Engineering the Evolution of Self-Organizing Behaviors in Swarm Robotics: A Case Study

Vito Trianni*,**
Université Libre de Bruxelles

Stefano Nolfi†
Istituto di Scienze e Tecnologie della Cognizione

Keywords
Evolutionary robotics, swarm robotics, self-organization, engineering emergence

A version of this paper with color figures is available online at http://dx.doi.org/10.1162/artl_a_00031. Subscription required.

Abstract Evolutionary robotics (ER) is a powerful approach for the automatic synthesis of robot controllers, as it requires little a priori knowledge about the problem to be solved in order to obtain good solutions. This is particularly true for collective and swarm robotics, in which the desired behavior of the group is an indirect result of the control and communication rules followed by each individual. However, the experimenter must make several arbitrary choices in setting up the evolutionary process, in order to define the correct selective pressures that can lead to the desired results. In some cases, only a deep understanding of the obtained results can point to the critical aspects that constrain the system, which can be later modified in order to re-engineer the evolutionary process toward better solutions. In this article, we discuss the problem of engineering the evolutionary machinery that can lead to the desired result in the swarm robotics context. We also present a case study about self-organizing synchronization in a swarm of robots, in which some arbitrarily chosen properties of the communication system hinder the scalability of the behavior to large groups. We show that by modifying the communication system, artificial evolution can synthesize behaviors that scale properly with the group size.

1 Introduction

The synthesis of controllers for autonomous robots is a complex problem that has been faced with a large number of different techniques [32]. Among the various possibilities, evolutionary robotics (ER) represents a viable approach for the automatic synthesis of robot controllers requiring little a priori knowledge about the solution of a given problem [26]. In the swarm robotics context, ER can be very useful for synthesizing efficient self-organizing behaviors and for obtaining systems with desired emergent properties, such as robustness to individual failures, flexibility and adaptivity to environmental changes, or scalability to different group sizes. In fact, ER operates by introducing variations at the level of the fine-grained characteristics of the robots, and by retaining or discarding these variations on the basis of the effect they have on the global behavior exhibited by the swarm.

* Contact author.
** IRIDIA-CoDE, ULB, Avenue F. Roosevelt, n. 50, CP 194/6, 1050 Brussels, Belgium. E-mail: vtrianni@ulb.ac.be
† ISTC-CNR, via San Martino della Battaglia 44, 00185 Rome, Italy. E-mail: stefano.nolfi@istc.cnr.it

This bottom-up approach contrasts with classic top-down approaches (e.g., divide-and-conquer), which require an a priori arbitrary decomposition of the group behavior into individual behaviors and interaction rules. Such a decomposition is difficult to perform, due to the indirect relationships between the rules executed by the robots—which determine the individual reaction to perceived sensory states—and the overall group behavior. ER solves this *design problem* by synthesizing self-organization from the bottom up, therefore relieving the experimenter of the need to guess both the appropriate behavior that each individual robot should produce, and the control mechanisms that can lead to the exhibition of such behavior [39, 36]. More importantly, in some cases the designer might be unable to identify the behavior that should be exhibited by each individual robot in order to produce, in interaction with the other robots, the desired swarm behavior [12].

On the other hand, ER does not completely exempt the experimenter from the design problem, since he or she still has to face the problem of designing the setup of the evolutionary experiment, together with those characteristics of the robot and of the environment that are predetermined and fixed. In this case, the design effort concerns the preconditions that make the evolution of the self-organizing process possible, along with its emergent properties. Although designing an experimental setup that enables the evolution of effective solutions should be in principle simpler than designing the solutions themselves, the lack of systematic methodologies that can guide such an experimental design may prevent an effective exploitation of the potential advantages of the ER approach, or might condition such exploitation on the intuition of the experimenter. We believe that there is a strong necessity of formalizing an engineering approach for the evolution of robot behaviors, above all within a relatively novel and complex domain like swarm robotics.

In this article, we make a step in this direction by discussing which are the relevant choices that must be made when setting up an evolutionary experiment in a swarm robotics context. In particular, we focus on the robot sensory-motor system, the genotype-to-phenotype mapping, the fitness function, and the ecology in which the evolutionary process is carried out. Moreover, we suggest that, as in any engineering exercise, an iterative approach can guide the design process through incremental improvements. In this iterative process, a fundamental role is played by testing and evaluation of the obtained results, an activity that can convey useful information about the critical aspects of the system that must be improved. In this connection, we discuss a case study in which self-organizing synchronization behaviors are evolved in a robotic system. The goal of the experiment is to understand which are the minimal behavioral and communication strategies that allow a group of robots to synchronize their individual periodic behavior [38]. In particular, we are interested in the scalability property of the evolved behaviors to large groups. By analyzing the initial results, we discovered that the arbitrary choice made in the communication system was hindering the evolved behaviors from suitably scaling to large groups. This finding allowed us to re-engineer the characteristics of the robots by identifying a new communication channel, and to run further evolutionary experiments that resulted in properly scalable behaviors.

Engineering emergence in ER can be broadly linked to recent studies on guided self-organization [28, 29]. In these studies, various techniques are proposed to obtain desired behaviors through guidance rather than control. In fact, self-organizing systems are complex and strongly nonlinear, and attempts to obtain desired effects through direct control are often ill suited [14]. It is therefore more convenient to steer self-organization toward desirable outcomes, exploiting the dynamics of the self-organizing process itself. Therefore, on the one hand self-organization is promoted in a task-independent fashion, and on the other hand guidance is given to the self-organizing process in order to steer it toward desired patterns.

Recent studies have investigated different approaches to guide self-organization. Information-driven evolution is an ER technique that allows one to produce robot controllers exploiting a task-independent fitness function, usually based on information-theoretic measures, such as Shannon's entropy or mutual information [15, 16, 30, 33]. The task-independent metric provides intrinsic selective pressures that favor sensory-motor coordination or correlations among different agents' behaviors. However, some constraint is necessary to channel evolution toward the desired behavior. Constraints can be given in the form of explicit reward functions to be merged with the task-independent ones, or can be implicit in the robot configuration or in the robot ecology. A similar approach, but on an ontogenetic timescale, is homeokinetic learning [17]. Here, the spontaneous generation of behavioral patterns through homeokinesis is guided by learning through reinforcement signals [18]. Homeokinesis refers to the ability to produce and maintain a definite kinetic regime. In the robotic domain, this is achieved through learning an internal representation of the current behavior (a self-model) and adapting the behavior in order to minimize the difference with the learned model [8]. The interactions between model and behavior produce a spontaneous discovery and maintenance of behavioral patterns [7]. However, reinforcement signals are necessary to channel the spontaneous generation of behaviors toward desired outcomes [17].

In the context of evolutionary swarm robotics, self-organization needs to be guided at both the phylogenetic and the ontogenetic timescale. That is, we need to specify the correct selective pressures in order to observe the evolution of the desired group behaviors. Moreover, we need to set up the robotic system in order to obtain desired emergent properties of the evolved self-organizing behaviors. In Section 2 we discuss which are the choices that need to be taken in setting up an evolutionary experiment within a swarm robotic context. We consider four main components: the robot sensory-motor system in Section 2.1, the genotype-to-phenotype mapping in Section 2.2, the fitness function in Section 2.3, and the ecology in Section 2.4. In Section 3, we discuss our iterative approach in engineering and re-engineering the evolution of self-organizing behaviors. An instance of this process is presented in the case study of Section 4, which is discussed in detail in order to show how the proposed methodology can be applied. Section 5 concludes the article.

2 Engineering Emergence in Evolutionary Swarm Robotics

As mentioned above, the design of an evolutionary experiment in a swarm robotics context requires one to make a number of choices. First of all, the experimenter should determine which are the characteristics of the robots that are predetermined and fixed (e.g., the available sensors and actuators and/or the robot body structure), and which are the characteristics subject to the evolutionary process. Secondly, the experimenter should determine the way in which these evolved features are represented in the artificial genotype—in other words, how the genotype maps to the phenotype, that is, to the robotic system. Thirdly, the experimenter should devise the performance metric or fitness function to evaluate the evolving genotypes, which determines the selective pressures that allow him or her to progressively evolve the desired solution. Finally, the experimenter should determine the ecology (i.e., the characteristics of the environment in which the robots are evaluated), which in turn introduces ecological selective pressures to which evolving robots are exposed.

Guidance to the evolution of self-organization can be obtained through suitable choices at the moment of the design of the experiment. In this section, we discuss the above issues in an engineering perspective. That is, we consider that the robotic hardware and the task to be performed by the robots are specified in advance, and the focus of the experiment is the synthesis of a controller for the robotic group.[1] We limit our discussion to the swarm robotics domain, and we refer the reader to the ER literature for the topics not covered here [9, 13, 19, 26]. A schematic view of the design choices for the evolution of self-organizing behaviors that we consider in this work is given in Table 1.

2.1 Sensory-Motor System

The body and the sensory-motor system of the robots crucially constrain the way in which the robots can interact with the physical and social environment, and set the preconditions for the evolution of the desired global behavior and for the emergence of group-level properties. When the objective of an

1 A thorough discussion of all possible instances of evolutionary robotic experiments is out of the scope of this article. Nevertheless, this discussion can be easily generalized to issues that are not covered in these pages.

Table 1. The choices to be made in setting up an evolutionary experiment within the swarm robotics domain. Refer to Sections 2.1 to 2.4 for more details.

Component	Choices
Sensory-motor system	• Selected sensors and actuators
	• Communication channels
	• Preprocessing of raw data
Genotype-to-phenotype mapping	• Direct versus indirect encoding
	• Genetic relatedness of the group (homogeneous versus heterogeneous)
Fitness function	• Functional versus behavioural
	• Implicit versus explicit
	• External versus internal
Ecology	• Sampling of ecological conditions
	• Symmetry breaking

experiment is, as in the case considered here, to enable a group of robots to perform a certain function or to solve a certain problem, the characteristics of the body and of the overall sensory-motor system are predetermined and fixed. The experimenter, however, should determine a subset of the available sensors and actuators that will be used. Moreover, it must be determined how the chosen sensors and actuators are interfaced with the robot control system. The selection of the appropriate sensors and actuators is usually straightforward. However, it can be desirable or even necessary to process the raw sensor data to obtain more compact or better-usable information. This preprocessing can be as simple as a linear scaling of the sensor reading, but can even be a complicated function of many sensory inputs. For instance, a color camera provides a very rich information, and some feature extraction algorithm is necessary in order to recognize useful patterns within the image. The choices of the quantity and quality of the information extracted by the camera, and in general by a preprocessing of raw data, can be of fundamental importance for the evolvability of the system, and should be carefully taken into account.

A similar discussion holds for communication. Robots may be provided with different communication devices, which enable one to choose the type of messages exchanged (e.g., implicit communication through infrared sensors detecting the other robots' bodies [31] or torque sensors affected by the movement of other robots [3], sound communication through microphones and speakers [37], or light communication through colored LEDs and cameras [6, 22, 23]). The choice of the communication channel may significantly affect the evolutionary process, as it has a strong influence on the ability of the robots to interact with each other, and therefore to self-organize. In Section 4, we present a case study in which an ill-suited communication system severely limits the scalability of the evolved behaviors.

Overall, the important remark is that the selection of the appropriate sensory-motor system can be extremely critical, as it sets the preconditions for the evolution of the desired group behavior. Not only does this choice determine the actual capabilities of the individual robots, but it also determines the interaction abilities among the robots, that is, through which channels information is exchanged among the robots, and what is the capacity of these information channels.

2.2 Genotype-to-Phenotype Mapping

In evolutionary computing methods (including ER), a genotype is usually a string of bits or real numbers that encodes 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 behavior. The experimenter therefore has to specify the genotype-to-phenotype mapping, that is, the rules or the processes that determine the relation between the genotype (the string of bits or numbers) and the phenotype (the robotic system) [9, 19].

A common approach in ER consists of a direct mapping between genotype and controller parameters, in which each free parameter of the controller 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 behavior exhibited by the robot, within the possibilities and the limits imposed by the hardware characteristics of the robots and by the architecture of the robot controllers that are designed by the experimenter. A widely used approach in the literature consists of encoding into the genotype a fixed number of parameters of a neural network, while the neural network size, topology, and connectivity remain fixed. The use of neural controllers allows the evolutionary process to operate on the network parameters that regulate the fine-grained interactions between the robot and the environment, as mentioned. In this way, the evolutionary process shapes the behavior exhibited by the robot in detail. The use of a fixed architecture and of a direct encoding has the advantage of being simple, but requires one to choose the architecture of the controller beforehand. This again might turn out to be critical, since the characteristics of the architecture (viz., the number of neurons and/or the presence of recurrent connections) might strongly influence the obtained result. Moreover, it is difficult for the experimenter to predict the type of behaviors that the robot should exhibit and consequently the characteristics that the architecture should have in order to enable the production of such behaviors.

A different approach consists of indirect encoding, for which the genotype *develops* in a corresponding phenotype according to some developmental rules that are encoded in the genotype. This approach is followed to mimic embryogenesis, and offers the possibility to specify complex control structures with compact genetic encodings (thereby reducing the search space), and to avoid fixing the controller structure beforehand. However, these advantages come at some cost: It is necessary to specify a suitable set of developmental rules, and to ensure that those rules have the necessary expressive power.

In collective and swarm robotics, whether a direct or an indirect encoding is used, it is anyway necessary to specify the characteristics of the whole robotic group. The relevant choice here 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, the genotype usually encodes the parameters of a single controller, which is duplicated several times and embodied in all the robots taking part to the experiment (i.e., all individuals forming the swarm are generated from the same genotype). This method has the advantage of being simple and compact with respect to the number of free parameters, since the parameters of all individuals are encoded in the same genes. Moreover, it eliminates the problems related to the identification of the individual contributions to the performance produced by the entire swarm, and removes the conflict of interest that might arise between genetically different individuals, as discussed in the following section.

In a heterogeneous group, the swarms is constituted by individuals generated by different genotypes or by different parts of the same genotype. The use of heterogeneous groups might be advantageous when the robots forming the group are to play well-differentiated roles. In this case, however, the different roles must be somehow encoded into the genotype. The simplest approach consists of deciding a priori 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 might express different parts of their genome [4]. These approaches allow one 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 populations 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, 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.

2.3 Behavioral Selective Pressures: The Fitness Function

The definition of the performance metric that rewards the desired behavior is usually task-dependent (i.e., a function that estimates the extent to which the swarm solves the given task).[2] There are multiple ways to define a fitness function for a given problem [25]. To discriminate between different types, Floreano and Urzelai proposed the use of a three-dimensional *fitness space*, in which the different dimensions refer to important features of a fitness function [11]:

- *Functional versus behavioral:* A functional fitness rewards a particular working modality (i.e., gives an indication of the actuators' outputs), while a behavioral fitness measures the quality of the behavior (i.e., gives an indication of the outcome of a sequence of actions).
- *External versus internal:* An external fitness is computed through variables that are available to an external observer (e.g., the absolute position of the robot in the environment), while an internal fitness is computed through variables available to the robot (e.g., the sensor readings). While external fitness functions can be easier to deploy, internal ones may reduce possible biases introduced by the designer;
- *Explicit versus implicit:* An explicit fitness measures the way in which a goal is achieved (e.g., the trajectory followed to get close to a light source), and therefore puts constraints on the displayed behaviors. An implicit fitness function, instead, measures the level of attainment of a goal (e.g., how close to the light source the robot ends). An implicit fitness gives more freedom to explore the solution space, therefore allowing one to find solutions that are not a priori envisioned by the experimenter.

In swarm robotics, the indirect relationship between individual actions and group organization makes it difficult to devise functional measures. A functional measure, in fact, is directly related to the causes of the observed behavior, 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 behavior 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 by averaging over the group or by 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 organization. Finally, implicit measures should be preferred when the relationship between the individual control rules and the group behavior is indirect or unknown, as they pose less constraints on the way the desired collective behavior is achieved.

In evolutionary swarm robotics, it is often the case that a homogeneous group of robots should present a self-organizing behavior. In this case, it is useful to evaluate the group-level properties through external fitness functions, rather than looking at the individual actions. It is also useful to evaluate the

2 For alternative approaches based on task-independent fitness functions or combination of the two, see [15, 16, 30, 33], already mentioned in Section 1.

group organization (i.e., the spatiotemporal pattern) with the use of implicit metrics, rather than explicitly rewarding the way in which the organization is achieved. Indeed, external and implicit fitness functions pose less constraints on the way in which the problem should be solved. On the contrary, if we consider the case in which a heterogeneous group should display teamwork with highly specific roles, internal and explicit metrics could be preferred, as they may allow one to develop the implementation details of specific solutions beforehand identified.

The use of heterogeneous swarms constituted by individuals generated from different genotypes poses additional constraints on the definition of the fitness function, since it requires estimating the contributions of genetically different individuals to the overall ability of the group. Without going too deep into the details, it is important here to note that this problem does not affect homogeneous groups and heterogeneous teams generated from a single genotype. In these cases, in fact, the group performance can be directly assigned to the single genotype that generated the entire swarm. 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. That, however, is a complex and time-consuming procedure, which creates 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 behavior [10, 20, 40].

2.4 Ecology

The ability of a swarm to perform a certain task does not depend only on the characteristics of the robot, but is also influenced by the characteristics of the environment (e.g., variability in space and time) and by the relation between the robots and the environment (e.g., the set of possible initial positions and orientations of the robots within the environment). The behavior of the swarm should be robust with respect to possible variations of the environment and of other parameters that contribute to specify the *ecological niche* in which the behavior is evolved. Indeed, the specification of the ecological niche of the robotic system can significantly affect the results of the evolutionary process [27]. A precise computation of the fitness would require testing the behavior 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 determination of experimental conditions that can let the group experience the interaction patterns relevant for obtaining robust behavior.

It is important to note that indirect selective pressures may be created through the specification of the ecological niche and through the sampling employed to estimate the fitness. Given that the group is evaluated for presenting a robust behavior within the parameter space of the ecological niche, the choice of the sampling may influence the evolutionary path. For instance, in [2] 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 specified in order to account for robust group behavior and to take into account implicit selective pressures.

A final issue to consider concerns symmetry breaking, which is involved in 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, characterized 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 an 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 by exploiting the inherent randomness of the robotic system, and therefore suitable behavioral strategies may be required. The evolutionary machinery needs to encounter such conditions often in order to synthesize the collective behavior 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 behaviors. An example of systematic testing in symmetric and asymmetric conditions for a two-robot system is given in [1].

3 The Engineering Design Cycle in Evolutionary Robotics

An attentive setup of the evolutionary experiments does not guarantee achievement of the desired results. In fact, some choices, although reasonable without any a priori knowledge of the evolutionary dynamics, may turn out to be too constraining, or may prevent the system from displaying certain properties. However, negative results should be exploited to acquire information on the system dynamics and re-engineer hand-designed characteristics accordingly. In fact, by understanding the properties of unsuccessful results it may be possible to recognize which are the critical aspects that constrain the system in suboptimal solutions. As a consequence, it is possible to proceed with an informed re-engineering of the evolutionary experiments.

This iterative process is not any different from any engineering design cycle. We can highlight three main phases: (i) the *setup* of the evolutionary experiment, (ii) the actual *evolution*, and (iii) the *analysis* of the results. In the setup phase, all the relevant choices are taken to devise the evolutionary experiment, as discussed above. The setup phase is fundamental within the design cycle. Here, the expertise of the experimenter is of utmost importance to orient the evolutionary design in the right direction. The main goal of this phase is recognizing which are the free parameters of the system that must be specified beforehand, and which are the ones that will be subject to evolution (through variation and selective reproduction). In doing so, many choices must be performed somewhat arbitrarily, especially when there is no a priori knowledge about the system and evolutionary dynamics.

In the evolution phase, a sufficient number of instances (or replications) of the evolutionary experiment are performed, each starting with a different random population of genotypes. On the one hand, multiple instances increase the chances of finding an optimal solution to the given problem. On the other hand, the quality of the design choices can be verified by looking at the percentage of instances that produced good results. In the best case, all instances of the experiment produce acceptable results. Instead, a small percentage of successful instances might indicate that there is room for improvement (i.e., the setup of the evolutionary experiments can be modified in order to increase the quality of the evolved solutions). Also, a suboptimal solution may reveal that some features of the robotic system chosen in the setup phase hinder the production of the desired behavior.

Many ER practitioners just cycle between the setup and evolution phases, exploiting intuition and experience in adapting the evolutionary setup to obtain better results. Less frequently, an attentive analysis of the negative results obtained in previous iterations is performed to identify the critical aspects of the evolutionary design. We believe that the analysis phase should always be conducted, and suitable methodological tools developed, in order to support the re-engineering of the evolutionary experiment in promising directions. Two analyses should be conducted to guide the evolutionary process toward the desired results. First, the analysis of the selective pressures and of the evolutionary dynamics can inform the experimenter about the limits of the system in producing better and better solutions starting from scratch. Second, the evolved suboptimal solutions should be analyzed to acquire knowledge about the characteristics of the robotic system and of the produced behavior, in order to understand whether or not it can display the desired properties. In this case, generalization tests should be performed to verify the ability of the system to work in conditions not experienced during the evolutionary phase. In fact, during evolution it is not always possible to test all the conditions that can be faced by the robotic system. Therefore, it is important to test the robustness to failures, flexibility in response to varying environmental conditions, or scalability to larger groups. In the following section, similar analyses are performed to understand the properties of the evolved system with respect to the scalability to larger groups. In this case study, the design cycle is closed by re-engineering the communication channel in order to obtain better scalability and better performance, which also results in more robust evolutionary dynamics.

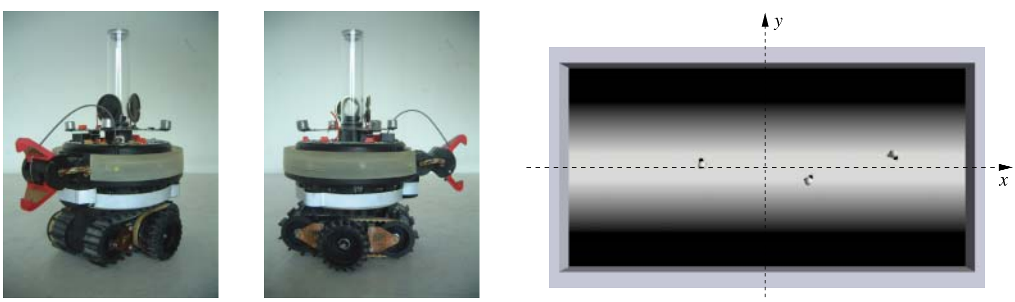
4 Case Study: Self-Organized Synchronization

Self-organized synchronization 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 behavior [34]. A well-known synchronization phenomenon is the flashing behavior of some firefly species in South-East Asia, which aggregate at dusk and engage in massively synchronous displays [5]. Models of this behavior describe fireflies as a population of pulse-coupled oscillators with equal or very similar frequencies. These oscillators can influence each other by emitting a pulse that shifts or resets their oscillation phase. The numerous interactions among the individual oscillator-fireflies are sufficient to explain the synchronization of the whole population (for more details, see [5, 21, 35]). This model has been often exploited to engineer systems capable of synchronous behavior in collective and swarm robotics [6, 41]. In this study, we have investigated which are the minimal behavioral and communicative conditions that can lead to synchronization in a group of robots, in which each individual presents a periodic behavior. For this purpose, we chose to provide robots with simple reactive controllers and basic communication abilities. The period and the phase of the individual behavior 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-organized synchronization, allowing minimal complexity of both the behavioral and the communication level (for more details, see [38]).

4.1 Phase 1: Evolutionary Setup

We make use of a simple evolutionary algorithm that works on a population of 100 binary genotypes, which are randomly generated. Each genotype is a string of bits that encode some parameters of the robotic system. Once the fitness of each genotype in the population has been evaluated, a new population is produced by a combination of selection with elitism and mutation. Recombination is not used. At each generation, the four best individuals of the population—the *elite*—are retained in the subsequent generation. The remainder of the new population is generated by the 20 individuals of the previous generation that scored the highest fitness. Each selected genotype reproduces at most five times by applying mutation with 3% probability of flipping a bit. The evolutionary process runs for 500 generations.

The evolutionary experiments are performed in simulation, using a simple kinematic model of the *s-bot* robot (see Figure 1a,b, and refer to [24] for details), and the results are afterward validated on



(a) (b) (c)

Figure 1. (a,b) The s-bot, the robot used in the experiments. (c) Snapshot of a simulation showing three robots in the experimental arena. The dashed lines indicate the reference frame used in the experiments.

the physical platform. The experimental scenario for the evolution of self-organizing synchronization requires that each robot in the group display a simple periodic behavior, which should be entrained with the periodic behavior of the other robots present in the arena. The individual periodic behavior consists of oscillations along the y direction of a rectangular arena (see Figure 1c). Oscillations are possible through the exploitation of a symmetric gradient in shades of gray painted on the ground. The gradient presents a white stripe for $|y| < 0.2$ m, and black stripe for $|y| > 1$ m.

4.1.1 Sensory-Motor System

For the purpose of engineering the evolutionary system, both the characteristics of the arena and the capabilities of the robots place several constraints on the experimental setup. According to these constraints, we select among the various possibilities the minimal set of sensors and actuators that are required to accomplish the task, that is, individual periodic oscillations over the gray gradient and synchronization of the oscillation phase. Certainly, the controller needs access to the wheels' motors, and we set $\omega_M \approx 4.5\ \mathrm{s}^{-1}$ as the maximum angular speed of the wheels. The gray gradient of the arena can be perceived by the robots through four infrared sensors placed under their chassis (ground sensors), which are appropriately scaled to encode the gray level in the range [0, 1], where 0 corresponds to black and 1 to white. The perception of the gradient through these sensors provides the robot with enough information to perform oscillations along the y axis. Additionally, robots need to use the infrared proximity sensors placed around their cylindrical body, in order to avoid collisions with walls or with other robots. These choices, which are mainly constrained by the arena setup and by the features of the physical robot, are sufficient for the individual behavior.

For what concerns the group behavior, instead, we need to provide the robots with suitable interaction modalities that can lead to synchronization of their movements. The choice of the communication system is the aspect we focus on in this article. In fact, the s-bot platform features various communication devices, and we need to select among them the one that best fits our experimental scenario. The robots are provided with speakers and microphones for sound communication. Moreover, the robots can exploit colored LEDs positioned around their turret to display a color pattern that can be perceived through an omnidirectional camera. Finally, the robots have wireless communication abilities. Therefore, there is large freedom in choosing the communication system. In order to maintain a minimal configuration, we decided to provide the robots with a *global* and *binary* communication system:

$$s(t) = \max_r S_r(t), \tag{1}$$

where $S_r(t) \in \{0, 1\}$ is the binary signal emitted by robot r at time t, and $s(t) \in \{0, 1\}$ is the binary signal perceived by all robots. In other words, each robot r can produce a signal $S_r(t)$. Signals produced by different robots cannot be distinguished, and result in a single signal $s(t)$ perceived by every robot in the arena, including the signaling one. Signals are perceived in a binary way: Either there is someone signaling in the arena, or there is no one. This communication channel is probably the poorest one, in terms of the amount of information, that can be conveyed. However, it is sufficient for our purposes, as we will see in the following. Note that this communication channel can be easily implemented with sound signals: A robot can emit a single-frequency tone with an intensity high enough to be perceived everywhere in the arena. Additionally, it is worth mentioning that, differently from the other sensors and actuators, the choice of the communication system is not constrained by the robotic hardware or by other aspects of the experimental setup, but is only dictated by the communication system we have chosen.

4.1.2 Genotype-to-Phenotype Mapping

As mentioned below, in self-organizing synchronization all individuals are (nearly) identical and display the same behavior. Synchronization emerges from the interactions among individuals, which adapt their behavior in order to entrain with each other. This feature constrains the genotype-to-phenotype mapping to produce a homogeneous group of robots. To do so, the genotype is mapped into a single control structure, which is cloned and downloaded onto all the robots taking part in the experiment.

The control structure chosen is a fully connected, feed-forward neural network—a perceptron network. The choice of this type of neural network stems from our working hypothesis, that is, searching for the minimal complexity of the control and communication system that can support synchronization behaviors. For this purpose, we choose a controller that directly transforms the sensory input into the motor output, without recurrent connections or internal states. The neural network has 11 sensory neurons directly connected to three motor neurons. The sensory neurons are simple relay units, and the output neurons are sigmoid units whose activation is computed as follows:

$$O_j = \sigma\left(\sum_i w_{ij} I_i + \beta_j\right), \qquad \sigma(z) = \frac{1}{1 + e^{-z}}, \tag{2}$$

where I_i is the activation of the ith input unit, β_j is the bias term, O_j is the activation of the jth output unit, w_{ij} is the weight of the connection between input neuron i and output neuron j, and $\sigma(z)$ is the sigmoid function. Six sensory neurons—I_1 to I_6—receive input from a subset of the infrared proximity sensors evenly distributed around the s-bot's turret. Four sensory neurons—I_7 to I_{10}—are dedicated to the readings of the four ground sensors. The state of all infrared and ground sensors is linearly scaled to the range [0.0, 1.0]. A simulated uniform noise within 5% of the input range is also added. The last sensory neuron I_{11} receives a binary input corresponding to the perception of sound signals. The activation states of the first two motor neurons—O_1 and O_2—is scaled onto the range $[-\omega_M, +\omega_M]$, where ω_M is the maximum angular speed of the wheels ($\omega_M \approx 4.5$ s^{-1}). The third motor neuron controls the speaker in such a way that a sound signal is emitted whenever the activation state O_3 is greater than 0.5. The connection weights w_{ij} and bias terms β_j are the genetically encoded parameters. Each parameter is represented with an 8-bit binary code mapped onto a real number ranging in [−10, +10]. In other words, we have a direct encoding of the genotype into the phenotype, as there is a bijective function that relates the genotype to the phenotype.

4.1.3 The Fitness Function

The performance of a genotype is evaluated by a two-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 2.3, this component is *behavioral*, *external*, and *implicit*. In fact, it rewards movement of the robot from the observer's perspective, without explicitly indicating how to perform a periodic behavior: The oscillatory behavior derives from the fact that the arena is surrounded by walls, so that oscillations during the whole trial are necessary to maximize $F_{\mathcal{M}}$. The second fitness component $F_{\mathcal{S}}$ rewards synchrony among the robots as represented by the cross-correlation coefficient between the distance of the robots from the x axis. This component as well is *behavioral*, *external*, and *implicit*: It is related to the group behavior, and measures a quantity—the cross-correlation—that is available only to the observer. In addition to the fitness computation described above, two indirect 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 for exploiting 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. For more details on the fitness computation, refer to [38].

4.1.4 The Ecology

The ecology in which the group behavior evolves also needs to be specified. The arena is a rectangle of 6 m × 3 m sides, completely surrounded by walls. The ground is painted white for $|y| < 0.2$ m, and linearly changes to black until $|y| = 1$ m. For larger distances, the arena is painted black. We are interested in behavioral and communication strategies that scale well with the group size. However, it is not possible to test every possible group size, and we therefore decided to fix the number of robots in the group at just three. This small number of robots allows us to obtain fast simulations and still support the evolution of group coordination. In order to have a good estimate of group performance, the genotype fitness is taken as the average of the group fitness computed over 10 different trials. In each trial, we vary the initial positions and orientations of the three robots by choosing them uniformly randomly within the arena. This allows us to let the robots experience many different initial conditions, and should result in robust and efficient synchronization behaviors.

4.1.5 Design and Evolution

Before presenting the obtained results, it is useful to discuss which features are fixed by the experimenter, and which are adaptively set by the evolutionary process. We have defined an experimental scenario that is intrinsically cooperative, because robots are homogeneous and are explicitly rewarded to display a desired group behavior. We have also fixed the sensory-motor configuration and the controller architecture. In particular, we have fixed the interaction modality between different robots, which mainly happens through the binary and global communication signal. Notwithstanding this, the motor and communicative behavior is not at all predetermined, but is the result of the evolutionary process. The individual behavior and the synchronization mechanisms are completely determined by the parameters of the neural controller (i.e., connection weights and biases). Individual behavior and communication signals coevolve and influence each other: The individual behavior determines how the robot moves and experiences the environment, which influences the signals emitted. In turn, perceived signals change the way in which the robot reacts to the environment. During evolution, the group behavior is shaped in order to maximize the user-defined utility metric, within the constraints imposed by the predetermined features. In the following, we will see how the communication channel we have chosen influences the obtained results.

4.2 Phase 2: Evolution

In order to test the robustness of the evolutionary setup we devised, we decided to run 20 different instances of the experiment. Each instance was initialized with a different randomly generated population of genotypes, and lasted 500 generations. At the end of the evolutionary process, to assess the quality of the evolved behaviors, we select a single genotype per instance to be chosen among the best individuals of the final generation. To do so, we evaluate the performance of the 20 best individuals of the final generation in 500 different trials, and we choose the individual with highest average fitness. In the remainder of this article, we refer to the best controllers evolved in replication i as c_i, $i = 1,\ldots, 20$. The performance of these controllers over the 500 post-evaluation trials, sorted according to decreasing median values, is shown in Figure 2. The obtained results show that in most replications the performance obtained is on average within the interval [0.7, 0.9], which indicates that robots are able to maximize both the movement fitness component $F_{\mathcal{M}}$, and the synchronization component $F_{\mathcal{S}}$.

In order to assess the difference in performance among the controllers evolved in different evolutionary replications, we used the performance data recorded over 500 trials to perform a series of pairwise Wilcoxon tests among all possible controller pairs. The results are plotted in Figure 2 as vertical lines spanning the controller numbers having a performance that is not statistically different (at 99% confidence). So, for example, controllers c_{13} and c_{15} are not statistically different from the performance point of view. Similarly, controller c_1 has a performance nearly equal to that of c_{18} and c_{19}, but it performs worse than controllers c_{10} and c_{14}. As can be seen in Figure 2, controller c_8 outperforms all other controllers.

The fitness values just tell us that the system is able to maximize the performance metric we devised. It does not tell us if and how synchronization is achieved, and whether or not the system

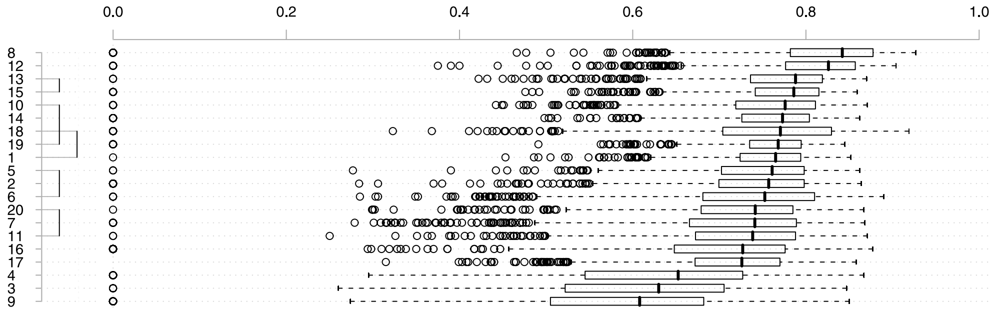


Figure 2. Post-evaluation results for the best evolved controllers c_i in each replication $i = 1,\ldots, 20$ of the evolutionary experiment. The performance is represented on the horizontal axis, and the controller number on the vertical axis. The boxplot displays the whole data set: Each box represents the interquartile range of the data, while the black vertical line inside the box marks the median value. The whiskers extend to the most extreme data points within 1.5 times the interquartile range from the box. The empty circles mark the outliers. Data from different controllers are sorted according to the median value. Moreover, statistical similarities are represented as vertical bars spanning the controller numbers (see text for detail).

presents desired properties like robustness or scalability. For this purpose, we performed behavioral and scalability analyses, which are discussed in the following section.

4.3 Phase 3: Analysis

By looking at the behavior exhibited by a group of three robots, we observed that each instance of the evolutionary experiment produced a successful synchronization behavior, in which robots display oscillatory movements along the y direction and synchronize with each other, according to the requirements of the devised fitness function. In general, it is possible to distinguish two phases in the evolved behaviors: an initial transitory phase during which robots achieve synchronization, and a subsequent synchronized phase. The transitory phase may be characterized by physical interferences between robots due to collision avoidance, if robots are initialized close to each other. The collision avoidance behavior 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 synchronization can be achieved. The synchronous phase is characterized by a stable synchronous oscillation of all robots, and small deviations from synchrony are immediately compensated for.

The individual's ability to perform oscillatory movements is based on the perception of the gradient painted on the arena floor, which gives information about the direction parallel to the y axis and about where to perform a U turn and move back toward the x axis, thereby avoiding movement into the black-painted area. Each evolved controller produces a signaling behavior that varies while the robots oscillate. The main role of the evolved signaling behavior is to provide a coupling between the oscillating robots, in order to achieve synchronization. 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 differences among themselves, and eventually synchronize. In a previous work [38], we developed a mathematical model and exploited dynamical systems theory to thoroughly analyze the synchronization behavior. We invite the reader to refer to that work for further details on the synchronization mechanism, which are out of the scope of the present article.

Having analyzed the synchronization behaviors evolved using three robots only, we tested scalability to large groups. To do so, we compared the performance of the evolved behavior while 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. By ensuring a constant initial density we limit the negative effects of overcrowding and we are able to compare the performance of robotic systems with varying group size. In order to keep a constant robot density equal to the one used in the evolutionary experiments, we lengthened the arena in the x direction, trying to keep a uniform initial density of 0.25 robots per square meter. Despite the increased arena length, we still kept the same communication system, that is, communication continued to be binary and global, with all robots affecting each other. This choice allows us to evaluate the scalability of a behavior as it was evolved, without modifying the features of the communication channel. 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 notable that most of the best evolved controllers have good performance for groups composed of six robots. Performance degrades for larger group sizes, and only a few controllers produce scalable behaviors up to groups formed by 96 robots.

The main problem that reduces the scalability of the evolved controllers is produced by the physical interactions among robots. Despite the constant initial density we imposed in order to limit the disruptive effect of collision avoidance, physical interactions nevertheless occur with a higher probability per timestep as the group size increases. Every collision avoidance action provokes a temporary desynchronization of at least two robots, which have to adjust their movements in order to regain synchronous oscillations with other robots. In such cases, the whole group is influenced by the attempt of a few robots to regain synchronization, due to global and binary communication.

To summarize, the above analysis shows that physical interactions and collision avoidance have a disruptive effect on the synchronization ability of the robots, and this effect is more and more visible as group size increases. However, the synchronization 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 ignored the physical interactions among the robots, as if each robot were placed in a different arena and perceived the other robots only through communication 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 good scalability, with only a slight decrease in performance due to the longer time required by larger groups to perfectly synchronize (namely, controllers evolved in replications 2, 8, 10, 12, 14, 18, and 19). This result confirms the analysis about the negative effects of physical interferences and collisions among robots. In fact, removing the necessity to avoid collisions leads to scalable self-organizing behaviors.

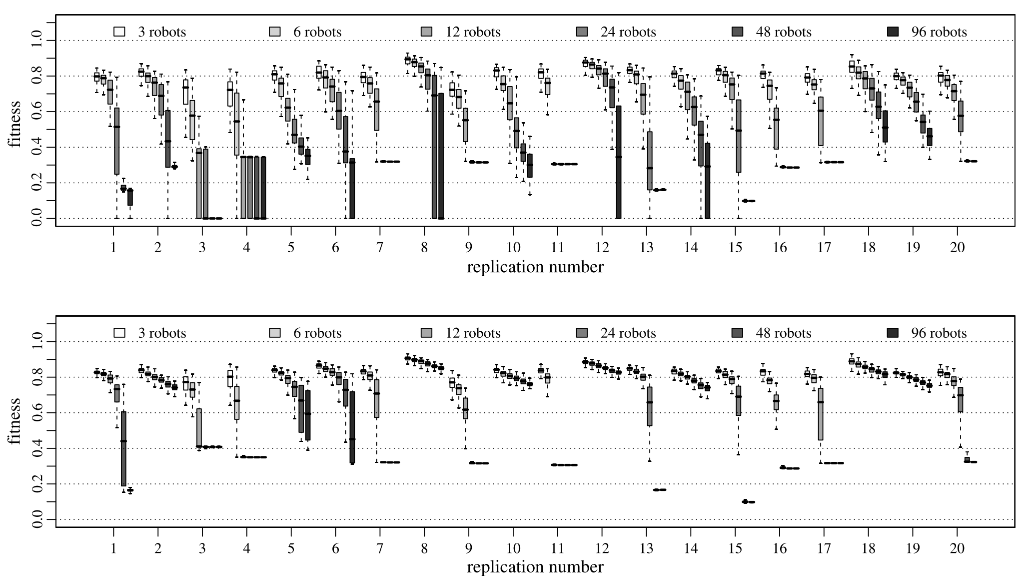


Figure 3. Scalability analysis. The boxplot shows, for each evolved controller, the performance obtained in tests with 3, 6, 12, 24, 48, and 96 robots. Each box represents the interquartile range of the data, while the black horizontal line inside the box marks the median value. The whiskers extend to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

Nevertheless, many other controllers (namely, controllers evolved in replications 3, 4, 7, 9, 11, 13, 15, 16, 17, 20) present a strange behavior. The performance presents high variability up to a certain group size. The variable performance indicates that in some cases the robots are able to synchronize, and in other cases not. With larger group sizes, the performance stabilizes to a low, constant value, independent from the initial conditions and the number of robots used. This value, which is characteristic of each nonscaling controller, represents the performance of the robotic system trapped in the basin of an *incoherent attractor*. In other words, the robotic system always converges into a dynamical condition in which no robot can synchronize with any other. By observing the actual behavior produced by these controllers, we realized 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 (signals are global and are perceived in a binary way, preventing a robot from recognizing different signal sources). If the perceived signal does not vary in time, it does not bring enough information to be exploited for synchronization, and the system remains desynchronized. This result is confirmed by the dynamical system analysis that we performed, which revealed how the individual signaling behavior is responsible for producing such communicative interference, allowing us also to predict which controllers present scalability by just looking at the individual behavior (for more details, see [38]).

4.4 Re-engineering for Scalability

The analysis of the unsuccessful controllers revealed that scalability cannot always be obtained, due to the physical and communicative interferences among robots. In particular, the communication channel we selected has a strong effect on the scalability of the system. Here, communication is global and binary, that is, the signal emitted by a robot is perceived by every other robot everywhere in the arena. Moreover, from the robot point of view, there is no difference between a single robot and a thousand signaling at the same time. Therefore, a single robot can influence the whole group. This has no negative effect as long as robots are synchronous, but can have severe consequences when a robot modifies its behavior due to collision avoidance following some physical interaction with other robots. Furthermore, the binary communication channel generates the communicative interference we described above, which prevents the group from synchronizing in certain conditions.

The main problems are therefore related to the absence of *locality* (i.e., signals are perceived everywhere in the arena) and of *additivity* (i.e., signals overlap without adding, making it impossible to recognize how many robots are contemporaneously signaling). The lack of locality and additivity is the main cause of failure of the scalability of the evolved synchronization mechanisms.[3] We therefore decided to re-engineer our evolutionary experiments by changing the communication system, which was chosen arbitrarily in the first place. Given that we are interested in studying global synchronization, we decided to re-engineer our experiments, focusing only on the additivity of the communication system. This allows us to make only minor changes to the experimental setup and directly compare the effects of the re-engineering approach.

We evolved self-organizing synchronization behaviors exploiting exactly the same setup as above, but changing the way robots signal and perceive emitted signals. Specifically, we change the binary communication system to a continuous one:

$$\tilde{s}(t) = \frac{1}{N}\sum_{r=1}^{N} \tilde{S}_r(t). \tag{3}$$

Now, robots always emit a signal $\tilde{S}_r(t) \in [0, 1]$, encoding a number in a continuous range. The emitted signals are perceived as the average $\tilde{s}(t)$ 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—

3 However, as we have seen, this problem affects only some of the analyzed controllers.

depends on the signaling behavior of the whole group: The bigger the group, the smaller the influence of the single individual. This communication channel can be easily implemented on the s-bots. For instance, signals could be sent as messages over a wireless network containing a real number in [0,1]. On the basis of the analysis performed so far, we expect that self-organizing synchronization behaviors can be evolved with such a communication system, and that these are more scalable.

Also in this case, we executed 20 instances of the evolutionary experiments, using groups of three robots. All replications were successful, and produced synchronization behaviors 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 their individual behavior so as to synchronize with other robots. We therefore executed a scalability analysis, which was performed with the same modalities as described above, and the obtained results are presented in Figure 4.

In the upper plot, scalability is tested including physical interactions. Also in this case, we note that collisions prevent the scalability of some controllers, in which good avoidance behavior was not evolved. Recall that when a collision is detected, the group scores a null performance. However, note that the use of an additive communication system leads to better performance even with large groups. Most controllers present good scalability for every tested group size, and only collisions substantially reduce the performance. Here, differently from what was observed before, physical interactions and collision avoidance do not have a severe effect on the performance of the whole group. In fact, the signals of the few nonsynchronous robots are averaged with those emitted by the rest of the group. As a consequence, the influence on the group of a single robot attempting to synchronize decreases with increasing group size. This leads to quick convergence to synchrony and to improved group performance.

To better understand the effects of the re-engineering approach, we also performed a scalability analysis for the evolved synchronization mechanisms, again removing the physical interactions among robots. The results plotted in the lower part of Figure 4 show that all evolved synchronization mechanisms scale perfectly, and they do not suffer from the communicative interference observed with binary signals. In fact, the perceived signal brings information about the average signaling behavior of all robots. As a consequence, synchronization is always achieved, no matter the group size. Note also that all controllers present a linear decrease in performance in correspondence to an exponential

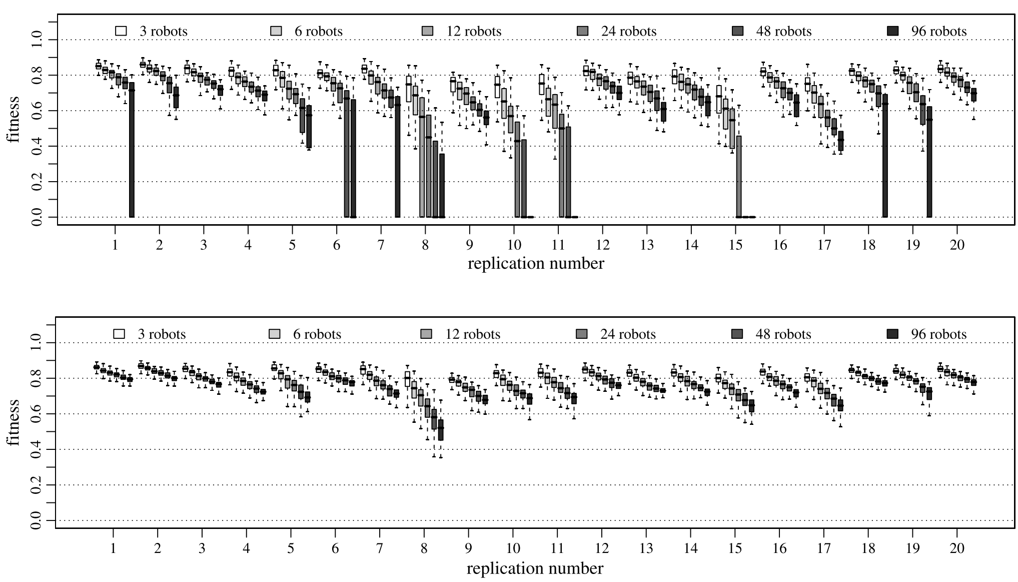


Figure 4. Scalability analysis for the continuous communication system. Top: scalability of the evolved controllers under normal conditions. Bottom: scalability of the synchronization mechanism.

growth of the group size. This observation suggests that the self-organizing synchronization mechanism is very efficient, and is only slightly affected by the group size.

5 Discussions and Conclusions

Evolutionary robotics can be a very useful methodology for the automatic synthesis of controllers for robotic systems. This is especially true in the swarm robotics domain, which is characterized by an indirect, nonlinear relationship between the control rules followed by an individual and the group behavior. However, ER cannot be applied blindly and effortlessly. Most importantly, ER does not exclude arbitrary choices in setting up experiments, which can strongly influence the outcome of the evolutionary process. The advantage given by ER is that, despite such arbitrary choices, it can find good or even optimal solutions for a given problem. However, much as in conventional engineering methods, multiple design loops may be needed to find such optimal results. In this article, we suggest that a structured, formal approach to the design of experiments in evolutionary swarm robotics can be very advantageous. On the one hand, we identified and discussed which are the relevant choices in designing the setup of an evolutionary experiment in the context of swarm robotics. On the other hand, we discussed the appropriateness of an engineering design cycle and showed its advantages in a practical case study.

The case study we presented was about the evolution of self-organizing synchronization in a robotic system. We showed that in setting up the experiments, some characteristics of the system were chosen somewhat arbitrarily, given that no a priori knowledge was available about the possible solutions to the given problem. The results obtained with the initial experimental setup proved that self-organizing synchronization can actually be achieved with minimal complexity at the level of the control and communication strategy. However, the analysis of the scalability results also pointed to some critical aspects of the system that hindered the group from scoring a good performance. We identified the problem as due to the communication system being global and binary, and to the effects of physical and communicative interference. To solve this problem, we re-engineered the arbitrarily chosen communication system, exploiting the knowledge acquired by analyzing the evolved behaviors. The newly devised continuous signals resulted in better synchronization behaviors, and in an optimally scaling communication system.

This article demonstrates that it is possible to engineer some features of a system undergoing artificial evolution on the basis of the outcome of the evolutionary process itself. Contrary to trial-and-error methods without any guidance, we showed that an attentive analysis of negative results conveys knowledge on how to modify the system for evolving better solutions. Note that this is not in contradiction with the need of little a priori knowledge in the design of the evolutionary experiment, as mentioned in the introduction. In fact, the knowledge we put into engineering the evolutionary system should not be related to the design of the solution, which is left to the evolutionary process, but rather to the preconditions required for obtaining good solutions. This is a subtle but germane difference, which should be discussed further. When we put knowledge about the solution into the experimental design—either explicitly in the fitness function, or implicitly in the experimental setup—we try to constrain evolution into a specific path. However, it may be very difficult to force the evolutionary dynamics and to obtain the desired solutions. And it may also be the case that the obtained solutions are too specific for the evolutionary conditions that have been set, missing important features like robustness, flexibility, and adaptivity. Additionally, if a specific behavior is desired and known, it may be easier to engineer it directly without relying on an automatic design methodology, and possibly use evolutionary methods just for parameter tuning.

On the contrary, we propose an iterative engineering methodology that allows us to exploit the knowledge acquired from negative results and previous evolutionary experiments to define the preconditions for obtaining better results. This does not mean forcing the system within specific evolutionary paths, but it rather liberates evolution from constraining attributes of the experimental setup. In this way, the solutions that can be generated should be more efficient, and should better generalize to conditions not directly experienced during evolution.

As a final remark, it is worthwhile to discuss the choice of the swarm robotics domain for the proposal of an engineering approach to ER. We actually believe that most of the methodological aspects mentioned in this article can be easily generalized to other domains. However, within swarm robotics a structured and formal methodology is even more compelling. This is related to the idea of evolving self-organizing processes through guidance rather than control: Swarm robotics has the additional complexity of dealing with emergent systems, with emergent properties resulting from complex interactions among individuals. In this case, knowing which are the degrees of freedom in setting up an experiment can be of utmost importance, as well as knowing in advance which are the possible effects of certain choices. In future work, theoretical and experimental studies should be conducted to devise a comprehensive methodological framework for the design of ER systems: much as in the software engineering field, engineering practices should be developed to guide the experimenter through the different phases of setup, evolution, and analysis of an evolutionary robotic system.

References

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

2. Ampatzis, C., Tuci, E., Trianni, V., & Dorigo, M. (2008). Evolution of signaling in a multi-robot system: Categorization and communication. *Adaptive Behaviour*, *16*(1), 5–26.

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

4. Bongard, J. C. (2000). The legion system: A novel approach to evolving heterogeneity for collective problem solving. In R. Poli, W. Banzhaf, W. Langdon, K. Miller, P. Nordin, & T. Fogarty (Eds.), *Genetic programming*. Berlin: Springer-Verlag.

5. Buck, J. (1988). Synchronous rhythmic flashing of fireflies. II. *The Quarterly Review of Biology*, *63*(3), 256–289.

6. Christensen, A. L., O'Grady, R., & Dorigo, M. (2009). From fireflies to fault tolerant swarms of robots. *IEEE Transactions on Evolutionary Computation*, *13*(4), 754–766.

7. Der, R., Hesse, F., & Martius, G. (2006). Rocking stamper and jumping snake from a dynamical system approach to artificial life. *Adaptive Behavior*, *14*(2), 105–115.

8. Der, R., Steinmetz, U., & Pasemann, F. (1999). Homeokinesis—A new principle to back up evolution with learning. In *Computational Intelligence for Modelling, Control, and Automation* (pp. 43–47). Amsterdam: IOS Press.

9. Floreano, D., Dürr, P., & Mattiussi, C. (2008). Neuroevolution: From architectures to learning. *Evolutionary Intelligence*, *1*(1), 47–62.

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

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

12. Funes, P., Orme, B., & Bonabeau, E. (2003). Evolving emergent group behaviors for simple human agents. In W. Banzhaf, T. Christaller, P. Dittrich, J. T. Kim, & J. Ziegler (Eds.), *Advances in Artificial Life. Proceedings of the 7th European Conference on Artificial Life (ECAL 2003)* (pp. 76–89). Berlin: Springer-Verlag.

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

14. Helbing, D., & Lämmer, S. (2008). Managing complexity: An introduction. In D. Helbing (Ed.), *Managing Complexity: Insights, Concepts, Applications* (pp. 1–16). Berlin: Springer-Verlag.

15. Klyubin, A., Polani, D., & Nehaniv, C. (2005). All else being equal be empowered. In M. Capcarrere, A. A. Freitas, P. J. Bentley, C. G. Johnson, & J. Timmis (Eds.), *Advances in Artificial Life. Proceedings of the 8th European Conference on Artificial Life (ECAL 2005)* (pp. 744–753). Berlin: Springer-Verlag.

16. Klyubin, A., Polani, D., & Nehaniv, C. (2005). Empowerment: A universal agent-centric measure of control. In *Proceedings of the 2005 IEEE Congress on Evolutionary Computation* (pp. 128–135). Piscataway, NJ: IEEE Computer Society Press.

17. Martius, G., & Herrmann, J. M. (2010). Taming the beast: Guided self-organization of behavior in autonomous robots. In S. Doncieux, B. Girard, A. Guillot, J. Hallam, J.-A. Meyer, & J.-B. Mouret (Eds.), *From Animals to Animats 11: 11th International Conference on the Simulation of Adaptive Behavior (SAB 2010)* (pp. 50–61). Berlin: Springer-Verlag.

18. Martius, G., Herrmann, J. M., & Der, R. (2007). Guided self-organization for autonomous robot development. In *Advances in Artificial Life. Proceedings of the 9th European Conference on Artificial Life (ECAL 2007)* (pp. 766–775). Berlin: Springer-Verlag.

19. Meyer, J.-A., Husbands, P., & Harvey, I. (1998). Evolutionary robotics: A survey of applications and problems. In P. Husbands & J.-A. Meyer (Eds.), *Proceedings of the First European Workshop on Evolutionary Robotics (EvoRobot98)* (pp. 1–21).

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

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

22. Mitri, S., Floreano, D., & Keller, L. (2009). The evolution of information suppression in communicating robots with conflicting interests. *Proceedings of the National Academy of Sciences*, *106*(37), 15786–15790.

23. Mitri, S., Floreano, D., & Keller, L. (2011). Relatedness influences signal reliability in evolving robots. *Proceedings of the Royal Society B*, *278*, 378–383.

24. Mondada, F., Pettinaro, G. C., Guignard, A., Kwee, I. V., Floreano, D., Deneubourg, J.-L., Nolfi, S., Gambardella, L. M., & Dorigo, M. (2004). SWARM-BOT: A new distributed robotic concept. *Autonomous Robots*, *17*(2–3), 193–221.

25. Nelson, A. L., Barlow, G. J., & Doitsidis, L. (2009). Fitness functions in evolutionary robotics: A survey and analysis. *Robotics and Autonomous Systems*, *57*(4), 345–370.

26. Nolfi, S., & Floreano, D. (2000). *Evolutionary robotics: The biology, intelligence, and technology of self-organizing machines*. Cambridge, MA: MIT Press.

27. Pfeifer, R., Lungarella, M., & Iida, F. (2007). Self-organization, embodiment, and biologically inspired robotics. *Science*, *318*(5853), 1088–1093.

28. Prokopenko, M. (2007). Design vs. self-organization. In M. Prokopenko (Ed.), *Advances in applied self-organizing systems* (pp. 3–17). Berlin: Springer-Verlag.

29. Prokopenko, M. (2009). Guided self-organization. *HFSP Journal*, *3*(5), 287–289.

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

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

32. Siciliano, B., & Khatib, O. (Eds.). (2008). *Springer Handbook of Robotics*. Berlin: Springer-Verlag.

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

34. Strogatz, S. H. (2003). *Sync: The emerging science of spontaneous order*. New York: Hyperion Press.

35. Strogatz, S. H., & Stewart, I. (1993). Coupled oscillators and biological synchronization. *Scientific American*, *269*(6), 102–109.

36. Trianni, V. (2008). *Evolutionary swarm robotics. Evolving self-organising behaviours in groups of autonomous robots*. Berlin: Springer-Verlag.

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

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

39. Trianni, V., Nolfi, S., & Dorigo, M. (2008). Evolution, self-organisation and swarm robotics. In C. Blum & D. Merkle (Eds.), *Swarm intelligence. Introduction and applications* (pp. 163–192). Berlin: Springer-Verlag.

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

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