Evolutionary robotics: what, why, and where to
Stephane Doncieux, Nicolas Bredeche, Jean-Baptiste Mouret, Gusz Ae Eiben

To cite this version:
Stephane Doncieux, Nicolas Bredeche, Jean-Baptiste Mouret, Gusz Ae Eiben. Evolutionary robotics: what, why, and where to. Frontiers in Robotics and AI, 2015, pp.10.3389/frobt.2015.00004. 10.3389/frobt.2015.00004 . hal-01131267

HAL Id: hal-01131267
https://hal.science/hal-01131267
Submitted on 23 Mar 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Evolutionary robotics: what, why, and where to

Stephane Doncieux\textsuperscript{1,2,*}, Nicolas Bredeche\textsuperscript{1,2}, Jean-Baptiste Mouret\textsuperscript{1,2} and Agostino E. (Gusz) Eiben\textsuperscript{3}

\textsuperscript{1} UMR 7222, ISIR, Sorbonne Universités, UPMC Univ Paris 06, Paris, France
\textsuperscript{2} UMR 7222, CNRS, ISIR, Paris, France
\textsuperscript{3} Department of Computer Science, VU University, Amsterdam, Netherlands

Edited by:
Geoff Nitschke, University of Cape Town, South Africa

Reviewed by:
Sabine Hauert, University of Bristol, UK
Joel Lehman, The University of Texas at Austin, USA
Joshua E. Auerbach, École polytechnique fédérale de Lausanne (EPFL), Switzerland

*Correspondence:
Stephane Doncieux, ISIR, UPMC, BC173, 4 Place Jussieu, 75252 Paris Cedex 05, France
e-mail: stephane.doncieux@upmc.fr

1. INTRODUCTION

In designing a robot, many different aspects must be considered simultaneously: its morphology, sensory apparatus, motor system, control architecture, etc. (Siciliano and Khatib, 2008). One of the main challenges of robotics is that all of these aspects interact and jointly determine the robot’s behavior. For instance, grasping an object with the shadow hand – a dextrous hand with 20 degrees of freedom (Shadow, 2013) – or the Cornell universal “jamming” gripper – a one degree-of-freedom gripper based on a vacuum pump and an elastic membrane that encases granular material (Brown et al., 2010) – are two completely different problems from the controller’s point of view. In the first case, control is a challenge, whereas in the second case it is straightforward. Likewise, a dedicated sensory apparatus can drastically change the complexity of a robotic system. 3D information, for instance, is typically useful for obstacle avoidance, localization, and mapping. Getting this information out of 2D images is possible (Saxena et al., 2009), but requires more complex processing than getting this information directly from 3D scanners. The sensory apparatus, morphology, and control of the robot are thus closely interdependent, and any change in a given part is likely to have a large influence on the functioning of the others.

Considering all these aspects at the same time contrasts with the straightforward approach in which they are all studied in isolation. Engineering mostly follows such a reductionist approach and does little to exploit these interdependencies; indeed, it often “fights” them to keep the design process as modular as possible (Suh, 1990). Most fields related to robotics are thus focused on one specific robot feature or functionality while ignoring the rest. Self localization and mapping (SLAM), for instance, deals with the ability to localize the robot in an unknown environment (Thrun et al., 2005), trajectory planning with how to make a robot move from position A to position B (Siegwart et al., 2011), etc.

The concept of embodied intelligence (Pfeifer and Bongard, 2007) is an alternative point of view in which the robot, its environment, and all the interactions between its components are studied as a whole. Such an approach makes it possible to design systems with a balance of complexity between their different parts, and it generally results in simpler and better systems. However, if the whole design problem cannot be decomposed into smaller and simpler sub-problems, the question of methodology is far from trivial. Drawing inspiration from nature can be helpful here, since living beings are good examples of systems endowed with embodied intelligence. The mechanism that was responsible for their appearance, evolution, is therefore an attractive option to form the basis of an alternative design methodology.

The idea of using evolutionary principles in problem solving dates back to the dawn of computers (Fogel, 1998), and the resulting field, evolutionary computing (EC), has proven successful in solving hard problems in optimization, modeling, and design (Eiben and Smith, 2003). Evolutionary robotics (ER) is a holistic approach to robot design inspired by such ideas, based on variation and selection principles (Nolfi and Floreano, 2000; Doncieux et al., 2011; Bongard, 2013), which tackles the problems of designing the robot’s sensory apparatus, morphology, and control simultaneously. The evolutionary part of an ER system relies on an evolutionary algorithm (see Figure 1). The first generation of candidate solutions, represented by their codes, the so-called “genotypes,” are usually randomly generated. Their fitness is then evaluated, which means (1) translating the genotype – the code – into a phenotype – a robot part, or its controller, or its overall morphology; (2) putting the robot in its environment; (3) letting the robot interact with its environment for some time and observing the resulting behavior; and (4) computing a fitness value on the basis of these observations. The fitness value is then used to select individuals to seed the
FIGURE 1 | Principal workflow of evolutionary robotics. The diagram presents a classic evolutionary robotics system, which consists of an evolutionary algorithm (left) and a robot, real or simulated (right). The evolutionary component follows the generic evolutionary algorithm template with one application-specific feature: for fitness evaluations, the (real or simulated) robot is invoked. The evaluation component takes a genotype from the evolutionary component as input and returns the fitness value of the corresponding robot as output. In this figure, the genotype encodes the controller of an e-puck robot (a neural net) and the fitness of the robot is defined by its behavior in a given maze, for instance, the distance of the robot’s final position to the maze output.

next generation. The selected would-be parents undergo randomized reproduction through the application of stochastic variations (mutation and crossover), and the evaluation–selection–variation cycle is repeated until some stopping criterion is met (typically a given number of evaluations or a predefined quality threshold).

Evolutionary robotics is distinct from other fields of engineering in that it is inherently based on a biological mechanism. This is quite unusual, as most engineering fields rely on classical approaches based on mathematics and physics. Biological mechanisms are less thoroughly understood, although much progress has been made in recent years. Experimental research in the area suffers from the fact that evolution usually requires many generations of large numbers of individuals, whose lives may last years or even decades. The only exceptions are organisms whose lifecycle is short enough to allow laboratory experiments (Wiser et al., 2013). In other cases, lacking the ability to perform direct experiments, biologists usually analyze the remains of past creatures, the genetic code, or commit to the theoretical approaches of population biology. ER offers an alternative synthetic approach to the study of evolution in such contexts. Using robots as evolving entities opens the way to a novel *modus operandi* wherein hypotheses can be tested experimentally (Long, 2012). Thus, the contributions of ER are not limited to engineering (Figure 2).

This article discusses the field of ER on the basis of more than two decades of history. Section 2 looks at the groups of researchers who stand most directly to benefit from ER, engineers, and biologists. Section 3 illuminates how the standard scientific method applies in ER, and Section 4 summarizes the major findings since the field began. They are followed by Section 5, which discusses the main open issues within ER, and Section 6, which elaborates on the expected results of ER for a broad scientific and engineering community.

### 2. EVOLUTIONARY ROBOTICS: FOR WHOM?

In this section, we look at the research communities, which stand to benefit most immediately from ER. In particular, we focus on engineers trying to design better robots, and biologists trying to understand natural evolution as the main beneficiaries.

#### 2.1. ENGINEERS

ER addresses one of the major challenges of robotics: how to build robots, which are simple and yet efficient. ER proposes to design robot behaviors by considering the robot as a whole from the very beginning of the design process, instead of designing each of its parts separately and putting them together at the end of the process.
(Lipson and Pollack, 2000). It is an intrinsically multi-disciplinary approach to robot design, as it implies considering the mechanics, control, perceptual system, and decision process of the robot all at the same level. By allowing the process to transcend the classical boundaries between sub-fields of robotics, this integrated view opens up new and original perspectives on robot design. Considering the robot’s morphology as a variable in behavior design, and not something designed a priori, greatly extends the design space.

ER’s integrative view on robot design creates new opportunities at the frontiers between different fields. Design processes including both morphology and control have thus been proposed (Lipson and Pollack, 2000), and studies have been performed on the influence of body complexity on the robustness of designed behaviors, e.g., Bongard (2011). Bongard investigated the impact of changes in morphology on the behavior design process and showed that progressively increasing morphological complexity actually helps design more robust behaviors. The influence of the sensory apparatus can likewise be studied using ER methods, in particular, in creating efficient controllers, which do not require sensory information that had been thought to be necessary (Hauert et al., 2008).

Applying design principles to real devices is critical in engineer- ing generally and in robotics, in particular. ER is an automated design process that can be directly applied to real robots, allevi- ating the need for accurate simulations. Hornby et al. (2005) thus had real Aibo robots learn to walk on their own using ER methods. Nevertheless, as tests on real robots are expensive in terms of both time and mechanical fatigue, the design process often relies at least partly on simulations. The reality gap (Jakobi et al., 1995; Miglino et al., 1995; Mouret et al., 2012; Koos et al., 2013b), created by the inevitable discrepancies between a model and the real system it is intended to represent, is a critical issue for any design method. A tedious trial and error process may be required to obtain, on the real device, the result generated thanks to the model. ER opens new perspectives with respect to this issue, as tests in simulation and on a real robot can be automatically intermixed to get the best of both worlds (Koos et al., 2013b). Interestingly, the ability to cross a significant reality gap opens the way to resilience: any motor or mechanical failure will increase the discrepancy between a robot model and the real system. Resilient robots can be seen as systems that are robust to the transfer between an inaccurate simulation and reality (Koos et al., 2013a). Likewise, the learning and adaptation properties of ER can be used to update the model of the robot after a failure, thus making it possible to adapt a robot controller without human intervention (Bongard et al., 2006). The originality of such approaches with respect to more classical approaches to handling motor or mechanical failures is that they do not require a priori identification of potential failures.

Recently, a range of new concepts have been proposed to go beyond the traditional, articulated, hard robot with rigid arms and wheels or legs. Such novel types of robots – be they a swarm of simple robots (Sahin and Spears, 2005), robots built with soft materials [so-called soft robots (Trivedi et al., 2008; Cheney et al., 2013, 2014)], or robots built with small modules (modular robots) (Zykov et al., 2007) – create new challenges for robotics. ER design principles can be applied to these unconventional robotic devices with few adaptations, thus making ER a method of choice in such cases (Bongard, 2013).

### 2.2. BIOLOGISTS

Understanding the general principles of evolution is a complex and challenging question. The field of evolutionary biology addresses this question in many ways, either by looking at nature (palaeo- logical data, the study of behavior, etc.) or by constructing models that can be either solved analytically or simulated. Mathematical modeling and simulation have provided (and are still providing) many deep insights into evolution, but sometimes meet their limits due to the simplifications, which are inherently required to construct such models (e.g., well-mixed population, or ad hoc dispersal strategies). Bridging the gap between real data and models is also a challenge, as observing evolution at work is very rarely possible. Although a small number of studies, such as the 25-year in vitro bacterial evolution experiment (Wiser et al., 2013) provide counter-examples, their limits are clear: evolution cannot be experimentally observed for long periods in species with a slow pace of evolution (such as animals), let alone in their natural habitat.

The challenge of studying evolution at work under realistic assumptions within a realistic time frame thus remains. John May- nard Smith, one of the fathers of modern theoretical biology, gave one possible, and rather convincing, answer: “so far, we have been able to study only one evolving system and we cannot wait for interstellar flight to provide us with a second. If we want to discover generalizations about evolving systems, we have to look at artificial ones” (Maynard-Smith, 1992). Simulation software like AVIDA (Lenski et al., 1999; Bryson and Ofria, 2013) and AEvol (Batut et al., 2013) offers examples that are traveling this path, providing simulation tools for studying various aspects of bacte- rial evolution. As with other artificial evolution setups (including evolutionary robotics), it is then possible to “study the biological design process itself by manipulating it” (Dennett, 1995).

In a similar fashion, evolutionary robotics provides tools for modeling and simulating evolution with unique properties: con- sidering embodied agents that are located in a realistic environ- ment makes it possible to study hypotheses on the mechanistic constraints at play during evolution (Floreano and Keller, 2010; Mitri et al., 2013; Trianni, 2014). This is particularly relevant for modeling behaviors, where complex interactions within the group (e.g., leading to the evolution of communication) and with the environment (e.g., leading to the evolution of particular morpho- logical traits) are at work. Evolutionary robotics can be used to test hypotheses about particular features with respect to both their origin (e.g., how does cooperation evolve?) and their functional relevance (e.g., does explicit communication improve coordina- tion?) in a realistic setup. Moreover, the degree of realism can be tuned depending on the question to be addressed, from testing a specific hypothesis to building a full-grown artificial evolutionary ecology setup, i.e., implementing a distributed online evolutionary adaptation process with (real or simulated) robots, with no other objective than to study evolution per se.

Within the last 10 years, evolutionary robotics has been used to study a number of key issues in evolutionary biology: the evolu- tion of cooperation, whether altruistic (Montanier and Bredeche, 2011, 2013; Waibel et al., 2011) or not (Solomon et al., 2012), the
evolution of communication (Floreano et al., 2007; Mitri et al.,
2009; Wischmann et al., 2012), morphological complexity (Bongard,
2011; Auerbach and Bongard, 2014), and collective swarming (Olson
et al., 2013), to cite a few. Another way of classifying the contributions
of evolutionary robotics to biology is in terms of whether they seek to
extend abstract models through embodiment (cf. most of the works
cited in this paragraph), or to carefully simulate existing features
observed from real data in order to better understand how and why
they evolved (Long, 2012).

Although the basic algorithms and methods remain similar, the
goal of these types of work is to understand biology rather than to
contribute to engineering. It is also important to note that most of
these studies have been published in top-ranked journals in biology,
proving that this branch of evolutionary robotics is recognized
as relevant by its target audience.

3. HOW TO USE ER METHODS AND CONTRIBUTE TO THIS
FIELD?

Evolutionary robotics is an experimental research field. Experiment-
ation amounts to executing an evolutionary process in a population
of (simulated) robots in a given environment, with some targeted robot
behavior. The targeted features are used to specify a fitness measure
that drives the evolutionary process. The majority of the ER field follows
the classic algorithmic scheme as shown in Figure 1. However, experiment-
ation can have different objectives. For instance, one could aim to solve
a particular engineering problem, such as designing a suitable morphology
and controller for a robot that efficiently explores rough terrain. Or
one could be interested in studying the difference between wheeled
and legged morphologies. Alternatively, one could investigate how
various selection mechanisms affect the evolutionary algorithm in
an ER context. Although these options differ in several aspects, in
any case the methodology should follow the generic principles
of experimentation and hypothesis testing. The corresponding
template can be summarized by the following items.

1. What is the question to be answered or the goal to be achieved?
   In other words, what is this work about, what is the main
   context?
2. What is the hypothesis regarding the question, or what is the
   appropriate success criterion, which will prove that the goal
   has been achieved?
3. What kind of results can validate or refute the hypothesis? Or,
   what data are needed to determine whether the success criterion
   has been met?
4. What experiment(s) can generate the results and data identified
   in the previous step? Control experiments – to reject alternative
   hypotheses that may also explain the results – are mandatory.
5. What conclusion can be drawn from the analysis of the results?

Two different aspects of ER require particular attention. First,
the stochastic nature of evolutionary algorithms implies multiple
runs under the same conditions and performing a good statisti-
cal analysis. Anything that happens only once can lead to no
other conclusion than “it is possible to obtain this result.” For
instance, when comparing two (or more) different settings, the
differences in outcomes may or may not be significant. Therefore,
the hypothesis of a difference between the processes that have
generated the results needs to be validated with appropriate statis-
tical tests (Bartz-Beielstein, 2006). Second, evolutionary robotics
experiments are, in general, performed on a simplified model of a
real robot or animal. Drawing any conclusion on the real robot or
animal requires discussing to what extent the model is appropri-
ate to study the research question (Hughes, 1997; Long, 2012). The
opportunistic nature of evolutionary algorithms makes this partic-
ularly mandatory, as the evolutionary process may have exploited
features that are specific to the simplified model, and which may
not hold on the targeted system, giving rise to a reality gap (Jakobi
et al., 1995; Koos et al., 2013b).

Although we have emphasized that all experimental research in
ER should follow the general template above, differences in objec-
tives can have implications for the methodology. For instance, a
study aiming at solving a particular engineering problem can be
successfully concluded by only inspecting the end result of the
evolutionary process, the evolved robot. Verifying success requires
the validation of the robot behavior as specified in the problem
description – analysis of the evolutionary runs is not relevant for
this purpose. On the other hand, comparing algorithmic imple-
mantations of ER principles requires thorough statistical analysis
of these variants. To this end, the existing practice of evolution-
ary computation can be very helpful. This practice is based on
using well-specified test problems, problem instance generators,
definitions of algorithm performance, enough repetitions with
different random seeds, and the correct use of statistics. This
methodology is known and proven, offering established choices
for the most important elements of the experimental workflow.
For instance, there are many repositories of test problems, several
problem instance generators, and there is broad agreement about
the important measures of algorithm performance, cf. Chapter 13
in Eiben and Smith (2003).

In the current evolutionary robotics literature, proof-of-concept
studies are common; these typically show that a robot controller
or morphology can be evolved that induces certain desirable or
otherwise interesting behaviors. The choice of target-
ged behaviors (fitness functions) and robot environments is to a
large extent ad hoc, and the use of standard test suites and bench-
marking is not as common as in evolutionary computing. Whether
adopting such practices from EC is actually possible and desirable
is an issue that should be discussed in the community.

4. WHAT DO WE KNOW?

Evolutionary robotics is a relatively young field with a his-
tory of about two decades; it is still in development. However,
it is already possible to identify a few important “knowledge
nuggets” – interesting discoveries of the field so far.

4.1. NEURAL NETWORKS OFFER A GOOD CONTROLLER PARADIGM
FOR ER

Robots can be controlled by many different kinds of controllers,
from logic-based symbolic systems (Russell and Norvig, 2009) to
fuzzy logic (Saffiotti, 1997) and behavior-based systems (Mataric
and Michaud, 2008). The versatility of evolutionary algorithms
allows them to be used with almost all of these systems, be it
to find the best parameters or the best controller architecture.
Nevertheless, the ideal substrate for ER should constrain evolution as little as possible, in order to make it possible to scale up to designs of unbounded complexity. As ER aims to use as little prior knowledge as possible, this substrate should also be able to use raw inputs from sensors and send low-level commands to actuators.

Given these requirements, artificial neural networks are currently the preeminent controller paradigm in ER. Feed-forward neural networks are known to be able to reproduce any function with arbitrary precision (Cybenko, 1989), and are well recognized tools in signal processing (images, sound, etc.) (Bishop, 1995; Haykin, 1998) and robot control (Miller et al., 1995). With recurrent connections, neural networks can also approximate any dynamical system (Funahashi and Nakamura, 1993). In addition, since neural networks are also used in models of the brain, ER can build on a vast body of research in neuroscience, for instance, on synaptic plasticity (Abbott and Nelson, 2000) or network theory (Bullmore and Sporns, 2009).

There are many different kinds of artificial neural networks to choose from, and many ways to evolve them. First, a neuron can be simulated at several levels of abstraction. Many ER studies use simple McCulloch and Pitts neurons (McCulloch and Pitts, 1943), like in "classical" machine learning (Bishop, 1995; Haykin, 1998). Others implement leaky integrators (Beer, 1995), which take time to into account and may be better suited to dynamical systems (these networks are sometimes called continuous-time recurrent neural networks, CTRNN). More complex neuron models, e.g., spiking neurons, are currently not common in ER, but they may be used in the future. Second, once the neuron model is selected, evolution can act on synaptic parameters, the architecture of the network, or both at the same time. In cases where evolution is applied only to synaptic parameters, common network topologies include feed-forward neural networks, Elman–Jordan networks [e.g., Mouret and Doncieux (2012)], which include some recurrent connections, and fully connected neural networks (Beer, 1995; Nolfi and Floreano, 2000; Bongard, 2011). Fixing the topology, however, bounds the complexity of achievable behaviors. As a consequence, how to encode the topology and parameters of neural networks is one of the main open questions in ER. How to encode a neural network so that it can generate a structure as complex, and also as organized, as a human brain? Many encodings have been proposed and tested, from direct encoding, in which evolution acts directly on the network itself (Stanley and Miikkulainen, 2002; Floreano and Mattiussi, 2008), to indirect encodings (also called generative or developmental encodings), in which a genotype develops into the neural network (Stanley and Miikkulainen, 2003; Floreano et al., 2008).

4.2. PERFORMANCE-ORIENTED FITNESS CAN BE MISLEADING

Most ER research, influenced by the vision of evolution as an optimization algorithm, relies on fitness functions, i.e., on quantifiable measures, to drive the search process. In this approach, the chosen fitness measure must increase, on average, from the very first solutions considered – which are in general randomly generated – toward the expected solution. Typical fitness functions rely on performance criteria, and implicitly assume that increasing performance will lead the search in the direction of desired behaviors. Recent work has brought this assumption into question and shown that performance criteria can be misleading. Lehman and Stanley (2011) demonstrated in a set of experiments that using the novelty of a solution instead of the resulting performance on a task can actually lead to much better results. In these experiments, the performance criterion was still used to recognize a good solution when it was discovered, but not to drive the search process. The main driver was the novelty of the solution with respect to previous exploration in a space of robot behavior features. Counter intuitively, driving the search process with the novelty of explored solutions in the space of behavioral features led to better results than driving the search with a performance-oriented measure, a finding that has emerged repeatedly in multiple contexts (Lehman and Stanley, 2008, 2011; Risi et al., 2009; Krich, 2010; Cuccu and Gomez, 2011; Mouret, 2011; Gomes et al., 2012, 2013; Gomes and Christensen, 2013; Lehman et al., 2013b; Liapis et al., 2013).

4.3. SELECTIVE PRESSURE IS AT LEAST AS IMPORTANT AS THE ENCODING

Many complex encodings have been proposed to evolve robot morphologies, control systems, or both [e.g., Sims (1994), Gruau (1995), Bongard (2002), Hornby and Pollack (2002), Stanley and Miikkulainen (2003), Doncieux and Meyer (2004), and Mouret and Doncieux (2008)]. Unfortunately, these encodings did not enable the unbounded complexity that had been hoped for.

There are two main reasons for this situation: (1) evolution is often prevented from exploring new ideas because it converges prematurely on a single family of designs, and (2) evolution selects individuals on the short term, whereas increases in complexity and organization are often only beneficial in the medium to long term. In a recent series of experiments, Mouret and Doncieux (2012) tested the relative importance of selective pressure and encoding in evolutionary robotics. They compared a classic fitness function to approaches that modify the selective pressure to try to avoid premature convergence. They concluded that modifying the selective pressure made much more difference to the success of these experiments than changing the encoding. In a related field – evolution of networks – it has also been repeatedly demonstrated that the evolution of modular networks can be explained by the selective pressure alone, without the need for an encoding that can manipulate modules [Kashlan and Alon (2005), Espinosa-Soto and Wagner (2010), Bongard (2011), and Clune et al. (2013), Section 5.2].

At the beginnings of ER, selective pressure was not a widely studied research theme. Fitness was viewed as a purely user-defined function for which, at best, the user could follow “best practices.” However, many recent developments have shown that the most evolutionary robotics experiments share common features and challenges, and can be tackled by generic techniques that modify the selective pressure (Doncieux and Mouret, 2014), like behavioral diversity (Mouret and Doncieux, 2012), novelty search [Lehman and Stanley (2011), Section 4.2], or the transferability approach [Koos et al. (2013b), Section 4.4]. All of these selective pressures take into account the robot’s behavior in their definition and thus drive the field away from the classical black-box optimization point of view (Figure 3). These encouraging results further suggest that a better understanding of selective pressures could help evolutionary robotics scale up to more complex tasks and
FIGURE 3 | Recent work on selective pressures suggest that taking the behavior into account in the selection process is beneficial. In particular, it may be helpful, rather than only considering fitness as indicated by some specific quantitative performance measure, to also take into account aspects of robot behavior, which are less directly indexed to successful performance of a task, e.g., novelty, diversity.

4.4. TARGETING REAL ROBOTS IS CHALLENGING

Simulation is a valuable tool in evolutionary robotics because it makes it possible for researchers to quickly evaluate their ideas, easily replicate experiments, and share their experimental setup online. A natural follow-up idea is to evolve solutions in simulation, and then transfer the best ones to the final robot: evolution is fast, because it occurs in simulation, and the result is applied to real hardware, so it is useful. Unfortunately, it is now well established that solutions evolved in simulation most often do not work on the real robot. This issue is called “the reality gap” (Jakobi et al., 1995; Miglino et al., 1995; Mouret et al., 2012; Koos et al., 2013b).

It has at least been documented with Khepera-like robots (obstacle avoidance, maze navigation) (Jakobi et al., 1995; Miglino et al., 1995; Koos et al., 2013b) and walking robots (Jakobi, 1998; Zagal and Ruiz-del Solar, 2007; Glette et al., 2012; Mouret et al., 2012; Koos et al., 2013a,b; Oliveira et al., 2013).

The reality gap can be reduced by improving simulators, for instance, by using machine learning to model the sensors of the target robot (Miglino et al., 1995) or by exploiting experiments with the physical robot (Bongard et al., 2006; Zagal and Ruiz-del Solar, 2007; Moекkel et al., 2013). Nevertheless, no simulator will ever be perfect, and evolutionary algorithms, as “blind” optimization algorithms, are likely to exploit every inaccuracy of the simulation. Jakobi et al. (1995) proposed a more radical approach: to prevent the algorithm from exploiting these inaccuracies, he proposed to hide things that are difficult to simulate in an “envelope of noise.” A more recent idea is to learn a function that predicts the limits of the simulation, which is called “a transferability function” (Mouret et al., 2012; Koos et al., 2013a,b). This function is learned by transferring a dozen controllers during the evolutionary process, and can be used to search for solutions that are both high-performing and well simulated (e.g., with a multi-objective evolutionary algorithm).

Another possible way to reduce the magnitude of the reality gap is to encourage the development of robust controllers. In this case, the differences between simulation and reality are seen as perturbations that the evolved controller should reject. Adding noise to the simulation, as in Jakobi’s “envelope of noise” approach (Jakobi et al., 1995), is a straightforward way to increase the robustness of evolved controllers. More generally, it is possible to reward some properties of simulated behaviors so that evolved controllers are less likely to over-fit in simulation. For instance, Lehman et al. (2013a) proposed to encourage the reactivity of controllers by maximizing the mutual information between sensors and motors, in combination with the maximization of classical task-based fitness. Last, controller robustness can be improved by adding online adaptation abilities, typically by evolving plastic neural networks (Urzelai and Floreano, 2001).

The reality gap can be completely circumvented by abandoning simulators altogether. Some experiments in 1990s thus evaluated the performance of each candidate solution using a robot in an arena, which was tracked with an external device and connected to an external computer (Nolfi and Floreano, 2000). These experiments led to successful but basic behaviors, for instance, wall-following or obstacle avoidance. Successful experiments have also been reported for locomotion (Hornby et al., 2005; Yosinski et al., 2011). However, only a few hundreds of evaluations can be realistically performed with a robot: reality cannot be sped up, contrary to simulators, and real hardware wears out until it ultimately breaks. A promising approach to scale up to more complex behaviors is to use a population of robots, instead of a single one (Watson et al., 2002; Bredeche et al., 2012; Haasdijk et al., 2014); in such a situation, several evaluations occur in parallel, therefore, the evolution process can theoretically be sped up by a factor equal to the number of robots.

These different approaches to bridging the reality gap primarily aim at making ER possible on a short time scale. However,
they also critically, and often covertly, influence the evolutionary process: they typically act as “goal refiners” (Doncieux and Mouret, 2014) – that is, they change the optimum objective function to restrict possible solutions to a subset that works well enough on the robots. For instance, the “envelope of noise” penalizes solutions that are not robust to noise – there may exist higher-performing solutions in the search space that are slightly less robust; the “transferability approach” can only find solutions that are accurately simulated – it cannot go beyond the capabilities of the simulator; and evaluating the fitness function directly on the robot might be influenced by uncontrolled conditions, for example, the light in the experimental room (Harvey et al., 1994).

4.5. COMPLEXIFICATION IS GOOD

Where to start when designing the structure of a robot? Should it be randomly generated up to a certain complexity? Should it start from the simplest structures and grow in complexity, or should it start from the most complex designs and be simplified over the generations? These questions have not yet been theoretically answered. One possibility is to start from a complex solution and then reduce its complexity. A neural network can thus begin fully connected and connections can be pruned afterwards through a connection selection process (Changeux et al., 1973). But a growing body of evidence suggests that starting simple and progressively growing in complexity is a good idea. This is one of the main principles of NEAT (Stanley and Miikkulainen, 2002), which is now a reference in terms of neural network encoding. Associated with the principle of novelty search, it allows the evolutionary process to explore behaviors of increasing complexity, and has led to promising results (Lehman and Stanley, 2011). Progressively increasing the complexity of the task, including the robot’s morphology (Bongard, 2010, 2011) and the context in which it is evaluated (Auerbach and Bongard, 2009, 2012; Bongard, 2011) has also been found to have a strong and positive influence on results. Furthermore, progressively increasing the complexity of the robot controller, morphology, or evaluation conditions aligns with the principles of ecological balance proposed by Pfeifer and Bongard (2007): the complexity of the different parts of a system should be balanced. If individuals from the very first generation are expected to exhibit a simple behavior, it seems reasonable to provide them with the simplest context and structure. This is also consistent with different fields of biology, either from a phylogenetic point of view (the first organisms were the simplest unicellular organisms) or from an ontogenetic point of view [maturation constraints reduce the complexity of perception and action for infants (Turkewitz and Kenny, 1985; Bjorklund, 1997)]. In any case, there is a need for more theoretical studies on these questions, as the observation of these principles in biology may result from physical or biological constraints rather than from optimality principles.

5. WHAT ARE THE OPEN ISSUES?

In this section, we give a treatment of the state of the art, organized around a number of open issues within evolutionary robotics. These issues are receiving much attention, and serving to drive further developments in the field.

5.1. APPLYING ER TO REAL-WORLD PROBLEMS

Real-world applications are not systematically followed by research papers advertising the approach taken by the engineers. It is thus hard to evaluate to what extent ER methods are currently used in this context. In any case, several successful examples of the use of ER methods in the context of real-world problems can be found in the literature. For instance, ER methods have been used to generate the behavior of UAVs acting in a swarm (Hauert et al., 2008), and the 2014 champion of the RoboCup simulation league relies on CMA-ES (Macalpine et al., 2015), an evolutionary algorithm. In these examples, ER methods were used either for one particular step in the design process or as a component of a larger learning architecture. Hauert et al. (2008) define a specific and preliminary step in the design process that relies on ER methods to generate original behaviors. Understanding how these behaviors worked led to new insights into how to solve the problem. ER methods were used only in simulation, and more classical methods were used to implement the solutions in real robots. Macalpine et al. (2015) define an overlapping layered learning architecture for a Nao robot to learn to play soccer, using CMA-ES (Hansen and Ostermeier, 2001) to optimize 500 parameters over the course of learning. Neither of these approaches implements ER as a full-blown holistic design method, but they show that its principles and the corresponding algorithms now work well enough to be included in robot design processes.

Using ER as a holistic approach to a real-world robotic problem remains a challenge because of the large number of evaluations that it implies. The perspective in this context is to rely either on many robots in parallel (see Sections 5.5 and 5.6 for the questions raised by this possibility) or at least partly on simulations. In the latter case, because of the inevitable discrepancies between simulation and reality and the opportunistic nature of ER, the challenge of the reality gap has to be faced. Several approaches have been proposed to cross this gap (Section 4.4). Relying on available simulations implies exploiting only the features that are modeled with sufficient accuracy. There have been a number of studies on building the model during the evolutionary run (Bongard et al., 2006; Zagal and Ruiz-del Solar, 2007; Moecckel et al., 2013), but the models generated in this way thus far are limited to simple experiments. Discovering models of robot and environment for more realistic setups remains an open issue.

ER applications have also emerged in a very different domain: computer graphics. From the seminal work of Karl Sims on virtual creatures (Sims, 1994) to more recent work on evolutionary optimization of muscle-based control for realistic walking behavior with simulated bipeds (Geijtenbeek et al., 2013), ER makes it possible to provide realistic and/or original animated characters.

5.2. NATURE-LIKE EVOLVABILITY

A central open question in evolutionary biology is what makes natural organisms so evolvable: that is, how species quickly evolve responses to new evolutionary challenges (Kaufman, 1993; Wagner and Altenberg, 1996; Kirschner and Gerhart, 2006; Gerhart and Kirschner, 2007; Pigliucci, 2008). And as ER takes inspiration from evolutionary biology, one of the main open questions in ER is how to design artificial systems that are as evolvable as natural species (Wagner and Altenberg, 1996; Tarapore and Mouret, 2014a,b).
To make evolution possible, random mutations need to be reasonably likely to produce viable (non-lethal) phenotypic variations (Wagner and Altenberg, 1996; Kirschner and Gerhart, 2006; Gerhart and Kirschner, 2007; Pigliucci, 2008). Viable variations are consistent with the whole organism. For instance, a variation of beak size in a finch requires consistent adaptation of both mandibles (lower and upper) and an adaptation of the musculoskeletal structure of the head. In addition, viable variations do not break down the vital functions of the organism: an adaptation of the beak should not imply a change in the heart. Likewise, a mutation that would add a leg to a robot would require a new control structure to make it move, but the same mutation should not break the vision system. This kind of consistency and weak linkage is impossible to achieve in a simple artificial evolutionary system in which each gene is associated to a single phenotypic trait (Wagner and Altenberg, 1996; Hu and Banzhaf, 2010).

In biology, the development process that links genes to phenotypes, called the genotype–phenotype map, is thought to be largely responsible for organisms’ ability to generate such viable variations (Wagner and Altenberg, 1996; Kirschner and Gerhart, 2006; Gerhart and Kirschner, 2007). Inspired by the progress made by the evo-devo community (Müller, 2007), many researchers in ER have focused on the genotype–phenotype map for neural networks (Kodjabachian and Meyer, 1997; Doncieux and Meyer, 2004; Mouret and Doncieux, 2008; Stanley et al., 2009; Verbancsics and Stanley, 2011) or complete robots (Bongard, 2002; Hornby and Pollack, 2002; Auerbach and Bongard, 2011; Cussat-Blanc and Pollack, 2012; Cheney et al., 2013). Essentially, they propose genotype–phenotype maps that aim to reproduce as many features as possible from the natural genotype–phenotype map, and, in particular, consistency, regularity (repetition, repetition with variation, symmetry), modularity (functional, sparsely connected subunits), and hierarchy (Lipson, 2007).

Intuitively, these properties should improve evolvability and should therefore be selected by the evolutionary process: if the “tools” to create, repeat, organize, and combine modules are available to evolution, then evolution should exploit them because they are beneficial. However, evolvability only matters in the long term and evolution mostly selects on the short term: long-term evolvability is a weak, second-order selective pressure (Wagner et al., 2004; Mouret and Doncieux, 2008; Stanley et al., 2009; Verbancsics and Stanley, 2011) or complete robots (Bongard, 2002; Hornby and Pollack, 2002; Auerbach and Bongard, 2011; Cussat-Blanc and Pollack, 2012; Cheney et al., 2013). Essentially, they propose genotype–phenotype maps that aim to reproduce as many features as possible from the natural genotype–phenotype map, and, in particular, consistency, regularity (repetition, repetition with variation, symmetry), modularity (functional, sparsely connected subunits), and hierarchy (Lipson, 2007).

Intuitively, these properties should improve evolvability and should therefore be selected by the evolutionary process: if the “tools” to create, repeat, organize, and combine modules are available to evolution, then evolution should exploit them because they are beneficial. However, evolvability only matters in the long term and evolution mostly selects on the short term: long-term evolvability is a weak, second-order selective pressure (Wagner et al., 2004; Mouret and Doncieux, 2008; Stanley et al., 2009; Verbancsics and Stanley, 2011) or complete robots (Bongard, 2002; Hornby and Pollack, 2002; Auerbach and Bongard, 2011; Cussat-Blanc and Pollack, 2012; Cheney et al., 2013). Essentially, they propose genotype–phenotype maps that aim to reproduce as many features as possible from the natural genotype–phenotype map, and, in particular, consistency, regularity (repetition, repetition with variation, symmetry), modularity (functional, sparsely connected subunits), and hierarchy (Lipson, 2007).

5.3. COMBINING EVOLUTION AND LEARNING

Animals’ ability to adapt during their lifetime may be the feature that most clearly separates them from current machines. Evolved robots with learning abilities could be better able to cope with environmental and morphological changes (e.g., damage) than non-plastic robots (Urzelai and Floreano, 2001): evolution would deal with the long-term changes, whereas learning would deal with short-term changes. In addition, many evolved robots rely on artificial neural networks (Section 4.1), which were designed with learning in mind from the outset. There is, therefore, a good fit between evolving neural networks in ER and evolving robots with online learning abilities. Most papers in this field have focused on various forms of Hebbian learning (Abbott and Nelson, 2000; Urzelai and Floreano, 2001), including neuro-modulated Hebbian learning (Bailey et al., 2000; Soltoggio et al., 2007, 2008).

Unfortunately, mixing evolution and online learning has proven difficult. First, many papers about learning in ER address different challenges while using the same terminology (e.g., “learning,” “robustness,” and “generalization”), making it difficult to understand the literature. A recent review may help clarify each of the relevant terms and scenarios (Mouret and Tonelli, 2014): it distinguishes “behavioral robustness” (the robot maintains the same qualitative behavior in spite of environmental/morphological changes; no explicit reward/punishment system is involved) from “reward-based behavioral changes” (the behavior of the robot depends on rewards, which are set by the experimenters). This review also highlights the difference between evolving a neural network that can use rewards to switch between behaviors that are measured by the fitness function, on the one hand, and being able to learn in a situation that was not used in the fitness function, on the other hand. Surprisingly, only a handful of papers evaluate the actual learning abilities of evolved neural networks in scenarios that were not tested in the fitness function (Tonelli and Mouret, 2011, 2013; Coleman et al., 2014; Mouret and Tonelli, 2014).

Second, crafting a fitness function is especially challenging. To evaluate the fitness of robots that can learn, the experimenter needs to give them the time to learn and test them in many different situations, but this kind of evaluation is prohibitively expensive (many scenarios, each of which has to last a long time), especially with physical robots. Recent results suggest that a bias toward network regularity might enable the evolution of general learning abilities without requiring many tests in different environments (Tonelli and Mouret, 2013), but these results have not yet been transferred to ER. In addition, evolving neural networks with learning abilities appears to be a deceptive problem (Risi et al., 2009, 2010) in which simple, non-adaptive neural networks initially outperform networks with learning abilities, since the latter are more complex to discover and tune. Novelty search (Section 4.2) could help mitigate this issue (Risi et al., 2009, 2010).
Last, evolution and learning are two adaptive processes that interact with each other. Overall, they form a complex system that is hard to tame. The Baldwin effect (Hinton and Nowlan, 1987; Dennett, 2003) suggests that learning might facilitate evolution by allowing organisms (or robots) to pretest the efficiency of a behavior using their learning abilities, which will create a selective pressure to make it easier to discover the best behaviors, which in turn should result in the inclusion of these behaviors in the genotype (so that there is no need to discover the behavior again and again). The Baldwin effect can thus smooth a search landscape and accelerate evolution (Hinton and Nowlan, 1987).

Nonetheless, from a purely computational perspective, the ideal balance between spending time on the fitness function, to give individuals the time to learn, and using more generations, which gives evolution more time to find good solutions, is hard to find. In many cases, it might be more computationally efficient to use more generations than to use learning to benefit from the Baldwin effect. More generally, evolution and learning are two processes that optimize a similar quantity – the individual’s performance. In nature, there is a clear difference in time scale (a lifetime vs. millions of years); therefore, the two processes do not really compete with each other. However, ER experiments last only a few days and often solve problems that could also be solved by learning: given the time scale of current experiments, learning and evolution might not be as complementary as in nature, and they might interact in a different way.

5.4. EVOLUTIONARY ROBOTICS AND REINFORCEMENT LEARNING

Although reinforcement learning (RL) and ER are inspired by different principles, they both tackle a similar challenge (Togelius et al., 2009; Whiteson, 2012): agents (e.g., robots) obtain rewards (resp. fitness values) while behaving in their environment, and we want to find the policy (resp. the behavior) that corresponds to the maximum reward (resp. fitness). Contrary to temporal difference algorithms (Sutton and Barto, 1998), but like direct policy search algorithms (Kober et al., 2013), ER only uses the global value of the policy and does not construct value estimates for particular state-action pairs. This holistic approach makes ER potentially less powerful than classic RL because it discards all the information in the state transitions observed during the “life” (the evaluation) of the individual. However, this approach also allows ER to cope better than RL with partial observability and continuous domains like robot control, where actions and states are often hard to define. A second advantage of ER over RL methods is that methods like neuro-evolution (Section 4.1) evolve not only the parameters of the policy but also its structure, hence avoiding the difficulty (and the bias) of choosing a policy representation (e.g., dynamic movement primitives in robotics) or a state-action space.

A few papers compare the results of using evolution-inspired algorithms and reinforcement learning methods to solve reinforcement learning problems in robotics. Evolutionary algorithms like CMA-ES (Hansen and Ostermeier, 2001) are good optimizers that can be used to optimize the parameters of a policy in lieu of gradient-based optimizers. In a recent series of benchmarks, Heidrich-Meisner (2008) compared many ER and RL algorithms to CMA-ES in evolving the weights of a neural network (Heidrich-Meisner, 2008; Heidrich-Meisner and Igel, 2009). They used classic control problems from RL (cart-pole and mountain car) and concluded that CMA-ES outperformed all the other tested methods. Stulp and Sigaud (2012) compared CMA-ES to policy search algorithms from the RL community and concluded that CMA-ES is a competitive algorithm to optimize policies in RL (Stulp and Sigaud, 2012, 2013). Taylor et al. (2006, 2007) compared NEAT (Stanley and Miikkulainen, 2002) to SARSA in the keep-away task from RoboCup. In this case, the topology of a neural network—that is, the structure of the policy – was evolved. Their results suggest that NEAT can learn better policies than SARSA, but requires more evaluations. In addition, SARSA performed better when the domain was fully observable, and NEAT performed better when the domain had a deterministic fitness function.

Together, these preliminary results suggest a potential convergence between RL for robotics and ER for behavior design. On the one hand, RL in robotics started some years ago to explore alternatives to classical discrete RL algorithms (Sutton and Barto, 1998) and, in particular, high-performance continuous optimization algorithms (Kober et al., 2013). On the other hand, EAs have long been viewed as optimization algorithms, and it is not surprising that the most successful EAs, e.g., CMA-ES, make good policy search algorithms for RL (Stulp and Sigaud, 2012), and are very close to popular algorithms from the RL literature, like cross-entropy search (Rubinstein and Kroese, 2004). Methods that learn a model of the reward function, like Bayesian optimization (Mockus et al., 1978), also offer a good example of convergence: they have been increasingly used in RL (Lizotte et al., 2007; Calandra et al., 2014), but they are also well studied as “surrogate-based methods” in evolutionary computation (Jin, 2011). Overall, evolution and policy search algorithms can be seen as two ends of a continuum: EAs favor exploration and creativity, in particular, when the structure of the policy is explored, but they require millions of evaluations; RL favors more local search, but requires fewer evaluations. There are, however, situations in which an ER approach cannot be substituted by a RL approach. For instance, RL is typically not the appropriate framework to automatically search for the morphology of a robot, contrary to evolution (Lipson and Pollack, 2000; Hornby and Pollack, 2002; Bongard et al., 2006; Auerbach and Bongard, 2011; Cheney et al., 2014); the same is true of modeling biological evolution (Floreano et al., 2007; Waibel et al., 2011; Long, 2012; Montanier and Bredeche, 2013; Auerbach and Bongard, 2014).

5.5. ONLINE LEARNING: SINGLE AND MULTIPLE ROBOTS

Evolutionary robotics is traditionally used to address design problems in an offline manner: there is a clear distinction between the design phase, which involves optimization and evaluation, and the operational phase, which involves no further optimization. This two-step approach is rather common in engineering and is appropriate for most problems, as long as the supervisor is able to provide a test bed that is representative with regard to the ultimate setup. However, it also assumes that the environment will remain stable after deployment, or that changes can be dealt with, for instance, thanks to robust behaviors or an ability to learn (cf. Section 5.3). However, this assumption does not always hold, as the environment may be unknown prior to deployment, or may change drastically over time.
Online versions of evolutionary robotics have been studied in order to address this class of problem (Eiben et al., 2010), both with single robots (Bongard et al., 2006; Bredeche et al., 2010) and multiple robots (Watson et al., 2002). Online ER benefits from the properties of its offline counterpart: black-box optimization algorithms that can deal with noisy and multimodal objective functions where only differential evaluations can be used to improve solutions. However, some features are unique to online optimization: each robot is situated in the environment, and evolutionary optimization should be conducted without (or least with a minimum of) external help. This has two major implications: first, that there are no stable initial conditions with which to begin the evaluation as the robot moves around (Bredeche et al., 2010); and second, that safety cannot be guaranteed as the environment is, by definition, unknown.

Going from single to multiple robots also raises particular challenges. Interestingly, distributed online evolutionary robotics [also termed "embodied evolution," a term originally coined by Ficici et al. (1999)] has drawn more attention than the single-robot flavor since the very beginning. On the one hand, the theoretical complexity of learning exact solutions with multiple robots [i.e., solving a decentralized partially observable Markov decision process (DEC-POMDP) in an exact manner] is known to be NEXP-complete (Papadimitriou, 1994) already in the offline setup (Bernstein et al., 2002). On the other hand, the algorithms at work in evolutionary robotics can easily be distributed over a population of evolving robots. While the result is limited to an approximate method for distributed online learning, which is subject to many limitations and open issues, it is possible to deploy such algorithms in a large population of robots, from dozens of real robots (Watson et al., 2002; Bredeche et al., 2012) to thousands of virtual robots (Bredeche, 2014). Beyond the obvious limitations in terms of reachability solutions (e.g., complex social organization, ability to adapt quickly to new environments, learning to identify no-go zones), one major open issue is bridging the gap between practice and theory, which is nearly non-existent in the field as it stands. This becomes particularly obvious in comparisons to more formal approaches such as optimal or approximate algorithms for solving DEC-POMDP problems, which have yet to scale up beyond very few robots, even with approximate methods [e.g., Amato et al. (2015)], but which benefit from a well-established theoretical toolbox [see Goldman and Zilberstein (2004) for a review].

5.6. ENVIRONMENT-DRIVEN EVOLUTIONARY ROBOTICS

In nature, an individual’s success is defined \textit{a posteriori} by its ability to generate offspring. From the evolutionary perspective, one has to consider complex interactions with the environment and other agents, which often imply the need to face a challenging tradeoff between the pressure toward reproduction and the pressure toward survival. Environment-driven evolutionary robotics deals with similar concerns, and considers multiple robots in open environments, where communication between agents generally happens in a peer-to-peer fashion, and where reproductive success depends on encounters between individuals rather than being artificially decided on the basis of a fitness value that is computed prior to reproduction. A particular individual’s success at survival and reproduction can thus only be assessed using a population-level measure, the number of offspring that the individual generates.

Focusing on the environment as a source of selection pressures can enable the emergence of original behavioral strategies that may have been out of the reach of goal-directed evolution. First, it is unlikely to be possible to formulate a fitness function in advance that adequately captures the description of behaviors relevant for survival and reproduction. Second, objective-driven selection pressure sometimes acts in a very different direction than environmental selection pressure. For example, a possible solution for maximizing coverage in a patrolling task is to keep agents away from one another, which is quite undesirable in reproductive terms, and may end up being counterproductive if behavioral targets include both survival and task efficiency. To some extent, environment-driven evolution can be considered as an extreme case of avoiding the pitfalls of performance-oriented fitness as discussed in Section 4.2.

While environment-driven evolution has been studied for almost 25 years in the field of artificial life (Ray, 1991), its advent in the field of evolutionary robotics is very recent (Bianco and Nolfi, 2004; Bredeche and Montanier, 2010), especially in real robots (Bredeche et al., 2012). Major challenges lie ahead: (1) from the perspective of basic research, including (of course) evolutionary biology, the dynamics of environment-driven evolution in populations of individuals, let alone robots, is yet to be fully understood; (2) from the perspective of applied research, addressing complex tasks in challenging environment could benefit from combining environment-driven and objective-driven selection, as hinted by preliminary researches in this direction (Haasdijk et al., 2014).

5.7. OPEN-ENDED EVOLUTION

Natural evolution is an open-ended process, constantly capable of morphological and/or behavioral innovation. But while this open-endedness can be readily observed in nature, understanding what general principles are involved (Wiser et al., 2013) and how to endow an artificial system with such a property (Ray, 1991; Bedau et al., 2000; Soros and Stanley, 2014) remain a major challenge. Some aspects of open-ended evolution have been tackled from the perspective of evolutionary robotics, from increasing diversity (see Section 4.2) and complexity (see Section 4.5) to continuous adaptation in an open environment (see Section 5.6), but a truly open-ended evolutionary robotics is yet to be achieved.

Why, in fact, should we aim toward an open-ended evolutionary robotics? This question can actually be addressed from several different viewpoints. From the biologist’s viewpoint, open-endedness could allow for more accurate modeling of natural evolution, where current algorithms oversimplify, and possibly miss, critical features. From the engineer’s viewpoint, on the other hand, open-endedness could ensure that the evolutionary optimization process will never stop, allowing it to bypass local optima and continue generating new – and hopefully more efficient – solutions.

6. WHAT IS TO BE EXPECTED?

In this section, we give an overview of the expected outcomes in the future of evolutionary robotics for the scientific and engineering community.
6.1. NEW INSIGHTS INTO OPEN QUESTIONS

6.1.1. How do the body and the mind influence each other?
The body and the mind have a strong coupling. A deficient body will require a stronger mind to compensate and vice versa. Living creatures have a balanced complexity between the two and such an equilibrium between the two could greatly simplify future robots (Pfeifer and Bongard, 2007). Evolutionary Algorithms are models of natural selection that are robust and can thus cope with changes in the morphology (Bongard et al., 2006; Koos et al., 2013a). The versatility of these algorithms and their ability to generate structures also allows them to generate both controllers and morphologies (Sims, 1994; Lipson and Pollack, 2000; Komosinski, 2003; de Margerie et al., 2007). Computational models relying on Evolutionary Algorithms can thus be built to provide insights into how the body and the mind influence each other. With such models, it has been shown that progressively increasing the complexity of the morphology has an impact on the robot controller adaptation process and on the complexity of the tasks it is able to fulfill (Bongard, 2010, 2011). Modifications of the environment were also shown to significantly influence this process. Likewise morphological and environmental adaptations revealed to be dependent, and thus either to coalesce to make the process even more efficient, or conflict and dampen efficiency (Bongard, 2011).

6.1.2. A path toward the emergence of intelligence
As stated by Fogel et al. (1966), “Intelligent behavior is a composite ability to predict one's environment coupled with a translation of each prediction into a suitable response in light of some objective.” Designing systems with this ability was the initial motivation of genetic algorithms (Holland, 1962a) and evolutionary programing (Fogel et al., 1966). A theory of adaptation through generation procedures and selection was thus developed (Holland, 1962b), and a “cognitive system” relying on classifier systems and a genetic algorithm with the ability to adapt to its environment was proposed (Holland and Reitman, 1977). Designing intelligence thus has a long history within the evolutionary computation community. It turns out that two different paths can actually be taken to reach this goal.

Darwin (1859) proposed natural selection as the main process shaping all living creatures. Within this theoretical framework, it is thus also responsible for the development and enhancement of the organ most involved in intelligence: the brain. Likewise, the optimization and creative search abilities of evolutionary algorithms can design robot “brains” to make them behave in a clever way. Much work in evolutionary robotics has been motivated by this analogy: the robot brain is generated after a long evolutionary search process, after which it is fixed and used to control the robot. Evolutionary algorithms can also design robot brains that include some plasticity so that, beyond its intrinsic robustness, the system maintains some adaptive properties (Floreano et al., 2008; Risi et al., 2010; Tonelli and Mouret, 2013).

But beyond this possibility of definitively setting the form of the organ responsible for intelligence once and for the robot’s entire “lifespan,” the evolutionary process could be used in an online fashion and run during the robot’s lifespan to make it more efficient with time and keep it adapted. In this context, evolution is at work over the course of a single individual lifetime, rather than through generational time. The population of solutions handled by the algorithm is located within the brain of a single individual. Bellas and Duro (2010) thus proposed a cognitive architecture relying on evolutionary processes for life-long learning. This online use of generation and selection algorithms to adapt the behavior of a system to its environment has also inspired neuroscientists, who have proposed models in which such mechanisms are involved to explain some of the most salient adaptive properties of our brains (Changeux et al., 1973; Edelman, 1987; Fernando et al., 2012).

6.1.3. Evolution of social behaviors
In nature, many species display different kinds of social behaviors, from simple coordination strategies to the many flavors of cooperation. The evolution of social behaviors raises the question of the necessary conditions as well as that of interaction with the evolution of other traits, which may or may not be required for cooperation, such as the ability to learn, to explicitly communicate, or even to recognize one's own kin. Evolutionary robotics may contribute in two different ways: by helping understand the why and how of social behaviors, and by producing new designs for robots capable of complex collective patterns.

From the perspective of evolutionary biology, evolutionary robotics can be used to study the particular mechanisms required for cooperation to evolve, and some work has already begun to explore in this direction [e.g., Waibel et al. (2011) on the evolution of altruistic cooperation]. For example, it is well known that cooperation can evolve, at least theoretically, whenever mutual benefits are expected, for example, when a large prey animal can only be caught when several individuals cooperate (Skyrms, 2003). However, it is not clear how the necessary coordination behavior (i.e., synchronizing moves among individuals) may, or may not, evolve depending on the particular environment (e.g., prey density, sensorimotor capabilities, etc.). As stated in Section 2.2, evolutionary robotics can help to model such mechanisms, further extending the understanding of this problem, which has previously been tackled with more abstract methods (e.g., evolutionary game theory), by adding a unique feature: modeling and simulating the evolved behaviors of individuals, emphasizing on the phenotypic traits rather than just the genotypic traits.

From an engineering perspective, the evolution of social behaviors opens up the prospect of a whole new kind of design in evolutionary swarm and collective robotics. Contributions so far have been focused on proofs-of-concept, concentrating either on the evolution of self-organized behaviors (Trianni et al., 2008) or on cooperation for pushing boxes from one location to another (Waibel et al., 2009). The evolution of teams capable of complex interactions and division of labor remains a task for future work in evolutionary robotics. By establishing a virtuous circle between biology and engineering, it can be expected that a better understanding of the underlying evolutionary dynamics can help to design systems that target optimality rather than adaptability. For example, it is well known in evolutionary biology that kin selection naturally favors the evolution of altruistic cooperation (Hamilton, 1964), and this can be used to tune how kin should be favored, or not, during an artificial selection process, to balance between optimality at the level of the robot (best individual
performance) and (part of) the population (highest global welfare of related individuals) (Montanier and Bredeche, 2011).

6.1.4. Existence proofs

There is no simple relationship between the properties of an animal or a robot (morphology and brain or control system) and the behavior it exhibits. A particular behavior may result from causes that are difficult to identify and, symmetrically, there may be many simple ways to achieve a particular behavior that are particularly hard to find. A well-known example is in Braitenberg (1984), where vehicles exhibit complex behaviors without a complex decision process or any kind of cognitive architecture. Likewise, a memory-less architecture has also been shown to exhibit a behavior requiring a memory thanks to careful exploitation of the environment (Ziemke and Thieme, 2002). The body–brain–environment loop is complex and frequently misleads our human intuition, biased by our anthropomorphic way of solving such problems. Evolutionary robotics can thus be used as an unbiased search process, systematically exploring solutions that a human would neglect or never think of. Counterintuitive solutions can emerge from this process and offer new insights into the requirements for the corresponding behavior.

6.2. NEW TOOLS AND METHODS FOR ROBOT DESIGN

As mentioned previously, evolutionary robotics can be perceived as a testing ground or experimental toolbox for studying various issues arising on the road to intelligent and autonomous machines. In this role, the evolutionary approach offers a design methodology that differs from conventional robotics. In general, evolutionary algorithms (EA) are praised for their ability to solve hard problems that are not well understood, such as problems that do not have neat analytical models, problems involving many parameters with non-linear interactions, and problems that suffer from noise and many local optima. Furthermore, EAs can be applied without much adjustment to the problem at hand. This makes them a good tool for investigating the design space of robotic applications. A punchy way of positioning EAs is: “An evolutionary algorithm is the second best solver for any problem”¹. The underlying observation is that, for any given problem, a superior problem-specific solver can be developed that would beat the evolutionary algorithm. However, this comes at a high price in time and effort for development, and the price for developing a superior solver for a particular problem can be (and often is) too high. In practice, one is often satisfied with a reasonably good solution. A perfect solution is not always required, or is an “expensive” problem solving algorithm. In such cases, evolution is a very suitable approach in general.

6.2.1. Design of morpho-functional machines

Traditional design methods rely on problem decomposition. Independently building the morphology, the sensory apparatus, the control, and the decision architecture allows researchers to focus on one aspect at a time, which is much simpler as separate research teams can work on each one in parallel. Such an approach has a strong drawback: it makes discovering and exploiting the synergies between these parts more difficult. Morpho-functional machines are machines that exploit such synergies (Hara and Pfeifer, 2003). In these machines, there is no clear separation between the morphology and the decision device. The morphology performs “computation,” which influences the behavior of a robot, but without a computer. Typical examples are passive walker robots, which use the dynamics of their legs to walk bipedally with little effort and computational power (Collins et al., 2003). Evolutionary robotics relies on evaluations of the complete robot in interaction with its environment. It implies that all of the robot’s parts are taken into account simultaneously. These methods thus naturally take into account the synergies between those parts and the robot’s environment, and the generated solutions rely on them (Doncieux and Meyer, 2005).

6.2.2. New representations for robot controllers and morphology

Evolutionary robotics requires defining a representation of the solutions to be explored. This representation should be compatible with random generation, mutation, and crossover, and should include solutions of interest. Finding appropriate representations is actually one of the most critical questions in evolutionary algorithms in general (Rothlauf, 2006). A great deal of effort has been dedicated in ER to the synthesis of neural networks, i.e., oriented graphs (Yao, 1999; Floreano et al., 2008). Many different encodings have been proposed to generate networks exhibiting modularity (Gruau, 1995; Doncieux and Meyer, 2004), regularity (Stanley et al., 2009), and even hierarchy (Mouret and Doncieux, 2008). For the evolution of robot morphologies, representations including both the robot structure and the control part have been proposed (Sims, 1994; Lipson and Pollack, 2000; Auerbach and Bongard, 2011).

A number of different kinds of representations have been considered (Floreano et al., 2008). Direct representations – also called direct encodings – manipulate the structure to be designed with no intermediate representation. In the generation of a neural network, mutations typically add or remove neurons and connections or change their parameters; crossovers are seldom used because of the permutation problem (Radciff, 1993), but when they are, they rely on techniques to match neural network parts before exchanging them (Stanley and Miikkulainen, 2002). Indirect representations – also called developmental representations or indirect or generative encodings – rely on a mapping between the genotype and the phenotype (Kowaliv et al., 2014). The main goal with such approaches is to define compact representations of complex systems with search operators that can typically exploit modularity (Gruau, 1995; Doncieux and Meyer, 2004; Mouret and Doncieux, 2008) and repetition with variation (Stanley et al., 2009).

How to define a compact representation that can cover each part of a robot, from its morphology, including its sensors and motors with their configuration, to its controller, is an open question in ER. The field can thus be expected to contribute to answering this question in the future. It should also be noticed that this question may not be independent of the selective pressures at play (Mouret and Doncieux, 2012; Clune et al., 2013; Doncieux and Mouret, 2014) and thus that representation and selective pressures should be considered simultaneously.

¹ Quoted from Sean Luke, personal oral communication.
6.2.3. Drivers for innovative design

What use is a wing or a leg before it is efficient enough to make the corresponding robot fly or walk? Can locomotion efficiency be the main and only driver for the design of both robot morphology and control? Biologists have looked deeply into these questions for living creatures, and proposed the concept of “exaptation” (Gould and Vrba, 1982). In exaptation, the purpose initially served by complex organs is different than one that they are observed to fulfill later. The initial functionality enables the changes that ultimately culminate in the development of the different and potentially more complex function. The wings of birds may thus have helped them escape from predators by climbing steep inclines (Dial, 2003), and tetrapods may have developed their legs to walk on the sea floor before conquering the land (Shubin, 2008). This suggests that the development of complex organisms is probably not driven by a single and monolithic pressure relying solely on global performance. Novelty search is an example of an approach, which shows that interesting results can be generated even if the goal description does not drive the search process at all (Lehman and Stanley, 2011).

Driving the search process and describing the expected solutions are then two different roles, and selective pressures can be defined for both of them, whether they are task-specific or task-agnostic (Doncieux and Mouret, 2014). This trend is recent in ER, and it can be expected to lead to new insights on the question of what drivers to use for innovative design.

6.3. New evolutionary mechanisms

As of today, the majority of ER applications employ a traditional black-box evolutionary algorithm, where the targeted robot features are evolved offline, before the operational period, and then implemented and not changed after deployment. In these cases, only the fitness evaluations are application specific, while the rest of the evolutionary algorithm is standard (see Figure 1). However, the application may require online evolution in groups of autonomous robots (Watson et al., 2002; Eiben et al., 2010) (Figure 4). This is the case if optimizing and fixing robot features prior to deployment is not possible, because the operational circumstances are not fully known in advance and/or are changing over time. In these cases, the traditional EA approach will not work, and new types of selection–reproduction schemes are necessary.

Such applications have a number of noteworthy properties with regard to the evolutionary algorithm. First, the targeted objective of evolution is twofold: usefulness and viability (Haasdijk et al., 2014). Usefulness means the optimization of some user-defined properties or skills, which are typically quantifiable. Viability requires adaptation to the given environment, where the criteria are not crisp and evaluation is based not on a quantifiable assessment but on the ability to survive and reproduce. The ideas of viability and survivability have drawn recent attention in the field of embodied evolution (Bredeche and Montanier, 2010), as well as in the larger field of evolutionary optimization (Maesani et al., 2014): considering more than just the best individuals enables the exploration of a larger set of candidate solutions, some of which may turn out to be more efficient in the end. Acting in this role, artificial evolution is indeed quite similar to the natural variant (Mayr, 2001).

Another special property is that the robots’ relationship to the evolutionary algorithm can be active. That is, rather than having its own controller being passively selected and chosen for reproduction by a centralized evolution manager, robots can select their mating partners, perform recombination on the two parent controllers, evaluate the child controllers, and select the best one for further usage. Obviously, different robots can apply different reproduction operators and/or different selection preferences, thus forming a new type of heterogeneous evolutionary system.

The third interesting property is that robots implicitly influence the evolutionary dynamics by structuring the population. Spatially structured EAs are, of course, nothing new (Tomassini, 2005). However, in evolving robot swarms this structure inherently changes over time and emerges from the bottom-up, without being explicitly controllable by the EA designer. For instance, maximum sensor and communication ranges imply (dynamically changing) neighborhoods that affect evolutionarily relevant interactions, e.g., mate selection and recombination. Obviously, sensor ranges and

![Figure 4](https://www.frontiersin.org)
the like are robot attributes that can be controlled by the designer or experimenter, but their evolutionary effects are complex, and to date there is no codified know-how about tweaking evolutionary algorithms through such parameters.

7. CONCLUSION
Evolutionary robotics as a research field has matured over the years. It has contributed to both engineering and biology, offering insights and opening new perspectives on holistic system design and related questions.

From an engineering point of view, it is a unique design process for systems endowed with embodied intelligence. It allows designers to work on the morphology, sensory apparatus, and controller at the same time, and takes into account each of these parts as they are – i.e., with their real features, including noise and nonlinearities, and not as they should be. ER therefore provides ideal tools to study the interdependencies between these different parts in the design process, and can easily be applied to unconventional robots (swarm, soft robots, modular robots, etc.). Likewise, it raises the question of which drivers should be used in such a holistic approach, and challenges the prevalence of goal-oriented fitness functions.

From a biological point of view, ER provides distinctive and powerful modeling tools for experimental work on evolution. Its agent-centered nature, associated to its behavior-based selection process, allows researchers to model phenomena that are difficult to study using a simplified and global statistics-based process. It provides the tools of choice to study the evolution of social behaviors, including cooperation and communication.

ER is an experimental field that relies on many evaluations to produce any result. Furthermore, its stochasticity implies that any experiment needs to be repeated several times before drawing any conclusion. This feature has limited the questions that have been considered in practice thus far – but this limitation should be carefully reconsidered. Evolutionary algorithms are easy to parallelize, so the length of an experiment can be drastically reduced if it is run in parallel on several processing units. The availability of powerful clusters of multi-core processors, together with the development of software frameworks that can exploit them (Mouret and Doncieux, 2010), opens new perspectives and allows researchers to consider questions that were out of reach for ER just a few years ago due to these practical limitations. This technological trend is expected to have a strong impact on ER, and lead to new results and insights into open questions in the future.

AUTHOR CONTRIBUTIONS
SD coordinated the paper. All authors equally contributed to the ideas and the writing.

ACKNOWLEDGMENTS
This work is supported by the ANR CreAdapt project (ANR-12-JS03-0009) and the European Union’s Horizon 2020 research and innovation program under grant agreement No 640891. AE received a grant from UPMC for a 1-month visit at ISIR/UPMC. The paper was initiated during this stay.

REFERENCES

Frontiers in Robotics and AI | Evolutionary Robotics March 2015 | Volume 2 | Article 4 | 14

Doncieux et al.


