gripper was omitted, as the main focus of these experiments was on scalable aggregation, and not on self-assembling. Each *s-bot* has control only on its two motorized wheels, schematically shown in Figure 2a. Additionally, each *s-bot* is equipped with a simulated speaker that can emit a tone for long range signaling. *S-bots* can perceive the intensity of sound using three sound sensors that simulate three directional microphones using a set of equations [3]. The tone emitted by an *s-bot* can be perceived by another *s-bot* from a distance of up to 75 *cm*. Beyond this value, the tone is covered by noise, simulated by adding a random component uniformly distributed within $\pm 5\%$ of the sensor saturation value. Short range detection of obstacles or of other *s-bots* is achieved using 8 proximity sensors, simulated using a sampling technique [19]. Also in this case, noise is simulated by adding a random component uniformly distributed within $\pm 5\%$ of the sensor saturation value. Figure 2b shows the position of the sensors used for this experiment. The environment consists of a square arena surrounded by walls. The size of the arena is chosen to be 3×3 meters and it is bigger than the perceptual range of the *s-bots*, in order to emphasize the locality of sensing.

The evolutionary algorithm used to evolve the controllers utilizes a population of 100 randomly generated binary genotypes. At every generation, the best 20 genotypes are selected for reproduction, and each generates 5 offspring. Each offspring is mutated with a 3% probability of flipping each bit. Recombination is not used. Parents are not copied in the population of the next generation. One evolutionary run lasts 100 generations.

Each genotype encodes the connections weights of a single layer perceptron, a neural network that directly connects the input neurons to the outputs. The perceptron has 12 sensory neurons, that encode the state of the 8 proximity sensors, of the 3 sound sensors and of a bias unit (i.e., a unit whose activation state is always 1.0). Each sensory neuron is directly connected with 2 motor neurons, which control the two wheels setting their speed within the range $[-6.5, +6.5]$ rad/s. Thus, the neural controller is made of $12 \times 2 = 24$ connections, each associated to a weight ranging in the interval $[-10, +10]$ and represented in the genotype with 8 bits. Therefore, the genotype is composed of $24 \times 8 = 192$ bits.

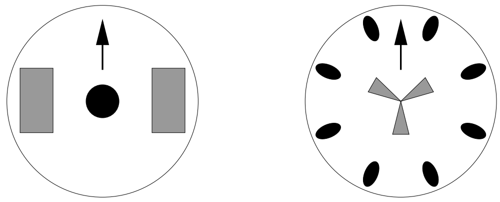
The fitness evaluation of a genotype is repeated 8 times (*epochs*), in order to better estimate the performance value. In each epoch the initial position and orientation of the *s-bots* in the arena is randomly chosen. Each epoch lasts 900 simulation cycles and each cycle simulates 100 ms of real time. In each epoch, the size of the group of *s-bots* is randomly chosen between 4 and 8. Varying the number of *s-bots* used during the evaluation of the genotypes is important to remove an invariant that could be exploited to synthesize aggregation behaviors that are not scalable [31]. The genotype is mapped into a neural network that is cloned and assigned to each *s-bot* that takes part in the experiment.

In order to evolve scalable aggregation behaviors, we devised a fitness function that takes into account the number n of *s-bots* used in each evaluation. This is justified by the need to have comparable performance measures, no matter the size of the group that is evaluated. The fitness F of a genotype is the average of the fitness evaluation F_e of each epoch e. In each epoch, the genotype is evaluated for its ability to minimize the average distance of all *s-bots* from the center of mass of the group. This is called *aggregation quality* $D(t)$. Additionally, a second component $S(t)$, called *motion quality*, has been introduced. The motion quality accounts for straight motion of *s-bots* and was introduced to avoid a turning-on-the-spot behavior of the *s-bots* when aggregated, which was observed to be a local optimum in which the evolved strategies often converged using the aggregation quality only.

The fitness F_e is measured averaging at the end of the epoch the product of the $D(t)$ and $S(t)$ components:

$$F_e = \frac{1}{W} \sum_{t=t_W}^{T} D(t) \cdot S(t), \tag{1}$$

where $T = 900$ is the total number of sensory-motor simulation cycles of one epoch, $t_W = T - W$ is the starting point of the time window in which the fitness is computed and $W = 800/n$ is the length of the time window. The fitness is measured at the end of the epoch, in order to leave to the *s-bots* enough time to search for each other. The length of the time window W varies with the group size n, being shorter



(a) (b)

Figure 2: A schematic *s-bot* seen from the top. The arrows show the front direction. (a) Actuators: the two gray rectangles indicate the motorized wheels. The black circle indicates the speaker, which continuously emits a tone. (b) Sensors: 8 proximity sensors (black ellipses) and three directional microphones (gray triangles).

for bigger group sizes. In this way, we do not penalize a slower aggregation process of bigger groups. In the following, we detail the computation of the two components $D(t)$ and $S(t)$.

- The aggregation quality $D(t)$ is related to the average distance of the *s-bots* from their center of mass. Thus, we first compute the distance $d_i(t)$ of each robot i from the center of mass of the group at simulation cycle t:

$$d_i(t) = \|\mathbf{X}_i(t) - \frac{1}{n}\sum_{j=1}^{n}\mathbf{X}_j(t)\|, \tag{2}$$

  where $\mathbf{X}_i(t)$ is the position vector of the i^{th} *s-bot* at time t. This value is used to compute the aggregation quality $D_i(t)$ of the i^{th} *s-bot* as follows:

$$D_i(t) = \begin{cases} 1 & \text{if } d_i(t) < r(n) \\ \frac{R(n)-d_i(t)}{R(n)-r(n)} & \text{if } r(n) \leq d_i(t) < R(n) \\ 0 & \text{if } d_i(t) \geq R(n) \end{cases}, \tag{3}$$

  where $r(n)$ is the radius (in centimeters) of the smallest circle that can contain n *s-bots*,[6] and $R(n) = r(n) + k$ ($k = 100$ was experimentally found to be a good value). This measure of the aggregation quality scales well with the group size, as it ensures that, no matter the group size, the maximum quality value is achievable. However, it is difficult to compute $r(n)$ for every group size (see also [35], this issue). Thus, we decided to approximate $r(n)$ with an upper bound $\tilde{r}(n)$, defined as the radius of the smallest circle that has n robots positioned on the perimeter:

$$\tilde{r}(n) = \frac{r_s}{\sin(\pi/n)}, \tag{4}$$

  where r_s is the radius of an *s-bot*. This upper bound is exact for group sizes from 2 to 6, but it diverges for bigger group sizes, for which it overestimates the exact value. This is not a problem for the fitness evaluation in our case, as it is computed using from 4 to 8 *s-bots*. Finally, the aggregation quality $D(t)$ of the group is computed averaging the aggregation quality of every robot.

- The motion quality $S(t)$ accounts for straight motion of *s-bots* and it is computed as the average motion quality $S_i(t)$ of each *s-bot* i:

$$S(t) = \frac{1}{n}\sum_{i=1}^{n} S_i(t) = \frac{1}{n}\sum_{i=1}^{n}\left(1.0 - \frac{|sl_i(t) - sr_i(t)|}{2 \cdot sm}\right), \tag{5}$$

  where $sl_i(t)$ and $sr_i(t)$ are respectively the speed of the left and right wheels of the *s-bot* i at simulation cycle t, and sm is the maximum possible speed. The measure $S_i(t)$ accounts for straight motion of the *s-bot* i, as it takes values near 1 if the two wheels have similar speed, and it is near 0 if the wheels turn in opposite directions.

4.2 Results

The evolutionary experiment was replicated 20 times, starting with different randomly initialized populations. We observed that aggregation behaviors were successfully generated in each replication. Figure 3 plots the average fitness of the

[6] Given that $d_i(t)$ is computed starting from the center of an *s-bot*, $r(n)$ is defined as the radius of the smallest circle that encloses the centers of n *s-bots*.

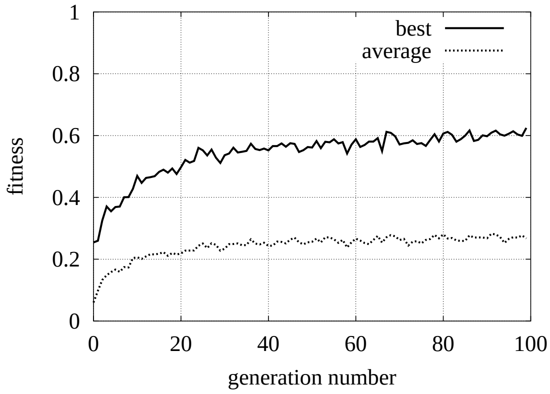


Figure 3: Performance across 100 generations, averaged over 20 replications. The fitness of the best genotype and the average fitness of the population are plotted against the generation number.

20 replications of the experiment. The best genotype of the population reaches in average 60% of the theoretical maximum value. It is important to bear in mind that the fitness is the result of a product, whose factors are values in the range $[0, 1]$. In our case, the value around 60% is the result of two components that have both high performance values, around 80%.

In order to assess the performance achieved by the evolved strategies, for each replication of the experiment, we selected the best 20 genotypes of the last generation, and we evaluated their fitness for 200 times (i.e., 200 epochs). Then, for each replication, we selected the genotype with the highest average fitness. The corresponding values are shown in Table 1, for all the 20 replications. It is possible to notice that these performances are slightly lower than the average fitness values of the best genotypes reached at the end of the evolutions, shown in Figure 3. This is mainly due to an over-estimation of the performance of the best genotype during the evolution. In fact, given a limited number of epochs, the fitness value F of a genotype is just an estimate of its real performance. Since only the best cases are retained by the selection operator, the performance measured during evolution is likely to represent an over-estimate of the real performance that can be obtained by those genotypes.

Table 1: Performance of the best genotype after the post-evaluation for each replication of the experiment. The values are the average fitnesses over 200 evaluations of the best genotype.

Replication	1	2	3	4	5
Fitness	0.487	0.451	0.413	0.590	0.371

Replication	6	7	8	9	10
Fitness	0.460	0.566	0.582	0.604	0.616

Replication	11	12	13	14	15
Fitness	0.588	0.554	0.562	0.636	0.572

Replication	16	17	18	19	20
Fitness	0.574	0.484	0.494	0.655	0.561

A qualitative analysis of the evolved controllers reveals that different replications result in slightly different behaviors. Some similarities can be observed among the evolved solutions. For example, solitary *s-bots* tend to explore the arena moving in large circles and turning away from obstacles when they are too close to them. The evolved solutions differ mainly in the behavior of *s-bots* when they are close to each other. In general, all evolved strategies rely on a delicate balance between attraction to sound sources and repulsion from obstacles, the former being perceived by sound sensors, the latter by proximity sensors. For the sake of simplicity, we will describe here the behavior of the controller produced by the tenth replication of the experiment.[7] This controller not only has a good performance, but it also presents the best scalability properties, as discussed later, in Section 4.3. In this case, the interaction between attraction and repulsion from other *s-bots* creates a "following behavior" that can be observed with small groups of *s-bots* (see Figure 4a). When the number of *s-bots* increases, this ordered "following behavior" is replaced by a disordered motion of the *s-bots*, which continuously change their relative positions, so that the aggregate continuously expands and shrinks, slightly moving across the arena (see Figure 4b). This feature of the evolved strategy is strictly related to scalability, as we discuss in the forthcoming section.

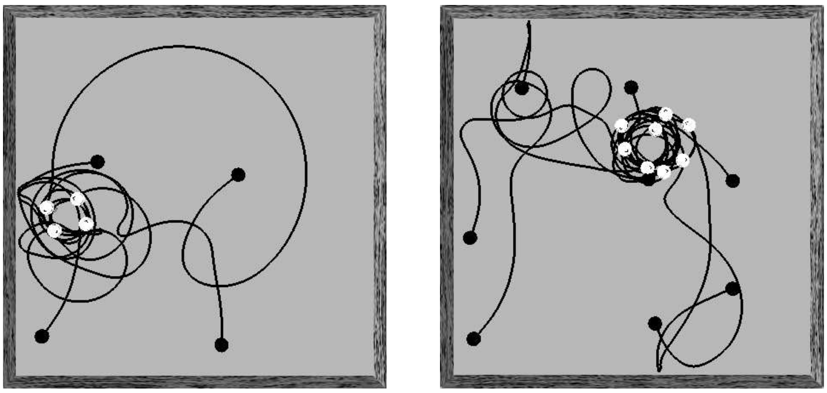


Figure 4: Aggregation behavior. (a) The aggregation of 4 *s-bots* usually produces groups moving in circles. (b) When the group is bigger, the movement is more disordered and the *s-bots* continuously change their relative positions.

4.3 Scalability

The scalability of the best controllers of each evolutionary run was evaluated for *s-bots* groups ranging from 4 to 40. A measure was defined in order to test the aggregation performance of different groups. For this purpose, the aggregation quality introduced in Section 4.1 was redefined using a different approximation for $r(n)$ (see (3) and (4)). In fact, as already mentioned, the upper bound $\tilde{r}(n)$ defined in (4) diverges from the real value with increasing group sizes, which determines an over-estimation of the performance of large groups. We defined a lower bound $\hat{r}(n)$, which is related to the area occupied by the *s-bots*:

$$\hat{r}(n) = r_s \cdot \left(\sqrt{n} - 1\right), \tag{6}$$

where r_s is the radius of an *s-bot*. As shown in Figure 5, the distance between the upper and lower bound increases with the group size. Also this lower bound diverges from the exact value for increasing group sizes, which determines an under-estimation of the performance of large groups. We decided to approximate $r(n)$ with the average $\overline{r}(n)$ of the two defined bounds, shown in Figure 5.

[7]See www.swarm-bots.org/scaling_aggregation.html for some movies of this behavior.

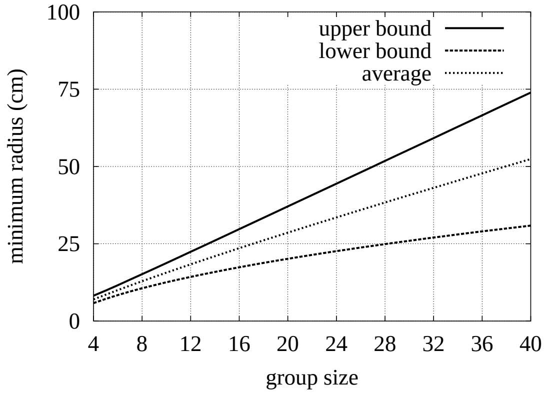


Figure 5: Approximations for $r(n)$. The continuous line plots the upper bound $\tilde{r}(n)$, the dashed line plots the lower bound $\hat{r}(n)$, and the dotted line plots the average $\overline{r}(n)$.

Having chosen a suitable approximation for $r(n)$, we defined the performance measure F_s as follows:

$$F_s = \frac{1}{nW} \sum_{t=t_W}^{T} \sum_{i=1}^{n} D_i(t), \tag{7}$$

where $t_W = T - W$ is the starting point of the evaluation time window. Here, $D_i(t)$ is computed as in (3), but using $\overline{r}(n)$ as an approximation of $r(n)$. In order to give enough time to the aggregation process of large groups, all the evaluations were performed over $T = 2000$ simulation cycles. The time window has $W = 100$ simulation cycles.

We performed 100 evaluations for different group sizes ($n = 4, 8, 12, \ldots, 40$). The results obtained showed that not all the evolved controllers have comparable performance. However, half of the tested controllers present a very good scalability. The best scalable strategy was the one produced by the tenth replication, already analyzed in the previous section. We have mentioned that this controller creates an aggregate that moves across the arena. This is a result of the complex motion of *s-bots* within the aggregate, which in turn is the result of the interaction between attraction to sound sources and repulsion from obstacles. The slow motion of the aggregate across the arena leads to scalability, as an aggregate can continue to move joining solitary *s-bots* or other already formed aggregates, eventually forming a single cluster of *s-bots*.

Figure 6 plots the performance of this controller as a function of the group size. We can see that the performance gracefully degrades when the group size increases over the limit used during evolution. It indicates that the aggregation behavior scales well and is not dependent on some particular settings. The best performance is obtained with 4 *s-bots*, and corresponds to the situation in which all the *s-bots* have an ordered circular motion, that allows them to stay very close to each other. The outliers correspond to situations in which the 4 *s-bots* never met each other in the limited time used for evaluation. When increasing the group size to 8 and 12 *s-bots*, we observe a drop in performance that is mainly due to the transition from the ordered to the disordered motion of the *s-bots* within the aggregate. In this case, the aggregate is more dynamic, continuously changing shape, size and position driven by the complex interactions among the *s-bots*. We observe also a higher variance in the data or more outliers, corresponding to the formation of two or more small aggregates that did not have enough time to join in a single

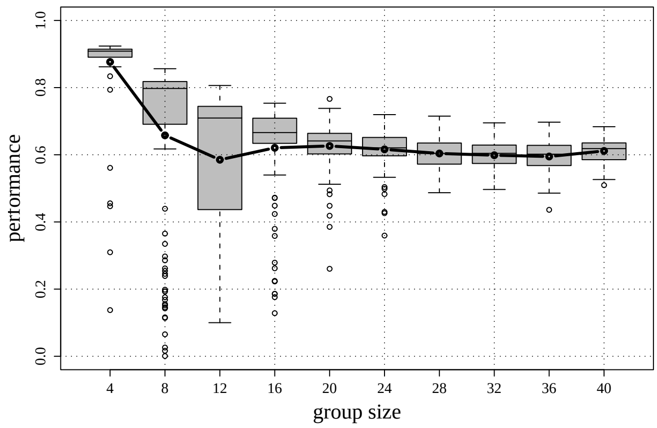


Figure 6: Scalability of the aggregation behavior. The performance for some group sizes (4, 8, 12, . . ., 40 *s-bots*) is shown. The box-plot shows 100 evaluations per box. The average values are indicated by the thick black line. Boxes represent the inter-quartile range of the data, while the horizontal bars inside the boxes mark the median values. The whiskers extends to the most extreme data points within 1.5 of the inter-quartile range from the box. The empty circles mark the outliers.

one. Further increasing the group size, we observe that the performance reaches a stable level. Less outliers are observed and also the variance is reduced, because the increasing density of *s-bots* in the arena makes it easier for smaller groups to aggregate into a single one.

5 Evolving Coordinated Movement

In this section, we consider a *swarm-bot*, made up of a collection of assembled *s-bots*, whose task is to display coordinated movement. In the experiments presented here, we study a *swarm-bot* composed of *s-bots* that are already connected through the grippers. The problem that the *s-bots* have to solve is that their wheels might have different initial directions or might mismatch while moving. In order to coordinate, *s-bots* should be able to collectively choose a common direction of movement having access only to local information.

We will show that evolution can find simple and effective solutions that allow the *s-bots* to move in a coordinate way independently of the topology of the *swarm-bot* and of the type of link with which the *s-bots* are connected (flexible or rigid, see below). Moreover, it will be shown that the evolved *s-bots* also exhibit obstacle avoidance behavior (when placed in an environment with obstacles) and object pulling/pushing behavior (when assembled to or around an object), and scale well to *swarm-bots* of a larger size.

5.1 Experimental Setup

The *swarm-bot* consists of four *s-bots* assembled in a linear structure, as shown in Figure 7. In these experiments, we used the simplified simulation model described in Section 3.

Differently from the previous experiments, *s-bots* have the possibility to rotate their turret with respect to their chassis (this can be done by means of a motorized "hinge joint" that can rotate around the vertical axis). Each *s-bot* is connected to another *s-bot* by means of a physical link between the turrets. The link consists of another "hinge joint" that has a rotation axis parallel to the horizontal plane and is perpendicular to the line formed by the four *s-bots*.

Each *s-bot* is provided with a *traction sensor*, placed in correspondence of the turret-chassis hinge joint, that returns the direction (i.e., the angle with respect to the chassis' orientation) and the intensity of the force of traction (henceforth called "traction") that the turret exerts on the chassis (see Figure 8). The traction intensity is scaled in [0, 1]. Traction is caused by the movements of both the connected *s-bots* and the *s-bot*'s chassis. Note that the turret of each *s-bot* physically integrates the forces that are applied to the *s-bot* by the other *s-bots*. As a consequence, the traction sensor provides the *s-bot* with an indication of the average direction toward which the group is trying to move as a whole. More precisely, it measures the mismatch between the directions toward which the entire group and the *s-bot*'s chassis are trying to move. The intensity of traction measures the size of this mismatch. Noise is simulated adding a random component uniformly distributed within the $\pm 5\%$ of the maximum traction intensity value.

Each *s-bot*'s controller is a neural network with 4 sensory neurons that encode the traction plus one bias unit. These 5 neurons are directly connected with 2 motor neurons that control the two motorized wheels and the turret-chassis motorized joint. The 4 sensory neurons encode the intensity of the traction from four

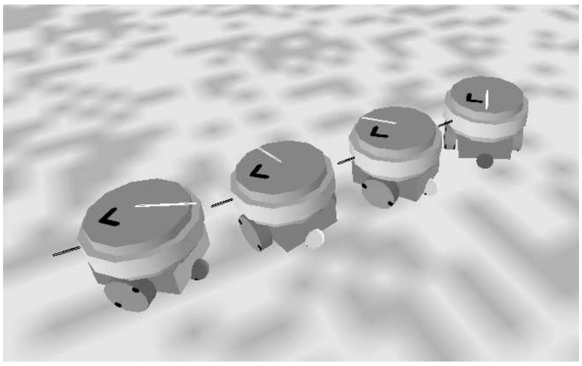


Figure 7: Four physically linked *s-bots* forming a linear structure. The line between two *s-bots* represents the physical link between them. The white line above each *s-bot* indicates the direction and intensity of the traction.

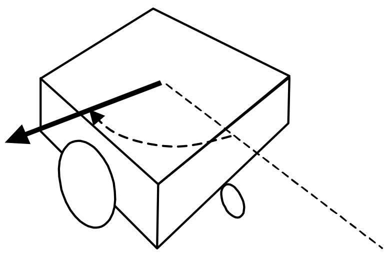


Figure 8: Traction force detected by the *s-bots*' traction sensor. The large and small circles respectively represent the right active wheel and front passive wheel. The dashed line and the full arrow respectively indicate the chassis' orientation and the direction and intensity of the traction. The dashed arrow indicates the angle between the chassis' orientation and the traction.

different preferential orientations with respect to the chassis (front, right, back and left). Each sensory neuron has an activation proportional to the cosine of the angle between the sensor's preferential orientation and the traction's direction when this angle is in $[-90, +90]$ degrees, and is 0 otherwise. This activation is then scaled by the traction intensity. The activation state of the motor units is normalized between $[-5, +5]$ rad/s and is used to set the desired speed of the two corresponding wheels and the turret-chassis motor.

The connection weights of the neural controller of the *s-bots* have been evolved. The initial population consists of 100 randomly generated genotypes that encode the connection weights of 100 corresponding neural controllers, that are used to control the *s-bots* involved in the experiment. Each connection weight is represented in the genotype by 8 bits that are transformed in a number in the interval $[-10, +10]$. Therefore, the total length of the genotype is $10 \times 8 = 80$ bits. The *swarm-bot* is allowed to "live" for 5 epochs, each lasting $T = 150$ simulation cycles. At the beginning of each epoch the chassis of the 4 *s-bots* are assigned random orientations. The 20 best genotypes of each generation are allowed to reproduce by generating 5 copies of their genotype with 3% of their bits replaced by a new randomly selected value. The evolutionary process lasts 100 generations. The experiment is replicated 20 times by starting with different randomly generated initial populations.

To favor the evolution of behaviors that let the *swarm-bot* move as fast and as straight as possible, we evaluate the fitness F_e in each epoch e as the Euclidean distance between the center of mass of the group at the beginning and at the end of the epoch:

$$F_e = \frac{\|\mathbf{X}(0) - \mathbf{X}(T)\|}{L(T)}, \tag{8}$$

$$\mathbf{X}(t) = \frac{1}{n}\sum_{j=1}^{n} \mathbf{X}_j(t), \tag{9}$$

where n is the number of *s-bots* involved in the experiment, $\mathbf{X}_j(t)$ are the coordinates of the j^{th} *s-bot* at simulation cycle t, $\mathbf{X}(t)$ are the coordinates of the center of mass of the group at simulation cycle t, and $L(T)$ is the maximum distance that a single *s-bot* can cover in T simulation cycles by moving straight at maximum speed (see [2] for more details).

5.2 Results

Figure 9 shows how the fitness of the population, averaged over the 20 replications of the experiment, changes across 100 generations. At the end of the evolution, the best controller of each replication was tested for 100 epochs, and the corresponding average performance is reported in Table 2. It can be noted that most replications of the experiment succeeded in finding a very good solution.[8]

Direct observation of the behavior shows that *s-bots* start pulling in different directions, orient their chassis in the direction where the majority of the other *s-bots* are pulling, move straight along the direction that emerges from this negotiation, and compensate successive mismatches in orientation that arise while moving. As shown in Figure 10, the direction that emerges from the negotiation between *s-bots* changes in different tests.

The analysis of how evolved controllers react to different direction and intensity of the traction indicates that they developed a simple strategy that can be described as follows: (i) When the chassis of the *s-bots* are oriented in the same direction, the intensity of the traction is null and the *s-bots* move straight at maximum speed.

[8] See `www.swarm-bots.org/coordinated_motion.html` for some movies of these behaviors.

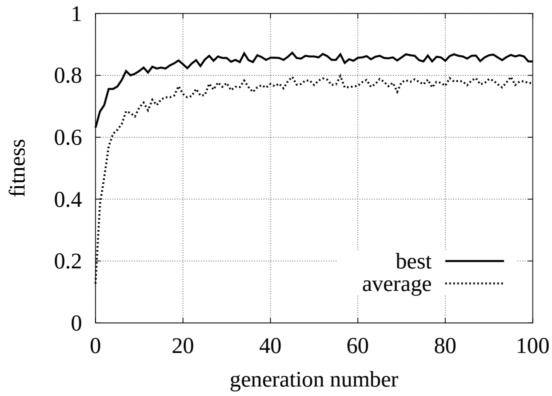


Figure 9: Performance across 100 generations. The continuous and dotted lines respectively plot the performance of the best genotype of each generation and the average performance of the population, averaged over the 20 replications.

Table 2: Average performances resulting from the post-evaluation analysis. For each of the 20 replications of the experiment, the best genotype of the last generation was selected for post-evaluation. Values in the table are the average over 100 (post-)evaluations.

Replication	1	2	3	4	5
Fitness	0.711	0.804	0.595	0.716	0.710

Replication	6	7	8	9	10
Fitness	0.765	0.734	0.563	0.793	0.577

Replication	11	12	13	14	15
Fitness	0.606	0.768	0.751	0.751	0.763

Replication	16	17	18	19	20
Fitness	0.749	0.742	0.813	0.737	0.754

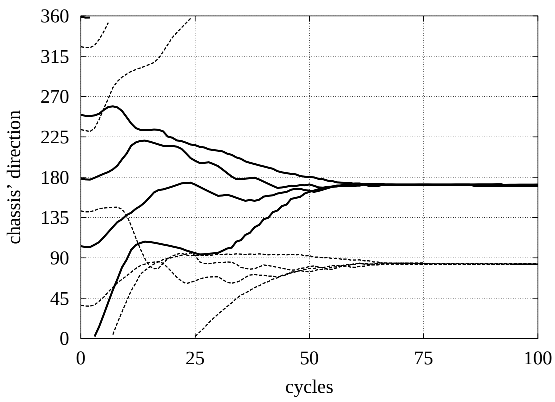


Figure 10: The graph shows the direction of the chassis of the four *s-bots* during 150 simulation cycles, starting with two different initial random orientations (continuous and dotted lines, respectively).

(ii) When the chassis of the *s-bots* are oriented in a similar, although non-identical, direction, the intensity of the traction is low. In this case, *s-bots* tend to turn

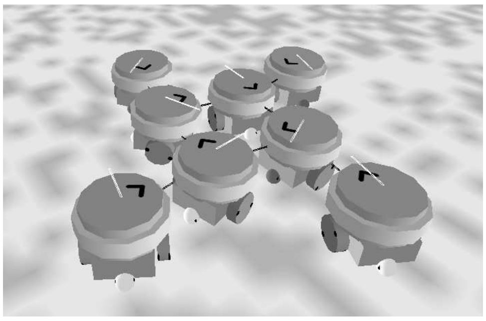


Figure 11: Eight *s-bots* connected by rigid links into a star formation.

toward the average direction in which the whole group is moving, that is, they tend to turn left when the traction comes from the left side and right when the traction comes from the right side. (iii) When the chassis of the *s-bots* are oriented in rather different directions, traction has a high intensity and its direction is highly misaligned with respect to the chassis. In this case, the *s-bots* rapidly change their direction of motion. The *s-bots* that have a larger mismatch with respect to the rest of the group perceive a stronger traction than the others, and this assures that a unique direction finally emerges for the whole group. For instance, three *s-bots* might be oriented North and one *s-bot* might be oriented South. In this case, the South-bound *s-bot* will change its direction more quickly than the other three North-bound *s-bots*.

5.3 Generalization and Scalability

As we claimed above, evolved controllers are capable of producing coordinated movements independently of the number of *s-bots*, of the topology with which they are connected, and of the type of links. For instance, by testing a group of eight *s-bots* connected to form the star formation shown in Figure 11, we observed that also in this case they can negotiate a unique direction of movement (see Figure 12).

The *s-bots* are capable of producing coordinated movement also when assembled by means of flexible, rather than rigid, links. Flexible links consist of two segments connected by a hinge joint that allows the connected *s-bots* to rotate on the ground plane around the middle point of the link. By testing eight *s-bots* connected by

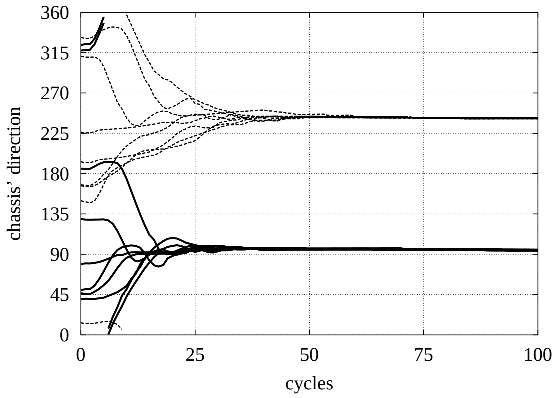
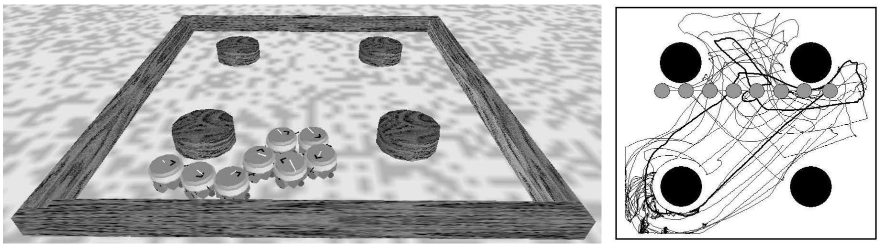


Figure 12: The direction of the chassis of the 8 *s-bots* of a star formation (continuous line) and snake formation (dotted line) during 150 simulation cycles.



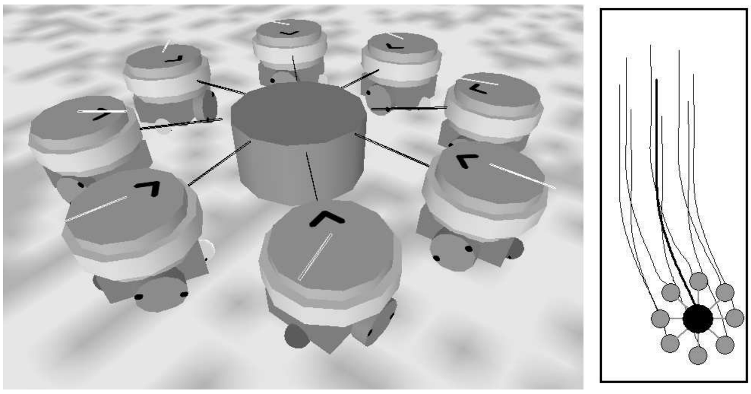
(a) (b)

Figure 13: Eight *s-bots* assembled into a snake formation displaying collective obstacle avoidance. (a) The parallelepipeds and the large cylinders represent walls and obstacles, respectively. (b) The small gray circles represent the initial position and shape of the *swarm-bot*. The square and the large full circles represent the walls and the obstacles. The lines show the trajectory of the *s-bots* during 600 simulation cycles.

flexible links so as to create a snake formation, we observed that they are still able to negotiate a unique direction and produce coordinated movement along such a direction. At the beginning of each trial, the formation changes shape as a consequence of the different orientation of the chassis of the *s-bots*, but after some time it settles to a stable configuration. Given that in structures assembled through flexible links the motors' actions performed by the *s-bots* might affect the shape of the *swarm-bot* rather than the traction perceived by other *s-bots*, these results seem to indicate that the evolved strategy is very robust and allows the *s-bots* to coordinate even when traction sensors provide incomplete information about the movements of the group.

Furthermore, by placing the *s-bots* in an environment with obstacles, we observed that they display individual and collective obstacle avoidance behaviors. In fact, when an *s-bot* hits an obstacle, the collision generates a force on the chassis in the direction opposite to the obstacle. This force is interpreted by the *s-bot* as a traction force. As a consequence, the *s-bot* tends to turn so as to cancel this "traction" force, thus avoiding remaining blocked by the obstacle. When the *s-bots* form a *swarm-bot*, the traction resulting from the collision is transmitted to the other *s-bots* through the physical links, forcing the whole group to reorganize and change direction, eventually avoiding the obstacle. Experiments show that a *swarm-bot* is able to avoid obstacles independently of the number of assembled *s-bots*, the way in which they are connected, and the type of links. Figure 13 shows the behavior of a snake formation connected with flexible links in an arena surrounded by walls and including four cylindrical obstacles. As shown in the figure, the *swarm-bot* is capable of coordinating and collectively avoiding walls. Since the *s-bots* are connected through flexible links, the *swarm-bot* tends to change its shape during the coordination phases and when colliding with obstacles. However, since the *s-bots* also tend to maintain their direction of movement, the *swarm-bot* is also capable of passing through narrow passages, if necessary deforming its shape according to the configuration of the obstacles. This collective obstacle avoidance behavior is very robust. Many of the evolved controllers tested in a snake formation never got stuck during long observation periods.

Finally, we observed that *s-bots* connected to an object, or connected so as to form a closed structure around an object, tend to pull or push the object in a coordinated fashion. Figure 14a shows an example of eight *s-bots* assembled to a cylindrical object through rigid links. If the object is not too heavy, the *s-bots* can coordinate and drag the object toward the direction that emerges from the



(a) (b)

Figure 14: (a) Eight *s-bots* connected to an object through rigid links. (b) Traces left by the *s-bots* (thin lines) and the object (thick line) during 150 simulation cycles. The gray and black circles represent the initial positions of the *s-bots* and of the object.

negotiation between their perceived traction forces (Figure 14b). This behavior can be explained by considering that evolved *s-bots* tend to follow the average direction of the group but also have a tendency to maintain their own direction of movement if the intensity of the perceived traction is not too high and the angle of the traction differs of about 180 degrees from the direction of movement.

A last set of tests was run to assess the scalability of the evolved strategies for coordinated motion. We measured the performance of *swarm-bots* made of an increasing number of *s-bots* (4, 8, 12, ..., 40) assembled in a grid-like formation. The performance of each group was measured 100 times with the same modalities used during evolution. The results of these tests are summarized in Figure 15. The graph shows that the average performance, indicated by the black line, is quite stable with respect to the group size. The maximum distance covered by the *swarm-bot*, indicated by the upper whiskers, tends to decrease slightly when increasing the number of *s-bots*. This suggests that bigger groups need a longer coordination phase for the negotiation of the common direction of motion. The outliers of the graph, represented by the small circles, indicate either the situations in which the group takes a long time to negotiate a common direction, or situations in which the group revolves around its center. The latter situation is a stable state for the group similarly to coordinated motion in a straight line, since it minimizes the intensity of traction perceived by the single *s-bots*. Small groups get into this situation more often than large groups, which have less chances to initiate such behavior, although they tend to negotiate for a longer time. A possible explanation of this is that in large groups it is less likely that the *s-bots* get into situations in which their wheels are aligned along concentric circles with respect to the group's center.

6 Path to Implementation

Following the methodology described in Section 3, the evolved controllers should be validated first using a detailed simulation model and then tested on the real *s-bots*. This last step is not described in this paper because the number of real *s-bots* available for experimentation was not sufficient for the replication of the performed experiments (although, as already mentioned, it was possible to test the quality of the detailed simulation models by running comparisons with the real *s*-

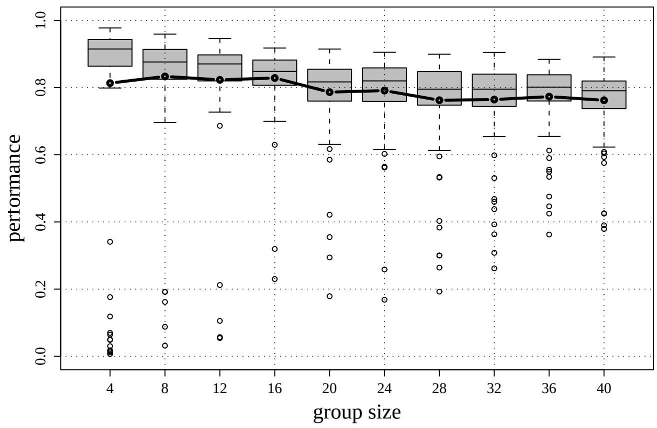


Figure 15: Scalability of the coordinated motion behavior. See text for details. For an explanation of the box-plot, see Figure 6.

bots, as illustrated in the companion paper [17]). Thus, in this section we present the validation on the detailed simulation model described in Section 3.

The validation of the aggregation behaviors with the detailed simulation model was performed using identical settings as in the scalability test presented in Section 4.3. Thus, 100 evaluations of the performance measure defined in (7) were performed for varying group sizes ($n = 4, 8, 12, \ldots, 40$). Figure 16 shows the results obtained. These data and the observation of the system reveal that the aggregation process is slower than with the simplified model. In fact, we observed that the presence of the tracks in the detailed model makes the rotation of the *s-bot* slower and less precise. This makes the "following behavior" described previously less efficient. This problem explains the results obtained for group size 4: the inefficiency of the "following behavior" has a greater impact for this group size, its role being more important when the density of *s-bots* in the arena is low. The aggregation performance increases for group size 8 and is comparable to the one of the simplified model (see Figure 6). For bigger group sizes, we observe a decrease in performance which rapidly stabilizes at a fairly good value.

Satisfactory results have been obtained for the coordinated motion task. Also in this case, the tests on scalability presented in Section 5.3 were replicated using the detailed simulation model. The results obtained are shown in Figure 17. In this case, the validation was clearly successful. For every group size we have a performance that is comparable to the one obtained with the simplified simulation model, shown in Figure 15. The reason why the variance is slightly higher in this case can be found again in the different dynamics in the turning of the *s-bot*, due to the presence of the tracks. The turning of the chassis, being less efficient, makes the coordination phase longer, which in turn corresponds to a lower performance. This is also the reason why more outliers are found for big group sizes, confirming that reaching a coordinate status was slower with the detailed model.

The experiments presented in this section confirm that the evolved strategies are robust enough to be ported on a different model with a tolerable decrease of performance. This result is very promising, in the perspective of a transfer to the physical robots.

Additionally, it is interesting to note that, as the different simulation models have comparable characteristics, an incremental evolution paradigm is applicable. Following this paradigm, evolution can be initially performed using a very simple and fast simulation model, in order to find a suitable solution for the given problem. Once such a solution has been found, evolution continues using the detailed simulation model or the real *s-bots*. In this way, it is possible to adapt the solutions obtained with the simple simulation model to the new situation more easily and in less time than starting from scratch with complex settings.[9]

However, in order to incrementally evolve controllers on real hardware, it is necessary to bear in mind that what was done in simulation should be feasible also in reality. In particular, we should be able to compute the fitness in the real world. This is not a particular problem as long as we use variables directly accessible to the robots, such as sensor readings. On the contrary, evolution may not be feasible if the fitness variables cannot be obtained in the real world. In this paper, the fitness computation was based on some global variables, that is, the absolute positions of all the robots. This information is not directly accessible to the *s-bots*, but can be easily obtained from an overhead camera. With such a setup, the incremental evolution of aggregation and coordinated motion strategies is possible also in the real world.

7 Related Work

Recently, there has been a growing interest in the research community for the study of complex robotic systems that could present features like versatility, robustness or capacity to perform complex tasks in unknown environments [8, 14, 25, 26]. In this section, we overview some of the recent studies belonging to the areas of swarm robotics and collective robotics, which are closely related to the experiments presented in this paper.

7.1 Swarm Intelligence and Swarm Robotics

The term *swarm intelligence* was coined by Beni and Wang [5] to describe a new approach to the control of distributed cellular robotic systems. Later, Bonabeau et

[9] A similar incremental evolution paradigm has already been successfully applied for the transfer of evolved controllers between different robotic platforms [10].

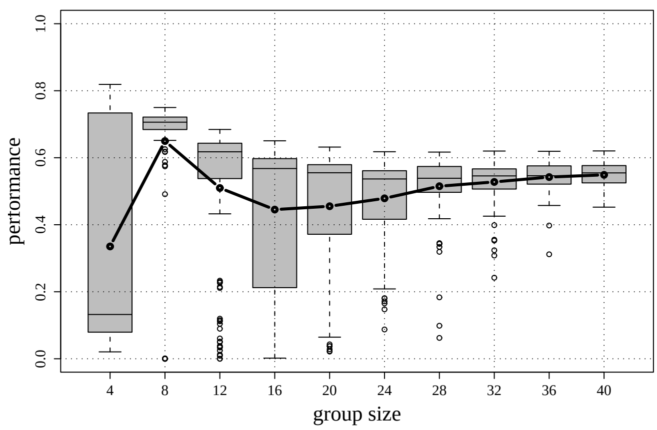


Figure 16: Aggregation behaviors tested with the detailed simulation model of the *s-bot*. For an explanation of the box-plot, see Figure 6.

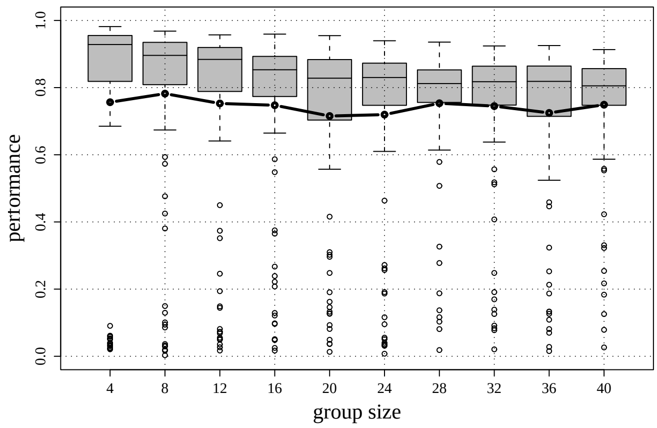


Figure 17: Coordinated motion behaviors tested with the detailed simulation model of the *s-bot*. For an explanation of the box-plot, see Figure 6.

al. [6] extended this definition to include "any attempt to design algorithms or distributed problem-solving devices inspired by the collective behavior of social insect colonies and other animal societies" ([6], page 7). This new definition promoted swarm intelligence as a new computational paradigm for solving a large variety of problems. Swarm robotics consists in the application of swarm intelligence to the control of robotic swarms, emphasizing decentralization of the control, limited communication abilities among robots, use of local information, emergence of global behavior and robustness.[10]

Within swarm robotics research, to the best of our knowledge, there is very little work on self-organized aggregation. Most of the research about aggregation refers to tasks like *foraging* or *object clustering*, in which robots have to form clusters of some objects initially scattered in the arena. In foraging, objects must be collected and retrieved in a particular area (the *home* or the *nest*). In clustering, the focus is put on the dynamics of the process, no matter the place in which the cluster is formed.

A number of papers study the self-organized clustering and sorting of objects in a closed arena, taking inspiration from the cemetery organization and brood sorting behaviors of ants [9]. Gaussier and Zrehen [11] manually designed reactive behaviors for controlling a group of Khepera robots, in order to cluster objects in an arena. Beckers et al. [4] and Holland et al. [12] studied the clustering and sorting of colored frisbees by a group of real robots. Frisbees were initially scattered on the ground, and the robots had to sort them in clusters of different colors. They designed a simple behavioral rule set for this purpose and concluded that the real-world physics was an essential component of the self-organization observed. Martinoli [15] studied the clustering of small cylinders by a group of real robots. In particular, he analyzed the interactions between the robots and the effect of the group size on the performance.

Efficiency in the foraging task is the main focus of the study of Sugawara et al. [29]. They showed that the use of a simple form of communication among robots could increase the efficiency of the swarm, if the distribution of pucks to be retrieved is not uniform in the environment. In a more recent work [30], Sugawara et al. studied puck clustering. Also in this case, a set of simple behavioral rules was

[10]For a definition of swarm robotics, see also the editorial of this special issue.

developed and a simple form of broadcast communication was used. Simulations showed that increasing the “interaction duration”, that is, the duration of the communication signal, led first to an increase and then to a decrease of the performance of the system. This indicates that there is an optimum interaction duration for the clustering process.

Concerning the coordinated motion task, it is worth mentioning the work of Sugawara et al. [28]. They proposed a simple behavioral model that, by varying some parameters of the system, could let a swarm of robots generate four different types of collective motion. The robots could either (i) form a fixed lattice and move in a straight line; or (ii) remain in an almost fixed lattice and present a wavy movement; or (iii) constantly change their relative positions, with a resulting irregular movement; or (iv) not maintain any particular structure without moving much.

7.2 Collective Evolutionary Robotics

Although artificial evolution has been often used for synthesizing behaviors for autonomous robots [19], its use as a methodology to evolve behaviors for groups of robots has been limited. Collective evolutionary robotics has often focused on coordinated motion in a group of robots, but physical connections among robots were never considered.

Reynolds [23] evolved the control system of a group of creatures, called *boids*, which were placed in an environment with static obstacles and a manually programmed predator. The control system was evolved to avoid collisions and to escape from predators. Although the results described in the paper are rather preliminary, some evidence indicates that coordinated motion strategies emerged. In a follow-up of this work, Ward et al. [33] evolved *e-boids*, groups of artificial fish capable of displaying schooling behavior. Two populations of predator and prey creatures were evolved and placed in a 2D environment containing randomly distributed food elements. The analysis of the distance between prey, prey and food, and predator and prey suggests that the emergence of the schooling behavior is correlated with: (i) an advantage in the ability to find food clumps, and (ii) a better protection from predation. Spector et al. [27] used genetic programming to evolve group behaviors for flying agents in a simulated environment.

Overall, the above mentioned works suggest that artificial evolution can be successfully applied to synthesize effective collective behaviors. Whether these results could be generalized to the synthesis of controllers for physical systems (robots), however, remains to be ascertained given that in those experiments creatures rely on sensory systems that provide information that is “perfect” (i.e., free from noise) and often “unrealistic” (i.e., hardly achievable on real hardware).

Recently, Quinn [21, 20] explored two ways of evolving controllers for a group of robots while studying a coordinated motion task using two simulated Khepera robots. In the first approach, called *clonal*, all members of the group share a same genome. This is the same approach we used in the experiments presented in this paper. The second approach, called *aclonal*, provides each member of the group with a different genome. In the aclonal evolution, the fitness of each robot is computed separately, whereas in the clonal evolution the fitness of a robot is calculated as the average fitness of the group. Results indicated that aclonal evolution produces better performing behaviors for this rather simple task. In fact, with aclonal evolution it was possible to obtain different controllers for different roles in the performance of the task. In a very recent work, Quinn et al. [22] evolved neural network controllers for small groups of homogeneous real robots, which have to perform a coordinated movement task. Analyzing the evolved behaviors, they were able to observe that robots adopt distinct roles in the group.

There are a few other works that are loosely related to the evolution of aggregation behaviors. For example, in an attempt to study the evolutionary origin of herding, Werner and Dyer [34] co-evolved two populations of predators and prey creatures that were selected for the ability to catch prey and find food, and to escape predators, respectively. By analyzing the result of a single evolutionary run, the author observed that after some generations, during which predators evolved an ability to catch the prey, creatures converged into small herds which were constantly splitting up and reforming.

Zaera et al. [36] carried out a series of experiments to study the use of evolution as a methodology to develop collective behaviors for "virtual fish" groups that swim in a rather realistic 3-D simulated environment. They were able to evolve aggregation and dispersal behaviors fairly easily, but they observed that these collective behaviors were not a result of interactions among the members of the group, but rather between the individual fish and the environment (the boundaries of the arena). Additionally, their attempts to evolve schooling behavior were not very successful.

8 Conclusions

This paper introduced a new robotic concept, called a *swarm-bot*, defined as an artifact composed of simpler autonomous robots, called *s-bots*. An *s-bot* is an autonomous robot with limited sensing, computational, and acting capabilities, capable of creating physical connections with other *s-bots*, thus forming a *swarm-bot* that is able to solve problems the single individual cannot cope with. We presented in this paper some of the results obtained in the attempt to control a *swarm-bot*. In particular, we chose to exploit *artificial evolution* for synthesizing the controllers for the *s-bots*, and for obtaining self-organization in the robotic system. The solutions found by evolution are simple and in many cases they generalize to different environmental situation. This demonstrates that evolution is able to produce a self-organized system that relies on simple and general rules, a system that is consequently robust to environmental changes and that scales well with the number of *s-bots* involved in the experiment.

Ongoing work is investigating the emergence of functional self-assembly, that is, the self-organized formation of structures that are functional to the accomplishment of a given task. For example, we are studying the use of artificial evolution for generating controllers that let a *swarm-bot* move toward a given target and assemble and disassemble on the basis of their current goal and of the environmental conditions [32]. From this point of view, the results reported in this paper on individual and collective obstacle avoidance behavior suggest that the problem of controlling single *s-bots* and teams of assembled *s-bots* might be solved with uniform and simple control solutions. Moreover, the results reported on the generalization ability of the evolved controllers suggest that the obtained behaviors might scale up to more complex situations.

Acknowledgments

This work was supported by the "SWARM-BOTS" project, funded by the Future and Emerging Technologies programme (IST-FET) of the European Commission, under grant IST-2000-31010. The information provided is the sole responsibility of the authors and does not reflect the Community's opinion. The Community is not responsible for any use that might be made of data appearing in this publication. Marco Dorigo acknowledges support from the Belgian FNRS, of which he is a Senior Research Associate, through the grant "Virtual Swarm-bots", contract no. 9.4515.03, and from the "ANTS" project, an "Action de Recherche Concertée" funded by the Scientific Research Directorate of the French Community of Belgium. The Swiss participants to the project are supported under grant 01.0012 by the Swiss Government.

References

[1] C. Anderson, G. Theraulaz, and J.-L. Deneubourg. Self-assemblage in insects societies. *Insectes Sociaux*, 49:99–110, 2002.

[2] G. Baldassarre, S. Nolfi, and D. Parisi. Evolution of collective behavior in a team of physically linked robots. In R. Gunther, A. Guillot, and J.-A. Meyer, editors, *Applications of Evolutionary Computing - Proceedings of the Second European Workshop on Evolutionary Robotics (EvoWorkshops2003: EvoROB)*, pages 581–592. Springer-Verlag, Berlin, Germany, 2003.

[3] G. Baldassarre, S. Nolfi, and D. Parisi. Evolving mobile robots able to display collective behaviour. *Artificial Life*, 9(3):255–267, 2003.

[4] R. Beckers, O.E. Holland, and J.-L. Deneubourg. From local actions to global tasks: Stigmergy and collective robotics. In R. Brooks and P. Maes, editors, *Proceedings of the Fourth Workshop on Artificial Life*, pages 181–189. MIT Press, Cambridge, MA, 1994.

[5] G. Beni and J. Wang. Swarm intelligence. In *Proceedings of the Seventh Annual Meeting of the Robotics Society of Japan*, pages 425–428, Tokio, Japan, 1989. RSJ press.

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

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

[8] Y. U. Cao, A. S. Fukunaga, and A. B. Kahng. Cooperative mobile robotics: Antecedents and directions. *Autonomous Robots*, 4:1–23, 1997.

[9] J.-L. Deneubourg, S. Goss, N. Franks, A. Sendova-Franks, C. Detrain, and L. Chretien. The dynamics of collective sorting: Robot-like ant and ant-like robot. In J.-A. Meyer and S. W. Wilson, editors, *From Animals to Animats. Proceedings of the First International Conference on Simulation of Adaptive Behavior (SAB90)*, pages 356–365. MIT Press, Cambridge, MA, 1990.

[10] D. Floreano and F. Mondada. Evolutionary neurocontrollers for autonomous mobile robots. *Neural Networks*, 11:1461–1478, 1998.

[11] P. Gaussier and S. Zrehen. A constructivist approach for autonomous agents. In N. Thalmann and D. Thalmann, editors, *Artificial Life and Virtual Reality*, pages 97–113. John Wiley & Sons, Chichester, UK, 1994.

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

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

[14] J. Liu and J. Wu. *Multiagent Robotic Systems*, volume 21 of *International Series on Computational Intelligence*. CRC Press, Boca Raton, FL, 2001.

[15] A. Martinoli and F. Mondada. Collective and cooperative group behaviours: Biologically inspired experiments in robotics. In *Proceedings of the Fourth International Symposium on Experimental Robotics*, pages 3–10. Springer-Verlag, Berlin, Germany, 1995.

[16] M.J. Matarić and D. Cliff. Challenges in evolving controllers for physical robots. *Robotics and Autonomous Systems*, 19(1):67–83, 1996.

[17] F. Mondada, G. C. Pettinaro, A. Guignard, I. V. Kwee, D. Floreano, J.-L. Deneubourg, S. Nolfi, L. M. Gambardella, and M. Dorigo. SWARM-BOT: A new distributed robotic concept. *Autonomous Robots*, 17(2–3), 2004.

[18] F. Mondada, G. C. Pettinaro, I. W. Kwee, A. Guignard, L. M. Gambardella, D. Floreano, S. Nolfi, J.L. Deneubourg, and M. Dorigo. SWARM-BOT: A swarm of autonomous mobile robots with self-assembling capabilities. In C.K. Hemelrijk and E. Bonabeau, editors, *Proceedings of the International Workshop on Self-organisation and Evolution of Social Behaviour*, pages 307–312, Monte Verità, Ascona, Switzerland, September 8–13, 2002.

[19] S. Nolfi and D. Floreano. *Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines*. MIT Press, Cambridge, MA, 2000.

[20] M. Quinn. A comparison of approaches to the evolution of homogeneous multi-robot teams. In *Proceedings of the 2001 Congress on Evolutionary Computation (CEC2001)*, pages 128–135. IEEE Press, Piscataway, NJ, 2001.

[21] M. Quinn. Evolving communication without dedicated communication channels. In J. Kelemen and P. Sosik, editors, *Proceedings of the Sixth European Conference on Artificial Life*, volume 2159 of *Lecture Notes in Computer Science*, pages 357–366. Springer-Verlag, Berlin, Germany, 2001.

[22] M. Quinn, L. Smith, G. Mayley, and P. Husbands. Evolving controllers for a homogeneous system of physical robots: Structured cooperation with minimal sensors. *Philosophical Transactions of the Royal Society of London, Series A: Mathematical, Physical and Engineering Sciences*, 361:2321–2344, 2003.

[23] C.W. Reynolds. An evolved, vision-based behavioral model of coordinated group motion. In J.-A. Meyer, H. Roitblat, and S. W. Wilson, editors, *From Animals to Animats 2. Proceedings of the Second International Conference on Simulation of Adaptive Behavior (SAB92)*, pages 384–392. MIT Press, Cambridge, MA, 1993.

[24] E. Şahin, T. H. Labella, V. Trianni, J.-L. Deneubourg, P. Rasse, D. Floreano, L. M. Gambardella, F. Mondada, S. Nolfi, and M. Dorigo. SWARM-BOT: Pattern formation in a swarm of self-assembling mobile robots. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*. IEEE Press, Piscataway, NJ, 2002.

[25] A. C. Schultz and L. E. Parker, editors. *Multi-Robot Systems: From Swarms to Intelligent Automata - Proceedings of the 2002 NRL Workshop on Multi-Robot Systems*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 2002.

[26] A. C. Schultz, L. E. Parker, and F. E. Schneider, editors. *Multi-Robot Systems: From Swarms to Intelligent Automata, Volume II - Proceedings of the 2003 International Workshop on Multi-Robot Systems*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 2003.

[27] L. Spector, J. Klein, C. Perry, and M.D Feinstein. Emergence of collective behavior in evolving populations of flying agents. In E. Cantù-Paz et al., editor, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2003)*, volume 2723 of *Lecture Notes in Computer Science*, pages 61–73. Springer-Verlag, Berlin, Germany, 2003.

[28] K. Sugawara, T. Arai, M. Sano, Y. Hayakawa, T. Mizuguchi, and T. Watanabe. Collective motion of interacting simple robots. In *Proceedings of the 27th Annual Conference of the Industrial Electronics Society (IECON'01)*, pages 428–432. IEEE Press, Piscataway, NJ, 2001.

[29] K. Sugawara and M. Sano. Cooperative acceleration of task performance: Foraging behavior of interacting multi-robots system. *Physica D: Nonlinear Phenomena*, 100(3/4):343–354, 1997.

[30] K. Sugawara and T. Watanabe. Swarming robots – Foraging behavior of simple multirobot system. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2002)*, pages 2702–2707. IEEE Press, Piscataway, NJ, 2002.

[31] V. Trianni, R. Groß, T. H. Labella, E. Şahin, and M. Dorigo. Evolving aggregation behaviors in a swarm of robots. In W. Banzhaf, T. Christaller, P. Dittrich, J. T. Kim, and J. Ziegler, editors, *Proceedings of the Seventh European Conference on Artificial Life*, volume 2801 of *Lecture Notes in Artificial Intelligence*, pages 865–874. Springer Verlag, Berlin, Germany, 2003.

[32] V. Trianni, E. Tuci, and M. Dorigo. Evolving functional self-assembling in a swarm of autonomous robots. In *From Animals to Animats 8. Proceedings of the Eight International Conference on Simulation of Adaptive Behavior (SAB04)*, 2004, to appear.

[33] C. R. Ward, F. Gobet, and G. Kendall. Evolving collective behavior in an artificial ecology. *Artificial Life*, 7(2):191–209, 2001.

[34] G. M. Werner and M. G. Dyer. Evolution of herding behavior in artificial animals. In J.-A. Meyer, H. L. Roitblat, and S. W. Wilson, editors, *From Animals to Animats 2. Proceedings of the Second International Conference on Simulation of Adaptive Behavior (SAB92)*, pages 393–399. MIT Press, Cambridge, MA, 1992.

[35] M. Wilson, C. Melhuish, A. B. Sendova-Franks, and S. Scholes. Algorithms for building annular structures with minimalistic robots inspired by brood sorting in ant colonies. *Autonomous Robots*, 17(2–3), 2004.

[36] N. Zaera, D. Cliff, and J. Bruten. (Not) evolving collective behaviours in synthetic fish. In P. Maes, M. Matarić, J.-A. Meyer, J. Pollack, and S. W. Wilson, editors, *From Animals to Animats 4. Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior (SAB96)*, pages 635–644. MIT Press, Cambridge, MA, 1996.