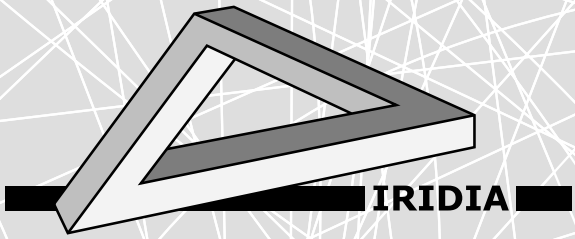


Université Libre de Bruxelles

Institut de Recherches Interdisciplinaires
et de Développements en Intelligence Artificielle

Cooperation through self-assembling in multi-robot systems

ELIO TUCI, RODERICH GROSS, VITO TRIANNI,
FRANCESCO MONDADA, MICHAEL BONANI, and
MARCO DORIGO

Technical Report No.

TR/IRIDIA/2005-3

November 2005

ACM Trans. on Autonomous and Adaptive Systems

Cooperation through self-assembling in multi-robot systems

ELIO TUCI, RODERICH GROSS, VITO TRIANNI,
FRANCESCO MONDADA, MICHAEL BONANI, and MARCO DORIGO

November 17, 2005

Abstract

This paper illustrates the methods and results of two sets of experiments in which a group of mobile robots, called *s-bots*, are required to physically connect to each other—i.e., to self-assemble—to cope with environmental conditions that prevent them to carry out their task individually. The first set of experiments is a pioneering study on the utility of self-assembling robots to address relatively complex scenarios, such as cooperative object trasport. The results of our work suggest that the *s-bots* possess hardware characteristics which facilitate the design of control mechanisms for autonomous self-assembly. The second set of experiments is an attempt to integrate within the behavioural repertoire of an *s-bot* decision making mechanisms to allow the robot to autonomously decide whether or not environmental contingencies require self-assembly. The results show that it is possible to synthesise, by using evolutionary computation techniques, artificial neural networks that integrate both the mechanisms for sensory-motor coordination and for decision making required by the robots in the context of self-assembly.

1 Introduction

Recently, there has been a growing interest in multi-robot systems since, with respect to a single robot system, they provide increased robustness by taking advantage of inherent parallelism and redundancy. Moreover, the versatility of a multi-robot system can provide the heterogeneity of structures and functions required to undertake different missions in unknown environmental conditions. Research in autonomous multi-robot systems often focuses on mechanisms to enhance the efficiency of the group through some form of cooperation among the individual agents. An innovative way of cooperation is given by self-assembly, that is, the capability of a group of mobile robots to autonomously connect to and disconnect from each other through some kind of device that allows physical connections.

Self-assembly can enhance the efficiency of a group of autonomous cooperating robots in several different contexts. Generally speaking, self-assembly is advantageous anytime it allows a group of agents to cope with environmental conditions which prevent them from carrying out their task individually. For example, robots designed for all-terrain navigation could make use of self-assembly to move in a particularly rough terrain by reducing the risk of toppling over (see figure 1a), as well as to bridge the gap between the two sides of a trough larger than the body of a single robot, reducing the risk of falling in (see figure 1b). In the context of object transport, a group of self-assembled robots might be capable of pushing/pulling an object which, due to its characteristics (e.g., mass, size, and shape), can not be transported by a single robot.

Despite its relevance within the context of multi-robot systems, the design of control policies for self-assembling robots has encountered difficulties. Section 2 shows that, up to now, there are no examples of self-assembling robots in which more than two autonomous mobile units manage to approach and to connect to each other. This lack of results is mostly due to hardware implementations which demand each robot of the group to be able to accurately coordinate its actions (a) to self-assemble, and (b) to facilitate the movement of the assembled robotic structure once connected.

The "marginal" role that self-assembly has been playing within multi-robot systems has been a motivation for us to carry out research work focused on the design of mechanisms underlying the motor

coordination required in self-assembly as well as on the decision making structures which allow the robots to decide when it is time to physically connect to each other. Indeed, the efficiency of a group of autonomous robots is strictly associated with the robots capability to exploit the most efficient strategies with respect to the environmental conditions. Self-assembly may improve the efficiency of the group if it is triggered by the perception of those environmental contingencies that jeopardise the accomplishment of a task if carried out in non-assembled structures. In order to do so, the robots should be equipped with mechanisms to allow them to autonomously decide whether or not the environmental conditions require self-assembly.

The paper illustrates the methods and results of two sets of experimental works in which robots are required to make use of self-assembly to cope with environmental conditions that prevent them to carry out their task individually. These robots are called *s-bots*. We use the term *swarm-bot* to refer to a multi-robot system composed of several *s-bots* physically connected.[1]

The goal of the first set of experiments is to validate on "real" hardware control mechanisms for self-assembly originally developed in a simulated environment (see section 4). In these experiments, a group of six *s-bots* is required to exploit self-assembly in order to transport a heavy object towards a target area. Each *s-bot* is driven by a controller made of two modules: the first one—referred to as "assembly module"—defines the rules for the connection to an object, or to already connected *s-bots*; the second one—referred to as "transport module"—defines the rules for collectively moving an object towards a target area. In general, we consider an instance of self-assembly to be the process which ends up in a structure whose elements (i.e., the *s-bots*) are physically connected to each other. In particular, in the considered cooperative transport scenario, self-assembly is such that at least one element of the assembled structure should be connected to the object to be transported. Experimental results show that the modular controllers can successfully generate the actions required by the *s-bots* to physically connect to the object and/or to each other and to move in a coordinated fashion once connections are established. The control policies we developed take advantage of the hardware design in order to achieve a successful self-assembly behaviour. We believe that our work represents a sensible step forward with respect to the state of the art in the design of self-assembling robots, in particular if we look at (a) the number of robots involved in self-assembly, (b) the reliability of the system, (c) the speed with which the robots form the assembled structures, and (d) the capability of the assembled structures to coordinate in order to transport a heavy object.

The results of the first set of experiments are particularly promising with respect to the effectiveness of the mechanisms underlying the coordination of movement of the single *s-bot* and of the *swarm-*

[1]The *s-bots* have been developed within the SWARM-BOTS project, a project funded by the Future and Emerging Technologies Programme (IST-FET) of the European Commission, under grant IST-2000-31010. See also `http://www.swarm-bots.org/`.

(a) (b)

Figure 1: A group of robots physically connected to each other, that (a) moves on rough terrain; (b) passes over a gap. These pictures are demonstrative of the capabilities of the self-assembling robots we developed.

bot as a whole. We also mention that this type of controller has been successfully used in a different context, to allow a group of *s-bots* to self-assemble to overcome steep hills which cause a single *s-bot* to topple backwards [29]. Notwithstanding the successful results, this modularised architecture is based on a set of *a priori* assumptions concerning the specification of the environmental conditions which trigger self-assembling. For example, (a) the objects that can be grasped must be red, and those that can not be grasped must be blue; (b) the action of grasping is carried out only if all the "grasping requirements" are fulfilled (see section 4.2.1 for details). If the experimenter could always know in advance in what type of world the agents will be located, assumptions such as those concerning the nature of the object to be grasped would not represent a limitation with respect to the domain of action of the robotic system. However, since it is desirable to have agents which can potentially adapt to variable circumstances or conditions that are partially or totally unknown to the experimenter, it follows that the efficiency of autonomous robots should be estimated also with respect to its capacity to cope with "unpredictable" events (e.g., environmental variability, partial hardware failure, etc.). We believe that a sensible step forward in this direction can be made by avoiding to constrain the system to initiate its most salient behaviours (e.g., aggregation, grasping of objects, self-assembly) in response to *a-priori* specified agent's perceptual states. As explained at the beginning of section 5, one way to take into account these principles is by exploiting artificial evolution to synthesise integrated (i.e., not-modularised) artificial neural network controllers.

Accordingly, the goal of the second set of experiments is to move a first step towards the development of integrated artificial neural networks that provide an *s-bot* with all the mechanisms required to perform tasks that demand self-assembly (see section 5). By exploiting this approach, we hope to reduce the amount of *a-priori* assumptions to the advantage of improving the capability of the robotic system to adapt to different and unforeseeable circumstances. Unfortunately, the simplifications introduced in the experimental set up do not allow testing on the real robots. This notwithstanding, we were able to achieve the important result of integrating in a single neural controller all the adaptive mechanisms required to solve the task—i.e., mechanisms for individual and collective behaviour, decision-making, self-assembly. Further work is certainly required in order to exploit this methodology to port the evolved controllers on the real *s-bots*. However, the results illustrated in section 5 look quite promising. They seem to suggest that in a near future we might be able to exploit integrated artificial neural networks designed by artificial evolution to improve the adaptiveness of our self-assembling robots.

1.1 Structure of the paper

In what follows, we first present a review of the work on self-assembling robots, with particular attention to both the hardware elements through which self-assembly is accomplished, and the characteristics of the controllers which bring forth the robot behaviour (see section 2).

In section 3, we provide a brief description of the most significant hardware characteristics of the *s-bots*.

In section 4, we describe methods and results of a first set of experiments in which a modular architecture has been employed to control the behaviour of the *s-bots* in a cooperative transport task. We discuss the results, and also the problems encountered with our approach.

In section 5, we illustrate research work in simulation about the design of collective decision mechanisms for self-assembling robots that might not present the limitations we discussed in the previous section.

Conclusions are drawn in section 6 and future work is discussed in section 7.

2 Related work

The design of the hardware and of the control policies for self-assembling robots is a particularly challenging task. In the robotic literature, there are several types of hardware platforms composed of

modules which are capable of connecting to each other through some kind of connection mechanism. The majority of such systems fall in the category of *self-reconfigurable robots* (see [38]). In most studies of self-reconfigurable robotic systems, single modules are pre-attached by the designer (e.g., PolyBot [36], CONRO [9], Crystalline [31], M-TRAN [27], and ATRON [24]). This review does not take into account these studies.

In the remain of this section, we mainly discuss those *self-reconfigurable robots* in which self-assembly is the results of autonomous movement of the single modules (see sections 2.1, 2.2, and 2.3). We also briefly overview recent work in stochastic reconfigurable robots in which the modules move passively and bound to each other upon random collisions (see section 2.4).

2.1 Chain-based Self-reconfigurable Robots

PolyBot [36, 39, 37] is a modular chain robot that can configure its form with no external mechanical assistance. Each module has one degree of freedom involving rotation of two opposite connection plates through a ± 90 degrees range. Additional passive cuboid segments with six connection plates are necessary to introduce branches to the structures and to establish connection with an (external) power supply. The active modules are equipped with IR sensors and emitters integrated in the connection plates, as well as with sensors to detect the positions of the rotational joints. Yim *et al.* [39] demonstrated the ability of a modular robot arm composed of six PolyBot modules to grasp another module on flat terrain. One end of this arm was attached to one of the walls of the arena. The joint angles for each segment of the arm were calculated by an inverse kinematics routine. This step requires knowledge about the goal position and orientation. Imprecision in the joints results in positional errors which increase with the length of the chain. Therefore, this method is applied only in the *long range* phase during which the corresponding modules get close to each other. The *median range* phase and the *short range* phase that follows make use of the IR sensors and emitters to support further alignment and approach.

CONRO is a homogeneous modular chain robot composed of modules that are fully self-contained [9]. The basic implementation of a CONRO module has three segments connected in a chain: a passive connector, a body and an active connector. Infrared emitters and receivers are located on both active and passive connectors to support the docking and to enable communication between connected modules. Rubenstein et al. [30] demonstrated the ability of two separate CONRO robots to perform an autonomous docking task. Each robot consisted of a chain of two, linearly-linked CONRO modules. The robots were put on an obstacle-free, flat terrain, at distances up to 15 cm. To ensure that the chains were able to perceive each other, they were set-up facing each other with an angular displacement not bigger than 45 degree. Using a simple control policy the robots put themselves in a proper alignment, then one robot was approaching the other. Finally, the docking was recognised and communicated to all the modules. The control was heterogeneous, both with respect to the modules of each robot, and concerning the different robots.

2.2 Lattice-based Self-reconfigurable Robots

Zykov *et al.* [40] presented a lattice-based self-reconfigurable robot capable of self-assembling. Each module is a cube, one half of it could swivel relative to the other half. Modules were powered externally from a grid-based supply fixed on the ground. Zykov *et al.* demonstrated self-replication of a four-module robot. The system required a well-ordered supply of additional modules. Also it could not adapt to situations in which the additional modules were supplied in other than pre-defined places. These constraints are mainly imposed by limitations in both the mobility and the perception, as it is the case in most lattice-based self-reconfigurable robots.

2.3 Mobile Self-reconfigurable Robots

Fukuda *et al.* proposed the concept of dynamically reconfigurable robotic systems and realized an implementation with CEBOT, the first cellular robotic system [12, 14]. CEBOT is a heterogenous system comprised of cells with different functions (e.g., move, bend, rotate, and slide). A series of prototypes has been implemented, including the CEBOT Mark I, II, III, IV and V. Fukuda *et al.* [13] presented a successful docking experiment in an obstacle-free, flat terrain: the object cell was put 60 cm away from the moving cell; the latter was oriented towards the object cell. The orientation of the object cell was displaced for 20 degree with respect to the moving cell. The moving cell was controlled with a hand-coded controller. To the best of our knowledge there are no quantitative results provided to assess the performance and the reliability of autonomous self-assembly in a group of CEBOT cells.

The work of Hirose et al. [23] describes a distributed robotic concept called Gunryu. Each robot unit is equipped with a versatile manipulation device and is capable of fully autonomous locomotion. In addition, the manipulator can be employed to establish a physical link with another robot unit. By exploiting this mechanism, a chain of connected units could potentially navigate through steep concave regions, or pass large troughs. A prototype of two units proved capable of locomotion on rough terrain under conditions in which single units failed. However, units have been mechanically linked by means of a passive arm. As a result, the robot units were not capable of self-assembling.

Super Mechano Colony (SMC) [10, 22] is a modular robotic concept composed of a single main body (called the mother-ship) and many child units attached to it. Child units are an integral part of the system's locomotion. In addition, the child robots can disband to accomplish separate, autonomous missions, and reconnect once the missions are accomplished. Damoto *et al.* [10] introduced the first prototype of an SMC system. Two motorised and two passive wheels are attached to the chassis, and allow for navigation on flat terrain. Each child robot can be equipped with CCD cameras. The mother-ship is equipped with passive wheels. Disconnecting and re-docking of a child unit to the mother-ship has been realized by letting it follow a fixed path by making use of dead-reckoning. The most recent development is the SMC rover [26]. It is a planetary rover with attachable child robots (called *Uni-Rovers*), each one composed of a single wheel and a manipulation arm (also used as connection mechanism). The current prototype is not equipped with visual sensors.

Similar to Gunryu, the Millibot Train is composed of multiple, linearly linked, modules [8]. Each module is equipped with caterpillar tracks. A prototype has been developed to study its mobility in climbing a step or in crossing a ditch. Since no external sensors had been integrated, the prototype was not capable of self-assembling [8].

Bererton and Khosla [5, 6] studied autonomous docking between two mobile robots in the context of self-repair. Although it might not be considered a self-reconfigurable system, the robots share some similarities with mobile self-reconfigurable systems. Hence, we decided to append this work to this section. To guide the docking procedure, a black and white camera is mounted on top of the approaching robot. However, image processing is performed externally on a PC. A simple state machine controls the robot to turn counter-clockwise until the target is perceived. Then, the robot approaches the target and aligns itself towards the receptacle. The robot drives forward until either a bumping sensor confirms a connection, or a timeout occurs.

2.4 Stochastic Self-reconfigurable Robots

Recently, there has been growing attention to the design and study of programmable parts that move passively and bound to each other upon random collision. White *et al.* implemented two systems in which the parts float passively on an air table that was fixed to an orbital shaker [34]. Units were un-powered and had no locomotion abilities. However, they became active once they bond to a main structure. Self-assembly has been demonstrated with up to three modules. In another system the parts where put in a fluid and the random motion was induced by the surrounding medium [35]. Self-assembly and self-reconfiguration of two un-powered blocks were studied.

Griffith *et al.* [17, 16] developed a system capable of self-assembly to study self-replication of strings of programmable, electromechanical parts. Modules slid passively on an air table and bound to each other upon random collisions. The system was capable of autonomous replication of a 5-bit string provided with an unordered supply of additional units. The replicants themselves were self-replicating artifacts.

Bishop *et al.* [7] demonstrated self-assembly with simple programmable parts that slid passively on an air table and bound to each other upon random collisions. Once attached, they executed a common graph grammar in a completely distributed fashion. Doing so, a collection of six programmable parts could form a hexagon.

2.5 Discussion

This literature review suggests that, up to now, there seem to be very few examples of self-assembling robots in which more than two robotic units manage to (a) autonomously approach and to connect to each other, and (b) accomplish self-assembly by starting from any arbitrary initial position of the modules. Only the work described in [30] shows two robotic units capable of autonomous movement and self-assembling. In all other works, only one unit is capable of autonomous movement and it assembles to a non-moving object cell (see [14, 5, 6]). Some publications report about robotic systems which are potentially capable of self-assembling. However, due to hardware and/or software limitations of the existing prototypes, no self-assembly can be achieved yet (see [23, 8]) or only if the modules are initially located in particular spatial positions and orientations [40].

There are multiple factors which have limited self-assembly to only two robotic units. Among these factors a main role is played, in our opinion, by the requirement of good alignment during the connection phase. For all the robotic systems reviewed, physical connections can be established only if the modules approach each other at specific orientations. That is, a great accuracy is required to align the connecting device in order for the robot to successfully connect. This requirement makes self-assembly an issue tightly linked to the capability of the robots to coordinate their movements. The coordination of motion during alignment becomes more complex when the connection has to be established between units already formed by assembled robots. In this case, the alignment is not any more a matter of coordinating the actions of two single units, but it requires the coordination of several units among which some are already assembled, and therefore constrained in their movements (see [8, 39, 30]).

Accurate motor coordination is the result of a tight interaction between the properties of the hardware and the robot control policy. However, if great accuracy of alignment is required for connection, even a robot which is "properly" equipped in terms of the nature and the degrees of freedom of its actuators and the variety and reliability of its sensors, may not be capable of autonomously achieving self-assembly. This is, for example, the case in the work of Bererton and Khosla [5, 6], in which, due to time constraints, the robots rely on an external PC for image processing of their camera vision system. Given the time interval within two consecutive actions, the robot was not capable of autonomously extracting from the image provided by the camera the elements of its surrounding world needed to decide what action to perform.

In the following section, we show how our research work on self-assembling robots managed to solve these problems. In particular, we show that, thanks to their sensors and motor devices, the *s-bots* "facilitate" the design of control systems to allow them to be able to autonomously build bigger robotic structures by exploiting physical connections.

3 The *s-bot*

Our experiments have been carried out using the *s-bots* (see figure 2a). The *s-bots* are mobile autonomous robots with the ability to connect to and to disconnect from each other (see [25] for a detailed description of the *s-bot* hardware). An artifact composed of a swarm of physically connected *s-bots* is referred to as a *swarm-bot*.

a b c

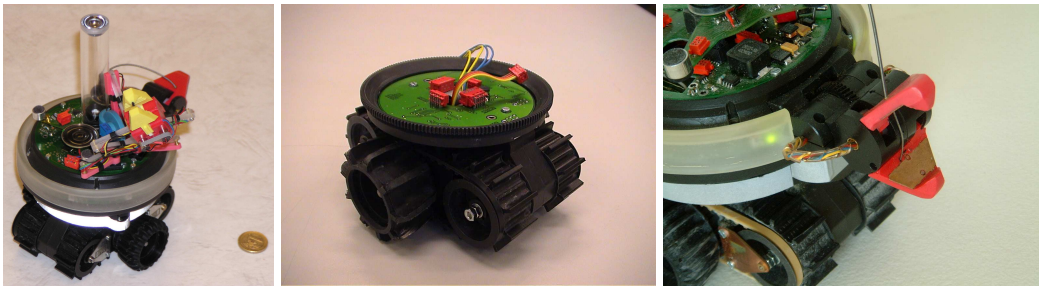
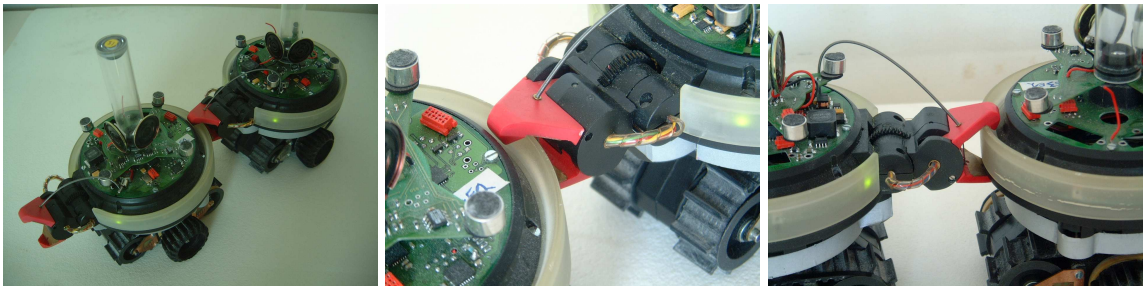


Figure 2: (a) An *s-bot* close to a 1 Euro coin; (b) the traction system of an *s-bot*; (c) the *s-bot*'s gripper.

The hardware design of an *s-bot* is particularly innovative, both concerning its actuators and its sensing devices (see figure 4). The *s-bot* is equipped with an innovative traction system which makes use of both tracks and wheels as illustrated in figure 2b. The wheel and the track on a same side are driven by the same motor, building a differential drive system controlled by two motors. This combination of tracks and wheels is labelled *Differential Treels*© *Drive*.[2] Such a combination has two advantages. First, it allows an efficient rotation on the spot due to the larger diameter and position of the wheels. Second, it gives to the traction system a shape close to the cylindrical one of the main body (turret), avoiding in this way the typical rectangular shape of simple tracks and thus improving the *s-bot* mobility and stability.

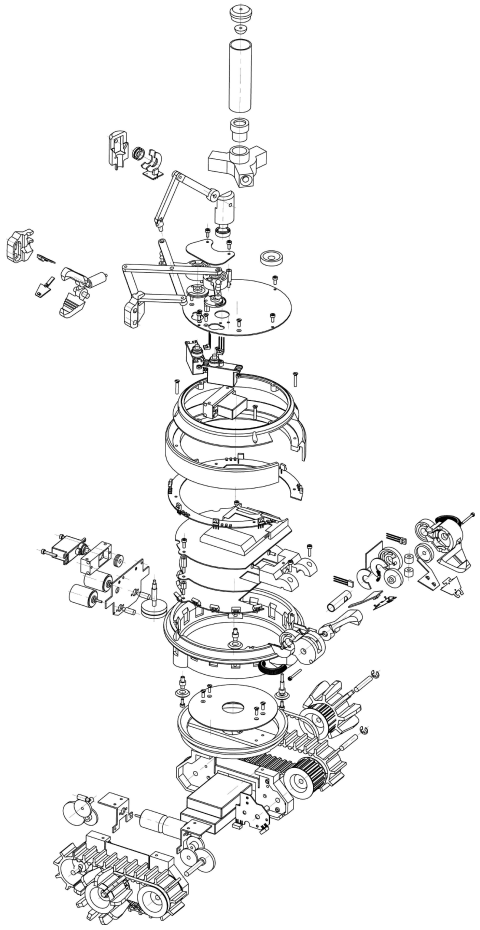
The *s-bot*'s traction system can rotate with respect to the main body—i.e., the robot's turret—by means of a motorised joint. The turret holds a gripper for establishing rigid connections between two *s-bots* or between an *s-bot* and an object (see figure 2c). The gripper is mounted on a horizontal active axis, and it has a very large acceptance area allowing it to realize a secure grasp at a wide angle range. The *s-bot* gripper can grasp another *s-bot* on a T-shaped ring placed around the *s-bot* turret (see figure 3a, b, and c). If it is not completely closed, such a grasp lets the two joined robots free to move with respect to each other while navigating. If the grasp is firm, the gripper ensures a very rigid connection which can even sustain the lifting up of another *s-bot*. An *s-bot* is provided with many sensory systems, useful for the perception of the surrounding environment or for proprioception. Infrared proximity sensors are distributed around the rotating turret, and can be used for detection of obstacles and other *s-bots*. Four proximity sensors are placed under the chassis, and can be used for perceiving holes or the terrain's roughness. Additionally, an *s-bot* is provided with eight light sensors, two temperature/humidity sensors, a 3-axes accelerometer and incremental

[2]Treels is a contraction of TRacks and whEELS



(a) (b) (c)

Figure 3: (a) Two connected *s-bots*; (b) and (c) detailed view of a connection between two *s-bots*.



- Diameter of the main body 116 mm, 100 mm in height
- All-terrain mobility using a *treels* © drive mechanism
- Rotation of the main body with respect to the motion base
- One degree of freedom rigid arm with gripper
- Three degrees of freedom flexible arm with gripper
- Optical barriers on grippers
- 15 IR proximity sensors around the *s-bot*
- 4 IR proximity sensors on the bottom of the robot
- 8x3-colour LEDs around the robot body
- 8 light sensors around the robot body
- Force sensor between *treels* © base and main body
- Torque sensor on wheels and body rotation
- 3-axes accelerometer
- Humidity sensor
- Temperature sensor
- One speaker and four microphones
- Omnidirectional camera
- Main board with 400 MHz x scale processor running Linux
- 15 microchips pic processor 20 MHz running real-time task
- Wireless communication
- The force of the rigid gripper is 14.72 N
- The elevation force of the rigid gripper is 6.87 N

Figure 4: On the left, mechanical drawing of the *s-bot*'s hardware components. On the right, a list of the technical characteristics of the *s-bot*.

encoders on each degree of freedom. Each *s-bot* is also equipped with audio and video devices to detect and communicate with other *s-bots*, such as an omni-directional camera, coloured LEDs around the *s-bot*'s turret, microphones and loudspeakers. Eight groups of three coloured LEDs each—red, green and blue—are mounted around the *s-bot*'s turret, and they can be used to display colours. The colour emitted by a robot can be detected by other *s-bots* by using an omni-directional camera, which allows to grab panoramic views of the scene surrounding an *s-bot*. As we will describe in section 4, the emission/perception of coloured cues plays a crucial role in the controllers we designed for self-assembling.

In addition to a large number of sensors for perceiving the environment, several sensors provide each *s-bot* with information about physical contacts, efforts, and reactions at the interconnection joints with other *s-bots*. These include torque sensors on all joints as well as a traction sensor to measure the pulling/pushing forces exerted on the *s-bot*'s turret. The traction sensor is placed at the junction between the turret and the chassis. This sensor measures the direction (i.e., the angle with respect to the chassis orientation) and the intensity of the force of traction (henceforth called "traction") that the turret exerts on the chassis. The traction perceived by one robot can be caused either by the force applied by the robot itself while pulling/pushing an object grasped through the gripper element, or by the mismatch of its movement with respect to the movement of other robots connected to it, or

by both the previous circumstances at the same time. The turret of an *s-bot* physically integrates, through a vector summation, the forces that are applied to it by another *s-bot*, as well as the force the *s-bot* itself applies to an object grasped. The traction sensor plays an important role in the context of coordinated movement of a group of physically connected *s-bots*—i.e., a *swarm-bot*. In particular, it can be employed to provide an *s-bot* with an indication of the average direction toward which the *swarm-bot* is trying to move. More precisely, the traction sensor measures the mismatch between the direction in which the *s-bot*'s own chassis is trying to move and the direction in which the whole group is trying to move (see [3, 11]).

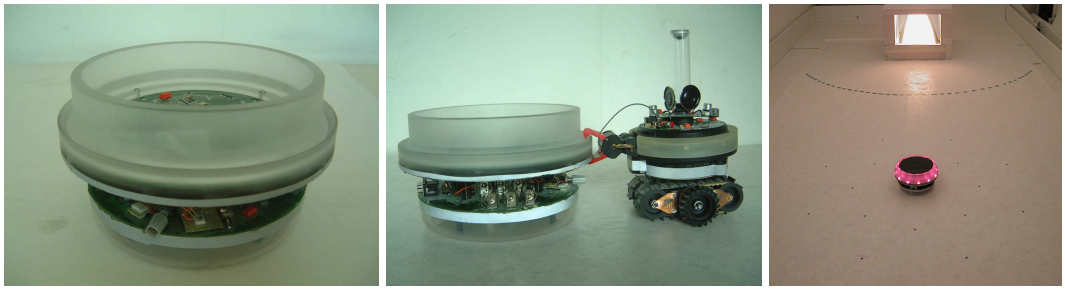
4 First set of experiments: self-assembling in cooperative transport

In this section, we describe a set of experiments in which a group of six self-assembling robots performs cooperative transport. Cooperative transport is extensively exploited by several species of ants to retrieve large and heavy items to the nest (see [1]). Usually, one ant finds a prey item, tries to move it, and, when unsuccessful for some time, recruits nest-mates. The ants grouped around the item "grasp" it and apply pulling/pushing forces until the item moves. Similarly to ants, the *s-bots* locate, approach, and finally transport an object towards a target zone indicated by a light source. Contrary to the group transport strategies employed by ants, in which each individual grasps the item, the *s-bots* transport the *prey* either by connecting directly to the object or to each other so to generate sufficient pulling/pushing forces to move the object itself. The way in which the six *s-bots* assemble around the object is dynamically determined during the development of the task. As discussed in section 1, we consider an instance of self-assembly to be the process which ends up in a structure whose elements (i.e., the *s-bots*) are physically connected to each other. In particular, in the considered cooperative transport scenario, self-assembly is such that at least one element of the assembled structure should be connected to the object to be transported. Therefore, cooperative transport may imply (although not necessarily) self-assembly. Whether or not the *s-bots* exploit self-assembly is empirically verified by counting the number of *s-bot* to *s-bot* connections in a group of agents assembled around the object. The *s-bots* are controlled by a modularised control system: the "assembly" module is in charge of controlling the behaviour of an agent during the assembly phase, in which the *s-bots* are required to directly connect to a cylindrical object, or to other *s-bots* already connected; the "transport" module is in charge of controlling the behaviour of an agent during the transport phase, in which the *s-bots* are required to coordinate their actions in order to generate sufficient forces to move the object towards the target. In the following, we first detail the methodology used in our work, and subsequently we illustrate the results.

4.1 The experimental setup

The cooperative transport task requires the *s-bots* to locate, approach and grasp an object—referred to as the *prey*, see figure 5a—that has to be subsequently transported from its initial location to a target zone. The *prey* has a cylindrical shape and is equipped with a T-shape ring of the same characteristics as the one mounted on the *s-bots*' turret. This ring makes possible for the *s-bots* to use the gripper to physically connect to the *prey* (see figure 5b). In our experimental setup, the *prey* is initially located at a distance of 225 cm from a light emitting beacon. The target zone is a circular area, centred around the beacon. The robots are successful if they manage to move the *prey* all the way down towards the target area within 5 minutes. If moved in a straight line, the distance covered by the *prey* to enter the target zone is 125 cm.

At the beginning of each trial, six *s-bots* are positioned within the arena, at a certain distance from the *prey*. The initial position of each *s-bot* is assigned randomly by uniformly sampling without replacement from a set of 16 specific starting points. The *s-bots* initial orientation is chosen randomly



(a) (b) (c)

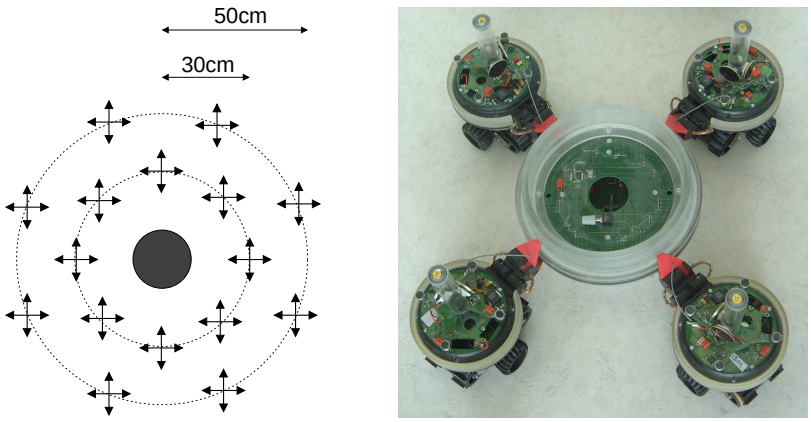
Figure 5: (a) The *prey*. (b) An *s-bot* connected to the *prey*. (c) Overview of the arena with the *prey* located at a distance of 225 cm from a light bulb which represents the centre of a circular target zone.

from a set of 4 specific directions. The 64 potential placements (16*4) of a single *s-bot* are illustrated in figure 6a.

The *prey* weighs 2310 g and cannot be moved by less than four *s-bots*. However, even four *s-bots* may not be sufficient to perform the task. In fact, the performance also depends on the way in which the *s-bots* are connected to the *prey* and/or to each other. Four *s-bots* connected in a "star-like" formation around the *prey* (see figure 6b) can move it with an average speed of about 1 cm s^{-1}.

4.2 The control policies for self-assembling

The control system described in this section has been previously designed in a relatively simple simulation environment [20], and subsequently transfered to the real *s-bot* [19, 18]. The controller is made of two sub-modules: the "assembly" module, which is in charge of controlling the *s-bot* until it is connected to the *prey* or to another *s-bot*; and the "transport" module, which allows the *s-bot* to move the *prey* towards the target area once a connection is established. The process of self-assembly is triggered by the perception of red objects. In fact, the *prey* and the *s-bots* already attached to the *prey* or to another *s-bot* have their ring coloured in red. The *s-bots* not yet connected have their ring coloured in blue. At the beginning of a trial, all the *s-bots*, controlled by the assembly module, move towards the nearest red object within their visual field and avoid collisions with not-connected *s-bots* by maintaining a certain distance to blue objects. If an *s-bot* managed to successfully connect to a red



(a) (b)

Figure 6: (a) Potential starting points and orientations of the *s-bots* around the *prey*. (b) Four *s-bots* connected in "star-like" formation around the prey.

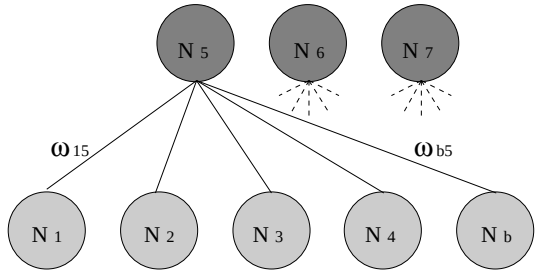
Algorithm I - The assembly module

```
1  activate colour ring in blue
2  repeat
3     (N1, N2) ← featureExtraction(camera)
4     (N3, N4) ← sensorReadings(proximity)
5     (N5, N6, N7) ← neuralNetwork(N1, N2, N3, N4)
6
7     if (N7 > 0.5) ∧ (grasping requirements fulfilled)
8        then
9              grasp
10             if (successfully connected)
11                then
12                      activate colour ring in red
13                      activate transport module
14                else
15                      open gripper
16             fi
17    fi
18    apply (N5, N6) to tracks
19 until timeout reached
```

object, it activates its colour ring in red. Therefore, it becomes itself an object with which to establish a connection. The transport module takes control of an *s-bot* as soon as the latter is successfully connected. However, there is no pulling/pushing if a connected *s-bot* perceives blue objects within its visual field. In the following, we detail the working of the two sub-modules.

4.2.1 The assembly module

The assembly module allows an *s-bot* to approach/connect with red objects and to avoid blue objects. This module is made of a feed-forward artificial neural network—a single-layer perceptron—and some hand-designed code to pre-process sensory input and to make sure that the output of the network is correctly "interpreted" by the *s-bots*' actuators. The parameters of the neural network—i.e., the connection weights—have been determined in simulation by using evolutionary algorithms. A detailed illustration of the simulation and the evolutionary algorithm used to design the artificial neural network and to develop the entire module can be found in [20]. As illustrated in figure 7, the neural network of



(a)

$$N_j = \frac{1}{1 + e^{(-x_j)}}; \quad j \in \{5, 6, 7\}$$

$$x_j = \sum_{i=1}^{4} \omega_{ij} N_i + \omega_{bj} N_b$$

(b)

Figure 7: (a) A graphical representation of the feed-forward two-layers artificial neural network of the assembly module. N_1, N_2, N_3, and N_4 are the nodes which receive input from the *s-bots* sensors. N_b is the bias term. N_5, N_6, and N_7 are the output nodes. (b) The equations used to compute the network output values.

Algorithm II - The transport module

```
1  (wait until no blue objects are perceived)
2  α ← 0
3  repeat
4     α ← computeTargetDirection(camera)
5     if (torque problem on turret or tracks)
6        then
7              execute recovery move
8        else
9              align chassis (α) and move
10    fi
11 until timeout reached
```

the assembly module has four input nodes N_1, N_2, N_3 and N_4, a bias N_b, three output nodes N_5, N_6, and N_7, and 15 connection weights (ω_{ij}). At each cycle, the network takes as input the *s-bot*' sensor readings. The input neuron N_1 and N_2 are set by extracting and pre-processing data from the *s-bot*'s vision system (Algorithm I, line 3). In particular, the feature extraction algorithm first checks whether any red or blue coloured object are perceived within a limited perceptual range bounded to the left and right side of the *s-bot*'s heading. Subsequently, the algorithm assigns a value to the input $N_1 \in \{0, 1\}$ and $N_2 \in \{0, 1\}$ according to the rules detailed in appendix A.1. The input variable $N_3 \in [0, 1]$ and $N_4 \in [0, 1]$ are set by taking the reading of the front-left-side and front-right-side *s-bot*'s proximity sensors (Algorithm I, line 4).

The network has three outputs $N_5 \in [0, 1]$, $N_6 \in [0, 1]$ and $N_7 \in \{0, 1\}$. The output neuron N_5 and N_6 set the angular speed of the left and right *s-bot*'s wheels. The values of the speed vector (N_5, N_6) are linearly scaled within the range defined by the *s-bot* speed limits. The output neuron N_7 is used to control the status of the gripper. In particular, the gripper is closed (a) if the output neuron $N_7 > 0.5$, and (b) if the gripper optical light barrier detects an object between the lower and the upper teeth of the gripper. While closing the teeth, the gripper is slightly moved up and down for several times to facilitate a tight connection. Failures of the grasping procedure can be detected by monitoring the aperture of the grasping device. In case of failure the gripper is opened again and the grasping procedure restarts from the beginning. If a red object is successfully gripped, then the *s-bot* sets the colour of its ring to red, and the transport module takes control of the robot. The *s-bot* life-span expires if it does not connect to a red object within 300 s (Algorithm I, line 19).

4.2.2 The transport module

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

The transport module exploits the camera vision system to establish the orientation of the light-emitting beacon with respect to the *s-bot*'s heading. By adjusting the orientation of the chassis with respect to the robot heading (i.e., the orientation of the turret) the *s-bot*'s controller sets the direction of motion α. The realignment of the chassis is supported by the motion of the tracks (Algorithm II, line 9).

During the transport, the *s-bot* monitors the magnitude of the torque acting on its tracks and the turret. If the torque values are particularly high, a recovery move is performed to prevent stagnation and the hardware from being damaged (Algorithm II, line 7). The recovery move lasts about 160 ms. During this time the *s-bot* moves slightly forward and backward. The life-span of a connected *s-bot* expires if it does not manage to bring the *prey* to the target-zone within 300 s (Algorithm II, line 11).

4.3 Results

In this section, we report data which represent a quantitative description of the performance of the *s-bots* engaged in the cooperative transport task. Recall that, in this task, six *s-bots* are required to assemble to and transport the *prey* from its initial position to a target zone. A trial can be divided in two different phases. In the first phase, the *s-bots*, controlled by the assembly module, try to establish a connection either directly to the *prey* or indirectly via a chain of other *s-bots*. This phase terminates once every *s-bot* has successfully established a connection. In the subsequent phase, the *s-bots*, controlled by the transport module, push/pull the prey towards the target. This phase terminates when the *prey* enters the target zone.[3]

We performed 30 replication of the experiment—i.e., 30 trials. A trial begins with the *s-bots* randomly placed around the *prey*, and it ends (a) successfully if the *s-bots* manage to transport the *prey* inside the target zone within the time-limits, or (b) unsuccessfully if, for any reason, the *s-bots* fail to transport the *prey* to the target-zone within the time-limits. Figure 8 shows a sequence of three pictures taken during a successful trial.

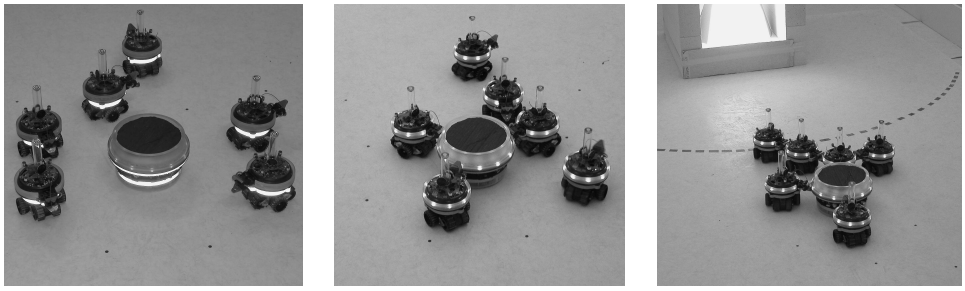
Figure 9a shows, for each trial, the number of *s-bots* which have successfully established a connection. In 26 out of 30 trials, all six *s-bots* connected. In trials n. 3, n. 12, and n. 29, a single *s-bot* failed to connect within the time limits. In trial n. 18, two *s-bots* failed to connect. Thus, out of the 180 connections required by the 30 trails—i.e., 6 connections per trial times 30 trials—we recorded only 5 failures. Due to the missing connection/s, in 4 out of 30 trials the *s-bots* did not manage to reach the transport phase. In fact, in these unsuccessful trials, the connected *s-bots* did not start to transport the *prey* due to the perception of an unconnected *s-bot*. Recall that, connected *s-bots* start transporting the *prey* only if they do not perceive any blue object—i.e., unconnected teammates.

Figure 9b shows, for each trial, the number of *s-bot* to *s-bot* connections. In this scenario, the process which generates this type of connections is considered an instance of self-assembly. As we can see, in each trial, included those in which the robots did not successfully transported the *prey* (i.e, trial n. 3, n. 12, n. 18, n. 29), we have at least two *s-bot* to *s-bot* connections.

Figure 9c shows the amount of time per trial spent by the *s-bots* in the two phases of the experiments mentioned above. Data concerning the 4 unsuccessful trials in which one or more *s-bots* fail to establish a connection are not shown. In 20 out of the 26 trials, the whole group could successfully self-assemble within 83 s, in the other trials self-assembly was successfully completed within 167 s.

Only in a single case out of those in which the *s-bots* connected successfully, the group failed to transport the *prey* entirely inside the target zone. In this unsuccessful trial, the transport was interrupted in the proximity of the target zone. This failure during the transport phase was probably

[3]The entire experiment has been recorded on video tape. Example movies are available at `http://www.swarm-bots.org/cooperative_transport.html`.



(a) (b) (c)

Figure 8: These pictures shows a sequence of actions, during a trial, in which a group of six *s-bots* randomly placed around the *prey* (a), initially locates, approaches and connects to the *prey* (b) and finally, once assembled, transports the *prey* to the target zone (c).

due to the light reflections in the immediate vicinity of the beacon which indicates the target zone. In fact, a too high intensity of the light disrupts the mechanism used by each *s-bot* to establish the direction of movement. Therefore, it may happen that, in the immediate vicinity of the target, the entire group loses efficiency in moving the *prey*. In all other cases, the *prey* entered the target zone within a short period of time. The average transport speed was 8.20 cm per s, which is about 55% of the maximum speed of a single *s-bot* moving without any load.

4.4 Discussion

The results of our experimental work have shown that the *s-bots* have the required characteristics to facilitate the design of control systems to allow them to self-assemble in a bigger physical structure. With respect to (a) the number of robots involved in self-assembly, (b) the reliability of the system, (c) the speed with which the agents generate the assembled structure, and (d) the capability of the assembled structures to coordinate their movement, our work represents a sensible step forward with respect to the state of the art in the design of controllers for self-assembling robots.

Moreover, our modular architecture has already proved successful in controlling the *s-bots* in a different scenario in which self-assembly is required to navigate a terrain with two different types of hills (more details on this research work can be found in [29]). In this task, simple hills can be overcome by a single *s-bot*, the difficult ones can not; that is, the *s-bots* topple backwards due to the steepness of the slope. The *s-bots* have to self-assemble in order to overcome the steep hill. The

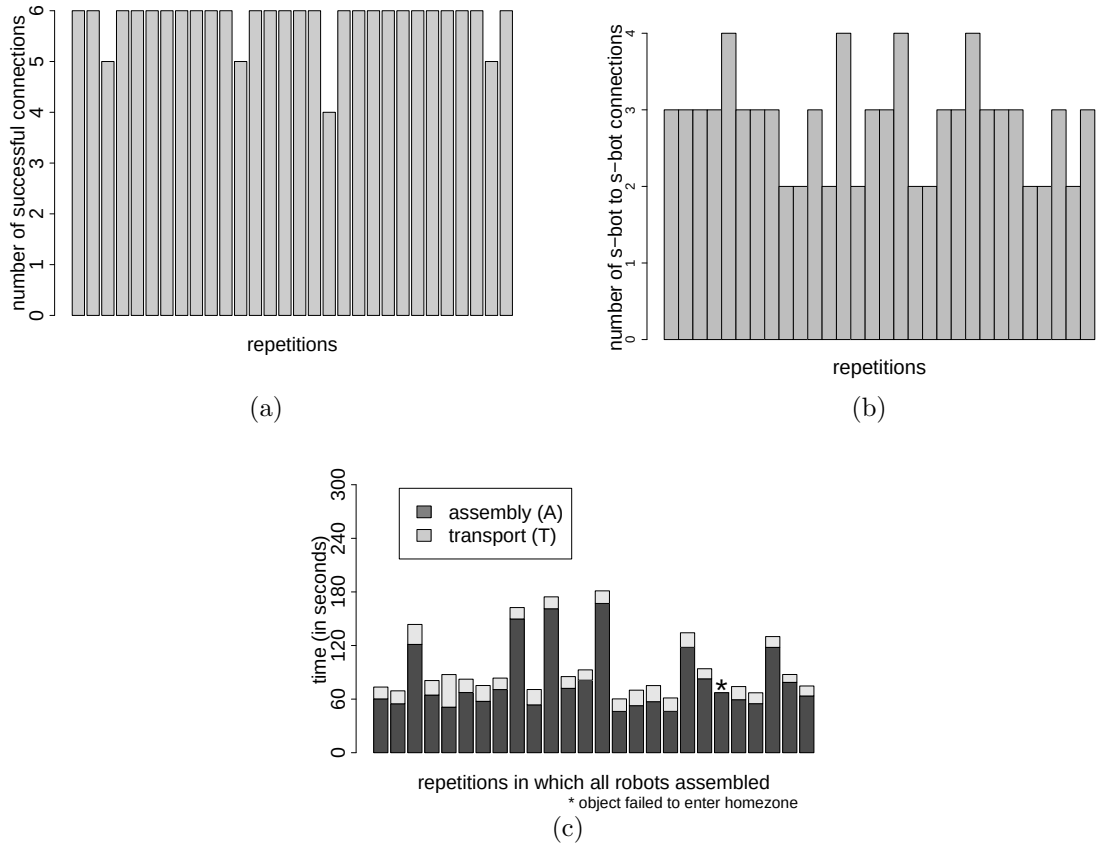


Figure 9: (a) Number of robots successfully connected. (b) Number of *s-bot* to *s-bot* connections. (c) Time period the group was busy self-assembling and transporting the *prey* inside the target zone.

experiment shows that the modular architecture previously described can be easily extended with other control mechanisms to allow the *s-bots* to exploit self-assembly in different context.

Although these results are particularly encouraging, we are not underestimating the limitations of our modular approach which may have a disruptive effect on the performance of the robotic system. For example, we have seen in the cooperative transport task that, if a "red" *s-bot* (i.e., an *s-bot* already connected) sees a "blue" *s-bot* (i.e., an *s-bot* not connected yet), the red one remains still. This mechanism has both positive and negative consequences. On the one hand, it facilitates the connection of the "blue" *s-bot* to the "red" *s-bots*, as all the red objects located in its surrounding do not move. On the other hand, as far as even a single *s-bot* fails to connect, and at the same time it remains within the visual field of other *s-bots* already attached, the transport phase can not begin, and consequently the trial ends unsuccessfully.

In order to overcome these problems, we are starting to investigate new collective decision mechanisms. For example, the decision to start a collective action (e.g., the group transport of an item, or moving uphill along a steep hill) might begin anytime a *swarm-bot* is capable of overcoming the difficulties which demand self-assembling, regardless of the number of *s-bots* connected. With this approach, we would let the system comply with its objectives without having to satisfy a set of *a priori* defined conditions, such as the requirement of having all the robots of a group connected to an item and/or to each other before starting the transport phase.

In our "work in progress" on the development of controllers for self-assembling robots, we are also exploring alternative methodologies, which try to minimise the amount of *a-priori* assumptions—made by the experimenter—regarding the domain of perception and action of the autonomous agents. The next section introduces our initial efforts on the design of controllers for self-assembling robots based on techniques which can potentially enhance the adaptiveness of a multi-robot autonomous system, reducing in this way the risk of incurring in the drawbacks discussed above.

5 The evolution of integrated neuro-controllers for self-assembling robots

The complexity of self-assembly resides in the nature of the perceptual and motor mechanisms with which each single robot should be equipped. In particular, a robot necessitates mechanisms to be able to autonomously (a) decide whether or not the environmental contingencies require self-assembly, (b) coordinate its movements to connect to and/or facilitate the connection from other *s-bots*, and (c) coordinate its movements once connections are established. As we said in the previous section, we are currently investigating different alternatives to enhance the adaptiveness of our self-assembling autonomous robots. One of our research direction is to explore the potentiality of integrated (i.e., not-modularised) artificial neural network controllers synthesised by evolution.

The rationale for employing these methodological tools can be found in the following two considerations. First, it is known to be particularly difficult to handcraft individual behavioural rules which arbitrate the response of an autonomous cooperative multi-robot system. Any time the individual behaviour is the result of the interaction between an agent and a dynamic environment, then it is difficult to predict which behaviour results from a given set of rules, and which are the rules behind an observed behaviour. Second, the adaptiveness of an autonomous multi-robot system is reduced if the circumstances an agent should take into account to make a decision concerning individual or collective behaviour are defined by a set of *a priori* assumptions. For example, when and with whom to self-assemble are two decisions which should be governed as much as possible by robots-environment contingencies not determined by the experimenter.

If we take into account the above mentioned considerations, it turns out that a promising approach to the design of controllers for autonomous cooperative multi-robot system is the evolutionary robotics (ER) approach (see [21, 28] for an introduction to ER). Research works in ER are generally based on the combination of artificial neural networks as robot controllers and artificial evolution as a design methodology. In particular, artificial evolution allows the automation of the design process

of the robot's control policies through the use of mechanisms inspired by natural selection. Artificial evolution partially bypasses the problem of decomposition at both the level of finding the mechanisms that lead to the emergent global behaviour and at the level of implementing those mechanisms in a controller for the robots. In fact, it relies on the evaluation of the system as a whole, that is, on the emergence of the desired global behaviour starting from the definition of the individual ones. The adaptiveness of the agent's mechanisms is determined by an evolutionary process which favours (through selection) those solutions which improve the "fitness" (i.e., a measure of an agent's ability to accomplish its task) of an agent and/or of a group of agents. The evolved mechanisms are also expected to cope with a certain amount of environmental variability experienced during evolution. Artificial neural networks provide evolution the building blocks to design the mechanisms an agent needs to perceive and act in its world. The evolved neuro-controllers allow an agent to distinguish and recognise the elements of its surrounding by exploiting perceptual cues which, "viewed" through its sensors, distinctively identify an object. Consequently, actions are initiated with respect to particular environmental conditions that emerge through the dynamics of the system components. Thus, these conditions might be *a-priori* unforeseeable by the experimenter. In the modularised approach illustrated in section 4, each agent perceives and acts according to conditions that are based on arbitrary associations, done by the experimenter, between sensorial cues and elements of the agent world (e.g., the red colour indicates objects to connect with). For example, the output of the neural network that controls the gripper is not directly used to set its state, but it is an element among others used to define the action to perform. In the approach we are going to present in this section, the evolved neural network is fully in charge of determining the state of the robot actuators and consequently its behaviour.

Notwithstanding its potentialities, the ER approach hasn't been previously used to design controllers for robots required to perform individual and collective responses such as self-assembling. In this section, we describe the methodology and we show the results of a set of simulations which represent a first step toward the synthesis through artificial evolution of integrated (i.e., not-modularised) artificial neural network controllers. The neuro-controllers should allow the *s-bots* (a) to autonomously decide which actions—i.e., individual or collective—to undertake with respect to the environmental conditions; and (b) to coordinate their actions to bring forth a bigger physical robotic structure. We underline that this section illustrates a study that represents a stepping stone toward the development of more advanced neuro-controllers for self-assembling. In spite of the simplifications introduced, we believe that this work contains all the required ingredients to evaluate the potentiality of the ER approach. The obtained results bring significant contributions, because this is one of the first works in which integrated artificial neural network controllers, synthesised by artificial evolution, proved capable of controlling robots that display a wide repertoire of individual and collective behaviours.

5.1 Methods

In the following subsections, we detail the characteristics of the task, the methodology employed to evolve *s-bots*' controllers and the evaluation function used.

5.1.1 Description of the task

Our study is focused on a scenario in which the *s-bots* should prove capable of performing individual and collective responses with respect to what the circumstances seem to require. In particular, we are interested in circumstances in which the *s-bots* should:

1. Independently perform a specific task. That is, if assembling is not required, *s-bots* should be capable of individually achieving their goal.

2. Aggregate in order to allow subsequent assembling. That is, if assembling is required by particular environmental contingencies, the *s-bots* should be capable of bringing forth the conditions

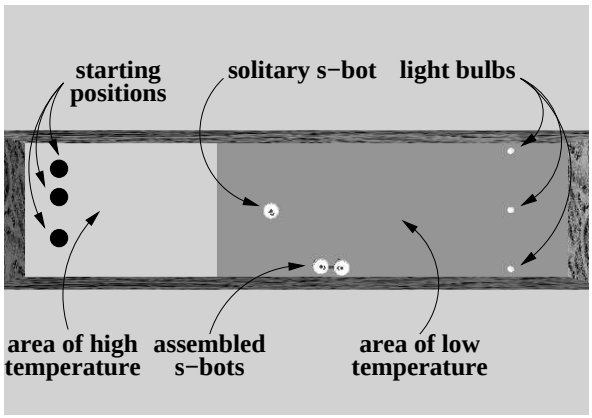


Figure 10: A graphical representation of the task. See text for details.

which facilitate self-assembly. Aggregation is the first steps in order to form an assembled structure—i.e., a *swarm-bot*.

3. Move coordinately in order to physically assemble. That is, each *s-bot* should find the correct position with respect to another *s-bot* in order to be able to establish a connection.

4. Move coordinately in order to contribute to the effectiveness of the behaviour of the assembled structure. That is, the *s-bots* should perform coordinate actions in order to achieve their common goal.

5. Disconnect. That is, once the environmental contingencies do not require any longer the assembled structure, the *s-bots* should disconnect and carry out their goal independently of each other.

An example of task with the above characteristics is the one in which a group of *s-bots* must move from a starting position to a goal location. During the movement the robots must traverse zones that may require or not to be in a self-assembled configuration (i.e., a *swarm-bot*). For example, the *s-bots* might start in a flat terrain zone in which the most efficient choice is to move independently of each other, then reach a rough terrain zone where by self-assembling into a *swarm-bot* they minimise the probability of toppling over, and finally enter the goal location area where the terrain is again flat and where they should therefore disband and continue moving independently of each other.

Committed to the principle of the Occam's razor, we tried to simplify as much as possible the characteristics of the above scenario without losing the significance of our work. In particular, the task we selected requires navigation within a rectangular corridor in order to approach light bulbs positioned on the opposite end with respect to the *s-bots*' starting positions (see Figure 10). The corridor (4 meters long, 1 meter wide) is divided in an area of high temperature and an area of low temperature (respectively, light and dark gray in Figure 10). Aggregation and assembling are required in order to traverse a low temperature area, within which a *swarm-bot* (i.e., assembled *s-bots*) navigates more effectively than a group of disconnected *s-bots*.

In our simulation, the "climatic" metaphor is just a simple way to model an environment made of two parts: one in which the *s-bots* should move not-assembled, and the other in which they should move in a *swarm-bot* formation (i.e., assembled). The "temperature" can be perceived by a single binary sensor which returns 1 if the *s-bots* are in a high temperature area, and 0 otherwise. This is a strong simplification with respect to more realistic scenarios, in which the *s-bots* might be required to employ more complex sensory-motor skills in order to perceive those environmental contingencies that require assembling. However, moving away from more "realistic" to our simulated scenario, the

peculiarity by which different areas of the environment require different responses (i.e., individual or collective) is kept unchanged.

In our simulation, the *s-bots* are allowed to make use only of a sub-set of all the sensors and actuators available to a real *s-bot*. Concerning the sensors, the *s-bots* can use their traction sensor, whose reading returns four variables encoding the traction force from four different preferential orientations with respect to the robot's chassis (front, right, back and left, see [2] for more details). The *s-bots* can also use two light sensors positioned on the front and on the back of their body. Notice that the light sensors are positioned on the turret, which might rotate with respect to the chassis. The simulated *s-bot* takes the readings from those light sensors which at any time happen to be at a specific orientation with respect to the chassis. Noise is simulated for all sensors, adding a random value uniformly distributed within the 5% of the sensors saturation value.

Concerning the actuators, *s-bots* can control the two wheels, independently setting their speed in the range $[-6.5, 6.5]$ rad/s. The loudspeaker can be switched on, simulating the emission of a continuous tone, or it can be turned off. *S-bots* are provided with a simulated gripper, that can be in either of two states: connected to another *s-bot* or open. Connections among *s-bots* are simulated creating a joint between the two *s-bots*' bodies. The creation of the joint between the *s-bots*' bodies directly follow a successful attempt to close the gripper. If the connection attempt fails, we force the gripper to stay open and ready for another connection. The connection procedure is idealised and is performed within a single time-step. Finally, the motor controlling the rotation of the turret is used, even though it is not directly controlled by the evolved neural network. When *s-bots* are not connected, this motor ensures the alignment between the turret and the chassis. On the contrary, when an *s-bot* is connected to other *s-bots* to form a *swarm-bot*, the turret can rotate freely.

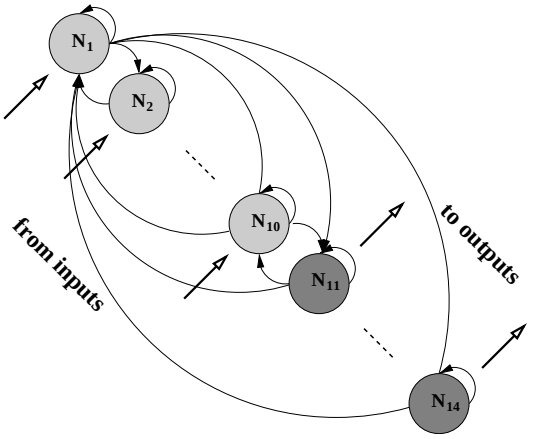
At the beginning of each trial, three *s-bots* are randomly positioned and oriented at one end of the corridor, in the area of high temperature. The light bulbs, located at the opposite end of the corridor, can be perceived by the *s-bots* from anywhere within the corridor. The intensity of the light which impinges upon the *s-bots* light sensors decreases quadratically with the distance from the light sources. The simulation is deliberately noisy, with noise added to all sensors. This is also extended to the environmental parameters: at the beginning of each trial, the point in which the temperature changes from high to low is redefined randomly within certain limits (see also [33] for further details).

5.1.2 The controller and the genetic algorithm

Groups of *s-bots* are controlled by artificial neural networks, whose parameters are set by an evolutionary algorithm. A single genotype is used to create a group of individuals with an identical control structure—i.e., a homogeneous group of robots. The *s-bot*'s controller is a fully connected, 14 neuron continuous time recurrent neural network (see also [4] for details). The neurons either receive direct sensor input or are used to set the state of an *s-bot*'s actuators (see figure 11). There are no internal neurons. All but four of the neurons receive direct input from the robot sensors. Each input neuron is associated with a single sensor, receiving a real value in the range $[0.0, 1.0]$, which is a simple linear scaling of the reading taken from its associated sensor.[4]

The four remaining neurons are used to control the *s-bot*'s actuators, after mapping their cell potential y_i onto the range $[0.0, 1.0]$ by a sigmoid function. Two of them are used to set the *s-bot*'s wheels speed, linearly scaling the output into $[-6.5, 6.5]$. The third motor neuron is used to set the state of the loudspeaker, which is turned on if the neuron output is higher than 0.5, and off otherwise. The last motor neuron controls the gripper actuator, trying to set up a connection if the neuron output is higher than 0.5, and keeping the gripper open otherwise. The strengths of the synaptic connections, the decay constants, bias terms and the gain factor are all genetically encoded parameters. Cell potentials are set to 0 each time a network is initialised or reset. State equations are integrated using the forward Euler method with an integration step-size of 0.1.

[4]Specifically, neurons N_1 to N_4 take input from the 4 variables encoding the traction force, neurons N_5 to N_7 take input from the sound sensors (i.e., the directional microphones), N_8 and N_9 from the virtual light sensors, and N_{10} from the temperature sensor.



(a)

$$\frac{dy_i}{dt} = \frac{1}{\tau_i}\left(-y_i + \sum_{j=1}^{14} \omega_{ji} z_j + g I_i\right)$$

where $\quad z_j = \dfrac{1}{1 + e^{-(y_j + \beta_j)}}$

(b)

Figure 11: (a) A graphical representation of the artificial neural network employed to control the *s-bots*. The nodes in light grey represent those which receive input from the *s-bots* sensors. The nodes in dark grey represent those whose activation values are used to set the *s-bots* actuators. (b) The equations governing the neuron internal state. Here, by analogy with real neurons, y_i is the cell potential, τ_i the decay constant, β_j the bias term, z_j the firing rate, ω_{ji} is the strength of synaptic connections from the j^{th} neuron to the i^{th} neuron, I_i the intensity of the sensory perturbation on sensory neuron i, g is a gain factor.

In order to set the parameters of the *s-bots*' controllers, a simple generational genetic algorithm (GA) is employed (see [15]). Initially, a random population of 100 genotypes is generated. Each genotype is a vector of 1800 binary values—8 bits for each of the 225 parameters, that is, 196 connections, 14 decay constants, 14 bias terms, and 1 gain factor. Subsequent generations were produced by a combination of selection with elitism and mutation. Recombination is not used. At every generation, the best 20 genotypes are selected for reproduction, and each generates 4 offspring. The genotype of the selected parents is copied in the subsequent generation; the genotype of their 4 offspring is mutated with a 5% probability of flipping each bit. One evolutionary run lasts 1000 generations.

The binary values of a genotype were mapped to produce CTRNN parameters with the following ranges:

- connection weights: $\omega_{ji} \in [-6, 6]$.
- biases: $\beta_j \in [-2, 2]$.
- gain factor: $g \in [1, 13]$.

Concerning the decay constants, the genetically encoded parameters were firstly mapped onto the range $[-1, 1]$ and then exponentially mapped onto $\tau_i \in [10^{-1}, 10]$.

5.1.3 The evaluation function

During the evolution, a genotype is mapped into a control structure that is cloned and downloaded to the *s-bots* taking part in the experiment. Groups of 3 *s-bots* are evaluated 5 times—i.e., 5 trials. Each trial differs from the others in the initialisation of the random number generator, which influences mainly the *s-bots* starting positions and the point beyond which the temperature drops from 1 to 0. In each trial θ, the lifetime of an *s-bot* is limited to 600 simulation cycles, corresponding to 60 s of real time. The behaviour of the *s-bots* is evaluated according to an evaluation function that averages the individual contribution of each *s-bot*. Individual contributions are designed in order to reward (a) phototaxis, looking at the distance covered along the corridor, and (b) self-assembly, looking at both

the strength an *s-bot* has at the end of a trial and at the size of the *swarm-bot* formed in order to reach the light bulbs (see appendix A.3 for a detailed description of the evaluation function).

Notice that, the effectiveness of the navigational strategies is evaluated by employing a performance measure which we refer to as "strength". At the beginning of a trial each *s-bot* has a certain strength. While performing the task, each *s-bots* keeps its strength by navigating disconnected in the area of high temperature, and assembled—i.e., by forming a *swarm-bot*—in the area of low temperature. If, while navigating, an *s-bot* exhausts its strength, it is not able to move any more. The *s-bots* do not have any information concerning their strength. However, the *s-bots* can reach the light bulbs before running out of strength if they properly react to the characteristics of the environment. In particular, an optimal strategy requires the *s-bots* (i) to individually move toward the light bulbs as long as the temperature remains high; (ii) to aggregate by exploiting the sound signalling system they are provided with as soon as the temperature drops; (iii) to continue their phototactic behaviour in an assembled structure (i.e., by forming a *swarm-bot*) throughout the low temperature area. A detailed description of how the *s-bot* strength varies while it is acting within the corridor is given in appendix A.2.

We would like to emphasise that "strength", as a performance measure, does not refer to any physical property of the *s-bots*. Moreover, it does not imply the use of unrealistic sensors, which cannot be instantiated on the real *s-bots*. In fact, the *s-bots* do not have any feedback about their own strength. The strength has been mainly introduced to evaluate the behaviour of a robot and to associate it to a fitness score. Thus, the strength plays an important role only in the evaluation procedure, because it locates the observed behaviour in a unidimensional metric space in which good strategies have a high score and bad strategies have a low score. This metric space, by playing a role in determining the fitness of the agents, helps the evolutionary algorithms to find a path towards the emergence of more adaptive controllers. We also make use of the "strength" in the results section to visualise what kind of strategy an *s-bot* is employing while it is moving towards the light. For example, a sudden drop in the strength level can be interpreted as the shift of a non-assembled robot from a high to a low temperature area.

5.2 Results

Ten evolutionary runs, each using a different randomly initialised population, were run for 1000 generations each. Two runs out of ten ended up successfully by producing controllers capable of displaying self-assembly. Figure 12 shows the fitness of the best group of *s-bots* and the average fitness of the population for each generation. Two prototypic runs are shown: a particularly successful one (top) and an unsuccessful one (bottom).

An analysis of the controllers produced by the unsuccessful runs revealed that these groups of *s-bots* have been only partially capable of solving the task. We observed that, while in these runs the *s-bots* were capable of phototaxis and obstacle avoidance, only in few runs they were able to properly react to the decrease in temperature. On the contrary, in the two successful runs, the groups of *s-bots* showed the complete repertoire of behaviours required by the task. In an additional series of post-evaluations we looked at the behavioural strategies employed by the best evolved group of *s-bots* to perform the task. In the first post-evaluation test, we simply observe, for each *s-bot*, how the strength level and the covered distance—the distance between the current position of an *s-bot* and the starting position, along the x axis—vary over time (see figure 13). Given the way in which these two variables change over time within a trial, we can infer that each *s-bot* undergoes four different behavioural phases: individual phototaxis, aggregation, self-assembly and collective phototaxis.

In the first phase—from cycle 0 to the time indicated by the empty circle—the three *s-bots*, located in the high temperature area and with full strength, perform individual phototaxis, as shown by the continuous line in figure 13. The second phase starts when the *s-bots* enter the low temperature area. Three phenomena can be observed: aggregation, decrease in the strength level and signalling behaviour. Aggregation is indicated by the covered distances of the three *s-bots* (see continuous lines in figure 13), which reach similar values before the end of the phase. The decrease in the strength

level indicates that the *s-bots* move independently. The *s-bots* react to the temperature decrease by switching on their loudspeaker, signalling their position to the other *s-bots* (see eq. 2 in appendix A.2). The rate of change of the *s-bot* strength value is also affected by the signalling behaviour of the *s-bot*. Since the strength level converges, for each *s-bot*, to a certain value higher than 0, we can deduce that the *s-bots* react to the temperature decrease by switching on their loudspeaker, signalling their position to the other *s-bots*. The sound signalling should in principle provide enough information to allow the *s-bots* to aggregate. However, we observed that the *s-bots* tend also to exploit environmental structures, such as the walls of the corridor, in order to get close to each other. The third phase corresponds to self-assembly. In figure 13, this phase is indicated by an increase in the strength level (dashed line), caused by the *s-bots* connecting to each other when located in the low temperature area (see eq. 1 in appendix A.2). In this particular case, *s-bots* 1 and 2 self-assemble first, while *s-bot* 3 joins the *swarm-bot* later. Collective phototaxis is performed during the last phase. Here, *s-bots* move assembled in a *swarm-bot* that approaches the light bulbs, as indicated in figure 13 by the synchronous increase of the covered distance (see continuous lines).

In the second post-evaluation test, we looked at the capability of the best evolved group of *s-bots* to disassemble—i.e., to switch from a *swarm-bot* formation to not-connected *s-bots*—as a reaction to an increase of the environmental temperature. Notice that this circumstance has never been encountered by the *s-bots* during the evolutionary phase. Therefore, disassembling should be considered an additional capability of the evolved controllers, which confers robustness to the system. We placed the *s-bots* in a corridor with four temperature areas: two high temperature and two low temperature areas (see Figure 14). The graphs in Figure 15 show how the covered distance and the strength level of each *s-bot* vary in time while the *s-bots* move down the corridor toward the light bulbs. In this case, we focus our attention on how the *s-bots* react to the transition from low to high temperature areas. In fact, the transitions from high to low temperature areas result in a variation of the covered distance and of the strength levels similar to what was observed and discussed for Figure 13.

The transition of the *s-bots* from low to high temperature areas is indicated in the graphs by

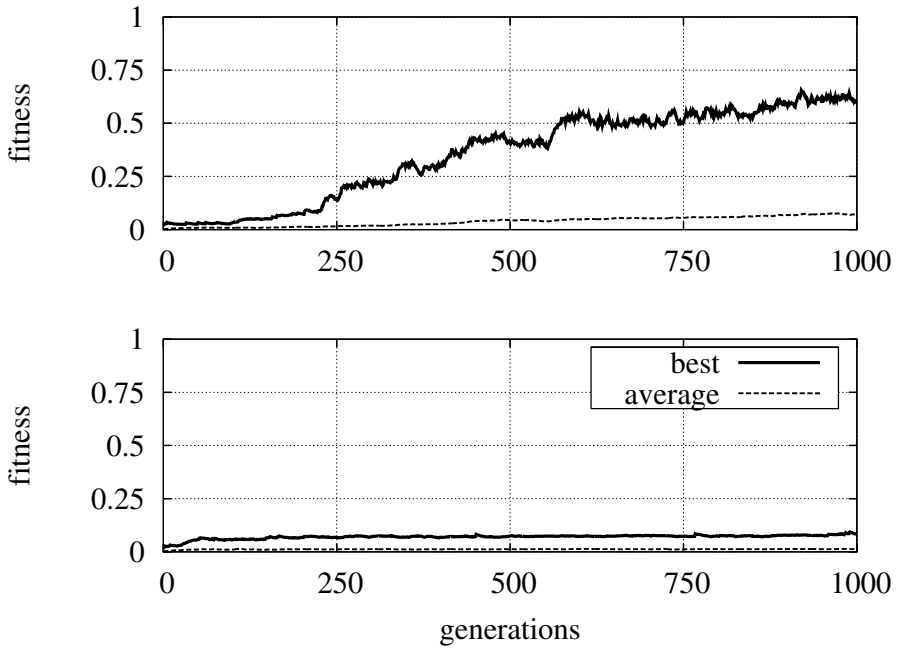


Figure 12: The graphs show the fitness of the best group of *s-bots* (thick line) and the normalised average fitness of the population (thin line), for each generation, for a successful run (top graph) and an unsuccessful one (bottom graph).

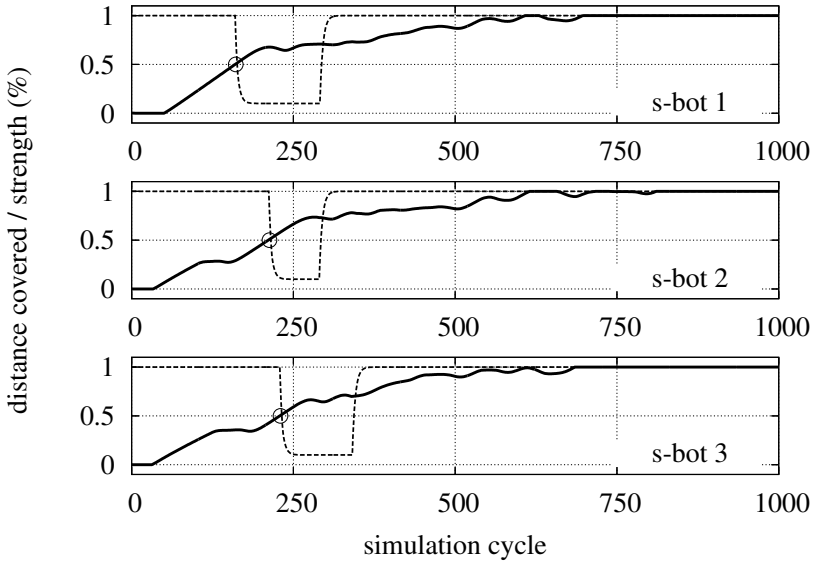


Figure 13: The graphs refer to a post-evaluation of the best evolved group of three *s-bots*. In particular, each graph shows how the covered distance along the corridor (continuous line) and the strength (dashed line) of an *s-bot* vary during a post-evaluation which lasts 1000 simulation cycles. The empty circles indicate the time when an *s-bot* enters the low temperature area.

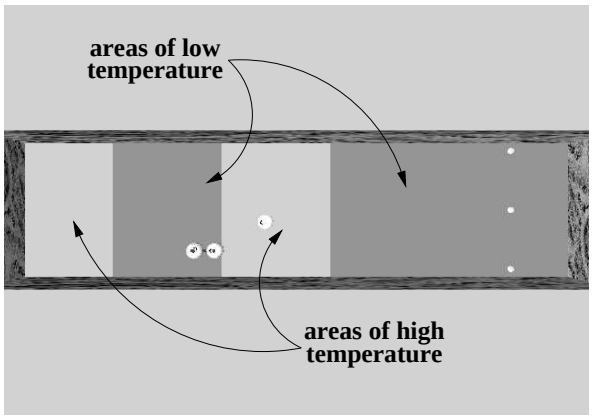


Figure 14: A graphical representation of the environment with two high temperature and two low temperature areas. This environment has been used for post-evaluation to check whether the *s-bots* capable of assembling were also capable of disassembling in response to an increase in the environmental temperature.

a filled circle. This transition is characterised by two different phases. Initially, a decrease in the strength level is observed, when an *s-bot*, still assembled in a *swarm-bot* formation, perceives the new environmental condition (high temperature). Subsequently, the *s-bots* progressively disconnect from each other, which results in a gain in the strength level (see eq. 3 in appendix A.2). In the particular case illustrated in Figure 15, *s-bot* 1 is the first to perceive the high temperature area and consequently to disassemble from the *swarm-bot*. It is possible to notice that *s-bot* 1, after disconnecting, moves back and forth, experiencing twice the low-to-high temperature transition. Similarly, *s-bot* 2 disconnects from *s-bot* 3 as soon as it ends up in the high temperature area. Consequently, *s-bot* 3 finds itself alone

in the area of low temperature. It is possible to notice that its strength drops, due to the fact that the *s-bot* has the loudspeaker turned on. Nevertheless, the *s-bot* still has enough strength to perform individual phototaxis and to approach the high temperature area. Once in the high temperature area, its strength increases again, indicating that the *s-bot* has switched off the loudspeaker. Its covered distance indicates that the *s-bot* approaches the light bulbs, reaching and finally connecting to the other 2 *s-bots*.

In conclusion, the post-evaluation tests showed that the group of three *s-bots* mentioned above successfully employs self-assembly to navigate the low-temperature area. Self-assembly is functional to the accomplishment of a particular task, that could not be individually solved by the *s-bots*. Simple and effective decision making mechanisms trigger (a) the aggregation and the subsequent assembling of the *s-bots* as soon as the latter enter the low temperature area; (b) disassembling of the *swarm-bot* as soon as the environmental contingencies that hinder individual actions cease to exist.

6 Conclusions

The experimental work illustrated in this paper summarises our research activities carried out with robots capable of physically connecting to each other—i.e., the *s-bots*. Owing to their characteristics, the *s-bots* facilitate the design of the mechanisms required for self-assembly. In particular, the functional properties of the gripper mounted on the *s-bots* turret, the T-shape ring which surrounds the *s-bots* body, the good mobility and the large sensory capabilities of the *s-bots* are the ingredients which make our robots particularly suitable to investigate the potential benefit of self-assembly in multi-robot systems. The empirical evidences shown in the paper seem to confirm our claim.

The results of the first set of experiments proves that our work represents a sensible step forward with respect to the state of the art in the design of controllers for self-assembling robots, in particular if we look at (a) the number of robots involved in self-assembly, (b) the reliability of the system, (c) the speed with which the agents generate the assembled structure, and (d) the capability of the

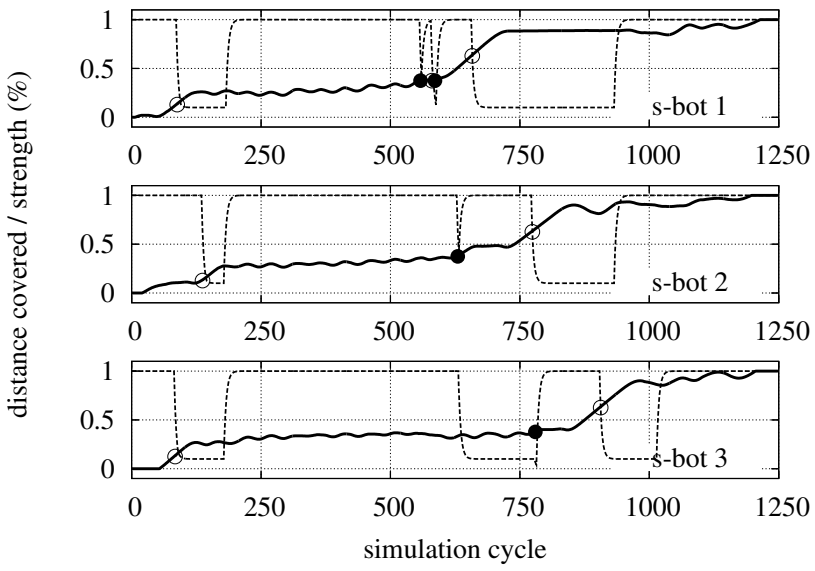


Figure 15: Each graph shows how the distance to the light bulbs (continuous line) and the strength (dashed line) of an *s-bot* vary during a post-evaluation which lasts 1250 simulation cycle. The empty circles indicate the time when an *s-bot* enters a low temperature area. The filled circles indicate the time when an *s-bot* enters a high temperature area.

assembled structures to coordinate in order to transport a heavy object. These experiments make clear that the control policies design is facilitated by the fact that the *s-bots* do not require a very precise alignment during the connection phase, and that they are equipped with a set of sensors which guarantees a sufficient accuracy as far as it concerns the capability of the robot to coordinate their movements, both before and after a connection is established.

In the second set of experiments, we started considering self-assembly within a framework in which the mechanisms for sensory-motor coordination are combined to decision making structures to allow the *s-bots* to decide when it is time to gather and pursue collective strategies. The aim of this work is to enhance the adaptiveness of a group of self-assembling robots by reducing to the minimum the *a-priori* assumptions concerning the nature of the control mechanisms that, by working on the agent's perceptual evidence, guide a multi-robot system in an intrinsically complex scenario. The results show that our methodology is promising: the evolved controllers are capable of displaying individual and collective obstacle avoidance, individual and collective phototaxis, aggregation and self-assembly. To the best of our knowledge, our experiments represent one of the first works in which self-assembly in a homogeneous group of robots has been achieved and evolved neural controllers successfully cope with such a complex scenario, producing different individual and collective responses based on the appropriate control of the state of various actuators triggered by the local information coming from various sensors.

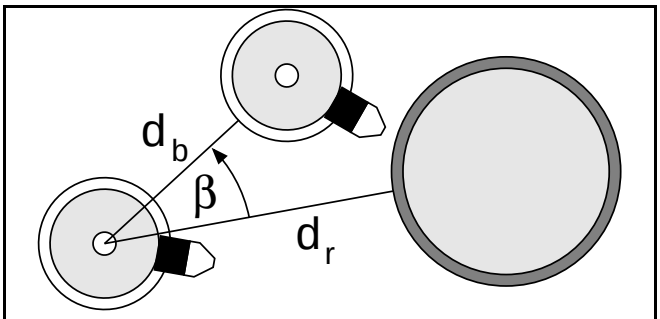
7 Future work

Overall, our work represents the beginning of a challenging research agenda in which we intend to further develop the capabilities of self-assembling robots. The results we achieved so far encourage us to further pursue our interests. There seems to be several directions for future work. Our main objective is in further developing the methods we described in section 5 to investigate scenarios in which the controllers can be directly ported and tested on the real *s-bots*. The following are just a few of the instances in which the functionality of the *swarm-bot* should be preferred to any other individual solution:

1. passing over a trough larger than the body of a single *s-bot*;
2. climbing a steep slope;
3. navigating on very rough terrain in which a single *s-bot* might topple over;
4. collective and cooperative transport of heavy items.

In order to face these challenges, the *s-bots* necessitate (a) the decision making structures to perceive those environmental contingencies that require self-assembling; (b) the mechanisms to bring forth the coordination of actions necessary for self-assembling; (c) the mechanisms to guarantee the efficiency of the assembled structures. Our intuition is that, in a near future, we might be able to design, through artificial evolution, an integrated neural network capable of providing a real *s-bot* all the above mentioned mechanisms. As far as it concerns (a) and (c), we are particularly optimistic. We believe that, even if the conditions that require self-assembling are perceived through articulated visual (i.e., through the camera image) and/or proprioceptive sensors (i.e., through the inclinometer reading), an artificial neural network can potentially process this input to let an agent initiate collective responses. Moreover, we have already tested in several circumstances the reliability of the traction sensor to coordinate the movement of a *swarm-bot* (see [32] for details).

At the moment, the uncertainties lie rather in (b). We still have to prove that, first, it is possible for an integrated network to coordinate the actions of a real *s-bot* during the docking phase by modulating the speed of the robot wheels and the state of the gripper; second, it is possible to design mechanisms to accomplish docking when the agent/object to grasp moves as well. A successful docking might be accomplished by exploiting the infra-red proximity sensors which can provide information on the



rule	conditions		N_1	N_2
1	$d_b < d_r \wedge d_b \leq D$	$-60^\circ < \beta < 90^\circ$	0	0
2	$d_b < d_r \wedge d_b > D$	$-25^\circ < \beta < 25^\circ$	0	0

Figure 16: The figure depicts a scenario in which an unconnected *s-bot*, represented by the small round object at the bottom left side, perceives another unconnected *s-bot* and the *prey*, represented respectively by the small round object at the top and by the big circle on the right.

distance and orientation between an agent and an object or another agent to grasp. Furthermore, the output of the network can be used to set the acceleration of the robot wheels instead of the speed. This might help the network to better modulate the movement of an agent during docking. Our future work will concentrate on studying these alternatives for the evolution of neural networks to allow the *s-bots* to self-assemble in response to the above mentioned scenarios.

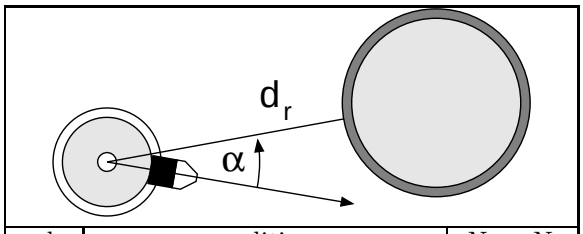
APPENDIX

A.1 The "feature extraction" algorithm from the *s-bot*'s camera image

The state of the first two input neurons (N_1, N_2) of the neural network that controls the robot during the assembly phase are set at each time-step by pre-processing the *s-bot*'s camera image. These readings are determined with respect to the presence/absence of red/blue objects within the *s-bot*'s visual field. In particular, the tuple (N_1, N_2) keeps the default value $(0, 0)$ in the following cases: (a) if there is no blue or red object within the *s-bot* visual field; (b) if only blue objects are perceived; (c) if a blue object is perceived closer than a red object, as depicted in figure 16. The table at the bottom of this figure details the conditions employed to set the values of the variables N_1 and N_2. The value d_b is the distance between the robot and the nearest blue object, while d_r is the distance between the robot and the nearest red object, and β corresponds to an angle (in degrees) which measures the distance between the red and the blue object. D is an estimate of the minimal distance between the *s-bot* and another object for which there is no risk of collision. The tuple (N_1, N_2) is set to values $(1, 1)$, $(0, 1)$, or $(1, 0)$ in case a red object is perceived by the *s-bot* and this red object is closer to the *s-bot* than any other blue object, as depicted in figure 17. The table at the bottom of the figure details the conditions employed to set the values of the variables N_1 and N_2. The values d_r and α (in degrees) correspond respectively to the distance of and the direction to the closest red object within the perceptual range of 45° left and right with respect to the *s-bot*'s heading. D is an estimate of the minimal distance between the *s-bot* and another object for which there is no risk of collision. This set of conditions is applied only if the set of conditions illustrated in figure 16 is not satisfied.

A.2 The performance measure "strength"

In the second set of experiments illustrated in section 5, the effectiveness of the navigational strategies of the *s-bots* is evaluated by employing a performance measure which we referred to as "strength".



rule	conditions		N_1	N_2
1	$d_r > D$	$\mid \alpha \mid \geq 20^\circ$	0	0
2	$d_r > D$	$\mid \alpha \mid < 20^\circ$	1	1
3	$0 \leq d_r \leq D$	$0^\circ \leq \alpha < 45^\circ$	0	1
4	$0 \leq d_r \leq D$	$-45^\circ < \alpha < 0^\circ$	1	0

Figure 17: The figure depicts a scenario in which an unconnected *s-bot*, represented by the small round object on the left, perceives on the left side of its camera-vision system the *prey*, represented by the big circle on the right.

Each *s-bot* s has an initial amount of strength $e_s = 1$, which must be higher than a certain threshold $\epsilon = 0.01$ for the *s-bot* to be able to move. The strength of each *s-bot* can increase or decrease depending on:

1. The temperature of the area in which the *s-bot* is currently located. The temperature is 1 if the *s-bot* is in a high temperature area, 0 if it is in a low temperature area.
2. The state of the *s-bot*'s loudspeaker. An *s-bot* emits a tone to signal its position to other *s-bots*. This signalling behaviour can facilitate the aggregation of the group, which is a prerequisite for the assembling.
3. Whether the *s-bot* is assembled or not.

More precisely, when *s-bot* s is assembled in a *swarm-bot* formation,
its strength decreases in the area of high temperature and increases in the area of low temperature, as described by the following equation:

$$e_s(t+1) = e_s(t) + \tau \cdot ((1 - \Gamma(t)) - e_s(t)), \tag{1}$$

where $e_s(t)$ is the strength of the s^{th} *s-bot* at cycle t, $\tau = 0.2$ is a time constant governing the speed of the strength variation and $\Gamma(t)$ is the temperature sensed by the s^{th} *s-bot* at cycle t in its current position. When an *s-bot* is not connected but it emits a sound signal, it loses strength in both areas. In the areas of low temperature its strength converges to a low but non-null value. This is described by the following equation:

$$e_s(t+1) = e_s(t) + \tau \cdot (k(1 - \Gamma(t)) - e_s(t)), \tag{2}$$

where $k = 0.1$ is a constant. In all the other situations, the *s-bot*'s strength increases in areas of high temperature and decreases in areas of low temperature:

$$e_s(t+1) = e_s(t) + \tau \cdot (\Gamma(t) - e_s(t)). \tag{3}$$

The time constant τ guarantees that the *s-bots*' strength varies smoothly according to the state of the system as described above. This smooth variation gives time to the control system of each *s-bot* to react to the new environmental situation in order to perform appropriate actions, before its strength drops under the threshold $\epsilon = 0.01$.

A.3 The evaluation function

In each trial θ, the behaviour of the *s-bots* is evaluated according to an evaluation function F_θ that takes into account the individual contribution of each *s-bot* s:

$$F_\theta = \frac{1}{n^3} \cdot \left(\sum_{s=1}^{n} d_s \cdot \sum_{s=1}^{n} e_s \cdot c \right), \tag{4}$$

where the factors d_s, e_s and c are explained below:

- d_s rewards *s-bots* that perform phototaxis; this fitness component is computed as follows:

$$d_s = \begin{cases} 0.1 \cdot \dfrac{x_{f,s} - x_{i,s}}{x_\Gamma - x_{i,s}} & \text{if } x_{f,s} \leq x_\Gamma, \\ 0.1 + 0.9 \cdot \dfrac{x_{f,s} - x_\Gamma}{x_M - x_\Gamma} & \text{if } x_\Gamma < x_{f,s} \leq x_M, \\ 1 & \text{otherwise,} \end{cases} \tag{5}$$

  where $x_{i,s}$ and $x_{f,s}$ are respectively the initial and final x coordinate of the s^{th} *s-bot* position, x_Γ is the x coordinate in which the temperature drops from 1 to 0, and x_M is the x coordinate of the light bulbs position.[5]

- e_s is the final strength possessed by the s^{th} *s-bot*, at cycle $t = 600$. The variation of the strength $e_s(t)$ of the s^{th} *s-bot* at cycle t is regulated by (1), (2) and (3) as discussed in appendix A.2.

  This fitness component rewards *s-bots* that end their lifetime with a high amount of strength. For example, if we compare groups of *s-bots* that managed to reach the end of the corridor close to the light bulbs, those which proved to be capable of assembling early in response to the decrease in the environmental temperature will get a higher fitness score than those which did not perform such collective response.

- c is the maximum size of a *swarm-bot* observed at the end of the trial, ranging from 1 (no connections among *s-bots*) to n (all *s-bots* connected in a single *swarm-bot*). This fitness component rewards *s-bots* that reach the end of the corridor assembled in a *swarm-bot* formation. Recall that, due to the characteristics of the environment—an initial area of high temperature is followed by an area of low temperature at the end of which the light bulbs are located—successful *s-bots* should terminate the trial in *swarm-bot* formation close to the opposite end of the corridor with respect to their starting position.

The average performance of the group F is computed averaging the evaluations F_θ performed in each trial θ. This value corresponds to the fitness of the genotype: it is used to select which genotypes will reproduce in the current generation, but is not in any sense a reinforcement directly available to the *s-bots*.

B 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 authors thank Alexandre Campo, Anders Lyhne Christensen, Christos Ampatzis, Halva Thomas Labella, Rehan O'Grady, and Shervin Nouyan for stimulating discussions and for their feedback during

[5] The coordinate system used has the x and y axes parallel respectively to the long and short wall of the corridor. The origin of the axes is positioned at the bottom left corner of the corridor.

the preparation of this paper. Marco Dorigo acknowledges support from the Belgian FNRS, of which he is a Research Director, and from the *ANTS* project, an Action de Recherche Concertée funded by the Scientific Research Directorate of the French Community of Belgium. 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.

References

[1] C. Anderson, G. Theraulaz, and J.-L. Deneubourg. Self-assemblages in insects societies. *Insectes Soc.*, 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, *Proc. of the Second European Workshop on Evolutionary Robotics*, pages 581–592. Springer Verlag, Berlin, Germany, 2003.

[3] G. Baldassarre, D. Parisi, and S. Nolfi. Coordination and behaviour integration in cooperating simulated robots. In S. Schaal, A. Ijspeert, A. Billard, S. Vijayakamur, J. Hallam, and J.-A. Meyer, editors, *From Animals to Animats 8. Proc. of the Eight Int. Conf. on Simulation of Adaptive Behavior (SAB04)*, pages 385–394. MIT Press, Cambridge, MA, 2004.

[4] R. D. Beer. A dynamical systems perspective on agent-environment interaction. *Artificial Intelligence*, 72:173–215, 1995.

[5] Curt Bererton and Pradeep Khosla. Towards a team of robots with repair capabilities: A visual docking systems. In *Int. Symposium on Experimental Robotics, (ISER)*, Lecture Notes in Control and Information Sciences, pages 333–342. Springer, Berlin, Germany, 2000.

[6] Curt Bererton and Pradeep Khosla. Towards a team of robots with reconfiguration and repair capabilities. In *IEEE Int. Conf. on Robotics and Automation*, volume 3, pages 2923–2928. IEEE Computer Society Press, Los Alamitos, CA, 2001.

[7] J. Bishop, S. Burden, E. Klavins, R. Kreisberg, W. Malone, N. Napp, and T. Nguyen. Programmable parts: A demonstration of the grammatical approach to self-organization. In *Proc. of the 2005 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 2644–2651. IEEE Computer Society Press, Los Alamitos, CA, 2005.

[8] H. Brown, M. Vande Weghe, C. Bererton, and P. Khosla. Millibot trains for enhanced mobility. *IEEE/ASME Trans. Mechatron.*, 7:452–461, 2002.

[9] A. Castano, W.-M. Shen, and P. Will. CONRO: Towards deployable robots with inter-robots metamorphic capabilities. *Auton. Robots*, 8(3):309–324, 2000.

[10] R. Damoto, A. Kawakami, and S. Hirose. Study of super-mechano colony: concept and basic experimental set-up. *Advanced Robotics*, 15(4):391–408, 2001.

[11] M. Dorigo, V. Trianni, E. Şahin, R. Groß, T. H. Labella, G. Baldassarre, S. Nolfi, J.-L. Deneubourg, F. Mondada, D. Floreano, and L. M. Gambardella. Evolving self-organizing behaviors for a swarm-bot. *Auton. Robots*, 17(2–3):223–245, 2004.

[12] T. Fukuda and S. Nakagawa. A dynamically reconfigurable robotic system (concept of a system and optimal configurations). In *Proc. of the 1987 IEEE Int. Conf. on Industrial Electronics, Control and Instrumentation*, pages 588–595. IEEE Computer Society Press, Los Alamitos, CA, 1987.

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

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

[15] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, Reading, MA, 1989.

[16] S. Griffith, D. Goldwater, and J. M. Jacobson. Self-replication from random parts. *Nature*, 437(7059):636, 2005.

[17] S. T. Griffith. *Growing Machines*. PhD thesis, MIT, MA, USA, September 2004.

[18] R. Groß, M. Bonani, F. Mondada, and M. Dorigo. Autonomous self-assembly in mobile robotics. *IEEE Trans. Robot.*, 2005. Submitted.

[19] R. Groß, M. Bonani, F. Mondada, and M. Dorigo. Autonomous self-assembly in a swarm-bot. In K. Murase, K. Sekiyama, N. Kubota, T. Naniwa, and J. Sitte, editors, *Proc. of the 3rd Int. Symp. on Autonomous Minirobots for Research and Edutainment (AMiRE 2005)*, pages 314–322. Springer, Berlin, Germany, 2006.

[20] R. Groß and M. Dorigo. Group transport of an object to a target that only some group members may sense. In Xin Yao, Edmund Burke, Jose A. Lozano, Jim Smith, Juan J. Merelo-Guervós, John A. Bullinaria, Jonathan Rowe, Peter Tiňo Ata Kabán, and Hans-Paul Schwefel, editors, *Parallel Problem Solving from Nature – 8th Int. Conf. (PPSN VIII)*, volume 3242 of *Lecture Notes in Computer Science*, pages 852–861. Springer Verlag, Berlin, Germany, 2004.

[21] I. Harvey, P. Husband, A. Thompson, and N. Jakobi. Evolutionary Robotics: the Sussex approach. *Robotics and Autonomous Systems*, 20:205–224, 1997.

[22] S. Hirose. Super mechano-system: New perspectives for versatile robotic systems. In Daniela Rus and Sanjiv Singh, editors, *Experimental Robotics VII*, volume 271 of *Lecture Notes in Control and Information Sciences*, pages 249–258. Springer, Berlin, Germany, 2001.

[23] S. Hirose, T. Shirasu, and E.F. Fukushima. Proposal for cooperative robot "Gunryu" composed of autonomous segments. *Robotics and Autonomous Systems*, 17:107–118, 1996.

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

[25] 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. *Auton. Robots*, 17(2–3):193–221, 2004.

[26] K. Motomura, A. Kawakami, and S. Hirose. Development of arm equipped single wheel rover: Effective arm-posture-based steering method. *Auton. Robots*, 18(2):215–229, 2005.

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

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

[29] R O'Grady, R Groß, and M. Dorigo. Self-assembly on demand in a group of physical autonomous mobile robots navigating rough terrain. In M.S. Capcarrere, A.A Freitas, P.J. Bentley, C.G. Johnson, and J. Timmis, editors, *Proc. of the Eighth European Conf. on Artificial Life (ECAL05)*, LNAI3630, pages 272–281. Springer-Verlag, Berlin, Heidelberg, Germany, 2005.

[30] M. Rubenstein, K. Payne, P. Will, and W.-M. Shen. Docking among indepenpent and autonomous CONRO self-reconfigurable robots. In *Proc. of 2004 IEEE Int. Conf. on Robotics and Automation (ICRA'04)*, volume 3, pages 2877–2882. IEEE Computer Society Press, Los Alamitos, CA, 2004.

[31] D. Rus and M. Vona. Crystalline robots: Self-reconfiguration with compressible unit modules. *Auton. Robots*, 10(1):107–124, 2001.

[32] V. Trianni, S. Nolfi, and M. Dorigo. Cooperative hole avoidance in a *swarm-bot*. *Robotics and Autonomous Systems*, 2005. In press.

[33] V. Trianni, E. Tuci, and M. Dorigo. Evolving functional self-assembling in a swarm of auton. robots. In S. Schaal, A. Ijspeert, A. Billard, S. Vijayakamur, J. Hallam, and J.-A. Meyer, editors, *From Animals to Animats 8. Proc. of the Eight Int. Conf. on Simulation of Adaptive Behavior (SAB04)*, pages 405–414. MIT Press, Cambridge, MA, 2004.

[34] P. J. White, K. Kopanski, and H. Lipson. Stochastic self-reconfigurable cellular robotics. In *Proc. of the 2004 IEEE Int. Conf. on Robotics and Automation (ICRA 2004)*, pages 2888–2893. IEEE Computer Society Press, Los Alamitos, CA, 2004.

[35] P. J. White, V. Zykov, J. Bongard, and H. Lipson. Three dimensional stochastic reconfiguration of modular robots. In *Proc. of Robotics Science and Systems*. MIT, Cambridge, MA, 2005.

[36] M. Yim, D. G. Duff, and K. D. Roufas. PolyBot: a modular reconfigurable robot. In *Proc. of the 2000 IEEE Int. Conf. on Robotics and Automation*, volume 1, pages 514–520. IEEE Computer Society Press, Los Alamitos, CA, 2000.

[37] M. Yim, K. Roufas, D. Duff, Y. Zhang, C. Eldershaw, and S. Homans. Modular reconfigurable robots in space applications. *Auton. Robots*, 14(2-3):225–237, 2003.

[38] M. Yim, Y. Zhang, and D. Duff. Modular robots. *IEEE Spectrum*, 39(2):30–34, 2002.

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

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