

## Open Issues in Evolutionary Robotics

**Fernando Silva**

fsilva@di.fc.ul.pt

Bio-inspired Computation and Intelligent Machines Lab, 1649-026 Lisboa, Portugal  
BioISI, Faculdade de Ciências, Universidade de Lisboa, 1749-016 Lisboa, Portugal  
Instituto de Telecomunicações, 1049-001 Lisboa, Portugal

**Miguel Duarte**

miguel\_duarte@iscte.pt

Bio-inspired Computation and Intelligent Machines Lab, 1649-026 Lisboa, Portugal  
Instituto de Telecomunicações, 1049-001 Lisboa, Portugal  
Instituto Universitário de Lisboa (ISCTE-IUL), 1649-026 Lisboa, Portugal

**Luís Correia**

luis.correia@ciencias.ulisboa.pt

BioISI, Faculdade de Ciências, Universidade de Lisboa, 1749-016 Lisboa, Portugal

**Sancho Moura Oliveira**

sancho.oliveira@iscte.pt

Bio-inspired Computation and Intelligent Machines Lab, 1649-026 Lisboa, Portugal  
Instituto de Telecomunicações, 1049-001 Lisboa, Portugal  
Instituto Universitário de Lisboa (ISCTE-IUL), 1649-026 Lisboa, Portugal

**Anders Lyhne Christensen**

anders.christensen@iscte.pt

Bio-inspired Computation and Intelligent Machines Lab, 1649-026 Lisboa, Portugal  
Instituto de Telecomunicações, 1049-001 Lisboa, Portugal  
Instituto Universitário de Lisboa (ISCTE-IUL), 1649-026 Lisboa, Portugal

---

### Abstract

One of the long-term goals in evolutionary robotics is to be able to automatically synthesize controllers for real autonomous robots based only on a task specification. While a number of studies have shown the applicability of evolutionary robotics techniques for the synthesis of behavioral control, researchers have consistently been faced with a number of issues preventing the widespread adoption of evolutionary robotics for engineering purposes. In this article, we review and discuss the open issues in evolutionary robotics. Firstly, we analyze the benefits and challenges of simulation-based evolution and subsequent deployment of controllers versus evolution on real robotic hardware. Secondly, we discuss evolutionary computation-specific issues that have plagued evolutionary robotics: (i) the bootstrap problem, (ii) deception, and (iii) the role of the genomic encoding and of the genotype-phenotype mapping in the evolution of controllers for complex tasks. Finally, we address the absence of standard research practices in the field. In addition to the review and discussion of the issues, we further discuss a number of promising avenues of research. Our underlying motivation is the reduction of the current gap between evolutionary robotics and mainstream robotics, and the establishment of evolutionary robotics as a canonical approach for the engineering of autonomous robots.

### Keywords

Evolutionary robotics, autonomous robots, controller synthesis, reality gap, online evolution, bootstrap problem, deception, genomic encoding, research practices.

## 1 Introduction

Evolutionary algorithms (EAs) have been the subject of significant progress since the introduction of the concept of evolutionary search by Turing (1950). Early pioneering work includes Barricelli's seminal contributions and computational experiments since 1953 on symbiogenesis (see Barricelli (1962); Fogel (2006)), and the studies by Fraser (1957) on the effects of selection in epistatic systems. Afterwards, the introduction of genetic algorithms by Holland (1962) as a computational abstraction of Darwin's theory of evolution promised to transfer the adaptation capabilities of natural organisms to different types of artificial agents, including autonomous robots. The envisioned possibilities inspired a new field of research, now called *evolutionary robotics* (ER) (Nolfi and Floreano, 2000).

ER aims to automatically synthesize robotic body plans and control software by means of evolutionary computation (Floreano and Keller, 2010). The field has diverged in two directions: one concerned with cognitive science (Harvey et al., 2005) and biology (Auerbach and Bongard, 2014; Elfwing and Doya, 2014; Lehman and Stanley, 2013), the other focused on using ER techniques for engineering purposes. Our interest lies in the second category, in which the long-term goal is to obtain a process capable of automatically designing and maintaining an efficient robotic system given only a specification of the task (Doncieux et al., 2011). Specifically, we focus on artificial evolution of *control systems* to govern the behavior of robots. In this respect, the use of evolutionary principles has been largely argued as necessary to replace inefficient preprogrammed approaches, see Harvey et al. (1997); Leger (2000); Lipson and Pollack (2000); Nolfi and Floreano (2000); Quinn et al. (2003) for examples.

In ER, the experimenter often relies on a self-organization process (Nolfi, 1998) in which evaluation and optimization of controllers are holistic, thereby eliminating the need for manual and detailed specification of the desired behavior (Doncieux et al., 2011). Traditional approaches consist of optimizing a population of genomes in *genotype space*. Each genome encodes a number of parameters of the robots' control system, the *phenotype*. For example, if the control system is an artificial neural network, the connection weights can be represented at genome level in a real-valued vector, while finite state machine-based controllers can be described by automaton-based representations (König et al., 2009). Optimization of genomes is based on Darwin's theory of evolution, namely blind variations and survival of the fittest, as embodied in the neo-Darwinian synthesis (Gould, 2002). The mapping from genotype to phenotype can capture different properties of the developmental process of natural organisms, and the phenotype can abstract various degrees of biological realism (Stanley and Miikkulainen, 2003). ER thus draws inspiration from biological principles at multiple levels.

After two decades of research in ER, controllers have been successfully evolved for robots with varied functionality, from terrestrial robots to flying robots (Floreano et al., 2005). Although there has been significant progress in the field, it is arguably on a scale that still precludes the widespread adoption of ER techniques (Silva et al., 2014b). Evolved controllers are in most cases not yet competitive with human-designed solutions (Doncieux et al., 2011), and have only proven capable of solving relatively simple tasks such as obstacle avoidance, gait learning, and search tasks (Nelson et al., 2009). In effect, a number of critical issues currently prevent ER from becoming a viable mainstream approach for engineers. In our view, the most relevant issues are: (i) the reality gap (Jakobi, 1997), which occurs when controllers evolved in simulation become ineffective once transferred to the physical robot, (ii) the prohibitively long time necessary to evolve controllers directly on real robots (Matarić and Cliff, 1996),

(iii) the bootstrap problem when solutions to complex tasks are sought (Nelson et al., 2009), (iv) deception (Whitley, 1991), (v) the design of genomic encodings and of the genotype-phenotype mappings that enable the evolution of complex behaviors (Meyer et al., 1998), and (vi) the absence of standard research practices in the field (Doncieux et al., 2011). Importantly, while issues such as the bootstrap problem and deception are inherent to the evolutionary computation approach, other issues such as the reality gap and the typically time-consuming evolution of controllers on real robots are specific to ER. In addition, the differences between ER and more traditional domains mean that there is currently little theory and formal methods that can be applied to ER. Consider, for instance, the analysis of fitness landscapes. In traditional domains, multiple methods can be used to elucidate the properties of the search space and of fitness landscapes. In ER, however, search spaces and corresponding fitness landscapes are challenging to characterize. Search spaces are generally rugged to an extreme degree, may have varying numbers of dimensions, and may potentially be non-static. In this way, full characterization of search spaces and of fitness landscapes is often an intractable problem (Nelson et al., 2009).

ER has the potential to cast the performance of EAs in a new perspective by focusing on its unique dimensions. For example, while traditional EAs are driven by a fitness function, an alternative class of methods has recently emerged in ER: because the goal is to find a suitable robot *behavior*<sup>1</sup>, the evolutionary process can also be driven by the search for novel behaviors, which potentially avoids bootstrapping issues and deception (Lehman and Stanley, 2011a; Lehman et al., 2013; Mouret and Doncieux, 2012). For example, to evolve controllers for maze-navigating robots, the fitness function can be defined based on how close the robot is to the goal at the end of an evaluation (Lehman and Stanley, 2011a). If a maze has no deceptive obstacles, this fitness function creates a monotonic gradient for the search to follow. However, mazes with obstacles that prevent a direct route may create local optima in the fitness landscape and deceive traditional fitness schemes. In this context, if the behavior of a maze navigator is characterized by its final position, searching for novel behaviors has the potential to avoid deception.

The key goal of this article is to highlight the open issues in ER. We review key contributions to the field, discuss the current challenges and limitations, and propose a number of research avenues for the development of ER as an engineering tool. While the automatic synthesis of complete robotic systems remains a long-term research goal, ER has the potential to become a canonical approach in mainstream robotics provided that the current issues faced by researchers in the field can be addressed. We expect that our review and discussion may help researchers to focus on addressing these issues in ER.

This article is organized as follows. In Section 2, we review and analyze the problem of the reality gap and the prohibitive amount of time required to evolve controllers directly in real robotic hardware. In Section 3, we discuss current approaches to overcome the bootstrap problem and deception. In Section 4, we examine the importance of the genomic encoding and of the genotype-phenotype mapping when scaling ER techniques to complex tasks. In Section 5, we argue for the adoption of revised robotics engineering and experimental science practices in order to accelerate progress in the

---

<sup>1</sup>There has been no consensus in previous studies concerning whether or not the behavior should be considered part of the phenotype (Mouret and Doncieux, 2012). Here, we distinguish between the concepts of *behavior* and *phenotype* (often used as a synonym of *controller*), to make our description of behavioral diversity-based methods and of genotype-phenotype mappings as clear as possible.

field. Finally, in Section 6, we conclude the article and summarize our contribution.

## 2 Evolving Controllers in Simulation and in the Real World

In the early years of ER, contributions such as that of Harvey et al. (1993) and of Floreano and Mondada (1994) laid the foundation for a number of important studies that followed. In particular, after Floreano and Mondada's study on the evolution of controllers on real hardware, a number of contributions focused on developing algorithms that explicitly conduct evolution *online*, and using the robot's typically limited computational resources, that is, *onboard*. The goal of the approach is to enable robots to operate in a self-contained manner (Bredeche et al., 2009), without relying on external entities for assessment of performance or additional computing power. To that end, the main components of the EA (evaluation, selection, and reproduction) are carried out autonomously by the robots without any external supervision. The main advantage of online evolution is that if the environmental conditions or task requirements change, the robots can modify their behavior to cope with the new circumstances. However, given its trial-and-error nature, EAs typically require the evaluation of a large number of candidate solutions to the task. Combined with the fact that each evaluation can take a significant amount of time on real robots, the approach still remains unfeasible (Silva et al., 2014b).

An alternative approach is to synthesize controllers *offline*, in simulation, and then transfer them to real robots post-evolution to avoid the time-consuming nature of performing all evaluations on real robotic hardware. A central issue with the simulate-and-transfer approach is the reality gap (Jakobi, 1997). Controllers evolved in simulation can become ineffective once transferred to the physical robot due to their exploitation of features of the simulated world that are different or that do not exist at all in the real world. Differences between simulation and the real world can vary, from inaccurate sensor modeling to simulation-only artifacts due to simplifications, abstractions, idealizations, and discreteness of physics implementations. For example, if a simulation employs an ideal sensor setting with no noise, an EA can synthesize a controller whose behavior is based on a very narrow range of sensor values. If the controller is then transferred to a physical robot with sensor value variations due to issues such as electronic or mechanical limitations, the physical robot is unlikely to behave as in the simulation. Overall, the difficulty of accurately simulating physical systems is well known in robotics (Mataric and Cliff, 1996). In ER, the reality gap is a frequent phenomenon and one of the main impediments for progress (Koos et al., 2013).

### 2.1 Crossing the Reality Gap

A number of approaches have been introduced to cross the reality gap, see Fig. 1. Miglino et al. (1995) proposed three complementary approaches: (i) using samples from the real robots' sensors, (ii) introducing a conservative form of noise in simulated sensors and actuators, and (iii) continuing evolution for a few generations in real hardware if a decrease in performance is observed when controllers are transferred. Using samples from real sensors increases the accuracy of simulations by using a more realistic sensor model, which in turn can decrease the difference between the sensory input experienced in simulation and in reality. Noise can be applied to promote the evolution of robust controllers that can better tolerate variations in the sensory inputs during task execution. Finally, if the performance of the controller decreases after transfer, continuing evolution on real hardware can potentially enable the synthesis of a well-adapted controller in a timely manner.

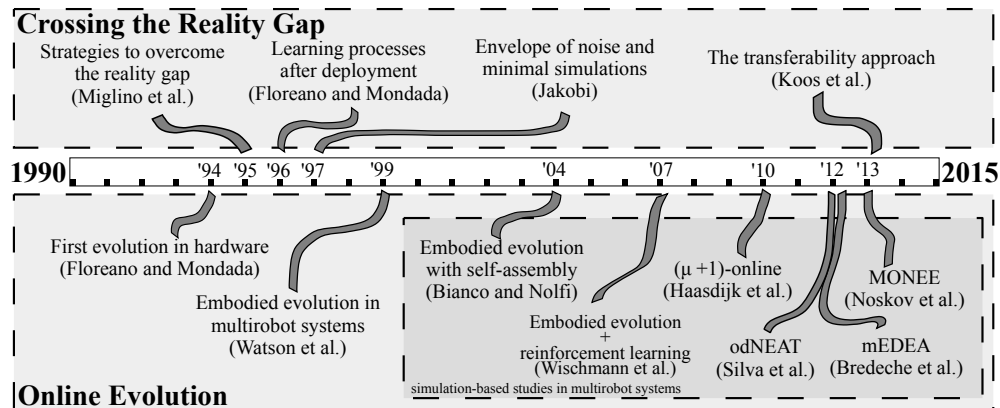


Figure 1: A chronogram of selected studies addressing the issue of the reality gap (top half) and online evolution (bottom half). Although one of the goals of online evolution is to confine the evolutionary process to real robotic hardware, experiments reported in recent studies have been conducted mainly in simulation.

After Miglino et al.'s study, sensor sampling and conservative noise methods have become widespread, but continuing evolution in real hardware has not been frequently used, despite promising results (Nolfi et al., 1994). From 1996 to 2001, a number of pioneering studies (Floreano and Mondada, 1996b, 1998; Floreano and Urzelai, 2000, 2001; Urzelai and Floreano, 2001) focused on how evolved Hebbian learning rules could be used to minimize the differences between simulation and the real world. In Floreano and Urzelai (2000, 2001); Urzelai and Floreano (2001), adaptive controllers were shown to cope well with the transfer from simulation to reality in a sequential light-switching task. Regardless of the promising results, using learning processes to bridge the reality gap has not been frequently studied in ER. Nonetheless, recent years have seen a resurgence of interest in controllers with online adaptation capabilities, as reviewed by Coleman and Blair (2012).

Despite the widespread adoption of the sensor sampling and of the conservative noise approaches, none of these methods is inherently scalable. For example, as discussed by Bongard (2013), consider that robots have to operate in an environment with irregular or asymmetric objects. The sensor sampling method would require that the objects are sampled from a large number of positions because there are multiple unique readings of the objects from the robot's vantage point. If the conservative noise method is used, the controllers have to be evaluated in a potentially large number of scenarios to ensure that the behavior evolved is robust. As the complexity of the robots increases, noise will have to be added to a greater number of sensors and actuators. Because there are additional noise sources, significantly more evaluations may be required to evolve effective controllers (Bongard, 2013). Furthermore, the amount of noise added to each set of sensors and actuators has to be carefully determined because too much noise may conceal the essential features for solving the task, while too little noise may prevent successful transfer to real hardware.

Jakobi (1997, 1998) introduced the concept of *minimal simulations*, in which the experimenter only implements features of the real world deemed necessary for successful evolution of controllers. All remaining features are hidden in an "envelope of noise" in order to minimize the effects of simulation-only artifacts that can prevent successful

Table 1: Summary of approaches introduced to cross the reality gap.

Approach	Tasks	Robot type
Sensor sampling (Miglino et al., 1995)	Navigation and obstacle avoid.	Wheeled
Conservative noise (Miglino et al., 1995)		
Evolution and learning (Floreato and Mondada, 1996b, 1998; Floreato and Urzelai, 2000, 2001; Urzelai and Floreato, 2001)	Navigation and obstacle avoid. Sequential light-switching	
Minimal simulations (Jakobi, 1997, 1998)	T-maze navigation Shape recognition Locomotion & obstacle avoid.	Wheeled Gantry 8-legged
Transferability approach (Koos et al., 2013)	T-maze navigation Locomotion	Wheeled 4-legged

transfer of evolved control to real robotic hardware. The approach was demonstrated in three tasks. The first task was a T-maze navigation task where a wheeled robot had to choose whether to turn left or right at an intersection depending on the location of a light source in the initial corridor. The second task was a shape discrimination task, in which a gantry robot had to distinguish between two shapes and move towards one of them. The third task was a locomotion and obstacle avoidance task for an eight-legged robot. It is not clear if Jakobi's approach scales well to complex tasks, since such tasks: (i) typically involve richer robot-environment interactions, and therefore more features, and (ii) require that the experimenter can determine the set of relevant features and build a task-specific simulation model. For example, if the tasks considered involve a large number of robots or robots with high-resolution sensory capabilities such as vision, minimal simulations call for considerable engineering effort because the critical simulation features become more difficult to ascertain and to model (Watson et al., 2002).

Koos et al. (2013) proposed the *transferability approach*, a multiobjective technique in which controllers are evaluated based on their performance in simulation and on the real robots. Contrarily to approaches that simply use individual fitness comparisons of reality versus simulation as a feedback to adapt the simulation model (Zagal and Ruiz-Del-Solar, 2007), the goal of the transferability approach is to *learn the discrepancies* between simulation and reality, and to constrain evolution in order to avoid behaviors that do not cross the reality gap effectively. The transferability approach relies on a surrogate model that is updated periodically by evaluating candidate solutions in real hardware. Koos et al. (2013) tested the approach in a T-maze navigation task with a differential-drive robot, and in a locomotion task with a quadruped robot. In both tasks, the transferability approach was able to find a solution to the task in relatively few generations (100 or less). However, the approach can become unfeasible if several hundreds or thousands of generations are required. Moreover, the difficulty in automatically evaluating controllers in real hardware represents an additional challenge.

Table 1 summarizes the tasks studied and the types of robots used in the assessment of different approaches for crossing the reality gap. Overall, Jakobi's minimal simulations (Jakobi, 1997, 1998) and Koos et al.'s transferability approach (Koos et al., 2013) are currently the most effective methods. Although these two approaches represent the current state of the art in formalized and engineering-oriented methods, the problem of how to effectively cross the reality gap has no generally recognized solution at this point in time. That is, while a number of different approaches have been proposed, none of them has been shown to be a general solution to the reality gap problem. The reality gap is therefore, to the extent that it currently occurs, a significant obstacle to widespread adoption of ER techniques for engineering purposes because it prevents controllers synthesized via simulation-based ER techniques from being used on real robots without additional, and often ad-hoc, tuning. In addition, the different approaches to help cross the reality gap have not been extensively compared and, their relative advantages, disadvantages, and adequacy have thus not yet been assessed in detail.

## 2.2 Speeding up Evolution in Real Hardware

One way to eliminate the reality gap is to rely exclusively on real robots. The first studies on online evolution in a real, neural network-driven mobile robot were performed by Floreano and Mondada (1994, 1996a). In their studies, the authors successfully evolved navigation and homing behaviors for a Khepera robot. Evolution was based on an online generational EA, but with the actual computation being performed on a workstation due to the limitations of the robot hardware. The synthesis of successful controllers required up to ten days of continuous evolution on real robots, thus indicating that the evaluation time is a key aspect in real-robot experiments. Even the evolution of controllers for a standard navigation and obstacle avoidance task required approximately 2.71 days (population of 80 genomes, 100 generations, 39 minutes per generation).

Since online evolution proved to be prohibitively challenging, researchers have focused on the problems posed by evolving controllers directly on physical robots (Matarić and Cliff, 1996). Several established methods exist to reduce the number of evaluations required or cut short an individual evaluation. Examples include as early stopping algorithms (Bongard, 2011) and racing techniques (Haasdijk et al., 2011). However, one of the goals of ER is to realize robots capable of operating in noisy or dynamic environments (Bongard, 2009), and that can execute multiple tasks in parallel or in sequence (Nolfi, 2002). As a result, there are several scenarios in which a proper estimate of controller performance cannot be made in a small amount of time (e.g. a few seconds) (Matarić and Cliff, 1996), thus making the evaluation time in real hardware a practical challenge.

To address the time-consuming nature of online evolution, different approaches have been introduced, as illustrated in Fig. 1. A pioneering approach called *embodied evolution* was introduced by Watson et al. (1999, 2002). In embodied evolution, the EA is distributed across a group of robots. The use of multirobot systems is motivated by an anticipated speed-up of evolution due to the inherent parallelism in such systems. The exchange of genetic information between robots is a form of knowledge transfer that offers a substrate for speeding up the evolutionary process and for collective problem solving (Bredeche et al., 2012; Silva et al., 2015b). Following Watson et al.'s studies on embodied evolution, a number of algorithms for online evolution in multirobot systems were introduced. Examples include an approach for self-assembling of robots

Table 2: Summary of online evolution approaches introduced starting from the point embodied evolution was developed (Watson et al., 1999, 2002)

Approach	Tasks	Real robot
Embodied evolution (Watson et al., 1999, 2002)	Phototaxis	Yes
Embodied evolution with self-assembly (Bianco and Nolfi, 2004)	Open-ended survival	No
Embodied evolution + reinforcement learning (Wischmann et al., 2007)	Predator vs. prey pursuit	No
$(\mu + 1)$ -online (Haasdijk et al., 2010)	Navigation and obstacle avoidance	No
mEDEA (Bredeche et al., 2012)	Dynamic phototaxis	Yes
odNEAT (Silva et al., 2015d, 2012)	Aggregation; dynamic phototaxis; navigation and obstacle avoidance	No
MONEE (Noskov et al., 2013)	Concurrent foraging	No

by Bianco and Nolfi (2004), the combination of embodied evolution and reinforcement learning by Wischmann et al. (2007),  $(\mu + 1)$ -online by Haasdijk et al. (2010), mEDEA by Bredeche et al. (2012), odNEAT by Silva et al. (2015d, 2012), and MONEE by Noskov et al. (2013).

After embodied evolution was introduced, different tasks and types of robots have been used in online evolution studies, as shown in Table 2. Despite the progress, few studies on online evolution have been conducted on real robots (Koos et al., 2013; Nelson et al., 2009). Researchers have evaluated their proposed algorithms mainly through online evolution in simulation. As a result, even though new algorithms have been introduced and online evolution approaches have matured, the state-of-the-art has not yet reached the point at which robots can adapt online in a timely manner (e.g. within minutes or hours), and the class of tasks addressed has not increased in complexity over the past years (Silva et al., 2015a, 2014b).

### 2.3 Combining Offline Evolution and Online Evolution

The ER studies reviewed in the previous section show an antagonistic relationship between offline evolution and online evolution. An interesting view is that offline evolution and online evolution can complement each other, that is, the benefits of each approach can be exploited to effectively bypass each other's limitations. The availability of relatively fast, generic, and open-source physics engines, such as ODE<sup>2</sup> and Bullet<sup>3</sup>, enable the development of simulations in which offline evolution can be used as an *initialization procedure*: approximate solutions are synthesized and transferred to

<sup>2</sup>ODE homepage: <http://www.ode.org>

<sup>3</sup>Bullet homepage: <http://bulletphysics.org/>



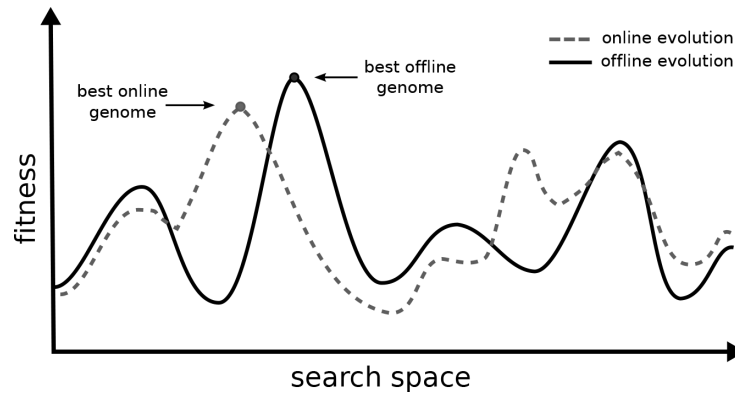


Figure 2: Simplified 1D illustration of how the fitness landscape can vary in simulation and in real robotic hardware (search spaces in ER typically have tens, hundreds, or even thousands of dimensions). The fitness of solutions found in simulation typically does not match the one obtained when controllers are transferred to a real robot, resulting in sub-optimal performance. This difference in performance is known as the reality gap, and it occurs because either controllers in simulation exploited unrealistic phenomena which are not present in the real world, or the modeling of the sensors, actuators, and environment is not sufficiently accurate.

real robots. Alternatively, minimal simulation-based approaches could be used but they would potentially require a larger engineering effort to construct an appropriate simulation environment (see Section 2.1). After the deployment of controllers, online evolution can serve as a *refinement procedure* that adapts solutions evolved offline to cope with differences between simulation and the real world as illustrated in Fig. 2, and to changing or unforeseen circumstances. To enable controllers to be further optimized online, the highest performing controllers of offline evolution could be specified, for instance, as the initial population of the online EA or defined as a functional module of the new controllers (Silva et al., 2014a). In this methodology, the key decision is therefore where to draw the line between which behavioral competences should be evolved offline and which should be evolved online.

As an example of the combination of offline evolution and online evolution, consider the case of multirobot exploration of unknown or dynamic environments, a field of research with a number of practical applications (de Hoog et al., 2010). Suppose that a group of robots has to explore a large environment, retrieve scattered objects, and transport them to a base station. Two fundamental competences need to be evolved: (i) navigation and obstacle avoidance, an essential feature for autonomous robots operating in real-world environments, and (ii) fine sensorimotor coordination to perform successful retrieval of objects. Offline evolution can be used to synthesize controllers that simultaneously avoid collisions with obstacles such as walls, and effectively search for and approach the target objects. Because fine sensorimotor coordination is an interaction difficult to model accurately in simulation (Silva et al., 2014b), such behavioral competence could be evolved on the real robots by building on top of the controllers synthesized offline. As a result of this combined offline-online evolution, the reality gap would potentially be reduced and the online evolutionary process would be accelerated because partial solutions to the task are already available.

An alternative approach is the onboard combination of simulation-based evolution and online evolution. In this scenario, each robot maintains: (i) a model of the environment and of other robots, and (ii) a simulator in which candidate solutions are tested. The highest performing solutions are then used by the robot, and any differences regarding the expected performance levels lead to the adaptation of the components of the internal model (O'Dowd et al., 2011), such as: (i) robot-robot correspondence, that is, physical aspects of the robots such as their morphology, (ii) robot-environment correspondence, that is, minimization of differences exhibited in the interactions between a robot and the environment, including sensor readings and actuation values, and (iii) environment-environment correspondence of the features of the environment. The result is a self-improving, self-contained system that can synthesize, evaluate, and deploy solutions. In effect, a number of studies have been conducted in this direction using simulation-based experiments (Bongard and Lipson, 2004, 2005; De Nardi and Holland, 2008), and real robots (Bongard et al., 2006; Bongard, 2009; O'Dowd et al., 2011).

To combine simulation-based evolution and online evolution on real robots, notable approaches have been introduced by Bongard et al. (2006); Bongard (2009), and by O'Dowd et al. (2011). Bongard et al. (2006); Bongard (2009) introduced an approach that employs three EAs: (i) the first algorithm optimizes a population of physical simulators in order to more accurately model the real environment, (ii) the second algorithm optimizes exploratory behaviors for the real robot to execute in order to collect new training data for the first EA, and (iii) the third EA then uses the best simulator to evolve locomotion behaviors for a real quadruped robot. Besides increasing the number of successful controllers, the combination of the three EAs yields an important advantage: enabling the robot to recover from unanticipated situations such as physical damage to one of its legs (Bongard et al., 2006).

O'Dowd et al. (2011) introduced a conceptually similar approach based on two EAs. The first EA is distributed across a population of robots, similarly to a physically embodied island model (Tanese, 1989). Each robot optimizes a simulation model of the environment, which can be transmitted to nearby robots. The second EA is private to each robot and is used to optimize the controllers. The action flow of the approach is composed of five steps repeated periodically: (i) optimizing a population of solutions using the embedded simulator, (ii) transferring the best controller to the real robot, (iii) assessing the performance of such controller, (iv) transmitting simulator genomes and real-robot fitness scores to other robots, and (v) optimizing the current population of genomes. O'Dowd et al.'s approach was assessed in a foraging task in which robots had to search for food items, and then deposit them at a nest site marked by a light source. A key result of the study was that the environment-environment correspondence achieved via the embedded simulator was able to cope well with changes in the real environment, namely when the light source was moved from one end of the environment to the other.

Overall, Bongard et al. (2006); Bongard (2009), and O'Dowd et al. (2011) have introduced new perspectives on how to cross the reality gap and speed up evolution of controllers on real robots. In the future, Moore's law may continue to contribute to an increase in computational power, and consequently to the improvement of onboard simulation capabilities. The potential gain in performance will, however, be dependent on how EAs will scale with task complexity. Future work on onboard simulation and evolution can also benefit from assessing how multiple collaborating robots can accelerate the modeling process and consequently the synthesis of suitable behaviors. Robots

could, for instance, exchange information about the environment as in O'Dowd et al.'s approach *and* about each other's body, or create collective models, thereby potentially reducing the time necessary for model search and for controller synthesis.

### 3 The Bootstrap Problem and Deception

A number of authors have reported that EAs can generate simple and efficient solutions seldom found when using classic engineering methods, see Doncieux and Meyer (2003); Floreano and Mondada (1994); Lipson and Pollack (2000) for examples. In this context, one of the main advantages of EAs is their ability to optimize robot controllers given only a fitness function based on a high-level description of the task to be solved. However, as pointed out by Mouret and Doncieux (2009a), a significant portion of successful ER studies using such fitness functions omit discussion of initial unsuccessful attempts to evolve more complex behaviors. The reason is that during the evolutionary synthesis of controllers, search may get stalled due to two issues: the bootstrap problem and deception.

The bootstrap problem (Gomez and Miikkulainen, 1997) occurs when the task is too demanding for the fitness function to apply any meaningful selection pressure on a randomly generated population of initial candidate solutions. All individuals in the early stages of evolution may perform equally poorly, causing evolution to drift in an uninteresting region of the search space. Deception (Whitley, 1991) occurs when the fitness function fails to build a gradient that leads to a global optimum, and instead drives evolution towards local optima. As a result, the evolutionary algorithm may converge prematurely to a sub-optimal solution, that is, it may *stagnate*.

Contrary to the ER-specific issues discussed in Section 2, the bootstrap problem and deception are inherent to the evolutionary approach in general. These problems are particularly problematic in ER because fitness plateaus are common (Smith et al., 2001) and deception often manifests itself if some degree of behavior adaptation is required during task execution (Risi et al., 2010). The more complex the task, the more susceptible evolution is to bootstrapping issues and to be trapped by local optima (Lehman and Stanley, 2011a; Zaera et al., 1996).

#### 3.1 Assisting the Evolutionary Process with Human Knowledge

From an engineering perspective, one solution to address the bootstrap problem and deception is to directly assist the evolutionary process (Mouret and Doncieux, 2008). In this context, three approaches have been widely adopted: (i) *incremental evolution*, in which a task is decomposed into different components in a top-down fashion, (ii) *behavioral decomposition*, in which the decomposition is performed in a bottom-up manner, and (iii) semi-interactive *human in-the-loop* approaches.

##### 3.1.1 Incremental Evolution

In incremental evolution, a task is decomposed into different components which are easier to solve individually. There are several ways to apply incremental evolution (Mouret and Doncieux, 2008), such as dividing the task into sub-tasks, or making the task progressively more difficult through environmental complexification (Christensen and Dorigo, 2006).

In typical incremental evolution approaches, the experimenter performs a manual switch between the execution of each component of the evolutionary setup, e.g. different sub-tasks or increasingly complex environments. If the switch from one component to the next is made too early, solutions found for the first component may not be suit-

able; if the switch is triggered too late, solutions may overfit the current component, and the global task therefore becomes even more difficult to solve. In addition, the order in which the different components are evolved can have a significant impact on the performance of the evolutionary process, see Auerbach and Bongard (2009); Bongard (2008) for examples. To minimize these biases, Mouret and Doncieux (2008) proposed to define the components of the global tasks independently in a multiobjective context, a scheme that removes the aforementioned adverse aspects of incremental approaches. The multiobjective approach was assessed in a combined obstacle avoidance and light-switching task that required eight sub-tasks to be solved. The only requirement was for the experimenter to specify the sub-components of the global task. However, as shown by Christensen and Dorigo (2006) in a combined phototaxis and hole avoidance task, if the components of the task are highly integrated, such specification can be difficult or even impossible to perform.

### 3.1.2 Behavioral Decomposition

In behavioral decomposition, the robot controller is divided into sub-controllers, and each sub-controller is either preprogrammed or evolved separately to solve a different sub-task. The final controller is then composed by combining the sub-controllers in a bottom-up fashion via a second evolutionary process. The behavioral decomposition approach has been used to synthesize, for example, homeostatic-inspired GasNet controllers (Moioli et al., 2008), genetic programming-based controllers (Lee, 1999), hierarchical or multilayer ANN-based controllers (Duarte et al., 2012; Larsen and Hansen, 2005; Togelius, 2004), and hybrid controllers in which evolved ANNs and preprogrammed behaviors are combined (Duarte et al., 2014a,b, 2015). Nonetheless, as in incremental evolution, successful behavioral decomposition: (i) requires detailed knowledge of the task because multiple evolutionary setups have to be defined and configured, and (ii) is prone to the introduction of potentially negative biases by the experimenter.

### 3.1.3 Human in-the-loop

A different approach to assist the evolutionary process consists of introducing a human in-the-loop to create a semi-interactive evolutionary process. The key idea is to enable users to guide evolution away from local optima by indicating intermediate states that the robot must go through during a task. A gradient is then created to guide evolution through the states, the assumption being that successfully reaching certain intermediate states should make a good stepping stone to controllers that will reach more advanced intermediate states. Recently, Celis et al. (2013) investigated this idea in a deceptive object homing task, where a robot has to go around an obstacle in order to reach its goal. The human expert helps evolution by adding waypoints to promote solutions that maneuver the robot around the obstacle. Aside from the formal definition provided by Celis et al., similar concepts have previously been used in an ad-hoc manner, see Bongard et al. (2012); Bongard and Hornby (2013); Risi and Stanley (2012a); Woolley and Stanley (2014) and examples therein. While semi-interactive approaches are appealing because the human user can, for instance, intervene to make his or her preferences or knowledge explicit, there are multiple open questions regarding how such approaches could be applied in complex tasks. If the complexity of the tasks for which solutions are sought increases, additional human intervention may be required and the approach will potentially face the issue of *human fatigue* (Takagi, 2001).

The bootstrap problem and deception are two general issues in the field of ER, and no approach to minimize them has been considered as predominant. One of growing

trends, as reviewed in this section, is to exploit human knowledge to address the problems of the evolutionary approaches. There are multiple reasons why such methodology represents a valuable design tool, one of the most important being that the experimenter can influence how human expertise and evolution are united to more easily overcome each other's limitations. For instance, in challenging tasks, evolution may be seeded with *building blocks* such as pre-evolved or manually-designed behaviors to enable a higher-level bootstrap process (Silva et al., 2014a). Although such a combined approach may lead to an effective synergy, it reduces ER's potential for automation in the design of robotic systems.

### 3.2 Promoting Diversity

In standard ER experiments, a fitness function is used both to define the goal and to guide the evolutionary search towards the goal. In this respect, the bootstrap problem and deception are typically related to finding suitable evolutionary dynamics. For example, a fitness function may accurately describe features of the solutions desired, but fail to build a suitable performance gradient connecting the intermediate controllers of the evolutionary process to a final solution.

A common approach to mitigate the bootstrap problem and deception is to employ a diversity maintenance technique. The key idea is that encouraging diversity may enable an EA to avoid being deceived by exploring multiple different paths through the search space. Diversity in evolutionary computation and ER studies is typically encouraged through metrics or selection mechanisms that operate at genotypic level (Lehman and Stanley, 2011a; Mouret and Doncieux, 2012). Canonical examples include promoting diversity in the space of genotypes, the age of genotypes, or the fitness of genotypes (Lehman et al., 2013). Although genotypic diversity methods facilitate exploration, they can still be deceived in ER tasks if genotypic differences are not correlated with behavioral differences (Lehman and Stanley, 2011a; Lehman et al., 2013).

#### 3.2.1 Novelty Search and Behavioral Diversity

In ER experiments, controllers can be scored based on a characterization of their behavior during evaluation, and not only based on a traditional fitness function, see Fig. 3. In a maze navigation task, the behavior could, for example, be characterized by: (i) the final position of the corresponding robot in the environment, which amounts to a characterization of length 2, or (ii)  $n$  equally-spaced samples of the robot's  $(x, y)$  position taken during an evaluation, in which case the resulting behavioral characterization is the vector  $(x_1, y_1, x_2, y_2, \dots, x_n, y_n)$  (Lehman and Stanley, 2011a).

Based on this insight, Lehman and Stanley (2008, 2011a) introduced *novelty search*, in which the idea is to maximize the novelty of behaviors instead of their fitness, that is, to explicitly search for novel behaviors as a means to bootstrap evolution and to circumvent convergence to local optima. Specifically, the novelty search algorithm uses an archive to characterize the distribution of novel behaviors that are found throughout evolution. The algorithm operates by: (i) computing the *novelty score* of a behavior by measuring the distance to the  $k$ -nearest neighbors, where  $k$  is a fixed parameter that is determined experimentally, and (ii) adding the behavior to the archive stochastically or if it is significantly novel, that is, if the novelty score is above some minimal threshold. Because behaviors from more sparse regions of the behavioral search space receive higher novelty scores, the gradient of search is always towards what is novel, with no explicit objective. This new perspective on how to guide the evolutionary process triggered a significant body of work and added a new dimension not only to ER (Cully

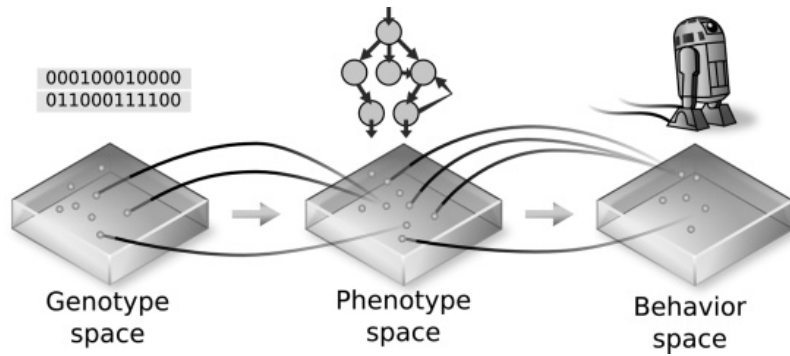


Figure 3: In traditional EAs, genotypic representations of candidate solutions are translated into phenotypes whose fitness is assessed. In ER, the behavior of the robot can also be characterized during task execution, which enables evolution to be guided by the search for novel behaviors instead of fitness. Adapted from Mouret and Doncieux (2012).

et al., 2015; Cully and Mouret, 2015; Doncieux and Mouret, 2014; Mouret and Doncieux, 2012; Mouret and Clune, 2015; Pugh et al., 2015), but to evolutionary computation in general (Goldsby and Cheng, 2010; Liapis et al., 2013; Naredo and Trujillo, 2013).

Given its divergent nature, novelty search has been shown to be unaffected by deception and less prone to bootstrapping issues than fitness-based evolution in a number of tasks, including maze navigation and biped locomotion with variable difficulty (Lehman, 2012; Lehman and Stanley, 2011a,c; Lehman et al., 2013), cognitive learning, memory, and communication tasks (Lehman and Miikkulainen, 2014), and collective robotics tasks such as aggregation and resource sharing (Gomes et al., 2013). Inspired by novelty search, different studies have introduced behavioral diversity-based methods that explicitly reward the novelty or the diversity of behaviors (Mouret and Doncieux, 2012). Novelty search and behavioral diversity methods are, however, significantly dependent on the *behavior characterization*, as shown by Kistemaker and Whiteson (2011), and can be challenging to apply when such a metric is not easy to define. That is, although the aforementioned methods operate independently of fitness, their effectiveness typically depends on a similar form of human knowledge. In addition, if the behavior space is vast, novelty search may not scale well (Doncieux and Mouret, 2014).

### 3.2.2 Reconciling Exploration and Exploitation Procedures

One way to potentially minimize the bootstrap problem and avoid deception is to direct the evolutionary process towards increasing exploration or exploitation of the search space. From the exploration-exploitation perspective, methods such as novelty search are an exploration procedure in the sense that they encourage a more expansive search, and fitness-based algorithms are an exploitation procedure as they typically focus on increasingly narrow regions of the search space.<sup>4</sup>

Different approaches have been introduced to reconcile exploration and exploitation procedures. One is the combination of the novelty score and of the fitness score using a weighted sum, as studied by Cuccu and Gomez (2011) in the deceptive tar-

<sup>4</sup>It should be noted that in an EA, the degree of exploration and of exploitation of the search space may also be conditioned by the genetic operators.

tarus task. Even though the study showed that combining the novelty and fitness scores may lead to better performance, the weights assigned to each score must be fine-tuned because different weighting values can cause significant variations in the results (Cuccu and Gomez, 2011). A second approach is *minimal criteria novelty search* (MCNS) by Lehman and Stanley (2010). MCNS is an extension of novelty search in which individuals must meet one or more domain-dependent fitness criteria to be selected for reproduction. The practical motivation behind MCNS is to reduce the size of behavior spaces to enable a more tractable search for suitable behaviors. The results of Lehman and Stanley's MCNS study on two maze navigation tasks support the hypothesis that reducing the behavior space during a search for novelty can often increase the efficiency of the search process. Although MCNS is an interesting method because it restricts the search for novelty to viable regions of the behavior space, its performance is contingent on a number of aspects, including: (i) choosing the minimal criteria should be done carefully because of the restrictions each criterion puts on the search space, and (ii) if no individuals are found at all that meet the minimal criteria, including in the initial population, there is no selection pressure and the evolutionary search enters a random drift (Lehman and Stanley, 2010).

An alternative approach are the Pareto-based multiobjective EAs (MOEAs) that simultaneously optimize behavioral diversity and fitness (Mouret and Doncieux, 2012; Mouret, 2011; Mouret and Doncieux, 2009b). The multiobjective approach automates the exploration and exploitation phases based on the tradeoffs between the behavioral diversity objective and the fitness objective. Throughout the search, fitness is maximized to the detriment of behavioral diversity at one extreme of the non-dominated Pareto front, while behavioral diversity is maximized at the expense of fitness at the opposite front. The continuum between the two extremes is therefore composed of multiple tradeoffs between performance and diversity. Importantly, while behavioral diversity and fitness are often conflicting objectives, using both objectives enables the search to move in multiple non-dominated directions instead of in a single direction, which allows to more easily bypass local optima (Knowles et al., 2001).

Recent results have demonstrated the potential of MOEAs (Mouret and Doncieux, 2012). As shown by Mouret (2011) in a deceptive maze navigation task, the multiobjective approach can fine-tune behaviors more effectively than pure novelty search. In an extensive study, Lehman et al. (2013) showed that as the task difficulty increases, methods that combine novelty and fitness in a multiobjective manner typically outperform methods that rely, for instance, on novelty search or on fitness-based search alone. Importantly, the multiobjective formulations described before combine behavioral diversity and fitness at a *global level*. A different formulation is *novelty search with local competition* (NSLC) (Lehman and Stanley, 2011b). In NSLC, the fitness objective is changed from being a global measure to being one relative to a controller's neighborhood of *behaviorally similar* individuals. In practice, when the novelty score of a controller is computed, the number of nearest neighbors with lower fitness is also counted. This number is assigned as the *local competition* objective for that individual, which measures a controller's performance with respect to its behavioral niche. In this way, the two objectives become the novelty score and the local competitiveness score.

In the context of evolution of robot body plans for virtual walking creatures, Lehman and Stanley (2011b) have shown that NSLC can effectively maintain and exploit a diversity of individuals at the expense of absolute performance. NSLC has since then been used as the basis of a number of recent algorithms such as: (i) the *Behavioral Repertoire* algorithm (Cully and Mouret, 2013), which employs NSLC to collect a

variety of different, high-performing behaviors throughout evolution instead of a single, general behavior, and (ii) the *Transferability-based* Behavioral Repertoire (Cully and Mouret, 2015), which combines the NSLC with the transferability approach (Koos et al., 2013) discussed in Section 2.1. The working principles of NSLC have also inspired the MAP-Elites algorithm (Cully et al., 2015; Mouret and Clune, 2015), which constructs a behavior-performance map. Given a behavior characterization with  $N$  dimensions, the MAP-elites algorithm transforms the behavior space into discrete bins according to a user-defined granularity level, and then tries to find the highest-performing individual for each point in the discretized space. However, as discussed by Pugh et al. (2015), the behavior characterization must be constrained to a low number of dimensions because the total number of bins increases exponentially with the dimensionality of the behavior characterization. Importantly, instead of a single controller, the complete behavior-performance map constructed by MAP-elites is deployed on the real robot. During task execution, if performance drops below a user-defined threshold due to, for instance, physical damage to the robot's body (Cully et al., 2015) or changes in the environmental conditions, the robot can iteratively select a promising behavior from the map, test it, and measure its performance until a suitable behavior is chosen.

Future research in the multiobjective combination of behavioral diversity and fitness may be an important stepping stone for leveraging ER's potential. However, defining effective behavior characterizations (Kistemaker and Whiteson, 2011) and fitness functions (Floreano and Urzelai, 2000) is usually the product of extensive trial-and-error experimentation. Thus, one important question in ER is to what extent such measures can be made generic and to what extent the amount of human input necessary can be minimized (Doncieux and Mouret, 2014). Over the past years, several studies have been conducted on this topic. Klyubin et al. (2005a,b); Prokopenko et al. (2006), Capdepuy et al. (2007), and Sperati et al. (2008) have studied task-independent fitness functions based on information theoretic measures such as Shannon's entropy and mutual information. Gomez (2009), Mouret and Doncieux (2012), and Gomes and Christensen (2013), on the other hand, have introduced different generic behavior characterizations for behavioral diversity-based algorithms. Complementarily, Gomes et al. (2014) have developed a method for the extraction of relevant behavior features based on formal descriptions provided by the experimenter. Overall, these contributions indicate that, even though ER is historically considered an empirical endeavor, a theoretical foundation for the field could be developed (Bongard, 2013). Importantly, the progressive introduction of task-independent fitness functions and behavior characterizations enables researchers to continue to pursue the goal of using ER as an automatic engineering tool for the synthesis of robot controllers.

#### 4 The Role of the Genomic Encoding in the Evolution of Complex Controllers

A wide variety of EAs have been used in ER studies. Take neuroevolution (Floreano et al., 2008; Yao, 1999), the optimization of artificial neural networks (ANNs) using EAs, as an example. ANNs are typically used as robotic controllers in ER (Nelson et al., 2009) due to their relatively smooth search space, their ability to represent general and adaptive solutions, and their capability to tolerate noisy input from the robots' sensors (Floreano and Mondada, 1994). Despite the widespread adoption of ANNs, however, there is no consensus on which type of neuroevolution algorithms are more appropriate for particular classes of tasks. The idea of using EAs to automate the design of neural networks dates back, at least, to 1989 (Angeline et al., 1994; Boers and Kuiper,



1992; Gruau, 1992; Harp et al., 1989; Kitano, 1990). Since then, neuroevolution has been successfully applied to control tasks in distinct domains (Floreano et al., 2008; Stanley and Miikkulainen, 2002; Yao, 1999). However, in some respect, ER studies have neglected some of the advantages of EAs and the effects of a number of key interacting components.

One particularly important component is the role of the genomic encoding and of the genotype-phenotype mapping, which have largely been left unstudied. The vast majority of ER studies employ *direct encoding* (Nelson et al., 2009), in which genotypes directly specify a phenotype. Even though direct encodings have been used with important results for controllers of relatively small size or with few parameters (Floreano et al., 2008), they are limited in their ability to evolve complex, large-scale controllers (Husbands et al., 1997; Meyer et al., 1998). Because each parameter of the controller has to be encoded and optimized separately, the size of the search space grows exponentially with the linear increase of controller size (Yao, 1999), which in turn leads to scalability issues. In effect, when ER emerged as a field of research, direct encodings were soon identified as one of the potential limitations for scaling evolutionary techniques to complex tasks (Husbands et al., 1997; Meyer et al., 1998).

#### 4.1 Evolving Complex Controllers with Indirect Encodings

*Indirect encodings*, also called generative or developmental encodings, enable representational efficiency in EAs by incorporating concepts from evolutionary developmental biology. The indirect encoding process is inspired by the genetic reuse that allows for structures to be represented compactly in DNA (Stanley and Miikkulainen, 2003). For example, while there are 20,000 to 25,000 human protein-coding genes (Southan, 2004), adult humans are composed of roughly 100 trillions of cells (Stix, 2006), and the cerebral cortex alone has approximately  $19 \times 10^9$  neurons (Herculano-Houzel, 2009) and  $6.7 \times 10^{13}$  synapses (Murre and Sturdy, 1995). In contrast with direct encodings, indirect encodings are more compact, as the same gene can be reused multiple times to construct different parts of the phenotype. That is, indirect encodings allow solutions to be represented as *patterns* of parameters, rather than requiring each parameter to be represented individually (Bentley and Kumar, 1999; Bongard, 2002; Risi, 2012; Seys and Beer, 2007; Stanley and Miikkulainen, 2003). In this way, evolution can search in a low-dimensional space and generate arbitrarily larger controllers.

Development is a prominent feature of biological organisms that enables the large-scale nervous systems underlying intelligence. Outside ER, several researchers have studied how to apply developmental processes to indirect encoding-based algorithms (Gruau, 1992; Kitano, 1990; Mattiussi and Floreano, 2007; Stanley and Miikkulainen, 2003; Suchorzewski, 2011). HyperNEAT by Stanley et al. (2009) and the variants that followed the original algorithm (D'Ambrosio et al., 2014; Risi and Stanley, 2012a,b) are powerful approaches to controller synthesis that employ an evolved generative encoding called Compositional Pattern Producing Network (CPPN) (Stanley, 2007). CPPNs are a versatile generative encoding that has been used to synthesize different types of controllers (Tarapore and Mouret, 2015), such as open-loop and closed-loop central pattern generators (Ijspeert, 2008), single-unit pattern generators (Morse et al., 2013), and artificial neural networks. This section focuses on the optimization of artificial neural networks because they are the prevalent control paradigm in evolutionary robotics (Nelson et al., 2009).

Formally, CPPNs are a composition of functions that encode the weight patterns of an ANN, as shown in Fig. 4. In conceptual terms, CPPNs are a variant of ANNs.

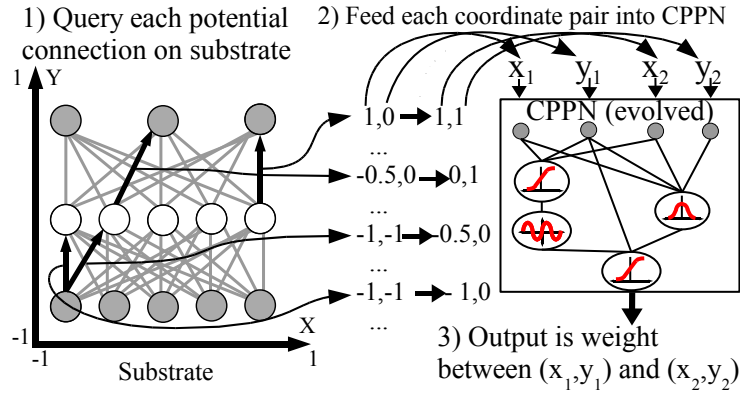


Figure 4: HyperNEAT connectivity patterns production. Neurons in the ANN are assigned coordinates that range from -1 to 1 in all dimensions of a *substrate*. The weight of each potential connection in the substrate is determined by querying the CPPN. The dark directed lines in the substrate represent a sample of connections that are queried. For each query, the CPPN takes as input the positions of two neurons, and outputs the weight of the connection between them. As a result, CPPNs can produce regular patterns of connections. From (Risi and Stanley, 2012a). © MIT Press.

The main difference between the two types of networks is that CPPNs rely on multiple different activation functions, which are composed to produce a pattern when CPPNs are queried over some input geometry (e.g a two-dimensional coordinate space). Each activation function in CPPNs represents a specific regularity such as symmetry, repetition, or repetition with variation (Stanley, 2007). For instance, a periodic function such as sine creates repetition, while a Gaussian function enables left-right symmetry.

In HyperNEAT, CPPNs produce connectivity patterns by interpreting spatial patterns generated within a hypercube as connectivity patterns in a lower-dimensional space. Neurons exist at *locations*, and one CPPN uses the coordinates of pairs of neurons to compute the connection weights for the entire network, as illustrated in Fig. 4. Artificial neurons are thus *spatially sensitive*. Because the connection weights between neurons are a function of the geometric position of such neurons, HyperNEAT can exploit the neural *topography* and not just the topology, and is able to automatically find the geometric aspects of a task (Clune et al., 2011; Stanley et al., 2009). Such exploitation of neural topography is based on a key concept from developmental biology that enables the evolution of regular neural structures: the fate of phenotypic components is a function of their geometric locations. In addition, the fact that CPPNs are *evolved* encodings approximates nature where the mapping from genotype to phenotype itself is also subject to evolution.

In ER domains, HyperNEAT yields important advantages over other algorithms. Given its ability to discover the task geometry, HyperNEAT has been shown capable of: (i) correlating the internal geometry of the ANN with the placement of robot sensors and actuators and of generalizing the learnt geometric principles to create functional larger-scale ANNs with additional inputs and outputs (D'Ambrosio and Stanley, 2007; Stanley et al., 2009), (ii) representing homogeneous and heterogeneous controllers for large groups of robots as a function of the control policy geometry, that is, the relationship between the role of the robots and their position in the group (D'Ambrosio et al.,

2010; D'Ambrosio and Stanley, 2013), and (iii) evolving high-performing controllers for simulated quadruped robots by exploiting regularities such as four-way symmetry, wherein all legs of the robots continuously move in unison with both front-back symmetry and left-right symmetry (Clune et al., 2009a,b, 2011). HyperNEAT's performance in the evolution of gaits for four-legged robots is particularly noteworthy because using other approaches, researchers typically need to manually identify the underlying task regularities and then force the encoding to exploit them (Clune et al., 2011; Gauci and Stanley, 2008; Stanley et al., 2009).

Even though HyperNEAT is able to exploit task geometry, its performance decreases on tasks that contain irregularities (Clune, 2010; Clune et al., 2011). For example, in the evolution of gaits for quadruped robots, if robots have faulty joints then HyperNEAT's ability to evolve coordinated behaviors is limited. As the number of faulty joints increases, HyperNEAT's performance continuously decreases and becomes statistically indistinguishable from that of its direct encoding counterpart NEAT (Stanley and Miikkulainen, 2002), which is both blind to task geometry and susceptible to the task dimensionality (800 parameters to be optimized) (Clune et al., 2011).

## 4.2 Hybridizing Encodings

Clune et al. (2011) recently proposed a hybridization of indirect and direct encodings, an algorithm called *switch-HybrID* to address HyperNEAT's ineffectiveness when irregularity is required. Switch-HybrID first evolves with HyperNEAT, and then switches to NEAT (Stanley and Miikkulainen, 2002) after a fixed, predefined number of generations. When the switch is made, each HyperNEAT ANN is translated to a NEAT genome. The evolutionary process then continues with NEAT until the end of the experiment. In Clune et al. (2011), switch-HybrID was shown to outperform HyperNEAT in three tasks: (i) two diagnostic tasks called *the target weights problem* and *the bit mirroring problem*, in which the degree of regularity can be varied, and (ii) in gait learning for quadruped robots, with and without faulty joints. The key idea is that because switch-HybrID is able to make subtle adjustments to otherwise regular patterns, it can account for irregularities, including faulty joints in quadruped robots (Clune et al., 2011).

The success of switch-HybrID suggests that indirect encodings may be more effective not as stand-alone algorithms, but in combination with a refining process that adjusts regular patterns in irregular ways. However, a key disadvantage of switch-HybrID is that the switch from HyperNEAT to NEAT is only made after a number of generations defined by the experimenter elapse. Similarly to devising optimal stopping criteria for EAs based on, for example, search space exploration, objective convergence, or population convergence (Aytug and Koehler, 2000; Goel and Stander, 2010; Jain et al., 2001), defining an appropriate switch point is a non-trivial task that requires domain-specific knowledge. In addition, the switch-point criterion limits the applicability of switch-HybrID in more open-ended domains such as when controllers are evolved online and must adapt to changing environmental conditions.

As an alternative to switch-HybrID, Silva et al. (2015c) recently introduced the R-HybrID algorithm in which ANN-based controllers are partially indirectly encoded and partially directly encoded. Specifically, genomes are composed of a direct encoding part, similar to NEAT, and an indirect encoding part, that is, a CPPN. Which parts of a given ANN are indirectly encoded via a CPPN or directly encoded is under evolutionary control. In this way, the evolutionary process has the potential to search for solutions across multiple ratios of indirect vs. direct encoding, and to automatically find an appropriate encoding combination to solve the current task. R-HybrID was shown

to: (i) outperform both NEAT and switch-HybrID, and to provide results comparable to HyperNEAT in the evolution of regular, large-scale controllers for a high-dimensional visual discrimination task (Stanley et al., 2009) that requires geometric principles to be evolved, and (ii) to typically outperform HyperNEAT, NEAT, and switch-HybrID in the coupled inverted pendulum (Hamann et al., 2011), a benchmark for modular robotics scenarios, and in a task inspired by the AX-CPT working memory test (Servan-Schreiber et al., 1996) that requires accumulating neural structure for cognitive behavior to emerge (Lehman and Miikkulainen, 2014; Ollion et al., 2012).

Even though R-HybrID and switch-HybrID have provided important insights on the effects of combining indirect and direct encodings, other refinement processes may also be important. One additional interesting possibility is the use of lifetime-learning algorithms (Niv et al., 2002; Risi and Stanley, 2010; Silva et al., 2014c; Soltoggio et al., 2008) to *complement* the refinement provided by direct encoding methods. The key motivation is that refining indirect encodings with a direct encoding alone may ultimately lead to the same scaling challenges that all direct encodings face in high-dimensional tasks. In this way, having a learning algorithm to complement the indirect and direct encodings may be important, especially as controllers such as ANNs scale closer to the size of biological brains.

## 5 Effective Robotics Engineering and Experimental Science Practices

The issues discussed throughout this paper are currently the main technical challenges in the field. Being a relatively young field of research, ER also faces the problem of positioning itself with respect to related fields. ER is partially a flavor of robotics engineering and partially an experimental science, and has not yet been able to become a canonical approach in mainstream robotics. In our view, one fundamental issue preventing the development of ER is the absence of widely adopted research practices in the field. For example, whereas there is an almost unanimous use of computer simulations in ER, there is not a prevalent simulation platform. In mainstream robotics, open-source platforms such as ROS (Quigley et al., 2009) and MOOS (Newman, 2008) have presented a solution for this problem, and accelerated innovation by facilitating, for instance, the design of new software toolboxes for programming robot controllers (Beetz et al., 2010), and control units for autonomous robots such as: (i) quadcopters (Achtelik et al., 2011; Lindsey et al., 2012), (ii) personal and assistive robots (Jain and Kemp, 2010; Meeussen et al., 2010), and (iii) water surface crafts (Curcio et al., 2005). In ER, research groups typically develop their own tools (Bredeche et al., 2013; Duarte et al., 2014c; Hugues and Bredeche, 2006; Mouret and Doncieux, 2010) or adapt existing platforms (Gerkey et al., 2003; Michel, 2004; Pinciroli et al., 2012) to their needs, which makes it difficult to conduct accurate reproductions of experiments or comparisons of results by independent researchers. Although the source code of the experiments can be distributed online, such software packages are typically not well documented, have a small user base, lack user support, and yield a significant learning curve because of their intricate details and configurability (Doncieux et al., 2011).

Besides the absence of standard simulation platforms, ER also suffers from the lack of benchmarks and test-beds. Typical benchmarks in mainstream robotics include RoboCup (Kitano et al., 1998) and the DARPA grand challenge (Seetharaman et al., 2006), among others (del Pobil, 2006), which are of limited applicability in ER (Hamann et al., 2011). Even though there are multiple so-called “common” tasks in ER such as simple navigation, navigation and obstacle avoidance, phototaxis, and gait learning in the case of legged robots (Nelson et al., 2009), there is no standard implementation of

these tasks, including specification of the type and size of the environments in which robots operate. Consequently, it is currently not possible for researchers to assess an algorithm on a set of benchmark or task instances. Such instances would be valuable not only for proofs of concept showing that a given algorithm has enough potential to be further investigated, but also to studies that analyze the strengths and limitations of a technique, how it fares against other algorithms, and its success and failure rates on a large number of different tasks.

For the state of the art in ER to advance, standard simulation platforms and benchmarks are important. Once those are established, the rigorous comparison of performance levels of different algorithms will require a greater emphasis on the experimental analysis of algorithms (see (Johnson, 2002) for a discussion on formal comparison methods). These methods may be based on, for example, task difficulty metrics and machine intelligence quotient definitions, which although not necessarily ER-specific may contribute to quantify progress in the field.

## 6 Conclusions and Discussion

ER emerged in the 1990s as a promising alternative to classic artificial intelligence for the synthesis of robust control systems for robots (Floreano and Mondada, 1994; Harvey et al., 1993; Husbands et al., 1997). As the field has moved from one-off successes towards consistent results and researchers attempted to progress from basic robot behaviors to increasingly complex ones, a number of issues have manifested themselves and impeded the widespread adoption of ER techniques for engineering purposes.

In this article, we have reviewed and discussed the current open issues in ER. We have discussed two types of issues: ER-specific issues and general issues related to evolutionary computation. In terms of ER-specific issues, we have discussed: (i) the reality gap effect that occurs when controllers are evolved in simulation and then deployed on real robots, (ii) the time-consuming evolution of controllers on real hardware, and (iii) the lack of standard research practices in the field. Regarding general evolutionary computation issues, and their impact in ER, we have discussed: (i) the bootstrap problem, (ii) deception, and (iii) the importance of the genomic encoding and of the genotype-phenotype mapping.

Overall, our review and discussion has shown that while there are not yet indisputable solutions to the current issues, some approaches with encouraging results have been introduced. To address the reality gap and the long time required to evolve controllers directly on real robots, the onboard combination of simulation-based evolution and online evolution (Bongard et al., 2006; Bongard, 2009; O'Dowd et al., 2011) stands out as the most promising approach. Specifically, because the utility of the simulation model is evaluated as a function of the online performance of the physical robot, controller evaluation and optimization can be conducted with increasing fidelity at a rate faster than real time. Regarding the bootstrap problem and deception, searching for solutions based on behavioral diversity methods (Lehman and Stanley, 2011a; Mouret and Doncieux, 2012) has been shown to be a path forward. Given the critical importance of how the search is guided (Doncieux and Mouret, 2014), multiobjective approaches are proving to be an effective way to combine behavioral diversity-based search and fitness-based search due to their ability to *automatically* direct evolution towards increasing exploration or exploitation of the search space. Because defining effective behavior characterizations and fitness functions typically requires a substantial amount of experimentation and human knowledge, developing generic metrics (Capdepuy et al., 2007; Gomes and Christensen, 2013; Gomes et al., 2014; Gomez, 2009; Klyubin et al.,

2005a,b; Mouret and Doncieux, 2012; Prokopenko et al., 2006; Sperati et al., 2008) so as to minimize the amount of human input necessary may prove important for future progress. Similarly, devising algorithms that can automatically find an appropriate genomic encoding and genotype to phenotype mapping to solve a given task (Silva et al., 2015c) are a potential way to facilitate the evolution of robot controllers for more complex tasks. It should not, however, be ignored how fundamentally challenging it is to devise a process capable of automatically evolving solutions given only an arbitrary task specification. Certain classes of tasks may therefore be challenging for future evolutionary techniques, in which case the use of human knowledge may be a viable option. In this respect, evolution may, for instance, be helped by being seeded with pre-evolved or manually-designed abilities (Silva et al., 2014b), such as sophisticated vision apparatus or cognitive behaviors like memory, learning, and high-level reasoning, in order to help expand the space of addressable tasks.

ER is currently not a mainstream topic in robotics (Stanley, 2011). In our view, this gap means that ER has an opportunity to earn its place as a canonical approach in robotics. If we, ER practitioners, manage to address the open issues and devise algorithms capable of *automatically* evolving effective controllers for different types of robots, there is the potential to revolutionize how control systems are synthesized. In addition, although this article has focused on the open issues regarding the synthesis of control systems for robots, it should also be stated as a concluding remark that Darwinian evolution is a powerful design process able to simultaneously optimize bodies and brains. In this respect, ER has different strengths when compared with traditional robotics and yields a higher potential, for instance, to solve tasks that require synthesizing controllers for robots with unconventional morphologies, such as four-legged and eight-legged robots (Bongard et al., 2006; Cully et al., 2015), and to coevolve control and morphology simultaneously (Lipson and Pollack, 2000).

## Acknowledgments

This work was partially supported by Fundação para a Ciência e a Tecnologia (FCT) under the grants SFRH/BD/76438/2011, SFRH/BD/89573/2012, UID/EEA/50008/2013, UID/Multi/04046/2013, and EXPL/EEI-AUT/0329/2013. The authors thank the anonymous reviewers for their constructive feedback and valuable comments.

## References

- Achtelik, M., Weiss, S., and Siegwart, R. (2011). Onboard IMU and monocular vision based control for MAVs in unknown in-and outdoor environments. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3056–3063. IEEE Press, Piscataway, NJ.
- Angeline, P., Saunders, G., and Pollack, J. (1994). An evolutionary algorithm that constructs recurrent neural networks. *IEEE Transactions on Neural Networks*, 5(1):54–65.
- Auerbach, J. and Bongard, J. C. (2009). How robot morphology and training order affect the learning of multiple behaviors. In *Proceedings of the IEEE Congress on Evolutionary Computation*, pages 39–46. IEEE Press, Piscataway, NJ.
- Auerbach, J. E. and Bongard, J. C. (2014). Environmental influence on the evolution of morphological complexity in machines. *PLoS Computational Biology*, 10(1):e1003399.
- Aytug, H. and Koehler, G. J. (2000). New stopping criterion for genetic algorithms. *European Journal of Operational Research*, 126(3):662–674.
- Barricelli, N. A. (1962). Numerical testing of evolution theories. *Acta Biotheoretica*, 16(1-2):69–98.

- Beetz, M., Mosenlechner, L., and Tenorth, M. (2010). CRAM – A cognitive robot abstract machine for everyday manipulation in human environments. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1012–1017. IEEE Press, Piscataway, NJ.
- Bentley, P. and Kumar, S. (1999). Three ways to grow designs: A comparison of evolved embryogenies for a design problem. In *Proceedings of the 1st Genetic and Evolutionary Computation Conference*, pages 35–43. ACM Press, New York, NY.
- Bianco, R. and Nolfi, S. (2004). Toward open-ended evolutionary robotics: evolving elementary robotic units able to self-assemble and self-reproduce. *Connection Science*, 16(4):227–248.
- Boers, E. and Kuiper, H. (1992). Biological metaphors and the design of modular artificial neural networks. Master’s thesis, Leiden University, Leiden, The Netherlands.
- Bongard, J. (2002). Evolving modular genetic regulatory networks. In *Proceedings of the IEEE Congress on Evolutionary Computation*, pages 1872–1877. IEEE Press, Piscataway, NJ.
- Bongard, J. (2008). Behavior chaining: Incremental behavior integration for evolutionary robotics. In *Proceedings of the 11th International Conference on Simulation and Synthesis of Living Systems*, pages 64–71. MIT Press, Cambridge, MA.
- Bongard, J., Beliveau, P., and Hornby, G. (2012). Avoiding local optima with interactive evolutionary robotics. In *Proceedings of the 14th Genetic and Evolutionary Computation Conference*, pages 1405–1406. ACM Press, New York, NY.
- Bongard, J., Zykov, V., and Lipson, H. (2006). Resilient machines through continuous self-modeling. *Science*, 314(5802):1118–1121.
- Bongard, J. C. (2009). Accelerating self-modeling in cooperative robot teams. *IEEE Transactions on Evolutionary Computation*, 13(2):321–332.
- Bongard, J. C. (2011). Innocent until proven guilty: Reducing robot shaping from polynomial to linear time. *IEEE Transactions on Evolutionary Computation*, 15(4):571–585.
- Bongard, J. C. (2013). Evolutionary robotics. *Communications of the ACM*, 56(8):74–83.
- Bongard, J. C. and Hornby, G. S. (2013). Combining fitness-based search and user modeling in evolutionary robotics. In *Proceedings of the 15th Genetic and Evolutionary Computation Conference*, pages 159–166. ACM Press, New York, NY.
- Bongard, J. C. and Lipson, H. (2004). Automated robot function recovery after unanticipated failure or environmental change using a minimum of hardware trials. In *Proceedings of the NASA/DoD Conference on Evolvable Hardware*, pages 169–176. IEEE Press, Piscataway, NJ.
- Bongard, J. C. and Lipson, H. (2005). Nonlinear system identification using coevolution of models and tests. *IEEE Transactions on Evolutionary Computation*, 9(4):361–384.
- Bredeche, N., Haasdijk, E., and Eiben, A. (2009). On-line, on-board evolution of robot controllers. In Collet, P., Monmarche, N., Legrand, P., Schoenauer, M., and Lutton, E., editors, *Proceedings of the 9th International Conference on Artificial Evolution*, pages 110–121. Springer, Berlin, Germany.
- Bredeche, N., Montanier, J. M., Liu, W., and Winfield, A. (2012). Environment-driven distributed evolutionary adaptation in a population of autonomous robotic agents. *Mathematical and Computer Modelling of Dynamical Systems*, 18(1):101–129.
- Bredeche, N., Montanier, J.-M., Weel, B., and Haasdijk, E. (2013). Roborobo! a fast robot simulator for swarm and collective robotics. *CoRR*, abs/1304.2888.
- Capdepuy, P., Polani, D., and Nehaniv, C. (2007). Maximization of potential information flow as a universal utility for collective behaviour. In *Proceedings of the IEEE Symposium on Artificial Life*, pages 207–213. IEEE Press, Piscataway, NJ.

- Celis, S., Hornby, G. S., and Bongard, J. (2013). Avoiding local optima with user demonstrations and low-level control. In *Proceedings of the IEEE Congress on Evolutionary Computation*, pages 3403–3410. IEEE Press, Piscataway, NJ.
- Christensen, A. L. and Dorigo, M. (2006). Incremental evolution of robot controllers for a highly integrated task. In *Proceedings of the 9th International Conference on Simulation of Adaptive Behavior*, pages 473–484. Springer, Berlin, Germany.
- Clune, J. (2010). *Evolving artificial neural networks with generative encodings inspired by developmental biology*. PhD thesis, Michigan State University, East Lansing, MI.
- Clune, J., Beckmann, B. E., Ofria, C., and Pennock, R. T. (2009a). Evolving coordinated quadruped gaits with the HyperNEAT generative encoding. In *Proceedings of the IEEE Congress on Evolutionary Computation*, pages 2764–2771. IEEE Press, Piscataway, NJ.
- Clune, J., Ofria, C., and Pennock, R. T. (2009b). The sensitivity of HyperNEAT to different geometric representations of a problem. In *Proceedings of the 11th Genetic and Evolutionary Computation Conference*, pages 675–682. ACM Press, New York, NY.
- Clune, J., Stanley, K., Pennock, R., and Ofria, C. (2011). On the performance of indirect encoding across the continuum of regularity. *IEEE Transactions on Evolutionary Computation*, 15(3):346–367.
- Coleman, O. J. and Blair, A. D. (2012). Evolving plastic neural networks for online learning: Review and future directions. In *Proceedings of the 25th Australasian Joint Conference on Artificial Intelligence*, pages 326–337. Springer, Berlin, Germany.
- Cuccu, G. and Gomez, F. (2011). When novelty is not enough. In *Applications of Evolutionary Computation*, pages 234–243. Springer, Berlin, Germany.
- Cully, A., Clune, J., Tarapore, D., and Mouret, J.-B. (2015). Robots that can adapt like animals. *Nature*, 521(7553):503–507.
- Cully, A. and Mouret, J.-B. (2013). Behavioral repertoire learning in robotics. In *Proceedings of 15th Genetic and Evolutionary Computation Conference*, pages 175–182. ACM Press, New York, NY.
- Cully, A. and Mouret, J.-B. (2015). Evolving a behavioral repertoire for a walking robot. *Evolutionary Computation*. In press.
- Curcio, J., Leonard, J., and Patrikalakis, A. (2005). SCOUT - a low cost autonomous surface platform for research in cooperative autonomy. In *Proceedings of MTS/IEEE OCEANS*, pages 725–729. IEEE Press, Piscataway, NJ.
- D’Ambrosio, D. B., Gauci, J., and Stanley, K. O. (2014). HyperNEAT: The first five years. In *Growing Adaptive Machines*, volume 557 of *Studies in Computational Intelligence*, chapter 5, pages 159–185. Springer, Berlin, Germany.
- D’Ambrosio, D. B., Lehman, J., Risi, S., and Stanley, K. O. (2010). Evolving policy geometry for scalable multiagent learning. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems*, pages 731–738. IFAAMAS, Richland, SC.
- D’Ambrosio, D. B. and Stanley, K. O. (2007). A novel generative encoding for exploiting neural network sensor and output geometry. In *Proceedings of the 9th Genetic and Evolutionary Computation Conference*, pages 974–981. ACM Press, New York, NY.
- D’Ambrosio, D. B. and Stanley, K. O. (2013). Scalable multiagent learning through indirect encoding of policy geometry. *Evolutionary Intelligence*, 6(1):1–26.
- de Hoog, J., Cameron, S., and Visser, A. (2010). Autonomous multi-robot exploration in communication-limited environments. In *Proceedings of the 11th Conference on Towards Autonomous Robotic Systems*, pages 68–75, Devon, UK. University of Plymouth.



- De Nardi, R. and Holland, O. E. (2008). Coevolutionary modelling of a miniature rotorcraft. In *Proceedings of the 10th International Conference on Intelligent Autonomous Systems*, pages 364–373. IOS Press, Amsterdam, The Netherlands.
- del Pobil, A. P. (2006). Why do we need benchmarks in robotics research? In *Proceedings of the Workshop on Benchmarks in Robotics Research, held as part of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. Available at: <http://www3.uji.es/~pobil/benchmarks1.pdf>.
- Doncieux, S. and Meyer, J.-A. (2003). Evolving neural networks for the control of a lenticular blimp. In *Proceedings of the Applications of Evolutionary Computing*, pages 626–637. Springer, Berlin, Germany.
- Doncieux, S. and Mouret, J.-B. (2014). Beyond black-box optimization: a review of selective pressures for evolutionary robotics. *Evolutionary Intelligence*, 7:71–93.
- Doncieux, S., Mouret, J.-B., Bredeche, N., and Padois, V. (2011). Evolutionary robotics: Exploring new horizons. In *New Horizons in Evolutionary Robotics*, volume 341 of *Studies in Computational Intelligence*, chapter 1, pages 3–25. Springer, Berlin, Germany.
- Duarte, M., Oliveira, S., and Christensen, A. L. (2012). Hierarchical evolution of robotic controllers for complex tasks. In *Proceedings of the IEEE International Conference on Development and Learning and on Epigenetic Robotics*, pages 1–6. IEEE Press, Piscataway, NJ.
- Duarte, M., Oliveira, S. M., and Christensen, A. L. (2014a). Evolution of hierarchical controllers for multirobot systems. In *Proceedings of the 14th International Conference on Synthesis and Simulation of Living Systems*, pages 657–664. MIT Press, Cambridge, MA.
- Duarte, M., Oliveira, S. M., and Christensen, A. L. (2014b). Hybrid control for large swarms of aquatic drones. In *Proceedings of the 14th International Conference on Synthesis and Simulation of Living Systems*, pages 785–792. MIT Press, Cambridge, MA.
- Duarte, M., Oliveira, S. M., and Christensen, A. L. (2015). Evolution of hybrid robotic controllers for complex tasks. *Journal of Intelligent & Robotic Systems*, 78(3–4):463–484.
- Duarte, M., Silva, F., Rodrigues, T., Oliveira, S. M., and Christensen, A. L. (2014c). JBotEvolver: A versatile simulation platform for evolutionary robotics. In *Proceedings of the 14th International Conference on Simulation and Synthesis of Living Systems*, pages 2010–2011. MIT Press, Cambridge, MA.
- Elfwing, S. and Doya, K. (2014). Emergence of polymorphic mating strategies in robot colonies. *PLoS One*, 9(4):e93622.
- Floreano, D., Dürr, P., and Mattiussi, C. (2008). Neuroevolution: from architectures to learning. *Evolutionary Intelligence*, 1:47–62.
- Floreano, D. and Keller, L. (2010). Evolution of adaptive behaviour by means of Darwinian selection. *PLoS Biology*, 8(1):e1000292.
- Floreano, D. and Mondada, F. (1994). Automatic creation of an autonomous agent: Genetic evolution of a neural-network driven robot. In *Proceedings of the 3rd International Conference on Simulation of Adaptive Behavior*, pages 421–430. MIT Press, Cambridge, MA.
- Floreano, D. and Mondada, F. (1996a). Evolution of homing navigation in a real mobile robot. *IEEE Transactions on Systems, Man, and Cybernetics*, 26(3):396–407.
- Floreano, D. and Mondada, F. (1996b). Evolution of plastic neurocontrollers for situated agents. In Maes, P., Mataric, M., Meyer, J.-A., Pollack, J., and Wilson, S., editors, *Proceedings of the 4th International Conference on Simulation of Adaptive Behavior*, pages 402–410. MIT Press, Cambridge, MA.
- Floreano, D. and Mondada, F. (1998). Evolutionary neurocontrollers for autonomous mobile robots. *Neural Networks*, 11(7):1461–1478.

- Floreano, D. and Urzelai, J. (2000). Evolutionary robots with on-line self-organization and behavioral fitness. *Neural Networks*, 13(4–5):431–443.
- Floreano, D. and Urzelai, J. (2001). Evolution of plastic control networks. *Autonomous Robots*, 11(3):311–317.
- Floreano, D., Zufferey, J.-C., and Nicoud, J.-D. (2005). From wheels to wings with evolutionary spiking circuits. *Artificial Life*, 11(1-2):121–138.
- Fogel, D. B. (2006). Nils barricelli – artificial life, coevolution, self-adaptation. *IEEE Computational Intelligence Magazine*, 1(1):41–45.
- Fraser, A. (1957). Simulation of genetic systems by automatic digital computers. *Australian Journal of Biological Sciences*, 10:484–491.
- Gauci, J. and Stanley, K. O. (2008). A case study on the critical role of geometric regularity in machine learning. In *Proceedings of the 23rd AAAI Conference on Artificial Intelligence*, pages 628–633. AAAI Press, Menlo Park, CA.
- Gerkey, B., Vaughan, R. T., and Howard, A. (2003). The Player/Stage project: Tools for multi-robot and distributed sensor systems. In *Proceedings of the 11th International Conference on Advanced Robotics*, pages 317–323, Coimbra, Portugal. FCT/UC.
- Goel, T. and Stander, N. (2010). A non-dominance-based online stopping criterion for multi-objective evolutionary algorithms. *International Journal for Numerical Methods in Engineering*, 84(6):661–684.
- Goldsby, H. J. and Cheng, B. H. C. (2010). Automatically discovering properties that specify the latent behavior of UML models. In *Proceedings of the 13th International Conference on Model Driven Engineering Languages and Systems*, pages 316–330. Springer, Berlin, Germany.
- Gomes, J. and Christensen, A. L. (2013). Generic behaviour similarity measures for evolutionary swarm robotics. In *Proceedings of the 15th Genetic and Evolutionary Computation Conference*, pages 199–206. ACM Press, New York, NY.
- Gomes, J., Mariano, P., and Christensen, A. L. (2014). Systematic derivation of behaviour characterisations in evolutionary robotics. In *Proceedings of the 14th International Conference on Synthesis and Simulation of Living Systems*, pages 212–219. MIT Press, Cambridge, MA.
- Gomes, J., Urbano, P., and Christensen, A. L. (2013). Evolution of swarm robotics systems with novelty search. *Swarm Intelligence*, 7(2–3):115–144.
- Gomez, F. and Miikkulainen, R. (1997). Incremental evolution of complex general behavior. *Adaptive Behavior*, 3–4(5):317–342.
- Gomez, F. J. (2009). Sustaining diversity using behavioral information distance. In *Proceedings of the 11th Genetic and Evolutionary Computation Conference*, pages 113–120. ACM Press, New York, NY.
- Gould, S. (2002). *The Structure of Evolutionary Theory*. Belknap Press, Cambridge, MA.
- Gruau, F. (1992). Genetic synthesis of boolean neural networks with a cell rewriting developmental process. In *Proceedings of the International Workshop on Combinations of Genetic Algorithms and Neural Networks*, pages 55–74. IEEE Press, Piscataway, NJ.
- Haasdijk, E., Atta-ul Qayyum, A., and Eiben, A. (2011). Racing to improve on-line, on-board evolutionary robotics. In *Proceedings of the 13th Genetic and Evolutionary Computation Conference*, pages 187–194. ACM Press, New York, NY.
- Haasdijk, E., Eiben, A., and Karafotias, G. (2010). On-line evolution of robot controllers by an encapsulated evolution strategy. In *Proceedings of the IEEE Congress on Evolutionary Computation*, pages 1–7. IEEE Press, Piscataway, NJ.

- Hamann, H., Schmickl, T., and Crailsheim, K. (2011). Coupled inverted pendulums: a benchmark for evolving decentral controllers in modular robotics. In *Proceedings of the 13th Genetic and Evolutionary Computation Conference*, pages 195–202. ACM Press, New York, NY.
- Harp, S., Samad, T., and Guha, A. (1989). Towards the genetic synthesis of neural network. In *Proceedings of the 3rd International Conference on Genetic Algorithms*, pages 360–369. Morgan Kaufmann, San Francisco, CA.
- Harvey, I., Di Paolo, E., Wood, R., Quinn, M., and Tuci, E. (2005). Evolutionary robotics: A new scientific tool for studying cognition. *Artificial Life*, 11(1–2):79–98.
- Harvey, I., Husbands, P., and Cliff, D. (1993). Issues in evolutionary robotics. In *Proceedings of the 2nd International Conference on Simulation of Adaptive Behavior*, pages 364–373. MIT Press, Cambridge, MA.
- Harvey, I., Husbands, P., Cliff, D., Thompson, A., and Jakobi, N. (1997). Evolutionary Robotics: the Sussex approach. *Robotics and Autonomous Systems*, 20(2–4):205–224.
- Herculano-Houzel, S. (2009). The human brain in numbers: a linearly scaled-up primate brain. *Frontiers in Human Neuroscience*, 3(31).
- Holland, J. (1962). Outline for a logical theory of adaptive systems. *Journal of the ACM*, 9(3):297–314.
- Hugues, L. and Bredeche, N. (2006). Simbad: an autonomous robot simulation package for education and research. In *Proceedings of the 9th International Conference on Simulation of Adaptive Behavior*, pages 831–842. Springer, Berlin, Germany.
- Husbands, P., Harvey, I., Cliff, D., and Miller, G. (1997). Artificial evolution: a new path for artificial intelligence? *Brain and Cognition*, 34(1):130–159.
- Ijspeert, A. J. (2008). Central pattern generators for locomotion control in animals and robots: a review. *Neural Networks*, 21(4):642–653.
- Jain, A. and Kemp, C. C. (2010). EL-E: an assistive mobile manipulator that autonomously fetches objects from flat surfaces. *Autonomous Robots*, 28(1):45–64.
- Jain, B. J., Pohlheim, H., and Wegener, J. (2001). On termination criteria of evolutionary algorithms. In *Proceedings of the 3rd Genetic and Evolutionary Computation Conference*, page 768. ACM Press, New York, NY.
- Jakobi, N. (1997). Evolutionary robotics and the radical envelope-of-noise hypothesis. *Adaptive Behavior*, 6(2):325–368.
- Jakobi, N. (1998). *Minimal Simulations for Evolutionary Robotics*. PhD thesis, University of Sussex, Sussex, UK.
- Johnson, D. S. (2002). A theoretician’s guide to the experimental analysis of algorithms. In *Proceedings of the 5th and 6th Discrete Mathematics and Theoretical Computer Science Implementation Challenges Workshops*, pages 215–250. American Mathematical Society, Providence, RI.
- Kistemaker, S. and Whiteson, S. (2011). Critical factors in the performance of novelty search. In *Proceedings of the 13th Genetic and Evolutionary Computation Conference*, pages 965–972. ACM Press, New York, NY.
- Kitano, H. (1990). Designing neural networks using genetic algorithms with graph generation system. *Complex Systems*, 4(4):461–476.
- Kitano, H., Asada, M., Noda, I., and Matsubara, H. (1998). RoboCup: Robot world cup. *IEEE Robotics & Automation Magazine*, 5(3):30–36.
- Klyubin, A. S., Polani, D., and Nehaniv, C. L. (2005a). All else being equal be empowered. In *Proceedings of the 8th European Conference on Artificial Life*, pages 744–753. Springer, Berlin, Germany.

- Klyubin, A. S., Polani, D., and Nehaniv, C. L. (2005b). Empowerment: A universal agent-centric measure of control. In *Proceedings of the IEEE Congress on Evolutionary Computation*, pages 128–135. IEEE Press, Piscataway, NJ.
- Knowles, J. D., Watson, R. A., and Corne, D. W. (2001). Reducing local optima in single-objective problems by multi-objectivization. In *Proceedings of the 1st International Conference on Evolutionary Multi-criterion Optimization*, pages 269–283. Springer, Berlin, Germany.
- König, L., Mostaghim, S., and Schmeck, H. (2009). Decentralized evolution of robotic behavior using finite state machines. *International Journal of Intelligent Computing and Cybernetics*, 2(4):695–723.
- Koos, S., Mouret, J.-B., and Doncieux, S. (2013). The transferability approach: Crossing the reality gap in evolutionary robotics. *IEEE Transactions on Evolutionary Computation*, 17(1):122–145.
- Larsen, T. and Hansen, S. (2005). Evolving composite robot behaviour - a modular architecture. In *Proceedings of the 5th International Workshop on Robot Motion and Control*, pages 271–276. IEEE Press, Piscataway, NJ.
- Lee, W.-P. (1999). Evolving complex robot behaviors. *Information Sciences*, 121(1-2):1–25.
- Leger, C. (2000). *Darwin2K: An Evolutionary Approach to Automated Design for Robotics*. Kluwer International Series in Engineering and Computer Science – Robotics : Vision, Manipulation and Sensors. Kluwer Academic Publishers, New York, NY.
- Lehman, J. (2012). *Evolution through the Search for Novelty*. PhD thesis, University of Central Florida, Orlando, FL.
- Lehman, J. and Miikkulainen, R. (2014). Overcoming deception in evolution of cognitive behaviors. In *Proceedings of the 16th Genetic and Evolutionary Computation Conference*, pages 185–192. ACM Press, New York, NY.
- Lehman, J. and Stanley, K. O. (2008). Exploiting open-endedness to solve problems through the search for novelty. In *Proceedings of the 11th International Conference on Simulation and Synthesis of Living Systems*, pages 329–336. MIT Press, Cambridge, MA.
- Lehman, J. and Stanley, K. O. (2010). Revising the evolutionary computation abstraction: minimal criteria novelty search. In *Proceedings of the 12th Genetic and Evolutionary Computation Conference*, pages 103–110. ACM Press, New York, NY.
- Lehman, J. and Stanley, K. O. (2011a). Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary Computation*, 19(2):189–223.
- Lehman, J. and Stanley, K. O. (2011b). Evolving a diversity of virtual creatures through novelty search and local competition. In *Proceedings of the 13th Genetic and Evolutionary Computation Conference*, pages 211–218. ACM Press, New York, NY.
- Lehman, J. and Stanley, K. O. (2011c). Novelty search and the problem with objectives. In *Genetic Programming Theory and Practice IX*, Genetic and Evolutionary Computation, chapter 3, pages 37–56. Springer, Berlin, Germany.
- Lehman, J. and Stanley, K. O. (2013). Evolvability is inevitable: Increasing evolvability without the pressure to adapt. *PLoS ONE*, 8(4):e62186.
- Lehman, J., Stanley, K. O., and Miikkulainen, R. (2013). Effective diversity maintenance in deceptive domains. In *Proceedings of the 15th Genetic and Evolutionary Computation Conference*, pages 215–222. ACM Press, New York, NY.
- Liapis, A., Yannakakis, G. N., and Togelius, J. (2013). Enhancements to constrained novelty search: Two-population novelty search for generating game content. In *Proceedings of the 15th Genetic and Evolutionary Computation Conference*, pages 343–350. ACM Press, New York, NY.

- Lindsey, Q., Mellinger, D., and Kumar, V. (2012). Construction with quadrotor teams. *Autonomous Robots*, 33(3):323–336.
- Lipson, H. and Pollack, J. (2000). Automatic design and manufacture of robotic lifeforms. *Nature*, 406:974–978.
- Matarić, M. and Cliff, D. (1996). Challenges in evolving controllers for physical robots. *Robotics and Autonomous Systems*, 19(1):67–83.
- Mattiussi, C. and Floreano, D. (2007). Analog genetic encoding for the evolution of circuits and networks. *IEEE Transactions on Evolutionary Computation*, 11(5):596–607.
- Meeussen, W. et al. (2010). Autonomous door opening and plugging in with a personal robot. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 729–736. IEEE Press, Piscataway, NJ.
- Meyer, J.-A., Husbands, P., and Harvey, I. (1998). Evolutionary robotics: A survey of applications and problems. In *Proceedings of the 1st European Workshop on Evolutionary Robotics*, pages 1–21. Springer, Berlin, Germany.
- Michel, O. (2004). Webots: Professional mobile robot simulation. *International Journal of Advanced Robotic Systems*, 1(1):39–42.
- Miglino, O., Lund, H., and Nolfi, S. (1995). Evolving mobile robots in simulated and real environments. *Artificial Life*, 2(4):417–434.
- Moioli, R., Vargas, P., Von Zuben, F., and Husbands, P. (2008). Towards the evolution of an artificial homeostatic system. In *Proceedings of the IEEE Congress on Evolutionary Computation*, pages 4023–4030. IEEE Press, Piscataway, NJ.
- Morse, G., Risi, S., Snyder, C. R., and Stanley, K. O. (2013). Single-unit pattern generators for quadruped locomotion. In *Proceedings of the 15th Genetic and Evolutionary Computation Conference*, pages 719–726. ACM Press, New York, NY.
- Mouret, J. and Doncieux, S. (2012). Encouraging behavioral diversity in evolutionary robotics: An empirical study. *Evolutionary Computation*, 20(1):91–133.
- Mouret, J.-B. (2011). Novelty-based multiobjectivization. In *New Horizons in Evolutionary Robotics*, volume 341 of *Studies in Computational Intelligence*, chapter 10, pages 139–154. Springer, Berlin, Germany.
- Mouret, J.-B. and Clune, J. (2015). Illuminating search spaces by mapping elites. *arXiv preprint arXiv:1504.04909*.
- Mouret, J.-B. and Doncieux, S. (2008). Incremental evolution of animats’ behaviors as a multi-objective optimization. In *Proceedings of the 10th International Conference on Simulation of Adaptive Behavior*, pages 210–219. Springer, Berlin, Germany.
- Mouret, J.-B. and Doncieux, S. (2009a). Overcoming the bootstrap problem in evolutionary robotics using behavioral diversity. In *Proceedings of the 11th IEEE Congress on Evolutionary Computation*, pages 1161–1168. IEEE Press, Piscataway, NJ.
- Mouret, J.-B. and Doncieux, S. (2009b). Using behavioral exploration objectives to solve deceptive problems in neuro-evolution. In *Proceedings of the 11th Genetic and Evolutionary Computation Conference*, pages 627–634. ACM Press, New York, NY.
- Mouret, J.-B. and Doncieux, S. (2010). Sferes v2: Evolving in the multi-core world. In *Proceedings of the IEEE Congress on Evolutionary Computation*, pages 1–8. IEEE Press, Piscataway, NJ.
- Murre, J. M. J. and Sturdy, D. P. F. (1995). The connectivity of the brain: multi-level quantitative analysis. *Biological Cybernetics*, 73(6):529–545.

- Naredo, E. and Trujillo, L. (2013). Searching for novel clustering programs. In *Proceedings of the 15th Genetic and Evolutionary Computation Conference*, pages 1093–1100. ACM Press, New York, NY.
- Nelson, A., Barlow, G., and Doitsidis, L. (2009). Fitness functions in evolutionary robotics: A survey and analysis. *Robotics and Autonomous Systems*, 57(4):345–370.
- Newman, P. M. (2008). MOOS – mission oriented operating suite. Technical Report 08, Massachusetts Institute of Technology, Cambridge, MA.
- Niv, Y., Joel, D., Meilijson, I., and Ruppin, E. (2002). Evolution of reinforcement learning in uncertain environments: A simple explanation for complex foraging behaviors. *Adaptive Behavior*, 10(1):5–24.
- Nolfi, S. (1998). Evolutionary robotics: Exploiting the full power of self-organization. *Connection Science*, 10(3–4):167–184.
- Nolfi, S. (2002). Evolving robots able to self-localize in the environment: the importance of viewing cognition as the result of processes occurring at different time-scales. *Connection Science*, 14(3):231–244.
- Nolfi, S. and Floreano, D. (2000). *Evolutionary robotics: The biology, intelligence, and technology of self-organizing machines*. MIT Press, Cambridge, MA.
- Nolfi, S., Floreano, D., Miglino, O., and Mondada, F. (1994). How to evolve autonomous robots: Different approaches in evolutionary robotics. In *Proceedings of the 4th International Workshop on Synthesis and Simulation of Living Systems*, pages 190–197. MIT Press, Cambridge, MA.
- Noskov, N., Haasdijk, E., Weel, B., and Eiben, A. E. (2013). MONEE: Using parental investment to combine open-ended and task-driven evolution. In *Proceedings of the 16th European Conference on the Applications of Evolutionary Computation*, pages 569–578. Springer, Berlin, Germany.
- O’Dowd, P. J., Winfield, A. F. T., and Studley, M. (2011). The distributed co-evolution of an embodied simulator and controller for swarm robot behaviours. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4995–5000. IEEE Press, Piscataway, NJ.
- Ollion, C., Pinville, T., and Doncieux, S. (2012). With a little help from selection pressures: evolution of memory in robot controllers. In *Proceedings of the 13th International Conference on Simulation and Synthesis of Living Systems*, pages 407–414. MIT Press, Cambridge, MA.
- Pincirololi, C. et al. (2012). ARGoS: a modular, parallel, multi-engine simulator for multi-robot systems. *Swarm Intelligence*, 6(4):271–295.
- Prokopenko, M., Gerasimov, V., and Tanev, I. (2006). Evolving spatiotemporal coordination in a modular robotic system. In *Proceedings of the 9th International Conference on Simulation of Adaptive Behavior*, pages 558–569. Springer, Berlin, Germany.
- Pugh, J. K., Soros, L., Szerlip, P. A., and Stanley, K. O. (2015). Confronting the challenge of quality diversity. In *Proceedings of the 17th Genetic and Evolutionary Computation Conference*, pages 967–974. ACM Press, New York, NY.
- Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., and Ng, A. Y. (2009). ROS: an open-source Robot Operating System. In *Proceedings of the Workshop on Open Source Software, held as part of the IEEE International Conference on Robotics and Automation*. Available at <http://ai.stanford.edu/~ang/papers/icraoss09-ROS.pdf>.
- Quinn, M., Smith, L., Mayley, G., and 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 A: Mathematical, Physical and Engineering Sciences*, 361(1811):2321–2343.

- Risi, S. (2012). *Towards Evolving More Brain-Like Artificial Neural Networks*. PhD thesis, University of Central Florida, Orlando, FL.
- Risi, S., Hughes, C., and Stanley, K. O. (2010). Evolving plastic neural networks with novelty search. *Adaptive Behavior*, 18(6):470–491.
- Risi, S. and Stanley, K. O. (2010). Indirectly encoding neural plasticity as a pattern of local rules. In *Proceedings of the 11th International Conference on Simulation of Adaptive Behavior*, pages 533–543. Springer, Berlin, Germany.
- Risi, S. and Stanley, K. O. (2012a). An enhanced hypercube-based encoding for evolving the placement, density, and connectivity of neurons. *Artificial Life*, 18(4):331–363.
- Risi, S. and Stanley, K. O. (2012b). A unified approach to evolving plasticity and neural geometry. In *Proceedings of the International Joint Conference on Neural Networks*, pages 1–8. IEEE Press, Piscataway, NJ.
- Seetharaman, G., Lakhota, A., and Blasch, E. (2006). Unmanned vehicles come of age: The DARPA grand challenge. *Computer*, 39(12):26–29.
- Servan-Schreiber, D., Cohen, J. D., and Steingard, S. (1996). Schizophrenic deficits in the processing of context: A test of a theoretical model. *Archives of General Psychiatry*, 53(12):1105–1113.
- Seys, C. W. and Beer, R. D. (2007). Genotype reuse more important than genotype size in evolvability of embodied neural networks. In *Proceedings of the 9th European Conference on Artificial Life*, pages 915–924. Springer, Berlin, Germany.
- Silva, F., Christensen, A. L., and Correia, L. (2015a). Engineering online evolution of robot behaviour. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems*, pages 2017–2018. IFAAMAS, Richland, SC.
- Silva, F., Correia, L., and Christensen, A. L. (2014a). Speeding up online evolution of robotic controllers with macro-neurons. In *Proceedings of the 17th European Conference on the Applications of Evolutionary Computation*, pages 765–776. Springer, Berlin, Germany.
- Silva, F., Correia, L., and Christensen, A. L. (2015b). A case study on the scalability of online evolution of robotic controllers. In *17th Portuguese Conference on Artificial Intelligence*, pages 189–200. Springer International Publishing, Berlin, Germany.
- Silva, F., Correia, L., and Christensen, A. L. (2015c). R-HybrID: Evolution of agent controllers with a hybridisation of indirect and direct encodings. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems*, pages 735–744. IFAAMAS, Richland, SC.
- Silva, F., Duarte, M., Oliveira, S. M., Correia, L., and Christensen, A. L. (2014b). The case for engineering the evolution of robot controllers. In *Proceedings of the 14th International Conference on Synthesis and Simulation of Living Systems*, pages 703–710. MIT Press, Cambridge, MA.
- Silva, F., Urbano, P., and Christensen, A. L. (2014c). Online evolution of adaptive robot behaviour. *International Journal of Natural Computing Research*, 4(2):59–77.
- Silva, F., Urbano, P., Correia, L., and Christensen, A. L. (2015d). odNEAT: An algorithm for decentralised online evolution of robotic controllers. *Evolutionary Computation*, 23(3):421–449.
- Silva, F., Urbano, P., Oliveira, S., and Christensen, A. L. (2012). odNEAT: An algorithm for distributed online, onboard evolution of robot behaviours. In *Proceedings of the 13th International Conference on Simulation and Synthesis of Living Systems*, pages 251–258. MIT Press, Cambridge, MA.
- Smith, T., Husbands, P., and O’Shea, M. (2001). Neutral networks in an evolutionary robotics search space. In *Proceedings of the IEEE Congress on Evolutionary Computation*, pages 136–143. IEEE Press, Piscataway, NJ.

- Soltoggio, A., Bullinaria, J. A., Mattiussi, C., Dürr, P., and Floreano, D. (2008). Evolutionary advantages of neuromodulated plasticity in dynamic, reward-based scenarios. In *Proceedings of the 13th International Conference on Simulation and Synthesis of Living Systems*, pages 569–576. MIT Press, Cambridge, MA.
- Southan, C. (2004). Has the yo-yo stopped? An assessment of human protein-coding gene number. *Proteomics*, 4(6):1712–1726.
- Sperati, V., Trianni, V., and Nolfi, S. (2008). Evolving coordinated group behaviours through maximisation of mean mutual information. *Swarm Intelligence*, 2(2-4):73–95.
- Stanley, K. and Miikkulainen, R. (2002). Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10(2):99–127.
- Stanley, K. and Miikkulainen, R. (2003). A taxonomy for artificial embryogeny. *Artificial Life*, 9(2):93–130.
- Stanley, K. O. (2007). Compositional pattern producing networks: A novel abstraction of development. *Genetic Programming and Evolvable Machines*, 8(2):131–162.
- Stanley, K. O. (2011). Why evolutionary robotics will matter. In *New Horizons in Evolutionary Robotics*, volume 341 of *Studies in Computational Intelligence*, chapter 3, pages 37–41. Springer, Berlin, Germany.
- Stanley, K. O., D’Ambrosio, D., and Gauci, J. (2009). A hypercube-based encoding for evolving large-scale neural networks. *Artificial Life*, 15(2):185–212.
- Stix, G. (2006). Owning the stuff of life. *Scientific American*, 294(2):76–83.
- Suchorzewski, M. (2011). Evolving scalable and modular adaptive networks with developmental symbolic encoding. *Evolutionary Intelligence*, 4(3):145–163.
- Takagi, H. (2001). Interactive evolutionary computation: Fusion of the capabilities of EC optimization and human evaluation. *Proceedings of the IEEE*, 89(9):1275–1296.
- Tanese, R. (1989). *Distributed Genetic Algorithms for Function Optimization*. PhD thesis, University of Michigan, Ann Arbor, MI.
- Tarapore, D. and Mouret, J.-B. (2015). Evolvability signatures of generative encodings: Beyond standard performance benchmarks. *Information Sciences*, 313:43–61.
- Togelius, J. (2004). Evolution of a subsumption architecture neurocontroller. *Journal of Intelligent & Fuzzy Systems*, 15(1):15–20.
- Turing, A. M. (1950). Computing machinery and intelligence. *Mind*, 59(236):433–460.
- Urzelai, J. and Floreano, D. (2001). Evolution of adaptive synapses: Robots with fast adaptive behavior in new environments. *Evolutionary Computation*, 9(4):495–524.
- Watson, R., Ficici, S., and Pollack, J. (1999). Embodied evolution: Embodying an evolutionary algorithm in a population of robots. In *Proceedings of the IEEE Congress on Evolutionary Computation*, pages 335–342. IEEE Press, Piscataway, NJ.
- Watson, R., Ficici, S., and Pollack, J. (2002). Embodied evolution: Distributing an evolutionary algorithm in a population of robots. *Robotics and Autonomous Systems*, 39(1):1–18.
- Whitley, L. (1991). Fundamental principles of deception in genetic search. In *Proceedings of the 1st Workshop on Foundations of Genetic Algorithms*, pages 221–241. Morgan Kaufmann, San Mateo, CA.
- Wischmann, S., Stamm, K., and Wörgötter, F. (2007). Embodied evolution and learning: The neglected timing of maturation. In *Proceedings of the 9th European Conference on Artificial Life*, pages 284–293. Springer, Berlin, Germany.



- Woolley, B. G. and Stanley, K. O. (2014). A novel human-computer collaboration: Combining novelty search with interactive evolution. In *Proceedings of the 16th Genetic and Evolutionary Computation Conference*. ACM Press, New York, NY.
- Yao, X. (1999). Evolving artificial neural networks. *Proceedings of the IEEE*, 87(9):1423–1447.
- Zaera, N., Cliff, D., et al. (1996). (Not) evolving collective behaviours in synthetic fish. In *Proceedings of the 4th International Conference on Simulation of Adaptive Behavior*, pages 635–644. MIT Press, Cambridge, MA.
- Zagal, J. C. and Ruiz-Del-Solar, J. (2007). Combining simulation and reality in evolutionary robotics. *Journal of Intelligent & Robotic Systems*, 50(1):19–39.