

Evolving Self-Assembly in Autonomous Homogeneous Robots: Experiments with Two Physical Robots

Christos Ampatzis^{*,**}
European Space Agency

Elio Tuci[†]
ISTC-CNR

Vito Trianni[†]
ISTC-CNR

Anders Lyhne Christensen[‡]
DCTI-ISCTE

Marco Dorigo[§]
CoDE-IRIDIA

Abstract This research work illustrates an 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 dynamic interactions. More specifically, we present a homogeneous control system that can achieve assembly between two modules (two fully autonomous robots) of a mobile self-reconfigurable system without a priori introduced behavioral or morphological heterogeneities. The controllers are dynamic neural networks evolved in simulation that directly control all the actuators of the two robots. The neurocontrollers cause the dynamic specialization of the robots by allocating roles between them based solely on their interaction. We show that the best evolved controller proves to be successful when tested on a real hardware platform, the *swarm-bot*. The performance achieved is similar to the one achieved by existing modular or behavior-based approaches, also due to the effect of an emergent recovery mechanism that was neither explicitly rewarded by the fitness function, nor observed during the evolutionary simulation. Our results suggest that direct access to the orientations or intentions of the other agents is not a necessary condition for robot coordination: Our robots coordinate without direct or explicit communication, contrary to what is assumed by most research work in collective robotics. This work also contributes to strengthening the evidence that evolutionary robotics is a design methodology that can tackle real-world tasks demanding fine sensory-motor coordination.

Keywords

Self-assembly, role allocation, neural network, artificial evolution, evolutionary robotics

1 Introduction

Self-assembly is a process that is ubiquitous in nature. According to Whitesides and Grzybowski [41], self-assembly is defined as “the autonomous organisation of components into patterns or structures

* Contact author.

** European Space Agency, Advanced Concepts Team, ESTEC, Keplerlaan 1, Postbus 2fff99, 2200 AG, Noordwijk, The Netherlands. E-mail: christos.ampatzis@esa.int

† ISTC-CNR, via San Martino della Battaglia 44, 00185 Roma, Italy. E-mail: elio.tuci@istc.cnr.it

‡ DCTI-ISCTE, Av. das Forças Armadas, 1649-026 Lisbon, Portugal. E-mail: anders.christensen@iscte.pt

§ CoDE-IRIDIA, Université Libre de Bruxelles (ULB), Av. F. Roosevelt 50, CP 194/6, 1050 Brussels, Belgium. E-mail: mdorigo@ulb.ac.be

without human intervention.” At the nano or microscopic scale, the interaction among components is essentially stochastic and depends on their shape, structure, or chemical nature. Nature also provides many examples of self-assembly at the macroscopic scale, the most striking being animals forming collective structures by connecting to one another. Individuals of various ant, bee, and wasp species self-assemble and manage to build complex structures such as bivouacs and ladders. Self-assembly in social insects typically takes place in order to accomplish some function (defense, object transport, passage formation, etc.; see [1]). In particular, ants of the species *Oecophylla longinoda* can form chains composed of their own bodies, which are used to pool leaves together to form a nest, or to bridge a passage between branches in a tree [22].

The robotics community has been largely inspired by cooperative behavior in animal societies when designing controllers for groups of robots that have to accomplish a given task. In particular, self-assembly provides a novel method of cooperation in groups of robots. Self-assembling robotic systems at the macroscopic scale display interesting properties, deriving mainly from the sensors and actuators that can be incorporated in the robots, providing increased component autonomy and more advanced and complex ways of interaction with the environment and other components (see [18]). Recently, the research work carried out in the context of the SWARM-BOTS project¹ proved that it is possible to build and control a group of autonomous self-assembling robots by using swarm robotics principles. Swarm robotics is a novel approach to collective robotics in which autonomous cooperating agents are controlled by distributed and local rules [7]. Research in swarm robotics focuses on mechanisms to enhance the efficiency of the group through some form of cooperation among the individual agents. In this respect, self-assembly can enhance the efficiency of a group of autonomous cooperating robots by overcoming the physical limitations of each individual robot. Within the SWARM-BOTS project, it has been shown that self-assembly can offer robotic systems additional capabilities useful for the accomplishment of the following tasks: (a) robots collectively and cooperatively transporting items too heavy to be moved by a single robot [17]; (b) robots climbing a hill whose slope would cause a single robot to topple over [32]; (c) robots navigating on rough terrain in which a single agent might topple over [32]. The application of such systems can potentially go beyond research in laboratories, space applications being one of the most obvious challenges (e.g., multi-robot planetary exploration and on-orbit self-assembly; see [24]).

This article illustrates an 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 dynamic interactions. More specifically, we focus on the problem of forming a physical structure with two robots that have to allocate distinct roles between them; these roles are the *gripper* (robot that grips) and the *grippee* (robot that receives the grip). In order to design the robots’ control system, we use *evolutionary robotics* (ER), a methodological tool to automate the design of robots’ controllers [31]. ER is based on the use of artificial evolution to find sets of parameters for artificial neural networks that guide the robots to the accomplishment of their task. In contrast to other design methods, ER does not require the designer to make strong assumptions concerning what behavioral and communication mechanisms are needed by the robots. The experimenter specifies the characteristics of a social context in which robots are required to cooperate. Then, the mechanisms for solitary and social behavior are determined by an evolutionary process that favors (through selection) those solutions that improve an agent’s or group’s ability to accomplish its task (i.e., the fitness measure).

In this study, we apply an artificial evolutionary process to synthesize dynamic neural network controllers (continuous-time recurrent neural networks—CTRNNs; see [6]) capable of autonomous decision making and self-assembling in a homogeneous group of robots. In particular, we train via artificial evolution a dynamic neural network that, when downloaded to real robots, allows them to

¹ The SWARM-BOTS project was 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>.

coordinate their actions in order to decide who will grip whom. Dynamic neural networks have been used in the past as a means to achieve specialization in a robot group (see [34, 38] for examples). Similarly, we study self-assembly in a setup where the robots interact and eventually differentiate by allocating distinct roles.

A first important contribution of this work is to show that an integrated (i.e., nonmodularized) dynamic neural network in direct control of all the actuators of the robots can successfully tackle real-world tasks requiring fine-grained sensory-motor coordination, such as self-assembly. Our work should be considered a proof of concept that contributes to strengthening the evidence that ER can safely be employed to obtain controllers for real robots engaged in real-world tasks requiring fine sensory-motor coordination.

It is important to note that some characteristics of the hardware may impose important constraints on the control of the modules of a self-assembling system. Some hardware platforms consist of morphologically heterogeneous modules that can only play a predefined role in the assembly process. In others, the hardware design does not allow, for example, the assembly of more than two modules, or requires extremely precise alignment during the connection phase, that is, it requires great accuracy. As argued by Tuci et al. [39], the swarm-bot platform, thanks to its sensors and actuators and its connection mechanism, does not severely constrain the design of control mechanisms for self-assembly. This platform consists of identical modules, each equipped with a gripper and a large area to receive connections from other modules. The lack of hardware constraints and the homogeneity of the robots requires that self-assembly must be achieved through a differentiation of roles, resulting in the definition of a gripper and a gripee. In previous work, such differentiation was either predefined [16], or based on stochastic events and a complex communication protocol [32]. The main contribution of this work lies in the design of control strategies for real assembling robots that are not constrained by either morphological or behavioral heterogeneities introduced by the hardware and the control method, respectively. To the best of our knowledge, there is no system in the robotic literature that can achieve self-assembly without a priori injected morphological or behavioral heterogeneities. Instead of a priori defining the mechanisms leading to role allocation and self-assembly, we let behavioral heterogeneity emerge from the interaction among the system's homogeneous components. Moreover, we show with physical robots that coordination and cooperation in self-assembly do not require explicit signaling of internal states, as assumed, for example, in [16]. In other words, we present a setup that requires minimal cognitive and communicative capacities on the part of the robots. We believe that by following such an approach, we can obtain more adaptive robotic systems with a higher level of autonomy, because the adaptiveness of an autonomous multi-robot system is reduced if the circumstances an agent should take into account to make a decision (concerning solitary and/or social behavior) are defined by a set of a priori assumptions made by the experimenter.

Our approach does not dictate the interaction rules and principles to be used to achieve the allocation of roles. The automatic process autonomously synthesizes the rules the robots have to employ, without the involvement of the experimenter. However, we should make clear that our experimentation and analysis do not take scalability into account. That is, the rules governing our system cannot be extrapolated to the control of groups of robots of higher cardinality. Expanding our design methodology for assembly in larger groups is a necessary step to be taken in future work; in this article, we present the rationale and motivations and we demonstrate the effectiveness of our approach for a pair of physical robots.

This article is organized as follows: In Section 2, we provide a brief overview of the state of the art in the area of self-assembling robots, and we discuss the limitations of these systems, justifying the methodological choices we have made. In the subsequent sections (Sections 3, 4, and 5), we describe the evolutionary machinery and the experimental scenario used to design neural network controllers. Then, in Section 6, we show the results of post-evaluation tests on physical robots controlled by the best-performing evolved controller, and we try to shed some light on the mechanisms underpinning the behavior of successful robots. The results presented are discussed in Section 7, and conclusions are drawn in Section 8.

2 Related Work

Following the distinction introduced in [18], self-assembling systems can be either self-propelled or externally propelled. The latter category includes components that need to be externally agitated in order to form structures (see [27], for example). On the contrary, in self-propelled systems, each module can be a component that approaches and assembles with other components. Mobile robots are a typical instantiation of such components, which can range from centimeter size to the nano scale. Although there is a considerable amount of literature treating self-assembly and pattern formation at the nano scale (see [2, 35], for example), here we focus our attention on self-propelled self-assembling systems at the macroscopic scale. Such systems are characterized by a high degree of autonomy linked to the more advanced and complex ways at their disposal to interact with their environment.

Several examples of robotic platforms in the literature consist of connecting modules. For a comprehensive review of self-propelled self-assembling robotic systems at the macroscopic scale, we direct the reader to the work of Yim et al. [43], Groß and Dorigo [18], Groß et al. [16], and Tuci et al. [39]. Following Yim et al. [43], it is possible to identify four different categories: chain-based, lattice-based, mobile, and stochastic reconfigurable robots. As this work focuses on mobile self-reconfigurable robots, in the following we provide a short overview of this category only. We then discuss the platform that is used in this study: the swarm-bot.

2.1 Mobile Self-Reconfigurable Robots

The first example of a mobile self-reconfigurable robot was CEBOT [13, 15]. CEBOT is a heterogeneous system composed of cells with different functions (move, bend, rotate, slide). Even though there are no quantitative results to assess the performance and reliability of this system, Fukuda et al. [14] have shown how docking can be done between a moving cell and a static object cell. Another robotic system capable of self-assembly is the *super-mechano-colony* [11, 20]. In this system, autonomous components, referred to as child units, can connect to and disconnect from a mother ship. Yamakita et al. [42] achieved docking by letting the child unit follow a predefined path. Groß et al. [19] recently demonstrated assembly between one and three moving child modules and a static module. Hirose et al. [21] presented a distributed robot called Gunryu. Each robot is capable of fully autonomous locomotion, and the assembled structure proved capable of navigating on rough terrain where a single unit would topple over. However, autonomous self-assembly was not studied, as the units were connected beforehand by means of a passive arm. Self-assembly is also not possible for the Millibot train [8], a system composed of multiple modules that are linearly linked. This is because no external sensor has been implemented. In all the above mobile self-reconfigurable systems, self-assembly either is not achieved at all or is only possible between one unit moving autonomously and a static object or unit.

We should also mention two important examples from the modular chain robot literature, CONRO and PolyBot. CONRO [9] has been used by Rubenstein et al. [36] to demonstrate autonomous docking between two robots. It should be noted, however, that the control was heterogeneous at all levels and the generality of the approach was limited by orientation and distance constraints. Yim et al. [44] demonstrated self-assembly with PolyBot: a six-module arm connected to a spare module on a flat terrain. One end of the arm and the spare module were fixed to the walls of the arena at known positions, and the motion of the arm relied on knowledge of the goal position and inverse kinematics.

2.2 Self-Assembly with the Swarm-Bot

The swarm-bot, a collective and mobile reconfigurable system (see [12, 28] and <http://www.swarm-bots.org>), consists of fully autonomous mobile robots called *s-bots*, which can physically connect to each other and to static objects (preys, also called *s-toys*). It is the only robotic platform for which self-assembly of more than two self-propelled robots has been demonstrated; most physical systems are still at the two-module self-assembly level to date (e.g., PolyBot, CONRO, and CEBOT,

mentioned above). Groß et al. [16] presented experiments with the s-bots that improved the state of the art in self-assembling robots, mainly with respect to the number of robots involved in self-assembly, the generality and reliability of the controllers, and the assembly speed. A significant contribution of this work is in the design of distributed control mechanisms for self-assembly relying only on local perception. In particular, self-assembly was accomplished with a modular approach in which some modules have been evolved and others handcrafted. The approach was based upon a signaling system that makes use of colors. For example, the decision concerning which robot makes the action of gripping (the *s-bot gripper*) and which one is gripped (the *s-bot gripee*) is made through the emission of color 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 gripee. Thus, it is the heterogeneity among the robots with respect to the color displayed, introduced a priori 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 gripee while the remaining s-bot grippers are progressively assembling. 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 ones. The switch from blue to red light indicates to the yet unassembled s-bots the metamorphosis of a behavioral or morphological heterogeneity. In other words, it requires either the presence of a prey lit up in red or the presence of a robot not sharing the controller of the others, which is forced to be immobile and to signal with a red color. O'Grady et al. [32] bypassed this requirement by handcrafting a decision-making mechanism based on a probabilistic transition between states. More specifically, the allocation of roles (which robot lights up red and triggers the process) depends solely on a stochastic process.

2.3 Motivations

The research works presented above have been very successful in that they also showed how assembled structures can overcome physical limitations of the single robots, for instance in transporting a heavy object or in navigating on rough terrain. However, this modularized architecture is based on a set of a priori assumptions concerning the specification of the environmental and behavioral conditions that trigger the self-assembling process. For example, (a) the objects that can be grasped must be red, and those that should not be grasped must be blue; (b) the action of grasping is carried out only if all the grasping requirements are fulfilled (among others, a combination of conditions concerning the distance and relative orientation between the robots; see [16] for details). If the experimenter could always know in advance in what type of world the agents would 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 that 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 (environmental variability, partial hardware failure, etc.). For example, failure to emit or perceive red light for robots guided by the controllers presented above would significantly hinder the accomplishment of the assembly task.

We believe that a sensible step in this direction can be made by not constraining the system to initiate its most salient behaviors (e.g., self-assembly) in response to the a priori specified agent's perceptual states. The work described in this article 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. By using dynamic neural networks shaped by artificial evolution, we managed to design mechanisms by which the allocation of the s-bot gripper and the s-bot gripee roles is the result of the dynamic interaction between the s-bots. Furthermore, coordination and role allocation in our system are achieved solely through minimal sensors (distance and angle information) and without explicit communication, contrary

to the works described above where the agents signal their internal states to the rest of the group. Also, due to the nature of the sensory system used, the robots cannot sense the orientation of their group-mates. In this sense, our approach is similar to (and inspired by) the one of Quinn [33] and Quinn et al. [34], where role allocation (leader-follower) or formation movement is achieved solely through infrared sensors. In addition, we show that the evolved mechanisms are as effective as the modular and hand-coded ones described in [16, 32] when controlling two real s-bots.

3 Simulated and Real S-Bot

The controllers are evolved in a simulation environment that models some of the hardware characteristics of the real s-bots [29]. An s-bot is a mobile autonomous robot equipped with many sensors useful for the perception of the environment and 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 11.6 cm, which can be actively rotated with respect to the chassis. The turret is equipped with a surrounding ring that receives connections from other s-bots through their grippers.

In this work, to allow robots to perceive each other, we make use of the omnidirectional camera mounted on the turret. 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 eight 45° sectors. A sector is referred to as CAM_i , where $i = 1, \dots, 8$ denotes the index of the sector. Thus, an s-bot to be perceived by the camera must light itself up in one of the three colors, using the LEDs mounted on the perimeter of its turret. An s-bot can be perceived in at most two adjacent sectors. Notice that the camera can clearly perceive colored blobs up to a distance of approximately 50 cm, but the precision beyond approximately 30 cm is rather low. Moreover, the precision with which the distance of colored blobs is detected varies with the color 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 postprocessing the readings of the optical barrier, we extract information about the status of the gripper and about the presence of an object between the gripper claws. More specifically, the postprocessing 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, and 0 otherwise; (b) the GG sensor, set to 1 if a robot is currently grasping an object, and 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 s-bot actuators are the two wheels and the gripper.

The simulator used to evolve the required behavior relies on a specialized 2D dynamics engine [10]. In order to evolve controllers that transfer to real hardware, we overcome the limitations of the

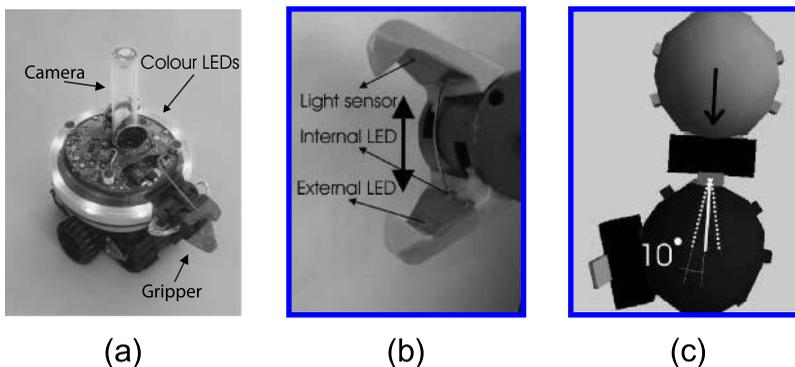


Figure 1. (a) The s-bot. (b) The gripper and sensors of the optical barrier. (c) Depiction of the collision manager. The arrow indicates the direction along which the s-bot-gripper should approach the s-bot-gripped without incurring collision penalties.

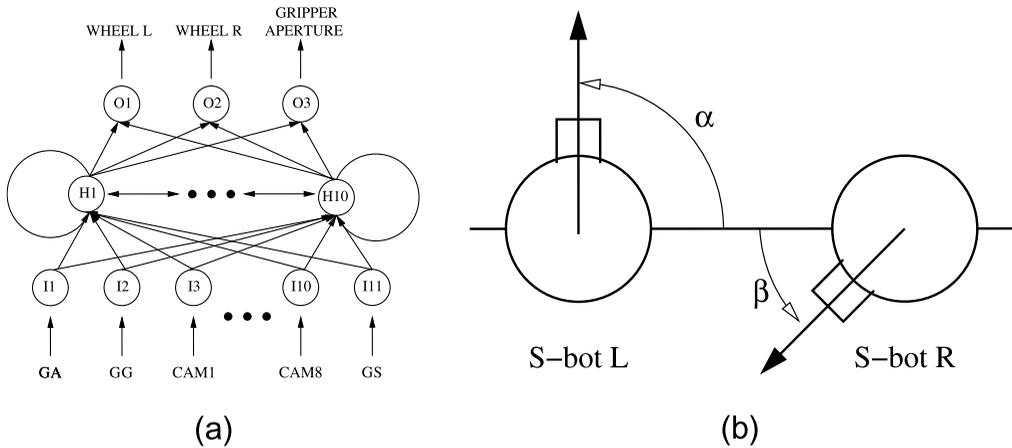


Figure 2. (a) Architecture of the neural network that controls the s-bots. (b) How the s-bots' starting orientations are defined given the orientation pair (α, β) . S-bot L and s-bot R are the robots whose initial orientations in any given trial correspond to the values of α and β , respectively.

simulator by following the minimal simulation approach proposed in [25]. In our setup, 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 could be interpreted as a collision). Instead of trying to accurately simulate the collisions, we force the controllers to minimize them and not to rely on their outcome. In case of a collision, the two colliding bodies are repositioned to their previous positions, and the behavior is penalized by the fitness function if the collision cannot be considered the consequence of an accepted grasping maneuver.

Concerning the simulation of the gripper, we modeled the two gripper claws as triangles extending from the body of the robot. As the gripper opens, these triangles are pulled into the robot body, whereas as it closes they grow out of it. Thus, the size of the collision object changes with the aperture of the gripper. In order for a grip to be called successful, we require that there be an object between the claws of the (open) gripper, as close as possible to the interior of the gripper, and that the claws close around it. In fact, we require that the object and the gripper socket holding the two claws collide. However, we do not penalize such a collision when the impact angle between the s-bots falls within the range $[-10^\circ, +10^\circ]$. Figure 1c shows how this impact angle is calculated and also depicts the simulated robots we use. In this way, we facilitate the evolution of approaching movements directed toward the turret of the robot to be gripped (see Figure 1c). Robots that rely on such a strategy when attempting to self-assemble in simulation can also be successful in reality. Other types of strategies based on rotating movements proved prone to failure when tested on real hardware. 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 speeds up the evolutionary process.

4 The Controller and the Evolutionary Algorithm

The agent controller is composed of a continuous-time recurrent neural network (CTRNN) of 10 hidden neurons and an arrangement of 11 input neurons and three output neurons (see Figure 2a and [6] for a more detailed 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 normalized readings of the eight camera sectors *CAM_s*, 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 every input neuron. Each output neuron receives one incoming synapse from every 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 neuron o_i (with $i \in [1, 3]$) is updated as follows:

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

In these equations, τ_i are the decay constants, ω_{ji} is the strength of the synaptic connection from neuron i to neuron j , β_k are the bias terms, and $Z(x) = (1 + e^{-x})^{-1}$ is a sigmoid function. τ_i , β_k , and ω_{ji} are genetically specified network parameters. $Z(o_1)$ and $Z(o_2)$, linearly scaled into $[-3.2 \text{ cm/s}, 3.2 \text{ cm/s}]$, are used to set the speed of the left and right motors, respectively. $Z(o_3)$ 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 initialized 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 initializing each component of each genotype to values randomly chosen from a uniform distribution in the range $[-10, 10]$. The population contains 100 genotypes. Generations following the first one are produced by a combination of selection, mutation, and elitism. For each new generation, the five highest-scoring individuals from the previous generation are chosen for reproduction. The new generations are produced by making 20 copies of each highest-scoring individual, with mutations applied only to 19 of them. Mutation entails that a random Gaussian offset be applied to each real-valued vector component encoded in the genotype, with a probability of 0.25.

5 The Experimental Setup and the Fitness Function

During evolution, each genotype is translated into a robot controller and cloned onto each agent. At the beginning of each trial, two s-bots are positioned in a boundless arena at a distance randomly generated in the interval $[25 \text{ cm}, 30 \text{ cm}]$. These distances are chosen because at this stage we study role allocation and self-assembly without addressing the issue of aggregation, and because of hardware constraints. In particular, two robots cannot be initialized closer than approximately 23 cm, given their diameter and the gripper's dimensions. Moreover, even if the robots can perceive each other at distances farther than 30 cm, the precision of the camera beyond that distance is quite low.² The initial orientations of the robots α and β (see Figure 2b) are predefined. Our initialization is inspired by the initialization used in [33]. In particular, we defined a set of (unordered) orientation pairs (α, β) as all the combinations with repetitions from a set

$$\Theta_n = \left\{ \frac{2\pi}{n} \cdot i \mid i = 0, \dots, n-1 \right\}, \quad (2)$$

where n is the cardinality of the set. In other words, we systematically choose the initial orientation of both s-bots drawing from the set Θ_n . The cardinality of the set of all the different pairs—where we

² In fact, beyond 30 cm it is possible to detect the presence of objects, but not to reliably determine their distance, due to the very noisy readings of the camera sensor.

consider $(\alpha, \beta) \equiv (\beta, \alpha)$ —corresponds to the total number of combinations with repetitions, and can be obtained by the following equation:

$$\frac{(n + k - 1)!}{k!(n - 1)!}, \tag{3}$$

where $k = 2$ indicates that combinations are pairs, and $n = 4$ lets us define the set of possible initial orientations $\Theta_4 = \{0^\circ, 90^\circ, 180^\circ, 270^\circ\}$. From this, we generate 10 different pairs (α, β) . Each group is evaluated four times at each of the 10 starting orientation pairs for a total of 40 trials. Each trial differs from the others in the initialization of the random number generator, which influences the robots' initial distance and their orientation by determining the amount of noise added to the orientation pairs (α, β) . During a trial, noise affects motors and sensors as well. In particular, uniform noise is added in the range ± 1.25 cm for the distance, and in the range $\pm 1.5^\circ$ for the angle of the colored blob perceived by the camera. 10% uniform noise is added to the motor outputs $Z(o_i)$. Uniform noise randomly chosen in the range $\pm 5^\circ$ 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), but a trial is also terminated if the robots incur 20 collisions.

The fitness assigned to each genotype after evaluation of the robot's behavior is the average of the fitness achieved in the 40 trials. In each trial, each group is rewarded by the following evaluation function, which seeks to assess the ability of the two robots to get closer to each other and to physically assemble through the gripper:

$$F = A \cdot C \cdot S. \tag{4}$$

Here A is the aggregation component, computed as follows:

$$A = \begin{cases} \frac{1.0}{1.0 + \arctan\left(\frac{d_{rr} - 16}{16}\right)} & \text{if } d_{rr} > 16 \text{ cm,} \\ 1.0 & \text{otherwise,} \end{cases} \tag{5}$$

with d_{rr} corresponding to the distance between the two s-bots at the end of the trial. This component helps to bootstrap evolution and to guide toward solutions in which the robots tend to approach each other.

C is the collision component, computed as follows:

$$C = \begin{cases} 1.0 & \text{if } n_c = 0, \\ 0.0 & \text{if } n_c > 20, \\ \frac{1.0}{0.5 + \sqrt{n_c}} & \text{otherwise,} \end{cases} \tag{6}$$

with n_c the number of robot-robot collisions recorded during the trial; the role of this collision component is to gradually punish collisions. The way in which collisions are modeled in simulation and handled by the fitness function is an element that favors the evolution of assembly strategies in which the s-bot gripper moves straight while approaching the s-bot gripee (see Section 3). This has been done to ease transferability to real hardware.

S is the self-assembly component, computed at the end of a trial ($t = T$ with $T \in (0, 250]$), as follows:

$$S = \begin{cases} 100.0 & \text{if } GG(T) = 1, \text{ for any robot,} \\ 1.0 + \frac{\sum_{t=0}^T K(t)}{T} & \text{otherwise,} \end{cases} \quad (7)$$

$K(t)$ is set to 1 for each simulation cycle t in which the sensor GS of any s-bot is active; otherwise $K(t) = 0$. The role of $K(t)$ within the self-assembly component (S) is very important: By rewarding the robot's sensing the turret of another robot, even if assembly is not achieved, we bootstrap evolution, since sensing an object within the open gripper is a prerequisite for establishing a connection.

Notice that, given the way in which F 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 gripee. All the components of the fitness function are meant to bootstrap evolution and to lead to collision-free self-assembly between the agents.

6 Results

As stated in Section 1, the goal of this research work is to design, through evolutionary computation techniques, dynamic neural networks to allow a group of two homogeneous s-bots to physically connect to each other. To pursue our objective, we run 20 randomly seeded evolutionary simulations for 10,000 generations. Although several evolutionary runs produced genotypes that obtained the highest fitness score (i.e., $F = 100$; see Section 5), the ranking based on the evolutionary performance has not been used to select a suitable controller for the experiments with real robots. The reason for this is that during evolution the best groups may have taken advantage of favorable conditions (robot initialization, noise level, etc.).

Thus, in order to select the genotype to be downloaded on the s-bots, we used the following procedure: We first identified the runs that produced genotypes that during evolution achieved the maximum fitness score ($F = 100$). Then, for these runs, which are 4 out of 20, the best evolved genotype from generation 5,000 to generation 10,000 was evaluated again on a series of 36,000 trials, obtained by systematically varying the s-bots' starting orientations. In particular, we evaluated the evolved genotypes using a wider set of initial orientations Θ_g , defined by Equation 2. The cardinality of this set of pairs is equal to 36^3 . Each starting condition (i.e., orientation pair) was tested in 1,000 trials, each time randomly choosing the robots' distance from a uniform distribution of values in the range [25 cm, 30 cm]. Noise was added to initial orientations, sensor readings, and motor outputs as described in Section 5.

From this pool of genotypes, we selected the one with the best average performance over the 36,000 post-evaluation trials. This genotype was decoded into an artificial neural network, which was then cloned and ported onto two real s-bots. In what follows, first we provide the results of post-evaluation tests aimed at evaluating the success rate of the real s-bots at the self-assembly task as well as the robustness of the self-assembly strategies in different setups (see Section 6.1). Subsequently, we illustrate the results of analyses carried out with simulated s-bots, aimed at unveiling operational aspects underlying the best evolved self-assembling strategy (see Section 6.2).

6.1 Post-Evaluation Tests on Real S-bots

The s-bots' controllers are evaluated four times on each of 36 different orientation pairs (α, β) , obtained by drawing α and β from Θ_g . As mentioned in Section 3, the s-bots have to turn on their

³ The cardinality is given by Equation 3 with $n = 8$, $k = 2$, and it is chosen to be higher than the one used during evolution in order to assess the ability of the controllers to generalize to a wider set of initial conditions.

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 initialized at a distance from each other of 25 and 30 cm, respectively. B30 and R30 refer to tests in which the s-bots light themselves up in blue and red, respectively, and are initialized at a distance of 30 cm from each other. Trials in which the physical connection between the s-bots requires more than one gripping attempt, due to inaccurate maneuvers I_i , are still considered successful. I_1 refers to a series of maladroit actions by both robots that makes it impossible for the s-bot gripper to successfully grasp the s-bot gripee's cylindrical turret. I_2 refers to those circumstances in which both robots assume the role of s-bot gripper and collide at the level of their grippers. I_3 refers to those circumstances in which, after grasping, the connected structure gets 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	Numbers of successful trials per gripping attempt and types of inaccuracy									No. of failures
	1st	2nd			3rd					
	Number	Number	I_1	I_2	I_3	Number	I_1	I_2	I_3	
G25	28 (77.78%)	7 (19.44%)	6	1	0	1 (2.78%)	2	0	0	0 (0.00%)
G30	29 (80.56%)	6 (16.67%)	3	3	0	1 (2.78%)	1	1	0	0 (0.00%)
B30	26 (72.22%)	5 (13.89%)	3	2	0	4 (11.11%)	8	0	0	1 (2.78%)
R30	21 (58.33%)	12 (33.33%)	8	0	2	4 (11.11%)	7	0	1	0 (0.00%)

colored LEDs in order to perceive each other through the camera. As discussed in Section 2.1, a significant advantage of our control design approach is that the specific color displayed has no functional role within the neural machinery that brings forth the s-bots' actions. However, the camera readings, as well as the optical barrier readings, vary with respect to the color of the perceived object. This effect can influence the performance of the s-bots. In order to test the robustness of our controllers against different LED colors, we decided to carry out experiments with LEDs emitting green, blue, and red color, with the robots initialized at 30 cm from each other. These tests are referred to as G30, B30, and R30, respectively. In the case of the green color, we also test our system for an initial distance of 25 cm—this test is referred to as G25. In each post-evaluation experiment, when one robot manages to grasp the other one, the trial is considered successful. Note that, for real s-bots, the trial's termination criteria were changed with respect to those employed with the simulated s-bots. We set no limit on the maximum duration of a trial, and no limit on the number of collisions allowed. In each trial, we let the s-bots interact until physically connected. As illustrated later in the section, these new criteria allowed us to observe interesting and unexpected behavioral sequences. In fact, the s-bots sporadically committed inaccuracies during their self-assembly maneuvers. Yet, the robots demonstrated the required capabilities to autonomously recover from these inaccuracies. In the following, we describe in detail the performance of the real s-bots in these post-evaluation trials.⁴

The s-bots proved to be successful in almost all tests. They managed to self-assemble in 135 out of 136 trials. Table 1 gives more details about the s-bots' performances. We notice that the proportion of successful trials at the first gripping attempt ranges from around 58% in the case of R30 to around 80% for G25. In a few trials, the s-bots managed to assemble after two or three grasping attempts (see Table 1, 3rd and 7th columns). The majority of the failed attempts were caused by inaccurate maneuvers—referred to as inaccuracies of type I_1 —in which a series of maladroit actions by both robots makes it impossible for the s-bot gripper to successfully grasp the s-bot gripee's cylindrical turret. In a few other cases the group committed a different inaccuracy—referred to as I_2 —in which both robots assume the role of s-bot gripper. In such circumstances, the s-bots head toward each other

⁴ Movies of the post-evaluation tests on real s-bots and data not shown in the article can be found at <http://iridia.ulb.ac.be/supp/IridiaSupp2008-002/>.

until a collision between their grippers occurs. Another type of inaccuracy emerged in test R30: In three trials, after grasping, the connected structure got slightly elevated at the connection point. We refer to this type of inaccuracy as I_3 . Note also that in a single trial, in test B30, the s-bots failed to self-assemble (see Table 1, last column). In this case, the s-bots moved so far away from each other that they ended up outside their perceptual camera range. This trial, in which the s-bots spent more than 1 min without perceiving each other, was terminated and considered unsuccessful. Overall, apart from the one failed trial, we observe that the s-bots always managed to recover from the inaccuracies and end up successful.

6.1.1 Behavioral Sequences

For each single test (i.e., G25, G30, B30, and R30), the sequences of s-bots' actions are rather different from one trial to another. However, these different histories of interactions can be succinctly described by a combination of a few distinctive phases and transitions between phases, which portray the observed phenomena. Figure 3 shows some snapshots from a successful trial that represent these phases. The robots leave their respective starting positions (see Figure 3a), and during the starting phase (see Figure 3b) they tend to get closer to each other. In the great majority of the trials, the robots move from the starting phase to what we call the role allocation phase (RA phase; see Figure 3c). In this phase, each s-bot tends to remain on the right side of the other. They slowly move, following a circular trajectory corresponding to an imaginary circle centered 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 assumes the role of the s-bot gripper, stops oscillating, and heads toward the other s-bot, which assumes the role of the s-bot gripee and orients itself properly in order to facilitate the gripping (gripping phase; see Figure 3d). The s-bot gripper approaches the s-bot gripee's turret and, as soon as its GS sensor is active, closes its gripper. A successful trial terminates as soon as the two s-bots are connected (see Figure 3e).

6.1.2 Inaccuracies and Recovery

As mentioned above, in a few trials the s-bots failed to connect at the first gripping attempt, by committing what we called inaccuracies I_1 and I_3 . These inaccuracies seem to indicate 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. Recovery 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. I_2 seems to be caused by the effects of the s-bots' starting positions on their behavior. 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. The way in which the robots perceive each other at starting positions seems to be the reason why they 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 gripee), and consequently they incur 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.

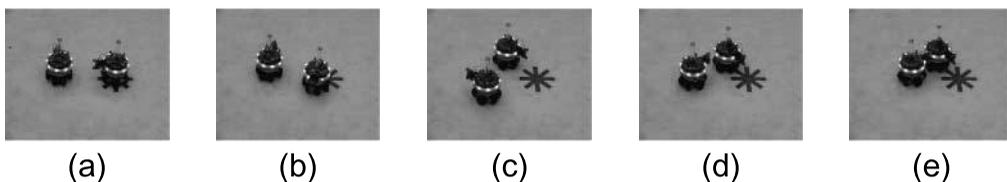


Figure 3. Snapshots from a successful trial: (a) initial configuration, (b) starting phase, (c) role allocation phase, (d) gripping phase, (e) success (grip).

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 3, in order to speed up the evolutionary process, the simulation in which controllers have been designed does not handle collisions with sufficient accuracy. In those cases in which, after a collision, the simulated robots had another chance to assemble, the agents were simply repositioned at a given distance to each other, and their controller was punished by the fitness function for incurring a collision. In spite of this, s-bots guided by the best evolved controllers proved capable of engaging in successful recovering maneuvers, which allowed them to eventually assemble.

The ability of the neural network to recover from inaccuracies must be directly linked to its generalization abilities and to the nature of the fitness function (see Section 5) that we employed to optimize its parameters. This fitness function rewards robots for achieving assembly under a comprehensive variety of initial conditions, without dictating the states through which the robots must go in order to self-assemble. This fact allows the CTRNN controller to operate in a continuous perception-action space, with the result that the network is able to generalize to conditions not encountered during evolution.

6.2 An Operational Description

In view of the results shown in Section 6.1, we believe that evolved neurocontrollers are a promising approach to the design of mechanisms for autonomous self-assembly. In the previous section, we demonstrated that the evolved mechanisms are as effective in controlling two real s-bots in the assembly task as those described in [16, 32]. However, it is important to remark that the operational principles of self-assembly used by the s-bots, controlled by this type of neural structure, are less transparent than the modular or hand-coded control described in [16, 32]. Further research work and experimental analysis are required to unveil the operational principles of the evolved neural controllers. What are the strategies that the s-bots use to carry out the self-assembly task? How do they decide which is the s-bot gripper, and which is the s-bot gripee? Although extremely interesting, providing an answer to this type of questions is not always a simple task. The methodologies we have in order to look for the operational mechanisms of evolved neural networks are limited to networks with a small number of neurons, or to cases in which the neural networks control simple agents that can only move in a one-dimensional world, or by discrete steps (see [4, 5, 26] for details). Due to the nature of our system, most of these methods cannot be directly employed to investigate which mechanisms control the process by which two homogeneous s-bots differentiate into s-bot gripper and s-bot gripee. In spite of these difficulties, we describe below the results of an initial series of studies focused on the relationship between the s-bots' starting orientations and the role allocation process.

Do the robots' orientations at the beginning of a trial influence the way in which roles (i.e., s-bot gripper versus s-bot gripee) are allocated? We start our analysis by looking at the results of the post-evaluation tests mentioned at the beginning of Section 6. In particular, we look at those data concerning the behavior of the s-bots controlled by the best performing genotype, that is, the genotype used to build the networks ported on the real robots. Recall that, in these tests, the simulated s-bots have been evaluated on a series of 36 starting orientation pairs (α, β) obtained from Θ_s . For each orientation pair the s-bots underwent 1,000 evaluation trials, each time randomly choosing the agents' distance from a uniform distribution of values in the range [25 cm, 30 cm]. Also notice that uniform noise randomly chosen in the range $\pm 5^\circ$ is added to the initial orientation of each s-bot. The 36 orientation pairs include eight symmetrical conditions in which $\alpha = \beta$. In symmetrical orientation pairs, the robots share the same perception at the beginning of the trial. That is, they perceive each other through the same sectors of their corresponding cameras. Asymmetrical orientation pairs are those in which $\alpha \neq \beta$.

Contrary to the real s-bots, the simulated robots, due to the way our simulator handles collisions, are not allowed to use any recovery maneuvers. That is, in these post-evaluation tests, the simulated s-bots are scored according to a binary criterion: A trial can be either successful or unsuccessful. Unsuccessful

trials are considered those in which the robots did not manage to self-assemble within the time limit, as well as those that terminated due to the occurrence of collisions that are not considered the result of an accepted grasping maneuver (see Section 3 for details).

In Figure 4a, we can see boxplots of the success rate of simulated s-bots, controlled by the best evolved genotype, for asymmetrical and symmetrical sets of trials (see caption for details). We observe that both medians are around 95%, and thus we can claim that the system manages to be very successful for both symmetric and asymmetric conditions.

Given that robots proved to be successful even in symmetrical trials, we can already exclude the possibility that the system works by following simple rules by which the role is determined by the initial individual perception. In other words, having the same initial perception does not hinder the robots from allocating different roles. Therefore, either the system has to be governed by more complex principles based on the combination of α and β , or the initial orientations do not influence the role allocation process. In the remainder of this section, we carry out an analysis that helps us further clarify this issue.

By looking at the frequency with which each s-bot (i.e., s-bot L and s-bot R) plays the role of s-bot gripper for any given values of α and β , our analysis is intended to unveil any relationship between the robots' initial orientations and the role they assume during the trial. In particular, we looked at the *role ratio*, which can be considered a property of each orientation pair. It indicates how often a given robot (i.e., s-bot L or s-bot R) played the role of s-bot gripper when repeatedly evaluated on a given orientation pair. In particular, the role ratio corresponds to the highest frequency of playing the s-bot gripper role between the one recorded by s-bot L and by s-bot R. Thus, the role ratio can vary between 50%, when the two robots played the s-bot gripper role with the same frequency, to 100%, when only one robot plays the s-bot gripper role in all the trials that start with the same perceptual scenario.

In Figure 4b, we provide a boxplot of the role ratio for asymmetrical and symmetrical sets of trials. We clearly see that while the totality of the orientation duplets corresponding to symmetrical starting positions is characterized by a role ratio around 50%, the large majority of the orientation pairs corresponding to asymmetrical starting positions are characterized by a role ratio near 100%. This means that while in the large majority of the asymmetrical trials the role of s-bot gripper is played by the same robot, in all symmetrical trials both robots play the role of s-bot gripper with more or less the same frequency.

Our analysis revealed that in the case in which the two robots have different initial perceptions, the role that each s-bot assumes can (usually) be predicted knowing the combination of α and β . This means that the initialization of the robots influences the final role allocation. It is important, though, to stress that it is the combination of the two orientations that determines the roles. In other words, perceiving the other robot at a specific distance and at a certain angle does not inform a robot about the role it will assume during the trial; this role equally depends on the initialization of the other robot. As a consequence, the robots go through a dynamic interaction that eventually leads to the allocation of roles. This dynamic interaction can be considered a sort of negotiation phase

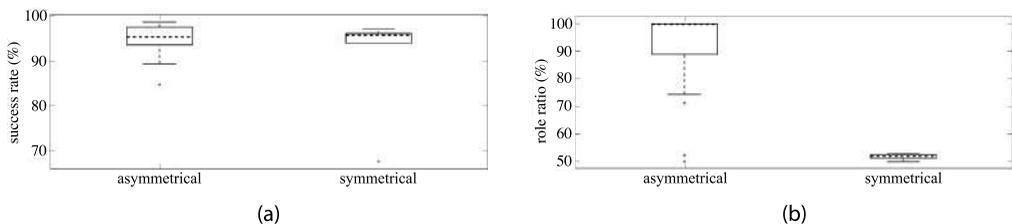


Figure 4. (a) Success rate for asymmetrical (28 out of 36 pairs) and symmetrical (8 out of 36 pairs) starting conditions. (b) Role ratio for asymmetrical and symmetrical starting conditions. Every observation in the boxplots corresponds to one orientation pair, and it represents either (a) the percentage of success or (b) the role ratio in 1,000 evaluation trials. Boxes represent the interquartile range of the data, while dashed horizontal bars in bold inside the boxes mark the median values. The whiskers extend to the most extreme data points within 1.5 times the interquartile range from the box. Crosses mark the outliers.

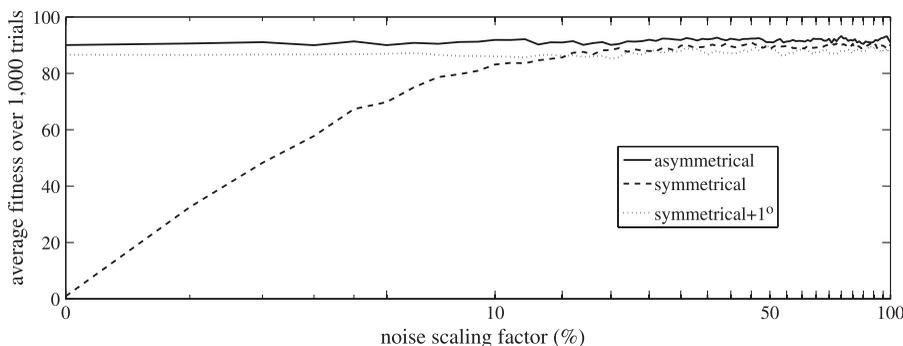


Figure 5. Logarithmic plot of the average fitness over 1,000 trials with the noise scaling factor for three different conditions: asymmetrical, symmetrical, and symmetrical + 1°.

between the two robots to allocate the roles of s-bot gripper and s-bot gripee between them. Such a negotiation unfolds in time during the whole trial.

On the other hand, in those cases in which the robots start with identical perceptions ($\alpha = \beta$), the experimenter cannot predict the outcome of the role allocation. So the question that should be asked is what drives the differentiation in the absence of initial perceptual asymmetries. In order to provide an answer to this question, we perform the following test: We gradually reduce the range of the random noise applied to sensors and actuators until no noise is present, and we record the average fitness of the system for 1,000 trials drawn from Θ_8 , for three different conditions: (i) asymmetrical ($\alpha \neq \beta$); (ii) symmetrical ($\alpha = \beta$); and (iii) symmetrical + 1° ($\alpha = \beta + 1^\circ$ —in this case, the initial orientations differ by only 1°).

In Figure 5, we plot the results of this test on a logarithmic scale. We can clearly see that (i) for asymmetrical trials (see continuous line), the noise scaling factor has no effect on performance and the initial asymmetry is what causes the differentiation of the controllers; (ii) for symmetrical + 1° (see dotted line), the same situation holds, that is, even 1° of difference in the initial perceptions can be enough to produce differentiation; (iii) for symmetrical trials (see dashed line), the noise scaling factor has a big effect on the number of trials where differentiation is achieved. In particular, this factor has to be over 10% in order for the performance to reach levels as high as with the other two cases.

Thus, we can say that in the symmetrical case, it is the random noise (real-world, or injected into the simulation) that introduces subtle asymmetries that lead to role allocation. In this particular case, the role of an s-bot is determined by stochastic phenomena, which justifies a role ratio of about 50%. Also, we have shown that a difference in initial perception as small as 1° is enough to produce differentiation and role allocation. This indicates that the neural network is amplifying the differences among the robots. Indeed, in the absence of noise and of significant asymmetries, it is the internal dynamics that amplifies the small asymmetries, which results in role allocation.

Finally, it should be mentioned that the type of solution described above is qualitatively similar to other solutions found by evolution. In particular, we analyzed successful genotypes of other evolutionary runs and we found that the robots rely on similar behavioral strategies to achieve assembly. However, solutions can differ on how the space of asymmetrical configurations is segmented; that is, for the same robot initialization, different solutions can lead to a different robot assuming a certain role. Also, rhythmic oscillation of the heading of the robots seems to be a common characteristic of successful strategies, even if the motion and trajectories of the robots can be different.

7 Discussion

In a context free of assumptions concerning the nature of the mechanisms underlying the agents' behavioral repertoire, our evolutionary robotics model exploits an automatic design process that mimics the mechanisms of natural evolution to define the control structures that allow the robots

to autonomously self-assemble by playing complementary roles (i.e., s-bot gripper and s-bot gripee). The results of post-evaluation analyses shown in Section 6.2 illustrate that the allocation of roles is the result of an autonomous negotiation phase between the two robots. The outcome of any action an agent chooses depends on the action of the other agent. In other words, neither of the two agents can know the role it will assume at the end of the trial, judging only from its initial perception.

We have shown on real hardware that explicit communication to directly access the “intention” of the other agent (through explicit signals, e.g., the ones used in [16]) is not a necessary condition for coordination. Our robots coordinate without direct and explicit communication. Noble [30] reached a similar conclusion with an evolutionary simulation model involving two simulated animals contesting the possession of a resource. Groß and Dorigo [17] have also concluded that cooperative behavior can be achieved without explicit means of communication. More specifically, in a cooperative transport task, simulated robots could find effective transport strategies exploiting indirect communication, that is, by interacting with each other through the object to be transported. Similarly, Ijspeert et al. [23] show that a homogeneous population of robots deprived of means for explicit communication can coordinate their actions and assume complementary roles in a stick-pulling experiment. In particular, each stick in order to be removed from the ground needs the engagement of two robots at different levels, and implicit communication takes place via the stick elevation. Our work shares with this research the fact that dynamic role allocation can be achieved without a priori introduced heterogeneities. Finally, our results are very similar to the results obtained by Quinn [33] and Quinn et al. [34], where role allocation (leader-follower) and formation movement are achieved solely through infrared sensors, and the control structure is once again an evolved dynamic neural network. In particular, Quinn [33] reports on role allocation between two robots for symmetrical and asymmetrical cases. While the author qualitatively explains how the difference in the initial perceptions influences the role allocation for asymmetrical cases, an analysis of the evolved behavior in the case of insufficient differences is not performed. In the analysis performed in Section 6.2, we have explained quantitatively, and to some degree qualitatively, the effect of the starting configuration on the final outcome of a trial (how roles are allocated); the great majority of asymmetrical configurations severely bias the role allocation process, while random noise is the element that produces differentiation and role allocation for symmetrical configurations.

Our system proved to be very effective in controlling two physical robots, due to its robustness against real-world noise and inter-robot differences. This robustness is demonstrated by the high performance in all our experiments, also due to the recovery mechanism. However, we did not test the robustness of the system against the initial distance of the robots, for two reasons: First, the goal of this research is to obtain self-assembly with physical robots and not the integration of self-assembly with aggregation. Second, as we have already explained in Section 5, the reliability of the camera sensor for distances farther than 30 cm is limited. This fact will affect the behavior of the physical agents, even if we expect the system to cope with this disruption to a certain extent, due to the recovery mechanism.

It should be noted that the robots’ initialization is an also important parameter for the evolutionary processes. Our choice was aimed to evolve a system that can cope with all possible orientation pairs. Altering the proportion of symmetrical and asymmetrical orientation pairs experienced throughout evolution might have an impact on the evolved role allocation strategies. These strategies also depend on the cardinality of the group, because evolving systems with more robots will definitely affect their nature. For example, in a system composed of many robots, symmetrical conditions might be extremely rare. Also, asymmetrical conditions could be so numerous that it might be easier for artificial evolution to design neurocontrollers relying more on stochastic events than the ones presented in this study. After all, stochastic events are the main driving force for self-assembly not only in nature, but also in the majority of engineered systems composed of a large number of components (see [2, 27] for examples).

Future work will also focus on the scalability of our system. Can the controllers we presented in the previous sections still manage to achieve assembly if there are more than two robots involved? The fitness function rewards two robots connecting to each other, but it does not explicitly impose the formation of one single structure: If we put more than three robots in our arena, nothing guarantees

the formation of one single swarm-bot. We did not perform a full scalability test, but only some initial experimentation with three robots, whose results are encouraging. In any case, if scalability is a desired property of our controllers, then it would be useful to make new evolutionary runs with more than two robots participating in the trials. This is because the controllers we evolved may be optimized for a two-robot case. This is a common problem in ER experimentation; for example, the controllers evolved in [34] are non-scalable; that is, they cannot successfully control a group of robots whose cardinality is larger than the one with which they were evolved [40]. However, we might expect our system to be able to cope with this challenge to a certain extent, due to its tolerance of inaccuracies and to the recovery mechanism.

Moreover, if we want to move from the study of self-assembly among s-bots to the study of self-assembly among swarm-bots, important issues that have been disregarded in the current work have to be taken into account. More specifically, the connected structure must have the ability to move coordinately: It should be able to perform coordinated motion [3], which means it should be equipped with more sensors and actuators (traction sensor and rotating turret for the case of the s-bot), in order to actively participate in the assembly process. For example, it could interact with other assembled structures or individual robots by either receiving connections from them or grasping them.

Finally, it should be mentioned that the research detailed in this article will be integral to the study of functional self-assembly, which will be tackled in future work. More in detail, we have managed to design a simulated environment that can model the relevant aspects of the fine-grained sensory-motor coordination required sufficiently well to achieve assembly with real robots. Thus, we will reuse this environment in order to study cases when the assembly should not be a priori demanded, but instead should be a consequence of the environmental contingencies. In other words, we will study more complex scenarios in which self-assembly is functional to the achievement of particular objectives that are beyond the capabilities of a single robot. For example, in [37] we started experimentation exploiting a setup where robots have to infer if the environment requires self-assembly by categorizing it individually or collectively. In particular, in one environment, the robots should move as an assembled structure in order to bridge a gap too large to be crossed by an individual robot and eventually reach a goal location.

8 Conclusion

In this article, we have presented the results of an evolutionary methodology for the design of control strategies for self-assembling robots. More specifically, to the best of our knowledge, the control method we have proposed for the physical connection of two robots is the only one existing in the literature where the role allocation between gripper and gripee is the result of an autonomous negotiation phase between homogeneous robots; there is no a priori injected behavioral or morphological heterogeneity in the system. Instead, the behavioral heterogeneity emerges through the interaction of the robots. Moreover, the communication requirements of our approach are reduced to the minimum; simple coordination by means of the dynamic interaction between the robots—as opposed to explicit communication of internal states—is enough to bring forth differentiation within the group. We believe that reducing the assumptions on necessary conditions for assembly is an important step to obtain more adaptive and more general controllers for self-assembly in autonomous robots.

The results of this work are a proof of concept: They prove that dynamic neural networks shaped by evolutionary computation techniques directly controlling the robots' actuators can provide physical robots all the required mechanisms to autonomously perform self-assembly. Contrary to the modular or hand-coded controllers described in [16, 32], the evolutionary robotics approach did not require the experimenter to make any a priori assumptions concerning the roles of the robots during self-assembly (i.e., either s-bot gripper or s-bot gripee) or about their status (e.g., either capable of moving or required not to move). Furthermore, in Section 6.1 we presented a system that exhibits recovery capabilities that could not be observed during the artificial evolution and that were not coded or foreseen by the

experimenter. Such a feature in our case comes for free, while in the case of Groß et al. [16] a recovery mechanism had to be designed as a specific behavioral module to be activated every time the robots failed to achieve assembly.

One major contribution of our work is the strengthening of the evidence that evolved neural networks can be a reliable and efficient way of controlling real robots engaged in real-world tasks requiring fine sensory-motor coordination, such as the establishment of a physical connection between two autonomous mobile robots. It is important to stress that the networks we used directly control all robots actuators, without the need for hand-coded filters mediating between the output of the network and the performed actuation.

Nevertheless, despite the advantages presented above, our system is not as transparent as a hand-coded control system, as we cannot easily break its behavior down to a set of rules or states. To do so seems to be very challenging and particularly difficult, especially when the network size is large and/or the movement of the robots takes place in a continuous and noisy world, such as the real world. However, we would like to stress that we do not consider this step a necessary precondition for the success of research work using evolutionary robotics as a design methodology. Our view is that it is more important to identify those choices that made the implementation and experimentation successful. That is, we put the stress on better understanding which principles make the evolutionary machinery able to produce efficient rules to guide groups of robots, rather than on identifying each and every one of these rules.

Acknowledgments

The authors would like to thank Dr. Roderich Groß, Dr. Francisco Santos, and Marco Montes de Oca for comments, recommendations, and fruitful discussions during the production of the manuscript.

Tuci and Dorigo acknowledge European Commission support via the ECAgents project, funded by the Future and Emerging Technologies program (grant IST-1940). Dorigo acknowledges support from the Belgian F.R.S.-FNRS, of which he is a research director. Dorigo and Ampatzis acknowledge support from the ANTS project, an Action de Recherche Concertée funded by the Scientific Research Directorate of the French Community of Belgium. Trianni and Dorigo acknowledge support via the Swarmanoid project, funded by the Future and Emerging Technologies program (grant IST-022888). 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.

This research work was done while all authors were with CoDE-IRIDIA, Université Libre de Bruxelles (ULB).

References

1. Anderson, C., Theraulaz, G., & Deneubourg, J.-L. (2002). Self-assemblages in insect societies. *Insectes Sociaux*, 49(2), 99–110.
2. Arbuckle, D., & Requicha, A. (2004). Active self-assembly. In *Proceedings of 2004 IEEE International Conference on Robotics and Automation (ICRA'04)* (pp. 896–901). Piscataway, NJ: IEEE Computer Society Press.
3. Baldassarre, G., Trianni, V., Bonani, M., Mondada, F., Dorigo, M., & Nolfi, S. (2007). Self-organized coordinated motion in groups of physically connected robots. *IEEE Transactions on Systems, Man and Cybernetics—Part B: Cybernetics*, 37(1), 224–239.
4. Beer, R. (2003). The dynamics of active categorical perception in an evolved model agent. *Adaptive Behavior*, 11(4), 209–243.
5. Beer, R. (2006). Parameter space structure of continuous-time recurrent neural networks. *Neural Computation*, 18, 3009–3051.
6. Beer, R. D., & Gallagher, J. C. (1992). Evolving dynamical neural networks for adaptive behavior. *Adaptive Behavior*, 1, 91–122.
7. Bonabeau, E., Dorigo, M., & Theraulaz, G. (1999). *Swarm intelligence: From natural to artificial systems*. New York: Oxford University Press.

8. Brown, H., Weghe, J. V., Bererton, C., & Khosla, P. (2002). Millibot trains for enhanced mobility. *IEEE/ASME Transactions on Mechatronics*, 7, 452–461.
9. Castano, A., Shen, W., & Will, P. (2000). CONRO: Towards deployable robots with inter-robot metamorphic capabilities. *Autonomous Robots*, 8, 309–324.
10. Christensen, A. (2005). *Efficient neuro-evolution of hole-avoidance and phototaxis for a swarm-bot*. DEA thesis, TR/IRIDIA/2005-14, Université Libre de Bruxelles, Bruxelles, Belgium.
11. Damoto, R., Kawakami, A., & Hirose, S. (2001). Study of super-mechano-colony: Concept and basic experimental set-up. *Advanced Robotics*, 15(4), 391–408.
12. Dorigo, M. (2005). Swarm-bot: A novel type of self-assembling robot. In *Proceedings of the 3rd International Symposium on Autonomous Minirobots for Research and Edutainment (AMiRE 2005)* (pp. 3–4). Berlin: Springer-Verlag.
13. Fukuda, T., & Nakagawa, S. (1987). A dynamically reconfigurable robotic system (concept of a system and optimal configurations). In *Proceedings of the 1987 IEEE International Conference on Industrial Electronics, Control and Instrumentation* (pp. 588–595). Piscataway, NJ: IEEE Computer Society Press.
14. Fukuda, T., Nakagawa, S., Kawachi, Y., & Buss, M. (1988). Self organizing robots based on cell structures—CEBOT. In *Proceedings of the 1988 IEEE International Workshop on Intelligent Robots* (pp. 145–150). Piscataway, NJ: IEEE Computer Society Press.
15. Fukuda, T., & Ueyama, T. (1994). *Cellular robotics and micro robotic systems*. London: World Scientific Publishing.
16. Groß, R., Bonani, M., Mondada, F., & Dorigo, M. (2006). Autonomous self-assembly in swarm-bots. *IEEE Transactions on Robotics*, 22(6), 1115–1130.
17. Groß, R., & Dorigo, M. (2008). Evolution of solitary and group transport behaviors for autonomous robots capable of self-assembling. *Adaptive Behavior*, 16(5), 285–305.
18. Groß, R., & Dorigo, M. (2008). Self-assembly at the macroscopic scale. *Proceedings of the IEEE*, 96(9), 1490–1508.
19. Groß, R., Dorigo, M., & Yamakita, M. (2006). Self-assembly of mobile robots—From swarm-bot to super-mechano colony. In *Proceedings of the 9th International Conference on Intelligent Autonomous Systems* (pp. 487–496). Amsterdam: IOS Press.
20. Hirose, S. (2001). Super-mechano-system: New perspectives for versatile robotic systems. In D. Rus & S. Singh (Eds.), *Proceedings of the 7th International Symposium on Experimental Robotics (ISER)* (pp. 249–258). Berlin: Springer.
21. Hirose, S., Shirasu, T., & Fukushima, E. (1996). Proposal for cooperative robot “Gunryu” composed of autonomous segments. *Robotics and Autonomous Systems*, 17, 107–118.
22. Hölldobler, B., & Wilson, E. O. (1978). The multiple recruitment systems of the African weaver ant, *Oecophylla longinoda* (latreille) (Hymenoptera: Formicidae). *Behavioural Ecology and Sociobiology*, 3, 19–60.
23. Ijspeert, A. J., Martinoli, A., Billard, A., & Gambardella, L. M. (2001). Collaboration through the exploitation of local interactions in autonomous collective robotics: The stick pulling experiment. *Autonomous Robots*, 11(2), 149–171.
24. Izzo, D., Pettazzi, L., & Ayre, M. (2005). Mission concept for autonomous on orbit assembly of a large reflector in space. In *56th International Astronautical Congress* (paper IAC-05-D14.03).
25. Jakobi, N. (1997). Evolutionary robotics and the radical envelope of noise hypothesis. *Adaptive Behavior*, 6, 325–368.
26. Keinan, A., Sandbank, B., Hilgetag, C., Meilijson, I., & Ruppin, E. (2006). Axiomatic scalable neurocontroller analysis via the Shapley value. *Artificial Life*, 12, 333–352.
27. Klavins, E. (2007). Programmable self-assembly. *IEEE Control Systems Magazine*, 27(4), 43–56.
28. Mondada, F., Gambardella, L. M., Floreano, D., Nolfi, S., Deneubourg, J.-L., & Dorigo, M. (2005). The cooperation of swarm-bots: Physical interactions in collective robotics. *IEEE Robotics & Automation Magazine*, 12(2), 21–28.
29. Mondada, F., Pettinaro, G., Guignard, A., Kwee, I., Floreano, D., Deneubourg, J.-L., Nolfi, S., Gambardella, L., & Dorigo, M. (2004). Swarm-bot: A new distributed robotic concept. *Autonomous Robots*, 17(2–3), 193–221.

30. Noble, J. (1998). Tough guys don't dance: Intention movements and the evolution of signalling in animal contests. In *Proceedings of the 5th International Conference on Simulation of Adaptive Behavior: From Animals to Animats 5* (pp. 471–476). Cambridge, MA: MIT Press.
31. Nolfi, S., & Floreano, D. (2000). *Evolutionary robotics: The biology, intelligence, and technology of self-organizing machines*. Cambridge, MA: MIT Press.
32. O'Grady, R., Groß, R., Mondada, F., Bonani, M., & Dorigo, M. (2005). Self-assembly on demand in a group of physical autonomous mobile robots navigating rough terrain. In *Proceedings of the 8th European Conference on Artificial Life (ECAL'05)* (pp. 272–281). Berlin: Springer Verlag.
33. Quinn, M. (2001). Evolving communication without dedicated communication channels. In *Advances in Artificial Life. Proceedings of the 6th European Conference on Artificial Life (ECAL'01)* (pp. 357–366). Berlin: Springer Verlag.
34. Quinn, M., Smith, L., Mayley, G., & Husbands, P. (2003). Evolving controllers for a homogeneous system of physical robots: Structured cooperation with minimal sensors. *Philosophical Transactions of the Royal Society of London, Series A: Mathematical, Physical and Engineering Sciences*, 361, 2321–2344.
35. Requicha, A. (2003). Nanorobots, nems, and nanoassembly. *Proceedings of the IEEE*, 91(11), 1922–1933.
36. Rubenstein, M., Payne, K., Will, P., & Shen, W.-M. (2004). Docking among independent and autonomous CONRO self-reconfigurable robots. In *Proceedings of the 2004 IEEE International Conference on Robotics and Automation (ICRA'04)*, Vol. 3 (pp. 2877–2882). Piscataway, NJ: IEEE Computer Society Press.
37. Trianni, V., Ampatzis, C., Christensen, A. L., Tuci, E., Dorigo, M., & Nolfi, S. (2007). From solitary to collective behaviours: Decision making and cooperation. In *Advances in Artificial Life. Proceedings of the 9th European Conference on Artificial Life (ECAL'07)* (pp. 575–584). Berlin: Springer Verlag.
38. Tuci, E., Ampatzis, C., Vicentini, F., & Dorigo, M. (2008). Evolving homogeneous neuro-controllers for a group of heterogeneous robots: Coordinated motion, cooperation, and communication. *Artificial Life*, 14(2), 157–178.
39. Tuci, E., Groß, R., Trianni, V., Bonani, M., Mondada, F., & Dorigo, M. (2006). Cooperation through self-assembling in multi-robot systems. *ACM Transactions on Autonomous and Adaptive Systems*, 1(2), 115–150.
40. Vicentini, F., & Tuci, E. (2007). Scalability in evolved neurocontrollers that guide a swarm of robots in a navigation task. In *Proceedings of the 2nd International Workshop on Swarm Robotics* (pp. 206–220). Berlin: Springer-Verlag.
41. Whitesides, G., & Grzybowski, B. (2002). Self-assembly at all scales. *Science*, 295, 2418–2421.
42. Yamakita, M., Taniguchi, Y., & Shukuya, Y. (2003). Analysis of formation control of cooperative transportation of mother ship by SMC. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA'03)*, Vol. 1 (pp. 951–956). Piscataway, NJ: IEEE Computer Society Press.
43. Yim, M., Zhang, Y., & Duff, D. (2002). Modular robots. *IEEE Spectrum*, 39(2), 30–34.
44. Yim, M., Zhang, Y., Roufas, K., Duff, D., & Eldershaw, C. (2002). Connecting and disconnecting for chain self-reconfiguration with polybot. *IEEE/ASME Transactions on Mechatronics*, 7(4), 442–451.