Evolving collective control, cooperation and distributed cognition

Vito Trianni and Stefano Nolfi
LARAL, Institute of Cognitive Sciences and Technologies
via S. Martino della Battaglia 44, 00185 Rome, Italy
vito.trianni@istc.cnr.it stefano.nolfi@istc.cnr.it

1 Introduction

Studies in collective robotics usually emphasise aspects like efficiency, robustness and flexibility of the system. These are all desirable features for a robotic system, which however do not come for free along with the distributed approach. Suitable design methodologies must be devised to obtain similar features in a collective robotic system, both for what concerns the robotic hardware and the control algorithms. In this chapter, we discuss the problem of synthesising distributed controllers for a group of robots by using evolutionary techniques.

The synthesis of a controller for a collective robotic system is a difficult problem, due to the indirect relationship between the desired group behaviour and the individual control rules. This is particularly true for self-organising behaviours, in which the spatio-temporal pattern observed at the system level emerges from the numerous interactions among the individual robots. A possible solution to the problem of designing for self-organisation is given by automatic techniques, which can synthesise the robot controller basing on some system-level utility metric defined by the user. Among the various machine learning methodologies, *Evolutionary Robotics* (ER) represents a viable approach to the solution of the design problem [Nolfi and Floreano, 2000, Trianni et al., 2008]. By evaluating the robotic system as a whole (i.e., by testing the global behaviour that results from the individual rules encoded into the individual genotype), ER provides an automatic process for identifying the mechanisms that produce and support the collective behaviour, and for implementing those mechanisms into the individual controller rules that regulate the robot/environment interactions. To this aim, it is necessary to identify initial conditions that assure the *evolvability* of the system, i.e., the possibility to progressively synthesise better solutions [Wagner and Altenberg, 1996]. Evolvability depends on a number of factors related both to the task to be faced and to the evolutionary algorithm in use. In Section 2, we point to the most common factors and to the influence they may have on the evolvability of the system. However, due to the high level of mutual dependency among these factors, it is difficult to determine the outcome of a certain choice without reference to the whole picture. For this reason, we present in detail a simple approach, which proved particularly successful for the synthesis of self-organising behaviours. We also present in Section 3 some case studies that demonstrate the possibility to evolve robotic systems displaying coordination, self-organisation and collective decision making. Moreover, we show how ER can be a powerful tool for the study of adaptive behaviour, communication and cooperation [Harvey et al., 2005, Nolfi, 2005]. Various complexity levels can be added to the basic system described above in order to evolve cooperative, cognitive behaviours in a collective. For instance, the individual ability to integrate information over time makes it possible to obtain complex group behaviours that can rely on both the individual and the group dynamics. In an evolutionary perspective, this can result in complex forms of cooperation particularly adapted to the experimental scenario.

In conclusion, in this chapter we show how artificial evolution applied to collective robotics can produce coordinated and cooperative behaviours. Future work in this direction should try to increase the complexity of the behaviours that can be evolved. There are two possible directions, in our view: on the one hand, more complex behaviours can be evolved by providing more capabilities and more structure to the individual controller. In this case, complex individual behaviours support the cooperation between individuals, for instance, through the development of a cooperative language that can help regulating the inter-individual interactions. We believe that another, very promising and yet-to-be-explored direction should fully rely on self-organisation. That is, capabilities of the individual robot should be relatively simple, but the group behaviour should be the result of the numerous interactions among individuals in the group. Brought to the limit, this approach sees robots as extremely simple devices able to support with their interactions complex cognitive behaviour at the group level.

This chapter is organised as follows: Section 2 describes the evolutionary approach to collective robotics, indicating the main issues in the application of artificial evolution to collective robotics (see Section 2.1), and presents a simple approach for the synthesis of self-organising behaviours. In Section 3, the introduced theoretical framework is instantiated in three case studies. Finally, Section 4 concludes the paper.

2 Evolutionary Methods in Collective Robotics

Evolutionary Robotics (ER) is an automatic approach for synthesising robotic systems provided with desired features and behavioural abilities [Nolfi and Floreano, 2000, Harvey et al., 2005]. ER is based on Artificial Evolution [Fogel et al., 1966, Holland, 1975, Schwefel, 1981, Goldberg, 1989]. It is an adaptive process inspired by the Darwinian principles of genetic variation and selective reproduction of the fittest: the individual that best adapts to its environment has more chances to reproduce and pass its genetic material to the subsequent generations. In this way, the species evolves toward better and better individuals. A similar process is exploited in the artificial counterpart, in which a population of individuals, each representing a potential solutions to a given problem, is "evolved" for many generations, until a good solution is found. In the following section, we briefly introduce artificial evolution and we focus on its application to the collective robotics domain. In Section 2.2 we introduce a simple approach for the evolution of coordinated and cooperative behaviours in a collective robotics setup.

2.1 The ER Approach

Artificial evolution is an unsupervised learning technique that operates on a population of potential solutions to a given problem, which "evolve" under the selective pressure enforced by a user-defined utility metric, thanks to the stochastic variability ensured by the user-defined genetic operators. More specifically, each individual of a population, generically called *genotype*, represents a solution for a given problem. Its *fitness*—i.e., the quality of the solution—is automatically evaluated in each generation thanks to a user-defined performance metric, usually referred to as *fitness function.* The best individuals of the population, identified through a *selection* operator, are allowed to "reproduce" by generating copies of their genotypes. The latter are modified using genetic operators, such as *crossover* (sexual reproduction) or *mutation* (asexual reproduction). In this way, the offspring is generated forming a new population that undergoes the same evaluation process, until a valid solution is found.

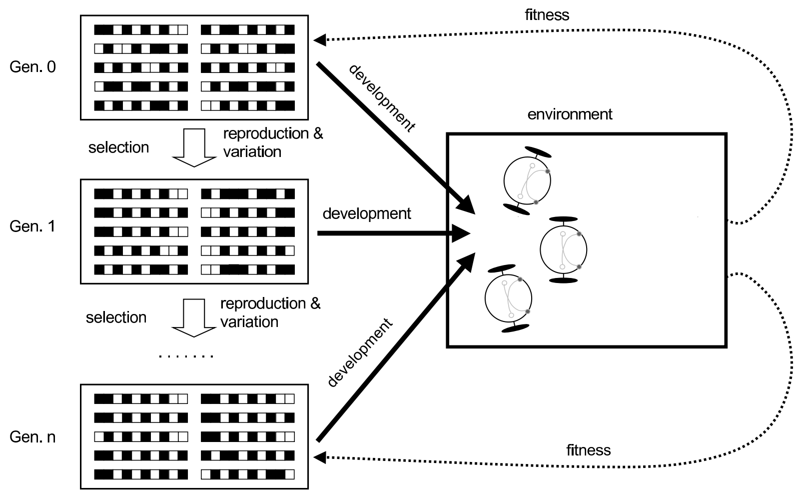


Figure 1: A schematic representation of the evolutionary robotics approach. Populations of genotypes evolve for multiple generations (left part of the figure). The fitness of a genotype is evaluated according to the behaviour of the corresponding robotic system—either physical or simulated—which is evaluated in its environment (right part of the figure). The evolutionary process can be run (i) in simulation, (ii) directly on the real robotic platform or (iii) on a mix of the two. In this chapter, we evolve the behaviours in simulation, and we validate the obtained results on the physical platforms.

In the case of ER, the general scheme described above must be adapted to produce solutions for a robotics problem. Robots can be considered as autonomous artificial organisms that evolve their skills in close interaction with their environment. An initial population of artificial genotypes is created randomly, each genotype encoding some parameters of the robotic system (see the generation zero represented in the top-left part of Figure 1). The behaviour displayed by the robotic system is evaluated with respect to the user-defined utility metric—the *fitness function*—that typically provides a reward commensurate with the robot's ability in solving the user defined task (right part of Figure 1). The best genotypes are selected and reproduce according to the genetic operators, and the process is iterated for a certain number of generations, or until the fitness of the individual robots overcome a certain threshold. The main issue in applying artificial evolution to a robotic setup is measuring the fitness associated to each genotype, which allows the evolutionary machinery to produce better and better solutions. Generally speaking, in order to measure the genotype fitness it is necessary to define (i) the mapping from the genotype to the *phenotype*, along with the architecture of the controller, (ii) the performance metrics that reward the desired behaviour (e.g., the actual fitness function), and (iii) the characteristics of the environment in which the robots are evaluated, which contribute in defining the ecological selective pressures.

2.1.1 Genotype-Phenotype Mapping and Robot Configuration

In evolutionary computing methods (including ER), a genotype is usually a string of bits or real numbers that encode a potential solution to a given problem. In ER, the genotype specifies the characteristics of a robotic system that should be able to display a desired behaviour. The experimenter therefore has to specify the genotype-to-phenotype mapping, i.e., the rules or the processes that determine the relation between the genotype—the string of bits or numbers—and the phenotype—the robotic system. A widely used approach consists in encoding into the genotype a fixed number of parameters of the robot controller (typically realized through an artificial neural network[1]), while keeping constant the controller structure and the robot sensory-motor configuration. This is the most common approach in the literature. Other approaches are possible, such as evolving the controller architecture [Stanley and Miikkulainen, 2002], evolving both the controller and the morphology of the robots [Sims, 1994, Lipson and Pollack, 2000]. With these instantiations, ER can contribute to the field of developmental/epigenetic robotics by studying the preconditions for the evolution of robotic systems that develop, adapt and learn new abilities over time in close interaction with their environment (for a survey of developmental robotics, see [Lungarella et al., 2003]). Another interesting approach consists in evolving self-assembling and self-replicating robots. In this case, it is possible to avoid specifying a fitness function, since evolution could be driven simply by the differential ability of the robots to survive and replicate. To date, however, self-replicating embodied agents have been demonstrated only through the use of hand-designed control rules [Fukuda and Ueyama, 1994, Zykov et al., 2005], and only preliminary results have been obtained exploiting evolution to synthesise the control rules that lead to self-replication [Bianco and Nolfi, 2004, Mytilinaios et al., 2004].

In this chapter, we consider the case of a direct encoding from the genotype to the phenotype, which is a neural network with fixed architecture. The sensory-motor configuration of the robot is *a priori* fixed by the experimenter and therefore defines the abilities of the robot to perceive and act in its environment. The genotype-to-phenotype relation consists of a direct mapping in which each free parameter of the network is encoded into a corresponding part (gene) of the genotype. This implies that artificial evolution operates on the parameters that regulate the fine-grained interactions between the robot and the environment, which in turn determine the behaviour exhibited by the robots.

In collective robotics, another characteristics that has to be determined concerns the genetic relatedness between the individuals forming the group, that is, whether they are *genetically homogeneous* (i.e., they are clones) or *heterogeneous* (i.e., they differ from each other). In a homogeneous group, robots are identical in both the controller and the sensory-motor configuration. In this case, robots might assume different roles (when needed) by exploiting (i) situated specialisation (i.e., robots assume and maintain different roles on the basis of the individual sensory-motor experience [Baldassarre et al., 2003]), (ii) internal dynamics (i.e., robots assume different roles on the basis of internal states of the controller resulting from the ability to integrate sensory-motor information over time [Ampatzis et al., 2009]), and (iii) learning (i.e., roles are determined by the online learning ability of the robot [Floreano and Urzelai, 2000]). When dealing with an homogeneous group, the genotype usually encodes the parameters of a single controller, which is copied in all the robots taking part into the experiment. In other words, a single genotype generates the parameters of the controller for the whole group of robots. This also simplifies the fitness assignment problem and eliminates conflicts of interest between genetically different individuals, as discussed in the following section.

The evolution of genetically heterogeneous groups can lead to the differentiation of the behaviour of the individual robots. This is potentially suitable for situations in which the robots forming the group should play well differentiated roles that do not vary over time. In this case, however, the different roles must be somehow encoded into the genotype. The simplest approach consists in *a priori* defining how many roles are necessary (at most, one role per robot in the group), and encoding in a single genotype all the parameters of all controllers. Alternatively, the robots might be genetically identical, but

[1]A different approach consists in evolving computer programs for autonomous robots (see the Genetic Programming literature [Koza, 1992]).

might *express* different parts of their genome [Bongard, 2000]. These approaches allow to evolve tightly cooperating teams, at the cost of substantially increasing the search space for the evolutionary algorithm. In order to reduce the search space, heterogeneous teams can be obtained from controllers evolved in different populations, which are updated in parallel. Each population is therefore dedicated to a specific role, and teams are formed by drawing from the different population with a certain strategy. Eventually, the best individual of each population is the representative of the corresponding role. A similar approach can be instantiated with a single population of genotypes: here, different roles are drawn from the same population. However, in this case, a strong convergence of the population would result in rather homogeneous teams. It would be required to use some technique to maintain enough diversity in the population, which would result in niches adapted to the required roles. In both cases, however, it is challenging to identify an effective way to assign the fitness to the different genotypes forming a team, as discussed below.

2.1.2 Behavioural Selective Pressures: The Fitness Function

The definition of the performance metric that rewards the desired behaviour is usually task-dependent. There are multiple ways to define a fitness function for a given problem. In order to evaluate the quality of the fitness function, Floreano and Urzelai propose the usage of a three-dimensional *fitness space*, in which the different dimensions refer to important features of a fitness function [Floreano and Urzelai, 2000]:

functional vs. behavioural: a functional fitness rewards a particular working modality (i.e., gives an indication on the actuators outputs), while a behavioural fitness measures the quality of the behaviour (i.e., gives an indication about the outcome of a sequence of actions);

external vs. internal: an external fitness is computed through variables that are available to an external observer (i.e., the absolute position of the robot in the environment), while an internal fitness is computed through variables available to the robot (i.e., the sensor readings). While external fitness functions can be easier to deploy, internal ones may reduce possible biases introduced by the designer;

explicit vs. implicit: an explicit fitness measures the way in which a goal is achieved (i.e., the trajectory performed to get close to a light source), and therefore puts constraints on the displayed behaviours. An implicit fitness function, instead, measures the level of attainment of a goal (i.e., how close to the light source the robot ends). An implicit fitness gives more freedom to explore the solution space, therefore allowing to find solutions that are not *a priori* envisioned by the experimenter.

It is worth mentioning that recent approaches in the literature propose the usage of task-independent fitness functions, in which the robots are rewarded using metrics that give no indication about the final behaviour [Sperati et al., 2008, Prokopenko et al., 2006]. The only drive for obtaining a desired behaviours is represented here by ecological selective pressures. This kind of fitness functions can be considered extreme cases that fall into the implicit category.

In a collective robotics setup, the indirect relationship between individual actions and group organisation makes it difficult to devise functional measures. A functional measure, in fact, is directly related to the causes of the observed behaviour, which are *a priori* unknown to the experimenter. Similarly, internal fitness functions may be more difficult to devise, given that they require the evaluation of the group behaviour from the perspective of the individual robots. However, a common approach is to devise an internal fitness function that is measured on each robot taking part to the experiment, obtaining individual fitness values that are aggregated either averaging over the group or selecting the best or the worst performing robot. In this way, it is possible to obtain a group measure starting from internal variables. This can be done only if the individual measure is directly related to the global organisation. Finally, implicit measures should be preferred when the relationship between the individual control rules and the group behaviour is indirect or unknown, as they pose less constraints on the way the desired collective behaviour is achieved. Examples of the fitness functions that follow the above classification are given in Section 3.

The definition of the fitness function is also influenced by the objectives of the experimenter. For instance, consider the case in which a homogeneous group of robots should present a self-organising behaviour: the collective behaviour should be the emergent result of the interactions among the individuals, which coordinate/cooperate to achieve a common goal. In this case, it is useful to evaluate the group level properties through external metrics, rather than looking at the individual actions. It is also useful to evaluate the group organisation (i.e., the spatio-temporal pattern), rather than the way in which the organisation is achieved, resorting to implicit metrics. In fact, external and implicit fitness function pose less constraints on the way in which the problem should be solved. On the contrary, if we consider the case in which an heterogeneous group should display a teamwork with highly specific roles, internal and explicit metrics could be preferred, as they may allow to develop the implementation details of specific solutions beforehand identified.

Some constraints to the fitness function definition are also given by the genotype-phenotype mapping. In particular, the way the group is formed can make it difficult to clearly assign a fitness value to the involved genotypes. This problem affects mainly teams, in which the individual contribution to the overall performance can significantly vary among the group members. It is therefore necessary to define a methodology to fairly estimate the individual contribution of each group member. Without going too deep into the details, it is important here to notice that this problem does not affect homogeneous groups and heterogeneous teams defined by one genotype. In these cases, in fact, the group performance can be directly assigned to the single genotype that encodes all the group controllers. Whenever the group members correspond to different genotypes, it is necessary to deploy a fitness function that directly measures the individual contribution. When this is not possible, the fitness of a single genotype must be evaluated by forming multiple groups, choosing the teammates randomly or with a specific strategy in order to have a good estimate of the individual contribution to the group performance. The latter, however, is a complex and time-consuming procedure, which adds further uncertainty over the estimation of the genotype fitness in varying environmental conditions, as discussed below. It is also worth mentioning that the use of heterogeneous groups constituted by genetically different individuals and the use of fitness functions that estimate the performance at the level of the single individuals tend to cause conflicts of interest between the individuals forming the group, which might prevent the evolution of stable coordinated/cooperative behaviour [Floreano et al., 2007, Mirolli and Parisi, 2008, Waibel et al., 2009].

2.1.3 Ecological Selective Pressures: The Environment Configuration

A typical problem of ER is the correct estimation of the performance of a genotype. The fitness function should evaluate the quality of the robot behaviour with respect to some variability of the environment. Typically, the behaviour must be robust with respect to varying initial position and orientation of the robot, and with respect to other parameters that contribute to define the *ecological niche* in which the behaviour is evolved. A precise computation of the fitness would require testing the behaviour systematically for every possible environmental condition in which the robot may find itself. This is normally not feasible, and therefore it is necessary to sample the space of the possible ecological conditions in an appropriate way, in order to obtain a reasonable fitness estimate. In a collective robotics setup, the problem is worsened by the presence of multiple robots, which increase the variability of the ecological niche. Interaction among individuals, physical interferences and collisions among robots may be very relevant to the accomplishment of the task, requiring the definition of experimental conditions that can let the group experience the interaction patterns relevant for obtaining a robust behaviour.

It is important to notice that indirect selective pressures may be created through the definition of the ecological niche and through the sampling employed to estimate the fitness. Given that the group is evaluated for presenting a robust behaviour within the parameter space of the ecological niche, the choice of the sampling may influence the evolutionary path. For instance, in section 3.3 we show how communication and cooperation emerge solely due to ecological selective pressures, as the fitness function does not contain any indication about cooperative strategies. Thus, the ecological niche and the sampling of the parameter space must be appropriately defined in order to account for robust group behaviour and to take into account implicit selective pressures.

A final issue to consider concerns symmetry breaking, which pertain many collective phenomena. Symmetry breaking refers to the situation in which a system passes from a disordered condition (which is symmetric in the sense that small changes do not change the overall appearance) to a more ordered one, characterised by some structure or pattern. For instance, a group of robots may pass from a disordered (symmetric) condition in which all robots are randomly oriented to a ordered one in which all robots have the same orientation. Symmetric conditions in a collective robotic system must be carefully identified: symmetry breaking may not be possible exploiting the inherent randomness of the robotic system, and therefore suitable behavioural strategies may be required. The evolutionary machinery needs to encounter such conditions often in order to synthesise the collective behaviour necessary to break the symmetry. For this reason, it is necessary to force the system into symmetric conditions, as well as into asymmetric ones, to evolve robust behaviours. An example of systematic testing in symmetric and asymmetric condition for a two-robot system is given in [Ampatzis et al., 2009].

2.2 An Evolutionary Approach to Self-Organising Behaviours

As we have seen above, ensuring the evolvability of the system requires to identify which are the conditions that can lead to the emergence of the desired behaviour. We have pointed to various choices available to the designer, which mainly pertain the experimental setup and the set of selective pressures that influence evolution. Each choice has certain influences over the evolutionary process and the quality of the collective behaviour that is afterwards obtained. In some cases, it is necessary to test different alternatives to obtain the desired system features. Due to the high level of mutual dependency, it is difficult to determine the outcome of a certain choice without reference to the whole picture. Instead of attempting a comprehensive analysis of all the factors that can influence the evolution of collective behaviour, we propose a simple approach that proved particularly suitable for the evolution of simple and robust group behaviours that rely on self-organisation to achieve the collective goal. This approach is based on the following choices:

neural network controllers: the robots' controller is realized through an artificial neural network;

direct genotype-phenotype mapping: the genotype encodes the parameters of a neural network, which takes as input the perceptual information collected through the sensors, and directly controls the actuators of the robot;

homogeneity of the group: all robots share the same sensory-motor apparatus and the same controller. In practice, all robots are identical copies;

minimal individual and communication capabilities: robots cannot rely on particularly powerful sensors, actuators or controllers. Sensors and actuators are noisy and unreliable. Individual abilities are constrained by the possibilities offered by of the neural network controllers and by the resulting sensory-motor coordination. Communication is sub-symbolic and noisy;

fitness function: fitness is behavioural, external and implicit.

The rationale behind the above choices is explained below. First of all, neural networks are capable of generalising to unexperienced conditions [Yao, 1999]. This aspect may play a crucial role especially in collective robotics applications, where it is difficult to predict and experience all possible configurations of the physical and social environment. The direct genotype-phenotype mapping is the simplest, still most common and effective option. To date, alternative methods have not proved more effective, and have not yet converged on a clearly validated methodology [Nolfi and Floreano, 2000]. The use of homogeneous groups leads to various advantages, as argued in the previous sections: it puts less constraints on the solutions that can be obtained through the evolutionary process, allows situated and temporary differentiation of roles, does not incur in credit assignment problems, and avoids conflicts of interest between individuals [Baldassarre et al., 2003, Mirolli and Parisi, 2008]. Minimal control and communication capabilities favour the emergence of collective behaviours that rely on individual sensory-motor coordination and on self-organisation. Typically, similar solutions are not based on complex internal processing capabilities of the individuals robots. Rather, they are the result of numerous interactions among relatively simple individuals. We believe that, by providing minimal complexity at the level of the individual capabilities, the evolutionary process is somehow forced to synthesise solutions based on self-organisation [Trianni, 2008]. Similarly, the usage of behavioural, external and implicit fitness function should maximise the chances to obtain solutions that rely on self-organisation. For this purpose, the fitness function should reward the group organisation (behavioural fitness) by relying on properties observable at the level of the group (external fitness) and related to the attainment of a collective goal (implicit fitness). The absence of *a priori* assumptions on the way the collective goal should be achieved opens the way to the synthesis of self-organising behaviours. Finally, along with the homogeneity of the group, the use of an external fitness creates the conditions for a group selection, which is particularly useful for the emergence of cooperative behaviours [Floreano et al., 2007].

3 Studies on Self-Organising Behaviours

Following the guidelines described above, we have studied various collective behaviours that involve coordination of activities, synchronisation, cooperation and emergent decision making. In this section, we detail the experimental results obtained in three different experimental setups: synchronisation and coordinated group behaviour in Section 3.1, coordinated motion and emergent collective decisions in Section 3.2, and finally cooperative categorisation in Section 3.3. In all cases, we highlight the self-organising features of the evolved systems, and relate the obtained results to the experimental methodology we propose.

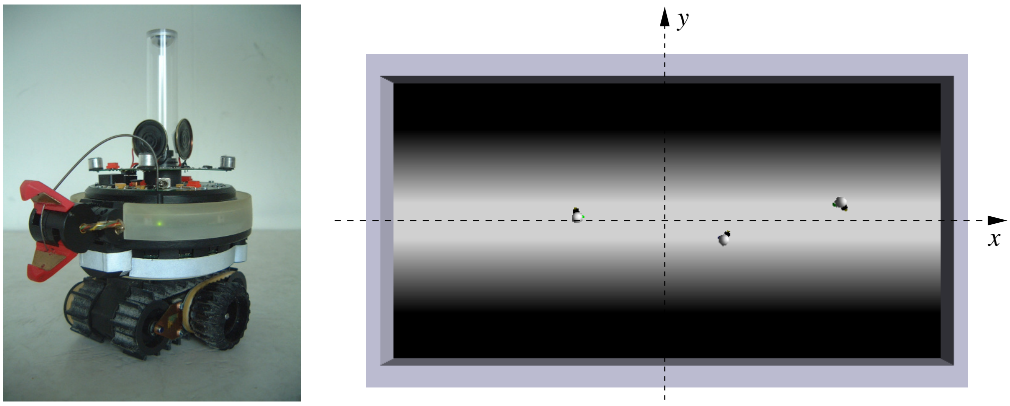


Figure 2: Left: the *s-bot*, the robot used in the synchronisation experiments. Right: snapshot of a simulation showing three robots in the experimental arena. The dashed lines indicate the reference frame used in the experiments.

3.1 Synchronisation

Self-organised synchronisation is a common phenomenon observed in many natural and artificial systems: simple coupling rules at the level of the individual components of the system result in an overall coherent behaviour [Strogatz, 2003]. In this study, we have investigated which are the minimal behavioural and communicative conditions that can lead to synchronisation in a group of robots, in which each individual presents a periodic behaviour. Contrary to models of the self-organised synchronisation observed in some firefly species, we do not postulate the need for internal dynamics [Mirollo and Strogatz, 1990, Wischmann et al., 2006]. Rather, the period and the phase of the individual behaviour are defined by the sensory-motor coordination of the robot, that is, by the dynamical interactions with the environment that result from the robot embodiment. We show that such dynamical interactions can be exploited for self-organised synchronisation, allowing to keep a minimal complexity of both the behavioural and the communication level (for more details, see [Trianni and Nolfi, 2009]).

Experimental setup The experimental scenario defined for the evolution of self-organising synchronisation requires that each robot in the group displays a simple periodic behaviour, which should be entrained with the periodic behaviour of the other robots present in the arena. The individual periodic behaviour consists in oscillations along the y direction of a rectangular arena (see Figure 2). Oscillations are possible through the exploitation of a symmetric gradient in shades of grey painted on the ground, which can be perceived by the robots through the infrared sensors placed under their chassis (ground sensors). The gradient presents a black stripe for $|y| > 1$, in which the robots are not supposed to enter. Collisions with walls or other robots are avoided using the infrared proximity sensors placed around the cylindrical body of the robots. Finally, synchronisation of the movements can be achieved by exploiting a binary communication system: each robot can produce a continuous signal that is perceived by every robot in the arena, including the signalling one. Signals are perceived in a binary way, that is, either there is someone signalling in the arena, or there is no one.

The evolutionary experiments presented in this study are performed in simulation, using a simple kinematic model of the *s-bot* robots [Mondada et al., 2004], and the results are afterwards validated on the physical platform. Artificial evolution is used to set the connection weights and the bias terms of a fully connected, feed forward neural network—a perceptron network. The evolved genotype is mapped into a control structure that is cloned and downloaded onto all the robots taking part in the experiment, therefore obtaining a **homogeneous** group of robots. During evolution, we use groups composed by three robots only. The performance of a genotype is evaluated by a 2-component function: $F = 0.5 \cdot F_{\mathcal{M}} + 0.5 \cdot F_{\mathcal{S}} \in [0, 1]$. The movement component $F_{\mathcal{M}}$ simply rewards robots that move along the y direction within the arena at maximum speed. With respect to the taxonomy introduced in Section 2.1.2, this component is **behavioural**, **external** and **implicit**. In fact, it rewards the movements of the robot from the observer perspective, without explicitly indicating how to perform a periodic behaviour: the oscillatory behaviour derives from the fact that the arena is surrounded by walls, so that oscillations during the whole trial are necessary to maximise $F_{\mathcal{M}}$. The second fitness component $F_{\mathcal{S}}$ rewards synchrony among the robots as the cross-correlation coefficient between the distance of the robots from the x axis. Also this component is **behavioural**, **external** and **implicit**: it is related to the group behaviour, and measures a quantity—the cross-correlation—that is available only to the observer. In addition to the fitness computation described above, two **ecological selective pressures** are present. First of all, a trial is stopped when a robot moves over the black-painted area, and we assign to the trial a performance $F = 0$. In this way, robots are rewarded to exploit the information coming from the ground sensors to perform the individual oscillatory movements. Secondly, a trial is stopped when a robot collides with the walls or with another robot, and also in this case we set $F = 0$. In this way, robots are evolved to efficiently avoid collisions.

Behavioural and scalability analyses We performed 20 evolutionary replications, each starting with a different population of randomly generated genotypes. Each replication produced a successful synchronisation behaviour, in which robots display oscillatory movements along the y direction and synchronise with each other, according to the requirements of the devised fitness function. In general, it is possible to distinguish two phases in the evolved behaviours: an initial transitory phase during which robots achieve synchronisation, and a subsequent synchronised phase. The transitory phase may be characterised by physical interferences between robots due to collision avoidance, if robots are initialised close to each other. The collision avoidance behaviour performed in this condition eventually leads to a separation of the robots in the environment, so that further interferences to the individual oscillations are limited and synchronisation can be achieved. The synchronous phase is characterised by a stable synchronous oscillations of all robots, and small deviations from synchrony are immediately compensated.[2] Each evolved controller produces a signalling behaviour that varies while the robots oscillate. The main role of the evolved signalling behaviour is to provide a coupling between the oscillating robots, in order to achieve synchronisation. In response to a perceived signal, robots react by moving in the environment, changing the trajectory of their oscillations. This results in a modulation of the oscillation amplitude and frequency, which allows the robots to reduce the phase difference among each other, and eventually synchronise (for further details, see [Trianni and Nolfi, 2009]).

Once analysed the synchronisation behaviours evolved using three robots only, we tested their ability to scale up with the group size. To do so, we compared the performance of the evolved behaviour varying the group size. To avoid overcrowding, we performed the scalability analysis in larger arenas, ensuring a constant density of robots across the different settings. We evaluated all best evolved controllers 100 times using six different group sizes (3, 6, 12, 24, 48 and 96 robots). The obtained results are presented in the top part of Figure 3. It is possible to notice that most of the best evolved controllers have a good performance for groups composed of 6 robots. Performance degrades for larger group sizes and only few controllers produce scalable behaviours up to groups formed by 96 robots. The main problem that reduces the scalability of the evolved controllers is

[2] Videos are available at `http://laral.istc.cnr.it/esm/trianni-nolfi-hcr/`.

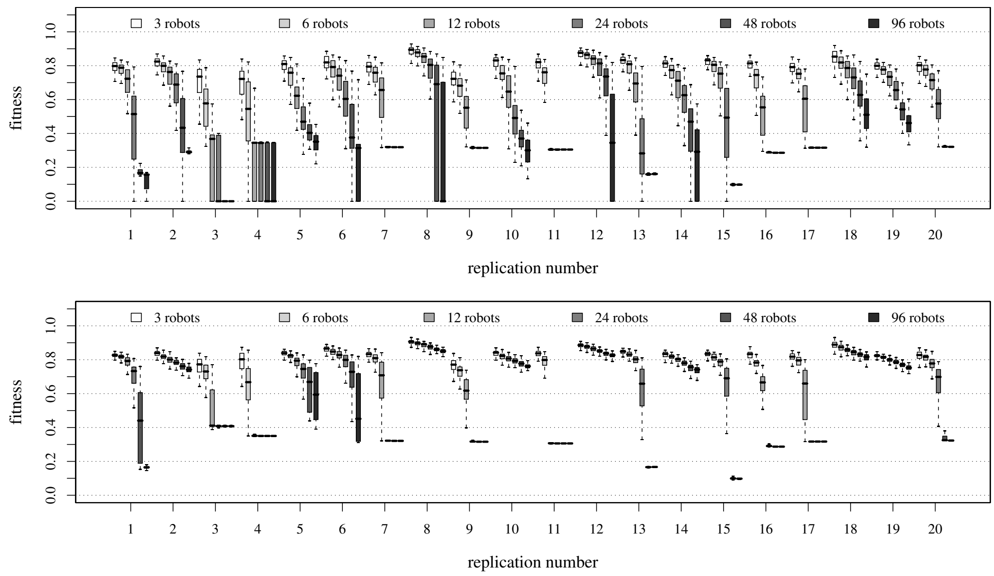


Figure 3: Scalability analysis. The boxplot shows, for each evolved controller, the performance obtained in tests with 3, 6, 12, 24, 48, and 96 robots. Each box represents the inter-quartile range of the data, while the black horizontal line inside the box marks the median value. The whiskers extend to the most extreme data points within 1.5 times the inter-quartile range from the box. Outliers are not shown. Top: scalability of the evolved controllers under normal conditions. Bottom: scalability of the synchronisation mechanism.

given by the physical interactions among robots. Despite the constant initial density we introduced in order to limit the disruptive effect of collision avoidance, physical interactions nevertheless occur with a higher probability per time step, as the group size increases. Every collision avoidance action provokes a temporary de-synchronisation of at least two robots, which have to adjust their movements in order to re-gain synchronous oscillations with other robots. The global and binary communication implies that the whole group is influenced by the attempt of few robots to re-gain synchronisation.

To summarise, the above analysis showed that physical interactions and collision avoidance have a disruptive effect on the synchronisation ability of the robots, and this effect is more and more visible as the group size increases. However, the synchronisation mechanism evolved may scale with the group size if we ignore physical interactions. To test this hypothesis, we performed an identical scalability analysis, but in this case we ignore the physical interactions among the robots, as if each robot was placed in a different arena and perceived the other robots only through sound signals. The obtained results are plotted in the bottom part of Figure 3. Differently from what was observed above, in this case many controllers present perfect scalability, with only a slight decrease in performance due to the longer time required by larger groups to perfectly synchronise. This result confirms the analysis about the negative impact of physical interferences and collisions among robots. In fact, removing the necessity to avoid collisions leads to scalable self-organising behaviours.

Nevertheless, many other controllers present poor scalability properties. It is possible to notice that the performance presents a high variability up to a certain group size. The variable performance indicates that in some cases the robots are able to synchronise, and in other cases not. With larger group sizes, the performance stabilises to a low, constant value, independent from the initial conditions and the number of robots used. This value, which is characteristic of each non-scaling controller, represents the performance of an

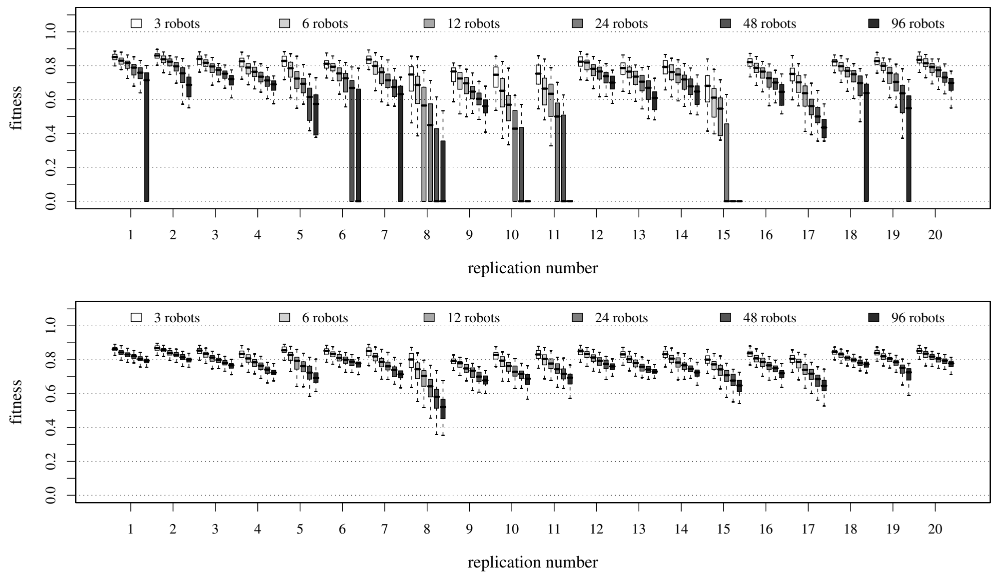


Figure 4: Scalability analysis for the continuous communication system. Top: scalability of the evolved controllers under normal conditions. Bottom: scalability of the synchronisation mechanism.

incoherent attractor for the robotic system. In other words, for every initial condition the robotic system converges into a dynamical condition in which no robot can synchronise with any other. By observing the actual behaviour produced by these controllers, we realised that the incoherent condition is caused by a communicative interference problem: the signals emitted by different robots overlap in time and are perceived as a constant signal (sound signals are global and are perceived in a binary way, preventing a robot from recognising different signal sources). If the perceived signal does not vary in time, it does not bring enough information to be exploited for synchronisation. This problem is the result of the global communication form in which the signal emitted by a robot is perceived by any other robot everywhere in the arena. Moreover, from the robot point of view, there is no difference between a single robot and a thousand signalling at the same time. The lack of locality and of additivity is the main cause of failure for the scalability of the evolved synchronisation mechanisms. However, as we have seen, this problem affects only some of the analysed controllers. In the remaining ones, the evolved communication strategies present an optimal scalability that is only weakly influenced by the group size.

Re-engineering for scalability We identified a cause of the lack of scalability in the communication system, which is neither additive nor local. Given that we are interested here in global synchronisation, we decided to re-engineer our experiments focusing on the additivity of the communication system. We evolved self-organising synchronisation behaviours exploiting exactly the same setup as above, but changing the way robots signal and perceive emitted signals: we change the binary communication system with a continuous one. Now, robots always emit a signal encoding a number in the continuous range [0,1]. The emitted signals are perceived as the average among all the perceived signals. By doing so, the influence of an individual robot on the global perceived signal—which is equal for all robots in the arena—depends on the signalling behaviour of the whole group: the bigger the group, the smaller the influence of the single individual. On the basis of the results obtained so far, we expect that self-organising synchronisation behaviour can be evolved with such a communication system, and that they are more scalable.

Also in this case, we performed 20 evolutionary runs for groups of three robots. All evolutionary runs were successful, and produced synchronisation behaviours that are qualitatively similar to those obtained with the binary communication system: robots perform oscillations over the painted gradient and react to the perceived signal by modifying the individual behaviour, in order to synchronise with other robots. The scalability analysis was performed with the same modalities as described above, and the obtained results are presented in Figure 4. In the above plot, scalability is tested including physical interactions, and also in this case, we notice that collisions prevent the scalability of some controllers. However, it is possible to notice that the usage of an additive communication system leads to better performance even with large groups. In fact, differently from what was observed before, physical interactions and collision avoidance do not have a severe impact on the whole group, as the signals of few non-synchronous robots are averaged with those emitted by the rest of the group. As a consequence, the influence on the group of a single synchronising robot decreases with increasing group size. This leads to an improved group performance.

We also performed a scalability analysis for the evolved synchronisation mechanisms, removing again the physical interactions among robots. The results plotted in the bottom part of Figure 4 show that all evolved synchronisation mechanisms perfectly scale, and they do not suffer from the communicative interference observed with binary signals. In fact, the perceived signal brings information about the average signalling behaviour of all robots. As a consequence, synchronisation is always achieved, no matter the group size. Notice also that all controllers present a linear decrease in performance in correspondence to an exponential growth of the group size. This observation suggests that the self-organising synchronisation mechanism is only slightly affected by the group size.

3.2 Coordinated Motion and Emergent Decisions

The second case study focuses on a particular behaviour, namely *coordinated motion*. In Nature, this behaviour is commonly observed, for instance in flocks of birds or in schools of fish (see [Camazine et al., 2001], chapter 11). We have studied coordinated motion in the particular context of the SWARM-BOTS project,[3] which aimed at the design and implementation of an innovative swarm robotics artifact—the *swarm-bot*—which is composed of a number of independent robotic units—the *s-bots*—that are connected together to form a physical structure (see Figure 5). When assembled together, the *s-bots* must coordinate in order to have an overall coherent motion of the *swarm-bot*. In this case, coordinated motion takes a particular flavour, due to the physical connections among the *s-bots*, which open the way to study novel interaction modalities that can be exploited for coordination [Baldassarre et al., 2007]. Coordinated motion is a basic ability and is essential for an efficient motion of the *swarm-bot* as a whole. It constitutes a basic building block for the design of more complex behavioural strategies, such as collectively moving and avoiding to fall out of the borders of the arena, or decide whether to pass over a gap or not .

Experimental Setup A *swarm-bot* can efficiently move only if the chassis of the assembled *s-bots* have the same orientation. The *s-bots* can independently rotate their chassis, and should prove capable of negotiating a common direction of movement and compensating possible misalignments that occur during motion. Each *s-bot* is provided with a *traction sensor*, which measures the pulling/pushing forces exerted by the robots assembled to its turret. At the beginning of a trial, the *s-bots* start with their chassis oriented in a random direction. Their goal is to choose a common direction of motion on the basis of the only information provided by their traction sensor, and then to move as far as possible from the starting position. The common direction of motion of the group

[3]For more details, see `http://www.swarm-bots.org`.

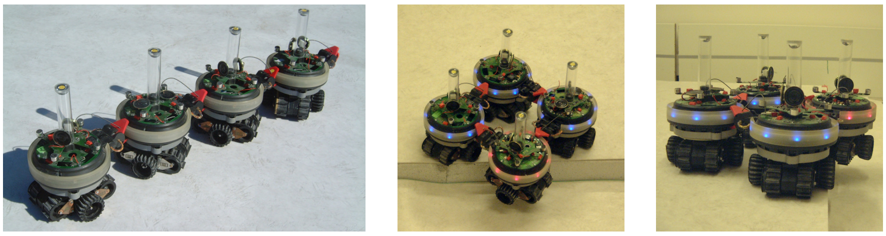


Figure 5: Left: four real *s-bots* forming a linear *swarm-bot*. Centre and Right: four *s-bots*coordinately move and avoid falling. Notice that one *s-bot* is suspended out of the border of the arena, and the physical connections among *s-bots* provide support.

should result from a self-organising process based on local interactions, which are shaped as traction forces. We exploit artificial evolution to synthesise a simple feed-forward neural network that encodes the motor commands in response to the traction force perceived by the robots. The evolutionary algorithm used in this case is identical to the one described above. Also in this case we make use of a **homogeneous** group of robots. The fitness of the genotype is computed as the average distance covered by the group during the trials. This fitness function is again **behavioural**, **external** and **implicit**, as it rewards the group behaviour looking at the final goal, that is, moving as far as possible from the initial position, without explicitly indicating how coordination should be achieved.

Coordinated motion in a *swarm-bot* Using the setup described above, 30 evolutionary runs have been performed in simulation. All the evolutionary runs successfully synthesised controllers that produced coordinated motion in a *swarm-bot*. Direct observation of the evolved strategies shows that at the beginning of each trial the *s-bots* try to pull or push the rest of the group in the direction of motion they are initially placed. This disordered motion results in traction forces that are exploited for coordination: the *s-bots* orient their chassis in the direction of the perceived traction, which roughly corresponds to the average direction of motion of the group. This allows the *s-bots* to rapidly converge toward a common direction and to maintain it.

In order to understand the mechanisms implemented by the evolved controller, we studied the individual behaviour by systematically varying the angle and the intensity of the traction force applied to the turret. We realised that the controller roughly implements two rules: (i) rotate the chassis in the direction of the perceived traction when the traction intensity is high and the traction direction is not aligned with the chassis direction; (ii) keep moving in the current direction when the traction intensity is low. This two rules are sufficient to break the symmetry and to observe a coordinated motion at the level of the group. In fact, they generate a positive feedback loop that allows to amplify initial random fluctuation and to reinforce the choice of a common direction of motion. In fact, at the beginning of each test, all *s-bots* start moving forward in the random direction they were initialised. Being assembled together, they generate traction forces that propagate throughout the physical structure. Each *s-bot* perceives a single traction force, that is, the resultant of all the forces applied to its turret, which roughly indicate the average direction of motion of the group. Following the simple rules described above, an *s-bot* rotates its chassis in order to align to the perceived traction force. In doing so, some *s-bots* will be faster than the others, therefore reinforcing the traction signal in their direction of motion. As a consequence, the other *s-bots* perceive an even stronger traction force, which speeds up the alignment process. Overall, this positive feedback mechanism makes all *s-bots* quickly converge toward a same direction of motion.

The self-organising behaviour described above is very effective and scalable, leading to coordinated motion of *swarm-bots* of different size and shape, despite it was evolved using a specific configuration (i.e., four *s-bots* in linear formation). We have tested the system in simulation using up to 36 robots physically assembled in a square structure, and we observed that coordinated motion would still occur, even though it takes usually longer to achieve coordination. Tests with real robots showed a good performance as well, confirming the robustness of the evolved controller. Overall, the tests with simulated and physical robots prove that the evolved controllers produce a self-organising system able to achieve and maintain coordination among the individual robots [Baldassarre et al., 2007]. The evolved behaviour maintains its properties despite the particular configuration of the *swarm-bot*. It also constitutes an important building block for *swarm-bots* that have to perform more complex tasks such as coordinately moving toward a light target [Baldassarre et al., 2006], and coordinately exploring an environment by avoiding walls and holes [Baldassarre et al., 2006, Trianni and Dorigo, 2006]. In the following, we analyse more in detail the "hole avoidance" extension of the coordinated motion task, and we show how it can lead to emergent collective decisions.

Hole avoidance and emergent collective decisions The "hole avoidance" task is a simple but challenging navigation problem, in which *s-bots* in a *swarm-bot* formation have to explore an arena presenting open borders in which they risk to fall (see Figure 5, centre and right). To do so, the *s-bots* are provided with infrared proximity sensors placed under the chassis of the robot, referred to as ground sensors, which detect the distance of the chassis from the ground. With these sensors, an *s-bot* can detect the empty space beneath whenever it is close to the border of the arena. The controller is a feed-forward neural network that directly connects the traction and ground sensors to the motor outputs. The parameters of the neural controller are evolved with the usual strategy. However, we exploited the knowledge gained evolving a simple coordinated motion in order to devise an **internal** fitness function. In this case, in fact, an external fitness would be complex to devise, as it is difficult to evaluate the avoidance behaviour without being too explicit about how falling should be avoided. We therefore devised a fitness function that rewards straight and fast motion of the *s-bots*, looking at the wheels' speed, and penalises the *s-bots* that do not coordinate their movements with the group or that spend too much time in the vicinity of the arena border. This last component is computed simply looking at the activation of the traction and the ground sensors: we minimise the perceived traction force—which implicitly corresponds to groups that move coordinately—and require that the ground sensors are always activated—which implicitly corresponds to robots that move far from the borders of the arena. We aggregate at the group level the values internally computed on each *s-bot* by selecting the minimal one. This ensures that the group performance is conservatively estimated. Overall, the fitness function is **behavioural**, **internal** and **implicit**. Additionally, we exploit an **ecological selective pressure** by penalising those cases in which the *swarm-bot* falls (for more details, see [Trianni and Dorigo, 2006]).

The behaviours produced by the evolved neural networks are characterised by an initial coordination phase that leads to a coherent motion of the *swarm-bot*, in a very similar way to the simple coordinated motion case. The *swarm-bot* can therefore move coordinately into the arena exploiting the information coming from the traction sensor. When close to the border of the arena, an *s-bot* can detect the edge through the ground sensors, and reacts by rotating the chassis and changing its direction of motion. This change in direction produces a traction force for the other *s-bots*, which triggers a new coordination phase that continues until the *s-bots* eventually choose a new direction of motion, leading the *swarm-bot* away from the arena border. In some cases, the reaction of a single *s-bot* may not be sufficient to influence the behaviour of the rest of the group. As a consequence, the *s-bot* may be pushed out of the arena. However, physical connections serve as support for this *s-bot*, while the rest of the group continues to perform hole avoidance and eventually leads the whole *swarm-bot* to a safer location.

This behaviour is mainly based on the properties of the traction sensor, which allows the *swarm-bot* to exploit the direct interactions among *s-bots*—shaped as traction forces—to communicate the presence of a hazard—the hole to be avoided [Trianni and Dorigo, 2006]. Traction forces are also at the basis of the self-organising process that leads to the collective decision about passing over a trough or avoiding it when it is too wide. Intuitively, if a through is small enough to be bridged, the *swarm-bot* could pass over it exploiting the physical connections among *s-bots*. However, a mechanism is necessary to estimate the width of the through and trigger an avoidance or a passing-over behaviour. Such an estimation can be collectively performed—and a decision collectively taken—by the *s-bots* forming the *swarm-bot*, without making use of the individual perception of the trough (e.g., by means of their camera or ground sensors), which would anyway be very limited. We designed a set of experiments in order to test the ability of a *swarm-bot* to bridge a gap of varying size. This test is intended to demonstrate how the simple controllers developed for hole avoidance generalise to a collective decision-making mechanism for discriminating between situations that can be faced by a *swarm-bot* from situations that could be too hazardous even for a large connected structure.

The *swarm-bot* is placed in an arena divided by a trough (see Figure 6). We test *swarm-bots* of different size—4, 9, and 16 *s-bots* connected in a square formation—that have to confront with a trough of width varying from 2 to 30 cm. We performed 100 evaluation trials per experimental setup, systematically varying the *swarm-bot* size and the trough width—i.e., 100 trials for each size/width pair. The results of this analysis are plotted in Figure 7. The plot shows, for each trough width, the performance of the three studied *swarm-bots*. We count the number of trials in which the *swarm-bot* successfully bridges the gap and passes on the other side. We also count the number of errors, that is, trials in which the *swarm-bot* falls into the troughs or remains stuck: even if the gap is bridged, the *swarm-bot* may not be able to efficiently coordinate in order to pass on the other side. In fact, once the gap is encountered and bridged by some of the *s-bots*, a new coordination phase is triggered which generally leads to the choice of a new direction of motion, that may let the *swarm-bot* retrace its steps. Furthermore, the coordination phase over the trough is time-consuming, and the *swarm-bot* may not be able to completely pass over the trough in the limited available time.

From the results shown in Figure 7 it is possible to notice how the success rate generally decreases as the width of the gap increases. Up to a certain width, the *swarm-bot*

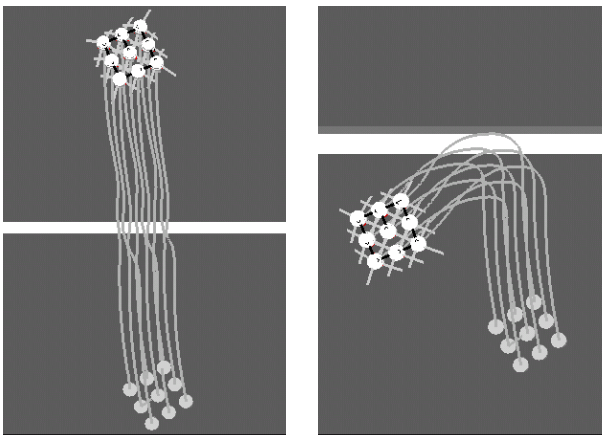


Figure 6: Trajectories drawn by a *swarm-bot* composed of 9 *s-bots* in a square formation. Left: the *swarm-bot* is able to pass over a 10 cm wide trough. Right: the *swarm-bot* avoids a 20 cm wide trough, which could be too large to be bridged.

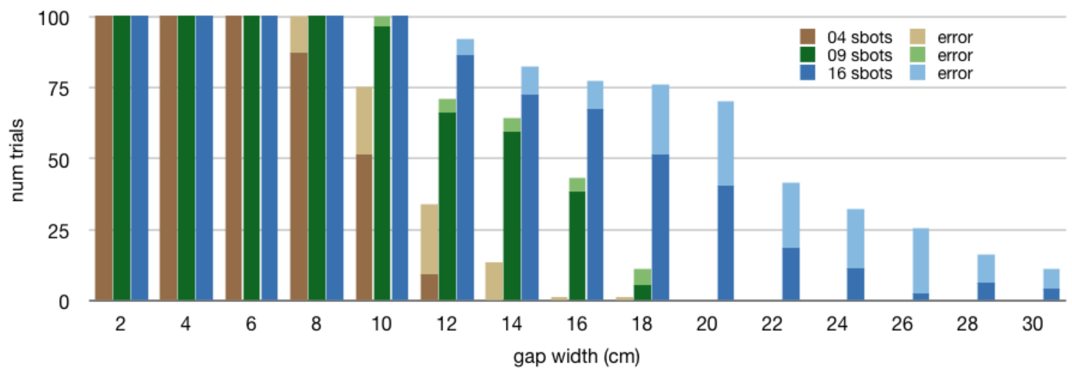


Figure 7: Performance of a *swarm-bot* passing over a trough. The stacked boxes represent the number of trials in which the *swarm-bot* manages to pass over the trough, and the number of trials in which an error occurs (i.e., the *swarm-bot* falls or remains stuck over the trough).

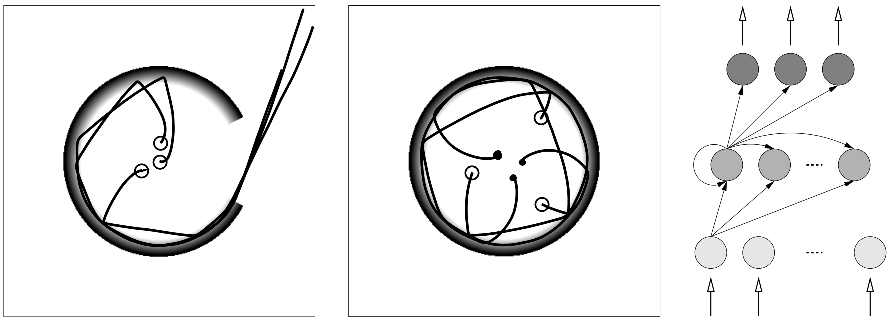
systematically passes over the through. This width depends on the *swarm-bot*'s size: the bigger the *swarm-bot*, the larger the gap that can be passed. For larger sizes, a transition can be observed in which the *swarm-bot* stops passing over the trough systematically and sometimes avoids it. Finally, for very large troughs the avoidance behaviour is usually preferred. The behaviour presented above can be considered conservative, as the avoidance is in general preferred to the passing over the trough. This is not surprising because the behaviour was evolved explicitly for the hole avoidance task. Therefore, a trough can be estimated too large to be bridged even when the *swarm-bot* is big enough to pass over it. However, looking at the success rate shown in Figure 7, we can notice that the *swarm-bots* perform reasonably well with respect to their physical constraints. In fact, given the size of a 4-individual *swarm-bot*, the maximum width of a trough that can be bridged is about 12 cm. Our results show that from this width on, the *swarm-bot* always performs an avoidance action, while the *swarm-bot* is able to pass over narrower troughs, even if not systematically. A similar situation can be observed for the case of 9 and 16 *s-bots*, which are respectively characterised by the maximum width of 18 and 30 cm.

Whether a trough is avoided or bridged depends on multiple factors, among which the orientation of the *swarm-bot* and its direction of motion when it first approaches the trough. In fact, the collective behaviour of passing over a trough relies on a delicate balance between the forces exerted by the *s-bots* that touch the ground and the missing influence of those *s-bots* that are suspended over the gap. The size of the *swarm-bot* also matters, as it has a bearing on the inertia of the whole group: the bigger the size of the *swarm-bot*, the bigger the inertia of the physical structure. Once the *swarm-bot* reaches an edge, its inertia will cause some *s-bots* to be pushed out, over the gap. In fact, few *s-bots* have a small effect on the overall behaviour of the group. When a sufficient number of *s-bots* is suspended out of the arena, the forces exerted by those *s-bots* that reach the edge can be perceived by the whole group, and they will trigger a change in the direction of motion of the *swarm-bot* in order to avoid falling. If some of the suspended *s-bots* reach the other side of the trough, they start again to have an influence on the rest of the group. First, they align with the current direction of motion, and afterwards they contribute to the gap passing behaviour pulling the whole structure on the other side of the gap. This emergent behaviour can be considered self-organised, as it depends on the interactions among individuals and on clear feedback loops: the conformist tendency of the *s-bots* in following the average direction of the group constitutes a positive feedback, while the tendency to avoid falling of individual *s-bots* and the missing influence of the suspended *s-bots* constitute the negative feedback. In conclusion, the collective behaviour of passing over a trough relies on the emergent decision-making mechanism that allows a *swarm-bot* to discriminate between those troughs that are small enough to be safely bridged and those that are not. In other words, through a self-organising process, the *swarm-bot* is able to collectively estimate the width of the trough, and consequently it is able to take the correct decision about the way to move.

3.3 Adaptation of Communication, Coordination and Categorisation

In the previous case studies, we have observed how artificial evolution can synthesise efficient self-organising behaviours that result from simple reactive controllers. In this section, we show how various complexity levels can be added to the basic system described above in order to evolve cooperative, cognitive behaviours in a collective system. Above all, by providing individual robots with more complex control and communication abilities, it is possible to obtain group behaviours that can rely on both the individual and the group dynamics. For instance, the ability to integrate information over time can provide robots with an excellent mean to balance individual with group abilities. In a evolutionary perspective, this can result in complex forms of cooperation particularly adapted to the experimental scenario. In fact, the actions of each robot are influenced by—and can influence themselves—the status of the other robots, which try to make their own decisions at the same time. This opens the way to cooperative solutions based on communication, which makes it possible to exploit not only the dynamical interactions among individuals, but also the way in which these interactions change over time. In this study, we demonstrate how a number of different strategies can be evolved displaying non-trivial individual and collective decision making. Moreover, we show that those solutions that exploit communication perform better, systematically achieving a consensus in the group and reducing the decision errors.

Experimental setup The task we study consists in a binary decision to be performed by three simulated robots, which have to recognise whether the arena they are placed in presents an opening or not. The arena is delimited by a circular band in shades of grey painted on the ground, which simulates some obstacles that the robots cannot overcome individually (see Figure 8a,b). The arena may present a *way out*, that is, a passage through which a solitary *s-bot* can exit (see Figure 8a). However, an *s-bot* does not have the perceptual abilities to detect the *way out* from every location in the arena: in fact, the grey level of the circular band can be perceived by the *s-bots* only locally through their ground sensors. Therefore, robots should first search for the *way out* and, if they do not find any as in Figure 8b, they should aggregate in one place. In short, we consider here the decision problem of switching from the individual behaviour of searching for the *way out* to the collective behaviour of aggregating in one place. *S-bots* can exploit an omnidirectional camera to perceive the other robots in their vicinity. Moreover, robots are provided with a global, binary communication system, like the one for the synchronisation experiments presented in Section 3.1. Each robot is controlled by a continuous time recurrent neural network (CTRNN, see [Beer, 1995]) with a multi-layer topology, shown in Fig. 8c. Four inputs take values from the camera, four from the ground sensors and one from sound perception, while two outputs control the wheels and one controls the sound signal. Moreover, the network is provided with a 5-neuron continuous time recurrent hidden layer. The weights of the synaptic connections between neurons, the bias terms and the decay constants of the hidden neurons are genetically encoded parameters. *S-bots* are rewarded to search and pass through the *way out* when placed in environment *A*, and to aggregate when they are placed in environment *B*. In this case we use a **behavioural**, **external** and **implicit** fitness function. However, we explicitly reward a different behaviour to be



(a) (b) (c)

Figure 8: (a,b) The experimental arena contains a circular band in shades of grey, which may or may not have the *way out*. Dark lines represent the trajectory of three robots, and the starting position on the trajectories is indicated by empty circles. (c) The three-layer architecture of the neural controller. The hidden layer is composed of continuous time neurons with fully recurrent connections.

performed when robots are placed in environment A and in environment B. For more details, see [Trianni et al., 2007]. The experiment was run in simulation.

Results We performed 20 replications of the experiment, most of which were successful. We selected the best controllers from the last generation of each evolutionary run—hereafter referred to as $C_1, ..., C_{20}$—which were evaluated for 2000 trials, half in environment A and half in environment B. The obtained results are summarised in Table 1: in both environments, we computed the average performance and its standard deviation (avg ± std), the rates of success %S (all robots achieve the desired distance D_O), failure %F (no robot achieves the desired distance D_O), partial success/failure %M (not all robots are successful or fail) and error %E (robots collide or cross the black edge of the circular band). In each trial, we also computed the *coverage*, which is defined as the percentage of the circular band that each robot covers in average during a trial: a value smaller than 1 indicates that the single robot does not search the whole circular band for the *way out*, while a value bigger than 1 indicates that the single robot performs more than one tour (see Fig. 9). The coverage—together with the success rate—is useful to quantitatively assess the quality of the evolved strategies.

Successful controllers produce good search behaviours when robots are in environment A: robots avoid collisions and move away from the centre of the arena. Once on the circular band, robots start looping in search of the *way out*, which is eventually found and traversed. On the contrary, if robots are placed in environment B, the absence of the *way out* is recognised by the robots through the integration over time of their perceptual flow, which includes the signals that the robots may emit. As a consequence, a behavioural transition can be observed from a searching behaviour (state $\mathcal{S}$) to an aggregation behaviour (state $\mathcal{C}$). The modalities with which the transition is performed significantly vary across the different solutions synthesised during different evolutionary runs.[4] However, looking at the behaviour produced by the evolved controllers, we recognised some similarities that let us classify the controllers in 4 classes.

Class $\mathbf{U} = \{C_4, C_6, C_{14}, C_{17}\}$ encompasses the "unsuccessful" controllers, that is, those controllers that solve the task only in part. These controllers generally produce appropriate search behaviours when robots are in environment A, as confirmed by the good performance and the high success rate (see Table 1). However, when robots are placed in environment B they fail in systematically aggregating, scoring a low performance and

[4]Videos are available at `http://laral.istc.cnr.it/esm/trianni-nolfi-hcr/`.

Table 1: Post-evaluation results. See text for details.

		environment A					environment B				
		avg ± std	%S	%M	%F	%E	avg ± std	%S	%M	%F	%E
U	c_4	0.82 ± 0.14	92.0	6.5	1.0	0.5	0.37 ± 0.11	19.4	18.9	61.7	0.0
	c_6	0.85 ± 0.06	98.6	1.2	0.0	0.2	0.31 ± 0.08	0.9	30.6	68.4	0.1
	c_{14}	0.83 ± 0.15	91.3	6.2	0.0	2.5	0.46 ± 0.15	2.5	65.1	24.0	8.4
	c_{17}	0.66 ± 0.07	74.3	25.4	0.1	0.2	0.39 ± 0.08	4.9	78.8	16.3	0.0
B	c_1	0.86 ± 0.11	97.7	0.8	0.0	1.5	0.69 ± 0.07	95.9	2.8	1.3	0.0
	c_5	0.85 ± 0.13	92.1	5.7	0.0	2.2	0.57 ± 0.14	66.8	16.9	16.1	0.2
	c_8	0.83 ± 0.15	90.3	7.6	0.4	1.7	0.57 ± 0.12	34.3	55.2	9.2	1.3
	c_{10}	0.88 ± 0.07	99.0	0.6	0.0	0.4	0.66 ± 0.07	94.1	2.1	3.7	0.1
	c_{16}	0.85 ± 0.14	94.4	4.1	0.0	1.5	0.74 ± 0.13	94.1	2.3	1.4	2.2
M	c_3	0.83 ± 0.15	85.8	11.7	0.0	2.5	0.63 ± 0.09	87.6	8.1	3.4	0.9
	c_7	0.79 ± 0.20	89.3	5.5	0.0	5.2	0.62 ± 0.25	49.5	34.2	10.5	5.8
	c_{11}	0.86 ± 0.07	98.9	0.6	0.0	0.5	0.61 ± 0.07	87.6	9.5	2.7	0.2
	c_{13}	0.85 ± 0.09	94.3	5.2	0.0	0.5	0.62 ± 0.07	93.0	5.3	0.8	0.9
	c_{19}	0.81 ± 0.15	94.8	2.3	0.6	2.3	0.67 ± 0.12	91.7	3.8	1.9	2.6
	c_{20}	0.87 ± 0.06	99.6	0.0	0.0	0.4	0.59 ± 0.07	79.3	11.3	9.3	0.1
C	c_2	0.86 ± 0.10	98.6	0.1	0.0	1.3	0.82 ± 0.12	97.1	0.4	0.9	1.6
	c_9	0.87 ± 0.08	99.2	0.0	0.0	0.8	0.78 ± 0.12	88.1	8.3	3.1	0.5
	c_{12}	0.87 ± 0.05	99.6	0.3	0.0	0.1	0.74 ± 0.11	87.8	6.4	5.4	0.4
	c_{15}	0.86 ± 0.08	99.3	0.0	0.0	0.7	0.78 ± 0.13	96.6	0.4	0.6	2.4
	c_{18}	0.84 ± 0.18	95.8	0.0	0.0	4.2	0.83 ± 0.17	95.3	0.3	1.0	3.4

a poor success rate. The second class $\mathbf{B} = \{C_1, C_5, C_8, C_{10}, C_{16}\}$ consists of controllers that produce a strategy named "bouncing" after the aggregation behaviour of the robots in state $\mathcal{C}$: robots search for each other by continuously bouncing off the circular band, so that they sooner or later meet and remain close. Communication is generally not exploited, and consequently each robot individually switches from state $\mathcal{S}$ to state $\mathcal{C}$, without any reference to the state of the other robots. The bouncing behaviour is resilient to possible individual failures in environment A: by bouncing off the circular band, robots can continue searching for the *way out*, even if less efficiently. The third class $\mathbf{M} = \{C_3, C_7, C_{11}, C_{13}, C_{19}, C_{20}\}$ encompasses controllers that produce a strategy named "meeting", due to the fact that robots aggregate by encountering at a meeting point, which is normally close to the centre of the arena. Except for C_7 and C_{19}, controllers of this class do not make use of communication. The main difference with class **B** controllers resides in the aggregation behaviour, which lets robots leave the band and move in circles close to the centre of the arena, waiting for the other robots to reach a similar position. This behaviour is not robust with respect to possible decision errors in environment A. As a consequence, evolution shaped the controllers of this class to be characterised by a higher coverage (see Fig. 9), which suggests that robots perform in average more than one loop over the circular band before switching to state $\mathcal{C}$. The last class $\mathbf{C} = \{C_2, C_9, C_{12}, C_{15}, C_{18}\}$ is named "cooperative" because it encompasses controllers that produce communicative behaviours exploited for cooperation in the decision making. In fact, robots are able to share the information they collect over time through their signalling behaviour. The robots initially emit a sound signal, and they stop only after looping on the circular band for some time. If any robot finds the *way out*, signalling continues, inducing all other robots to remain in state $\mathcal{S}$ and to keep searching for the *way out*. This leads to a high success rate in environment A, and no complete failures are observed (see Table 1). When the *way out* is not present, all robots eventually stop signalling, allowing the transition to state $\mathcal{C}$ and

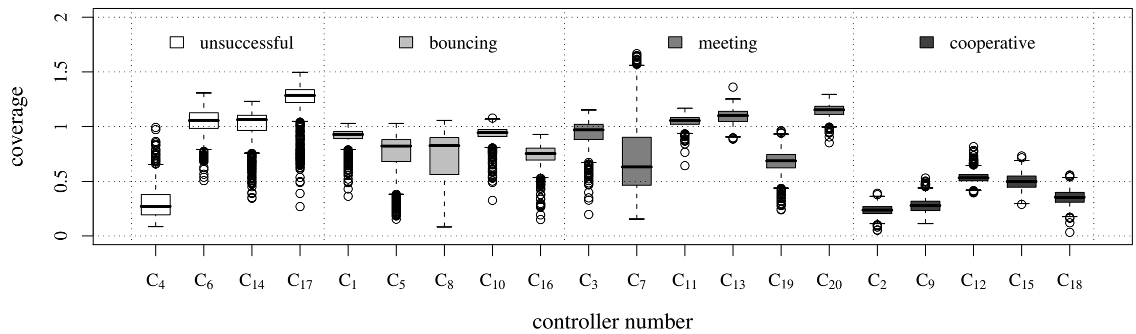


Figure 9: The *coverage* of the evolved controllers. Boxes represent the inter-quartile range of the data, while the horizontal lines inside the boxes mark the median values. The whiskers extend to the most extreme data points within 1.5 times the inter-quartile range from the box. The empty circles mark the outliers.

triggering the aggregation behaviour. By sharing the information through communication, robots can collectively search the circular band, splitting the task among them: as shown by the coverage data in Fig. 9, each robot covers from a quarter to half circle when placed in environment B. This allows to consistently reduce the search time, achieving high performance and high success rates. Communication is fundamental here, because it provides robustness to the decision making process and it makes the system more efficient by reducing the time necessary to take the decisions to switch from solitary to collective behaviours.

It is important to note here the interaction between individual sensory-motor coordination, individual categorisation and communication. By contemporaneously evolving these three features, it is possible to observe the interplay of different selective pressures that shape the individual and the collective response. The first selective pressure is related to the searching behaviour, which allows to develop individual sensory-motor coordination necessary to navigate in the environment and efficiently find the *way out* in environment A. Some evolutionary runs remain stuck at this level (i.e., those that produce controllers within class **U**), as there is no observable behavioural transition from state $\mathcal{S}$ to state $\mathcal{C}$ for the individual robots. The second selective pressure that comes into play is the necessity to individually categorise the environment, therefore integrating the perceptual flow over time in order to recognise that there is no *way out* in environment B (e.g., most of the controllers belonging to class **B** and class **M**). Finally, the **ecological selective pressures**—given by the limited time available in each trial—have an influence on the efficiency of the categorisation process: groups that categorise the environment quickly have more time to accomplish the task. Here, communication comes into play: the strategy exploited by class **C** controllers is not efficient *per se*, but it is efficient as soon as it allows to reduce the individual coverage, as this leaves more time to the group to aggregate in environment B. It results that communication is initially neutral for the task, as it does not give a selective advantage. However, once a signalling mechanism is in place (e.g., signalling when the *way out* is found), it is exploited by evolution for refining both the transition from state $\mathcal{S}$ to state $\mathcal{C}$, which is performed only when there is no signalling robot, and the individual coverage, which is reduced from generation to generation to increase the efficiency of the overall behaviour. These complex evolutionary dynamics are observable in this setup thanks to the dynamical properties of the individual controllers that are able to integrate information over time. Evolution can act on additional free parameters, that is, the time constants of the CTRNN that define the leaky integration abilities of the controller. By acting on both individual and collective dynamics, complex solutions like the one observed in class **C** controllers cab be synthesised.

4 Conclusions

In this chapter, we have shown how artificial evolution applied to collective robotics can produce coordinated and cooperative behaviours. We have described the main methodological choices that need to be performed when setting up an evolutionary experiment, and we have proposed a particular technique that proved successful for the evolution of self-organising behaviours. In the studies presented in this chapter, self-organisation is the result of simple individual behaviours and simple interactions among robots, both shaped by evolution in order to achieve and support the group organisation. Such self-organising behaviours present interesting generalisation abilities, above all when they exploit feedback loops given by the physical interactions among the robots and between the robots and the environment, as presented in Section 3.2.

Given that the evolutionary machinery just works on the parameters of the individual controller, it is of fundamental importance the attentive definition of the ecological conditions in which evolution is carried out. In particular, the definition of suitable communication modalities can make the difference. In fact, contrary to physical constraints that cannot be modified at will (e.g., friction or gravity), it is usually possible to define the communication protocol—i.e., the way in which signals are emitted and perceived—in particular when communication is implemented through sound, light or other wireless signalling technologies. However, this freedom should be suitably managed: ER favours simple sub-symbolic communication forms, and aims at contextually developing the behavioural and communication strategies, which can co-evolve as a single whole. For instance, in the experiment presented in Section 3.1, we observed that a global binary signal is sufficient for synchronisation, even though it does not carry explicit information about the position of the signalling robot in the arena. In this case, communicative and non communicative behaviours co-evolve and adapt one to the other, exploiting the fine-grained interactions between the robots and the physical and social environment. In Section 3.3, we have also shown that co-evolution opens the way to communication forms that are tightly linked with the sensory-motor coordination of the robots and their individual cognitive abilities (e.g., integration over time of perceptual information for decision making). Finally, we have observed how changing the communication protocol can have a strong impact on the properties of the group behaviour: the additive communication exploited for promoting scalability in Section 3.1 does not require additional complexity at the level of the individual behaviour, but helps in providing a more robust and scalable synchronisation mechanism.

The study of scalability of the synchronisation behaviour also demonstrated that it is possible to engineer some features of a system undergoing artificial evolution on the basis of the outcomes of the evolutionary process itself. We showed that an attentive analysis of negative results conveys knowledge on how to modify the characteristics of the system that are designed by the experimenter and are not varied during the evolutionary process so to allow evolution to find better solutions. We believe that this result could be generalised towards an engineering approach to ER, which can provide guidelines for the design of evolutionary experiments. This is particularly relevant for collective and swarm robotics, in which the desired behaviour of the group is an indirect result of the control and communication rules followed by each individual.

An engineering approach to ER may help also in overcoming the current limitations of the approach. Currently, the main problem is scaling in complexity beyond simple and idealised scenarios toward real world problems. This is the grand challenge for ER in the future. There are two possible directions, in our view: on the one hand, more complex behaviours can be evolved by providing more capabilities and more structure to the individual controllers. In this case, complex individual behaviours support the cooperation between individuals, for instance, through the development of a cooperative language that can help regulating the inter-individual interactions [De Greef and Nolfi, 2009]. We believe that another, very promising and yet-to-be-explored direction should fully rely on self-organisation for producing distributed, cognitive robotic systems. That is, the capabilities of the individual robot should remain relatively simple, but the group should display cognitive abilities, such as decision-making, categorisation or attention, as the result of the numerous interactions among the individuals. Moreover, by evolving swarm robotic systems that display cognitive processes, it could be possible to shed light on the distributed mechanisms that support cognition in collectives. Current trends in the scientific community recognise in the study of collective behaviours the possibility to identify the distributed mechanisms underlying certain cognitive processes such as decision-making or attention (see [Couzin, 2009, Goldstone and Gureckis, 2009, Marshall and Franks, 2009]). These studies claim that, at a certain level of description, operational principles used to account for the behaviour of natural swarms may turn out to be extremely powerful tools to identify the neuroscientific basis of cognition (i.e., the explanatory principles). Both the above scientific and the technological drives led to the introduction of *Swarm Cognition* as a novel approach to the study of cognitive processes emerging from the interaction of low-level cognitive units, be they natural or artificial [Trianni and Tuci, 2009]. In this framework, evolutionary swarm robotics allows to explore in a synthetic setup the relationship between embodied cognition and information processing: a swarm robotic system merges these two aspects within the numerous interactions among the system components, which all together perform cognitive processing in continuous interaction with the environment. It is therefore interesting to identify which are the components of the collective cognitive process that are directly related to the embodiment of the robots, and which are the components that are instantiated in the interactions among robots.

References

C. Ampatzis, E. Tuci, V. Trianni, A. L. Christensen, and M. Dorigo. Evolving self-assembly in autonomous homogeneous robots: Experiments with two physical robots. *Artificial Life*, 15(4):465–484, 2009.

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

G. Baldassarre, V. Trianni, M. Bonani, F. Mondada, M. Dorigo, and S. Nolfi. Self-organised coordinated motion in groups of physically connected robots. *IEEE Transactions on Systems, Man and Cybernetics - Part B: Cybernetics*, 37(1):224–239, 2007.

Gianluca Baldassarre, Domenico Parisi, and Stefano Nolfi. Distributed Coordination of Simulated Robots Based on Self-Organization. *Artificial Life*, 12(3):289–311, July 2006.

R. D. Beer. A dynamical systems perspective on agent-environment interaction. *Art. Intell.*, 72:173–215, 1995.

R. Bianco and S. Nolfi. Toward open-ended evolutionary robotics: evolving elementary robotic units able to self-assemble and self-reproduce. *Connection Science*, 4:227–248, 2004.

J. C. Bongard. The legion system: A novel approach to evolving heterogeneity for collective problem solving. In R. Poli, W. Banzhaf, W. Langdon, K. Miller, P. Nordin, and T. Fogarty, editors, *Genetic Programming*. Springer Verlag, Berling, Germany, 2000.

S Camazine, Jean-Louis Deneubourg, N Franks, J Sneyd, and G Theraulaz. *Self-Organization in Biological Systems*. Princeton University Press, Princeton, NJ, 2001.

I.D. Couzin. Collective cognition in animal groups. *Trends in Cognitive Sciences*, 13(1): 36–43, 2009.

J. De Greef and S. Nolfi. Evolution of implicit and explicit communication in a group of mobile robots. In S. Nolfi and M. Mirolli, editors, *Evolution of Communication and Language in Embodied Agents*. Springer Verlag, Berling, Germany, 2009.

D. Floreano and J. Urzelai. Evolutionary robots with on-line self-organization and behavioral fitness. *Neural Networks*, 13(4–5):431–443, 2000.

D. Floreano, S. Mitri, S. Magnenat, and L. Keller. Evolutionary conditions for the emergence of communication in robots. *Current Biology*, 17(9):514–519, 2007.

L. J. Fogel, A. J. Owens, and M. J. Walsh. *Artificial Intelligence through Simulated Evolution*. John Wiley & Sons, New York, NY, 1966.

T. Fukuda and T. Ueyama. *Cellular Robotics and Micro Robotics Systems*. World Scientific, Singapore, 1994.

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

R. L. Goldstone and T. M. Gureckis. Collective behaviour. *Trends in Cognitive Science*, 1(3):412–438, 2009.

I. Harvey, E. A. Di Paolo, R. Wood, M. Quinn, and E. Tuci. Evolutionary robotics: A new scientific tool for studying cognition. *Artificial Life*, 11(1–2):79–98, 2005.

J. H. Holland. *Adaptation in natural and artificial systems*. University of Michigan Press, Ann Arbor, MI, 1975.

J.R. Koza. *Genetic Programming*. MIT Press, Cambridge, MA, 1992.

H. Lipson and J.B. Pollack. Automatic design and manufacture of robotic lifeforms. *Nature*, 406:974–978, 2000.

Max Lungarella, Giorgio Metta, Rolf Pfeifer, and Giulio Sandini. Developmental robotics: a survey. *Connection Science*, 15(4):151–190, 2003.

J. A. R. Marshall and N. R. Franks. Colony-level cognition. *Current Biology*, 19(10): 395–396, 2009.

M. Mirolli and D. Parisi. How producer biases can favor the evolution of communication: An analysis of evolutionary dynamics. *Adaptive Behavior*, 16(1):27–52, 2008.

R. E. Mirollo and S. H. Strogatz. Synchronization of pulse-coupled biological oscillators. *SIAM Journal on Applied Mathematics*, 50(6):1645–1662, 1990.

Francesco Mondada, Giovanni C Pettinaro, Andre Guignard, Ivo W Kwee, Dario Floreano, Jean-Louis Deneubourg, Stefano Nolfi, Luca Maria Gambardella, and Marco Dorigo. SWARM-BOT: A new distributed robotic concept. *Autonomous Robots*, 17(2):193–221, 2004.

E. Mytilinaios, M. Desnoyer, D. Marcus, and H. Lipson. Designed and evolvable blueprints for physical self-replicating machines. In J. Pollak, M. Bedeau, P. Husband, J. Ikegami, and R. A. Watson, editors, *Artificial Life IX. Proeceeding of the Ninth International Conference on the Simulation and Synthesis of Living Systems*, pages 15–20. MIT Press, Cambridge, MA, 2004.

S. Nolfi. Emergence of communication in embodied agents: Co-adapting communicative and non-communicative behaviours. *Connection Science*, 17(3–4):231–248, 2005.

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

M. Prokopenko, V. Gerasimov, and I. Tanev. Evolving spatiotemporal coordination in a modular robotic system. In S. Nolfi, G. Baldassarre, R. Calabretta, J.C.T. Hallam, D. Marocco, J.A. Meyer, O. Miglino, and D. Parisi, editors, *From Animals to Animats 9: 9th International Conference on the Simulation of Adaptive Behavior (SAB 2006)*, pages 558–569. Springer Verlag, Berlin, Germany, 2006.

H.-P. Schwefel. *Numerical Optimization of Computer Models.* John Wiley & Sons, New York, NY, 1981.

K. Sims. Evolving 3d morphology and behavior by competition. In R.A. Brooks and P. Maes, editors, *Artificial Life IV*, pages 28–39. MIT Press, Cambridge, MA, 1994.

V. Sperati, V. Trianni, and S. Nolfi. Evolving coordinated group behaviours through maximization of mean mutual information. *Swarm Intelligence*, 2(2–4):73–95, 2008.

K.O. Stanley and R. Miikkulainen. Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10(2):99–127, 2002.

S. H. Strogatz. *Sync: The Emerging Science of Spontaneous Order*. Hyperion Press, New York, NY, 2003.

V. Trianni. *Evolutionary Swarm Robotics. Evolving Self-Organising Behaviours in Groups of Autonomous Robots*, volume 108 of *Studies in Computational Intelligence.* Springer Verlag, Berlin, Germany, 2008.

V. Trianni and M. Dorigo. Self-organisation and communication in groups of simulated and physical robots. *Biological Cybernetics*, 95:213–231, 2006.

V. Trianni and S. Nolfi. Self-organising sync in a robotic swarm. A dynamical system view. *IEEE Transactions on Evolutionary Computation, Special Issue on Swarm Intelligennce*, 13(4):722–741, 2009.

V. Trianni and E. Tuci. Swarm cognition and artificial life. In *Advances in Artificial Life. Proceedings of the 10th European Conference on Artificial Life (ECAL 2009)*, 2009.

V. Trianni, C. Ampatzis, A. L. Christensen, E. Tuci, M. Dorigo, and S. Nolfi. From solitary to collective behaviours: Decision making and cooperation. In F. Almeida e Costa et al., editor, *Advances in Artificial Life. Proceedings of the 9th European Conference on Artificial Life (ECAL 2007)*, volume 4648 of *Lecture Notes in Artificial Intelligence*, pages 575–584. Springer Verlag, Berlin, Germany, 2007.

V. Trianni, S. Nolfi, and M. Dorigo. Evolution, self-organisation and swarm robotics. In C. Blum and D. Merkle, editors, *Swarm Intelligence. Introduction and Applications*, Natural Computing Series, pages 163–192. Springer Verlag, Berlin, Germany, 2008.

G.P. Wagner and L. Altenberg. Perspective: Complex adaptations and the evolution of evolvability. *Evolution*, pages 967–976, 1996.

M. Waibel, L. Keller, and D. Floreano. Genetic team composition and level of selection in the evolution of cooperation. *IEEE Transactions on Evolutionary Computation*, 13 (3):648–660, 2009.

S. Wischmann, M. Huelse, J. F. Knabe, and F. Pasemann. Synchronization of internal neural rhythms in multi-robotic systems. *Adaptive Behavior*, 14(2):117–127, 2006.

Xin Yao. Evolving artificial neural networks. *Proceedings of the IEEE*, 87(9):1423–1447, 1999.

V. Zykov, E. Mytilinaios, B. Adams, and H. Lipson. Self-reproducing machines. *Nature*, 435(7038):163–164, 2005.