Self-Assembly in Physical Autonomous Robots: the Evolutionary Robotics Approach

Elio Tuci[1], Christos Ampatzis[1], Vito Trianni[2] Anders Lyhne Christensen[1], and Marco Dorigo[1]

[1] IRIDIA, CoDE, Université Libre de Bruxelles, Belgium
{etuci,campatzi,alyhne,mdorigo}@ulb.ac.be
[2] ISTC-CNR, Rome, Italy, vito.trianni@istc.cnr.it

Abstract. 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 priory* assumptions. Based on this premise, this research work illustrates an alternative approach to the design of controllers for self-assembling robots in which the self-assembly is initiated and regulated by perceptual cues that are brought forth by the physical robots through their dynamical interactions.

1 Introduction

Swarm robotics represents a novel way of doing collective robotics in which autonomous cooperating agents are controlled by distributed and local rules [1]. That is, each agent uses individual mechanisms and local perception to decide what action to take. Research in swarm robotics 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 [2]. In [3] the term self-assembly is meant to refer to "processes that involve pre-existing components (separate or distinct parts of a disordered structure), are reversible, and can be controlled by proper design of the components". Self-assembly can enhance the efficiency of a group of autonomous cooperating robots in several different contexts. In general, 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 (see [4] for a review of the field).

Owing to the development of an innovative robotic platform called *s-bot*, the research work carried out in the context of the SWARM-BOTS project—see http://www.swarm-bots.org/ for details on the project—proved that self-assembly can offer robotic systems additional capabilities useful for the accomplishment of the following tasks: a) *s-bots* collectively and cooperatively transporting heavy items [5]; b) *s-bots* climbing a steep slope [6]; c) *s-bots* navigating on very rough terrain in which a single agent might topple over [6]. A significant contribution of these works is in the design of distributed control mechanisms. In accordance with the swarm robotics principles, these research works show the

effectiveness of hand-coded sets of individual rules in producing complex group level behaviour such as self-assembly. In particular, the self-assembly process is based upon a signalling system which makes use of colours. For example, the decision concerning which robot makes the action of gripping, that is the *s-bot-gripper*, and which one is gripped, that is the *s-bot-grippee*, is made through the emission of red and blue light signals, according to which the *s-bots* emitting blue light are playing the role of *s-bot-gripper* and those emitting red light the role of *s-bot-grippee*. In [2], it is the heterogeneity among the robots with respect to the colour displayed, *a-priory* introduced by the experimenter, that triggers the self-assembly process. That is, a single *s-bot* "born" red among several *s-bots* "born" blue is meant to play the role of *s-bot-grippee* while the remaining *s-bot-grippers* are progressively switching their colour and role once assembled. In [6], a stochastic mechanisms is employed so that, at each time cycle, with a certain probability, each *s-bot* lights itself up in red given that no other red objects are perceived. In this way, it is the first *s-bot* emitting red light that triggers the self-assembly process. Once successfully assembled to another *s-bot*, each blue light emitting robot was programmed to turn off the blue LEDs and to turn on the red one. The switch from blue to red light signals to the non-assembled yet *s-bots* the "metamorphosis" of a robot from *s-bot-gripper* to *s-bot-grippee*.

As mentioned in [5], although the results of these research works are particularly encouraging, the authors should not underestimate the limitations of some of the aspects of the proposed approach. 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 (e.g., the object or robot to grasp should not move). 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 their 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., self-assembly) in response to *a-priory* specified agent's perceptual states. The work described in this paper represents a significant step forward in this direction.

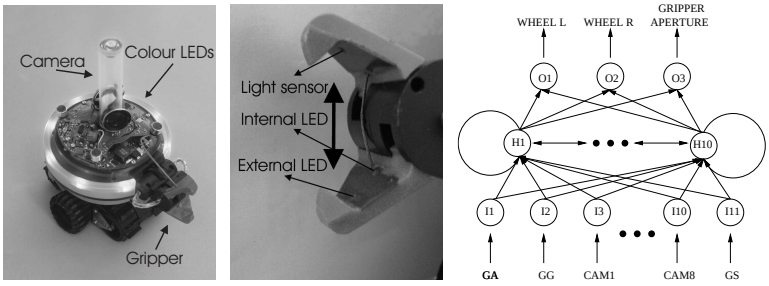
Our research work illustrates the details of an alternative methodological approach to the design of homogeneous controllers (i.e., where a controller is cloned in each robot of a group) for self-assembly in physical autonomous robots in which no assumptions are made concerning how agents allocate roles at the self-assembly task. By using dynamical neural networks, shaped by artificial evolution, we managed to design mechanisms by which the allocation of the *s-bot-gripper* and the *s-bot-grippee* roles is purely the result of an autonomous negotiation phase between the two *s-bots*. In other words, the self-assembly pro-

cess is triggered and regulated by perceptual cues that are brought forth by the agents through their dynamical interactions. We show that, the evolved mechanisms are as effective as the hand-coded one described in [2,6] when controlling a group of two real *s-bots*. In the following sections, we first describe the evolutionary machinery and the experimental scenario used to design neural network controllers. Then, we show the results of post-evaluation tests on physical robots (i.e., the *s-bots*) controlled by the best evolved controller. Conclusions are drawn in section 6.

2 Simulated and real *s-bot*

The controllers are evolved in a simulation environment which models some of the hardware characteristics of the real *s-bots* (see [7]). An *s-bot* is a mobile autonomous robot equipped with many sensors useful for the perception of the surrounding environment or for proprioception, a differential drive system, and a gripper by which it can grasp various objects or another *s-bot* (see figure 1a). The main body is a cylindrical turret with a diameter of about 12cm, which can be actively rotated with respect to the chassis.

In this work, to allow robots to perceive each other, we make use of the omni-directional camera. The image recorded by the camera is filtered in order to return the distance of the closest red, green, or blue blob in each of the eight 45° sectors. Each sector is referred to as CAM_i. Thus, an *s-bot* to be perceived by the camera must light itself up in one of the three colours using the LEDs mounted on its turret. Notice that the camera can clearly perceive coloured blobs up to a distance of approximately 50cm, but the precision above 33cm is rather low. Moreover, the precision with which the distance of coloured blobs is detected varies significantly with respect to the colour of the perceived object. We also make use of the optical barrier which is a hardware component composed of two LEDs and a light sensor mounted on the gripper (see figure 1b). By post-processing the readings of the optical barrier we extract valuable information concerning the status of the gripper and about the presence of an object between



(a) (b) (c)

Fig. 1. (a) The s-bot. (b) The gripper and sensors of the optical barrier. (c) Architecture of the neural network that control the *s-bots*.

the gripper claws. More specifically, the post-processing of the optical barrier readings defines the status of two virtual sensors: a) the *GS* sensor, set to 1 if the optical barrier indicates that there is an object in between the gripper claws, 0 otherwise; b) the *GG* sensor, set to 1 anytime a robot has gripped an object, 0 otherwise. We also make use of the *GA* sensor, which monitors the gripper aperture. The readings of the *GA* sensor range from 1 when the gripper is completely open to 0 when the gripper is completely closed.

The simulator used to evolve the required behaviour relies on a specialized 2D dynamics engine (see [8]). In order to evolve controllers that transfer to real hardware, we overcome the limitations of the simulator by following the approach proposed in [9]; motion is simulated with sufficient accuracy, collisions are not. Self-assembly relies on rather delicate physical interactions between robots that are integral to the task (e.g., the closing of the gripper around an object can be seen as a collision). Instead of trying to accurately simulate the collisions, we force the controllers to minimise them and not to rely on their outcome. In other words, in case of a collision, the two colliding bodies are repositioned to their previous positions, and the behaviour is penalised by the fitness function if the collision can not be considered the consequence of an accepted grasping manoeuvre. Having taken care of the collisions involved with gripping, the choice of a simple and fast simulator instead of one using a 3D physics engine significantly speed up the evolutionary process[3].

3 Controller and Evolutionary Algorithm

The agent controller is composed of a continuous time recurrent neural network (CTRNN) of ten hidden neurons and an arrangement of eleven input neurons and three output neurons (see figure 1c and [10] for a deeper illustration of CTRNNs). Input neurons have no state. At each simulation cycle, their activation values I_i—with $i \in [1, 11]$—correspond to the sensors' readings. In particular, I_1 corresponds to the reading of the *GA* sensor, I_2 to the reading of the *GG* sensor, I_3 to I_{10} correspond to the normalised reading of the eight camera sectors CAM_i, and I_{11} corresponds to the reading of the *GS* sensor. Hidden neurons are fully connected. Additionally, each hidden neuron receives one incoming synapse from each input neuron. Each output neuron receives one incoming synapse from each hidden neuron. There are no direct connections between input and output neurons. The state of each hidden neuron y_i—with $i \in [1, 10]$—and of each output neurons o_i—with $i \in [1, 3]$—are updated as follows:

$$\tau_i \frac{dy_i}{dt} = -y_i + \sum_{j=1}^{11} \omega_{ji} I_i + \sum_{k=1}^{10} \omega_{ki} Z(y_k + \beta_k); \quad o_i = \sum_{j=1}^{10} \omega_{ji} Z(y_j + \beta_j); \quad (1)$$

In these equations, τ are the decay constants, ω_{ij} the strength of the synaptic connection from neuron j to neuron i, β the bias terms, and $Z(x) = (1+e^{-x})^{-1}$ is

[3] Further methodological details and movies of the post-evaluation tests on real *s-bots* can be found at `http://iridia.ulb.ac.be/supp/IridiaSupp2008-002/`

a sigmoid function. τ, β, and ω_{ij} are genetically specified networks' parameters. $Z(o_1)$ and $Z(o_2)$ linearly scaled into [-3.2 3.2] are used to set the speed of the left and right motors. $Z(o_3)$ is used to set the gripper aperture in the following way: if $Z(o_3) > 0.75$ the gripper closes; if $Z(o_3) < 0.25$ the gripper opens. Cell potentials are set to 0 when the network is initialised or reset, and circuits are integrated using the forward Euler method with an integration step-size of 0.2.

Each genotype is a vector comprising 263 real values. Initially, a random population of vectors is generated by initialising each component of each genotype to values randomly chosen from an uniform distribution in the range [0,1]. The population contains 100 genotypes. Generations following the first one are produced by a combination of selection, and mutation. For each new generation, the five highest scoring individuals from the previous generation are chosen for breeding. The new generations are generated by making 20 copies of each highest scoring individual with mutations. Mutation entails that a random Gaussian offset is applied to each real-valued vector component encoded in the genotype, with a probability of 0.25. All the genetically specified network parameters are linearly scaled in the range $[-10, 10]$.

4 The fitness function

During evolution, each genotype is translated into a robot controller, and cloned onto each agent. Each group made of two simulated *s-bots*, is evaluated for 40 trials obtained by repeating for 4 times a set of 10 trials. At the beginning of each trial, the *s-bots* are positioned in a boundless arena at a distance randomly generated in the interval $[25cm, 30cm]$, and with predefined initial orientations[3]. Each trial (e) differs from the others in the initialisation of the random number generator, which influences the robots' starting position and orientation. Noise is added to motors and sensors. In particular, uniform noise is added in the range $\pm 1.25cm$ for the distance, and in the range $\pm 1.5°$ for the angle of the coloured blob perceived by the camera. 10% uniform noise is added to the motor outputs $Z(o_1)$. Uniform noise randomly in the range $\pm 5°$ is also added to the initial orientation of each *s-bot*. Within a trial, the robots life-span is 50 simulated seconds (250 simulation cycles). Each group is rewarded by an evaluation function $FF = \sum_{e=1}^{E} F_e$ with $E = 40$ which seeks to assess the ability of the two robots to get closer to each other and to physically assemble through the gripper. F_e is computed as follows:

$$F_e = A_e * C_e * S_e; \qquad A_e = \begin{cases} \frac{1.0}{1.0+atan(\frac{d_{rr}}{0.16})} & \text{if } d_{rr} > 16\text{cm}; \\ 1.0 & \text{otherwise}; \end{cases} \tag{2}$$

$$C_e = \begin{cases} 1.0 & \text{if } n_c = 0; \\ 0.0 & \text{if } n_c > 20; \\ \frac{1.0}{0.5+\sqrt{n_c}} & \text{otherwise}; \end{cases} \quad S_e = \begin{cases} 100.0 & \text{if } t = T, GG = 1; \\ 1.0 + \frac{29.0 \sum_{t=0}^{T} K(t)}{T} & \text{otherwise}; \end{cases} \tag{3}$$

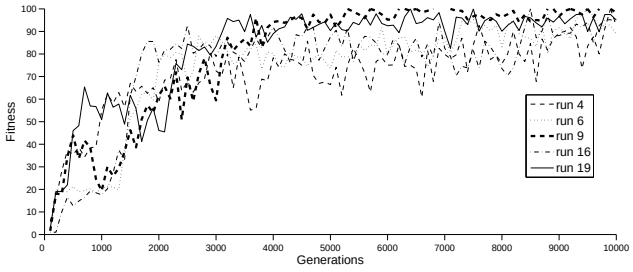


Fig. 2. Fitness of the best genotypes per generation of the best 5 evolutionary runs.

A_e is the aggregation component, C_e is the collision component, and S_e is the self-assembly component. d_{rr} is the distance between the two *s-bots* at the end of the trial e; n_c is the number of *s-bot-s-bot* collisions recorded during trial e; $T = 250$ corresponds to the maximum length of a trial in simulation cycles; $K(t)$ is set to 1 for each simulation cycle t in which the sensor GS of any *s-bot* is active, otherwise $K(t) = 0$. Notice that, given the way in which FF is computed, no assumptions are made concerning which *s-bot* plays the role of *s-bot-gripper* and which one the role of *s-bot-grippee*. The way in which collisions are modelled in simulations is an element that favours the evolution of assembly strategies in which the *s-bot-gripper* moves straight while approaching the *s-bot-grippee*. This has been done to ensure transferability to real hardware.

5 Results

As mentioned in section 1, the goal of this research work is to design, through evolutionary computation techniques, dynamical neural networks to allow a group of two homogeneous *s-bots* to physically connect to each other. To pursue our objective, we run, for 10000 generations, twenty randomly seeded evolutionary simulations. Given the way in which the fitness is computed (see section 4), the maximum fitness score a group of robots can obtain during evolution corresponds to 100. In figure 2, we plot the fitness score of the best genotypes at each generation of the five runs which managed to achieve the maximum score. We noticed that, with the exception of run n. 9, the fitness of the best agents oscillates quite a lot throughout the evolution. Through a first series of post-evaluations carried out in simulation, we re-evaluated all the best evolved genotypes from generation 5000 to generation 10000 of the best five evolutionary runs shown in figure 2. Based on the results of these tests, it turned out that the genotype corresponding to the homogeneous group with the highest success rate at the self-assembly task is one taken from evolutionary run n. 9. The best performing genotype has been decoded into an artificial neural network which is cloned and ported on both real *s-bots*. Then, the *s-bots* are evaluated in four different sets of 36 trials each. The number of trials results from a systematic variation of the robots' initial orientations. This is done in order to rule out any effect associated

with the relative orientation of the robots at the beginning of the trial on their capability to perform self-assembly. Note that, for real *s-bots*, the trial's termination criteria was changed with respect to those employed with the simulated *s-bots*. In particular, there is no time-limit on the maximum duration of a trial, and no limit on the maximum number of collisions allowed. In each trial, we let the *s-bots* interact either until physically connected, or until the occurrence of events—illustrated below—that hinder the robots to achieve their objective. The new termination criteria allowed us to observe interesting and unexpected behavioural sequences[3]. The first two post-evaluation sets are referred to as test G25 and test G30. These are tests in which the *s-bots* light themselves up in green and are initialised at a distance from each other of 25cm and 30cm, respectively. The *s-bots* proved to be 100% successful in both tests. That is, they terminated all the trials physically assembled. Table 1 gives more details about the *s-bots*' performances in these trials. In particular, we notice that the number of successful trials at the first gripping attempt is 28 trials and 29 trials out of 36 for G25 and G30 respectively (see Table 1, 2^{nd} column). In few trials, the *s-bots* managed to assemble after two/three grasping attempts (see Table 1, 3^{rd} and 7^{th} column). The failed attempts were mostly caused by inaccurate manoeuvres of type I_1, in which a series of maladroit actions by both robots makes impossible for *s-bot-gripper* to successfully grasp the *s-bot-grippee*'s cylindrical turret. In few other cases, the group committed inaccuracy I_2. That is, both robots assume the role of *s-bot-gripper*. In such circumstances, the *s-bots* head towards each

Table 1. Results of post-evaluation tests on real *s-bots*. G25 and G30 refer to the tests in which the *s-bots* light themselves up in green and are initialised at a distance from each other of 25cm and 30cm, respectively. B30 and R30 refer to the tests in which the *s-bots* light themselves up in blue and red respectively, and are initialised at a distance of 30cm from each other. Trials in which the physical connection between the *s-bots* requires more than one gripping attempts, due to inaccurate manoeuvres I_i, are still considered successful. In inaccuracy I_1, a series of maladroit actions by both robots makes impossible for *s-bot-gripper* to successfully grasp the *s-bot-grippee*'s cylindrical turret. In I_2 both robots assume the role of *s-bot-gripper*, which results in a collision between the robots' grippers. In I_3, after grasping the connected structure get slightly elevated at the connection point. Failures correspond to trials in which the robots do not manage to return to a distance from each other smaller than their visual field.

Test	Num. suc. trials per gripping attempt and types of inaccuracy									N. failure
	1^{st}	2^{nd}				3^{rd}				
	$N.^\circ$	$N.^\circ$	I_1	I_2	I_3	$N.^\circ$	I_1	I_2	I_3	
G25	28	7	6	1	0	1	2	0	0	0
G30	29	6	3	3	0	1	1	1	0	0
B30	26	5	3	2	0	4	8	0	0	1
R30	21	12	8	0	2	4	7	0	1	0

other until a collision between their respective grippers occurs. Note that, both in G25 and in G30, the *s-bots* always managed to recover from the inaccuracies and to successfully terminate the trials.

In a further series of tests on real *s-bots* we deliberately changed the colour of the LEDs mounted on the *s-bots* turret. Although the *s-bots* have to lighten their turret to perceive each other through their camera, the specific colour displayed has not functional role within the neural machinery that brings forth the *s-bots* actions. The aim of these tests is to show that the mechanisms that underpin the *s-bots* self-assembling strategies do not depend on the specific colour displayed by the LEDs. In other words, we test the robustness of the evolved mechanisms with respect to changes in the perceived colour. To prove what said above, we repeated the 36 trials with the *s-bots* initially placed at a distance of 30cm from each other, a first time with the LEDs displaying blue light—this test is referred to as B30—and a second time with the LEDs displaying red light—this test is referred to as R30. The *s-bots* proved to be very successful both in B30 and R30 (see Table 1, 2^{nd} column). In the large majority of the trials the *s-bots* managed to self-assemble at the first grasping attempt. In few trials, two or three grasping manoeuvres were required (see Table 1, 3^{rd} and 7^{th} column). A new type of inaccuracies emerged in test R30. That is, in three trials, after grasping, the connected structure got slightly elevated at the connection point. We referred to this type of inaccuracies as I_3. Notice also that in a single trial, in test B30, the *s-bots* failed to self-assemble (see Table 1, 11^{th} column). The reason of failure is what we refer to as the Lost-phase; that is, the *s-bots* moved so far away from each other that they ended up significantly outside the perceptual range of their respective camera. The trials in which the *s-bots* spend more than 1 minute in Lost-phase has been terminated once this time interval was expired, and the trial has been considered unsuccessful.

For each single test (i.e., G25, G30, B30, and R30), the sequences of *s-bots*'s actions are rather different from one trial to the other. However, these different histories of interactions can be succinctly described by a combination of few distinctive phases and transactions between phases which exhaustively "portray" the observed phenomena. During the starting phase the robots tend to get closer to each other by leaving their respective starting positions. In the great majority of the trials, the robots move from the starting phase to what we call the role-allocation phase (RA-phase). In this phase, each *s-bot* tends to remain on the right side of the other. They slowly move by following a circular trajectory corresponding to an imaginary circle centred in between the *s-bots*. Moreover, each robot rhythmically changes its heading by turning left and right. The RA-phase ends once one of the two *s-bots*—that is, the one that is assuming the role of the *s-bot-gripper*—stops oscillating and heads towards the other *s-bot*—that is, the one that is assuming the role of the *s-bot-grippee*—which instead keeps on moving in a similar way as during the RA-phase. The *s-bot-gripper* approaches the *s-bot-grippee*'s turret and, as soon as its *GS* sensor is active, it closes its gripper. A successful trial terminates as soon as the two *s-bots* are connected.

As mentioned above, in few trials the *s-bots* failed to connect at the first gripping attempts by committing what we called inaccuracies I_1 and I_3. These inaccuracies seems to denote problems in the sensory-motor coordination during grasping. Recovering from I_1 can only be accomplished by returning to a new RA-phase, in which the *s-bots* negotiate again their respective roles, and eventually self-assemble. Recovering from I_3 is accomplished by a slight backward movement of both *s-bots* which restores a stable gripping configuration. Given that I_3 has been observed only in R30, it seems plausible to attribute the origin of this inaccuracy to the effects of the red light on the perceptual apparatus of the *s-bots*. In particular, it could be that, due to the red light, the *s-bot-gripper* perceives through its camera the *s-bot-grippee* at a distance longer than the actual one. Alternatively, it could be that the red light perturbs the regular functioning of the optical barrier and consequently the readings of the *GS* and *GG* sensors. Both phenomena may induce the *s-bot-gripper* to keep on moving towards the *s-bot-grippee* up to the occurrence of I_3, even though the distance between the robots and the status of the gripper of the *s-bot-gripper* would require a different response. I_2 seems to be caused by the effects of the *s-bots*' starting positions on their behaviour. Recall that, in those trials in which I_2 occurs, after a short starting phase, the *s-bots* head towards each other until they collide with their grippers without going through the RA-phase. It looks like that it is the way in which the robots perceive each other at starting positions that induces them to skip the RA-phase. Without a proper RA-phase, the robots fail to autonomously allocate between themselves the roles required by the self-assembly task (i.e., *s-bot-gripper* and *s-bot-grippee*), and consequently they incur in I_2. In order to recover from I_2, the *s-bots* move away from each other and start a new RA-phase in which roles are eventually allocated. In the future we will further investigate the exact cause of the inaccuracies.

As shown in Table 1, except for a single trial in test B30 in which the *s-bots* failed to self-assemble, the robots proved capable of recovering from all types of inaccuracies. This is an interesting result because it is evidence of the robustness of our controllers with respect to contingencies never encountered during evolution. Indeed, as mentioned in section 2, in order to speed up the evolutionary process, the simulation in which controllers have been designed is not taking into account the dynamics of collisions. In those cases in which, after a collision, the simulated robots had another chance to assemble, the agents were simply re-positioned at a given distance to each other. In spite of this, *s-bots* guided by the best evolved controllers proved capable of engaging themselves in successful recovering manoeuvres which allowed them to eventually assemble.

6 Conclusions

The results of this work are a proof-of-concept: they proved that dynamical neural networks shaped by evolutionary computation techniques can provide physical robots all the required mechanisms to autonomously perform self-assembly. Owing to this design approach—known in the literature as evolutionary robotics,

see [11]—the self-assembly is initiated and regulated by perceptual cues that are brought forth by the two homogeneous robots through their dynamical interactions. Moreover, in spite of the system being homogeneous, role allocation—i.e., who is the *s-bot-gripper* and who is the *s-bot-grippee*—is successfully accomplished by the robots through an autonomous negotiation phase. Contrary to the hand-coded controllers described in [2,6], the evolutionary robotics approach did not require the experimenter to make any *a-priory* assumptions concerning the roles of the robots during self-assembly (i.e., either *s-bot-gripper*or *s-bot-grippee*) or about their status (i.e., either capable of moving or required not to move). Moreover, the evolved mechanisms proved to be robust with respect to changes of the colour of the light displayed by the LEDs. To cope with the same type of changes the hand-coded controllers described in [2,6] would require to be re-designed. Future work will focus on the analysis of the evolved mechanisms as well as on the study of more complex scenarios in which self-assembly is functional to the achievement of particular objectives that are beyond the capabilities of a single *s-bot*.

References

1. Dorigo, M., Şahin, E.: Swarm robotics – special issue editorial. Autonomous Robots **17**(2–3) (2004) 111–113
2. Groß, R., Bonani, M., Mondada, F., Dorigo, M.: Autonomous self-assembly in swarm-bots. IEEE Transactions on Robotics **22**(6) (2006) 1115–1130
3. Whitesides, G.M., Grzybowski, B.: Self-assembly at all scales. Science **295** (2002) 2418–2421
4. Groß, R.: Self-assembling robots. PhD thesis, Univeristé Libre de Bruxelles, Faculté des sciences appliquées (2007) Available at `http://theses.ulb.ac.be/ETD-db/collection/available/ULBetd-09302007-222011/`.
5. Tuci, E., Groß, R., Trianni, V., Bonani, M., Mondada, F., Dorigo, M.: Cooperation through self-assembling in multi-robot systems. ACM Transactions on Autonomous and Adaptive Systems **1**(2) (2006) 115–150
6. O'Grady, R., Groß, R., Mondada, F., Bonani, M., Dorigo, M.: Self-assembly on demand in a group of physical autonomous mobile robots navigating rough terrain. In et al., M.C., ed.: Proc. of the 8^{th} European Conference on Artifcial Life (ECAL05). Number 3630 in LNCS, Berlin, GE:Springer-Verlag (2005) 272–281
7. Mondada, F., Pettinaro, G., Guignard, A., Kwee, I., Floreano, D., Deneubourg, J.L., Nolfi, S., Gambardella, L., Dorigo, M.: Swarm-bot: A new distributed robotic concept. Autonomous Robots **17**(2–3) (2004) 193–221
8. Christensen, A.: Efficient neuro-evolution of hole-avoidance and phototaxis for a swarm-bot. DEA thesis TR/IRIDIA/2005-14, Université Libre de Bruxelles, Bruxelles, Belgium (September 2005)
9. Jakobi, N.: Evolutionary robotics and the radical envelope of noise hypothesis. Adaptive Behavior **6** (1997) 325–368
10. Beer, R.D., Gallagher, J.C.: Evolving dynamical neural networks for adaptive behavior. Adaptive Behavior **1** (1992) 91–122
11. Nolfi, S., Floreano, D.: Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines. MIT Press, Cambridge, MA (2000)