ALMA MATER STUDIORUM · UNIVERSITÀ DI BOLOGNA

Master in Artificial Intelligence

Master thesis
in
Intelligent Robotics Systems

A Comparative Analysis of Darwinian Asexual and Sexual Reproduction in Evolutionary Robotics

Author:
Carlo Longhi

Supervisor:
Andrea Roli
Co-Supervisor:
Jie Luo

Academic Year 2021/2022

ABSTRACT

Evolutionary Robotics systems draw inspiration from natural evolution to solve the problem of robot design. A key moment in the evolutionary process is reproduction, when the genotype of one or more parents is inherited by their offspring. Existent approaches have used both sexual and asexual reproduction but a comparison between the two is still missing. In this work, we study the effects of sexual and asexual reproduction on the controllers of an Evolutionary Robotics system. In our system, both morphologies and controllers are jointly evolved to solve two separate tasks. We adopt the Triangle of Life framework, in which the controllers go through a phase of learning before reproduction. Using extensive simulations we show that sexual reproduction of the robots' brains is preferable over asexual reproduction as it obtains better robots in terms of fitness. Moreover, we show that sexually reproducing robots present different morphologies and behaviors than the asexually reproducing ones, even though the reproduction mechanism only affects their brains. Finally, we study the effects of the reproduction mechanism on the robots' learning capabilities. By measuring the difference between the inherited and the learned brain we find that robots that evolved using sexual reproduction have better inherited brains and are also better learners.

Keywords Embodied Intelligence, Machine Learning, Artificial Intelligence, Evolutionary Robotics

Contents

1 Introduction **4**

2 Related Works **7**

2.1 Evolutionary Computing . . . 7
2.1.1 Evolutionary Robotics . . . 8
2.1.2 Triangle of Life . . . 8
2.2 Robot Design . . . 8
2.2.1 Morphology Design . . . 9
2.2.2 Controller Design . . . 9
2.3 Evolutionary Loop . . . 10
2.3.1 Evolving Robots . . . 10
2.3.2 Behavior . . . 11
2.3.3 Environment . . . 11
2.3.4 Genetic Encoding . . . 11
2.3.5 Reproduction Mechanism . . . 12
2.3.6 Learning . . . 12

3 Methods **14**

3.1 Robot Morphology (Body) . . . 14
3.1.1 Body Phenotype . . . 14
3.1.2 Body Genotype . . . 14
3.2 Robot Controller (Brain) . . . 15
3.2.1 Brain Phenotype . . . 15
3.2.2 Brain Genotype . . . 15
3.3 Evolution . . . 16
3.3.1 Initialization . . . 16
3.3.2 Controller Learning . . . 16
3.3.3 Evaluation . . . 18
3.3.4 Reproduction . . . 21
3.3.5 Survivor Selection . . . 21

4 Experimental Setup **22**

4.1 Performance Measures . . . 22
4.2 Morphological Descriptors . . . 22

5 Results **24**

5.1 Performance . . . 24
5.2 Morphology . . . 25
5.3 Learning Delta . . . 25

5.4 Behavior 25

6 Conclusion and Future Work **28**

1 Introduction

Designing a robot is a task that requires taking into consideration many different aspects: its morphology, its controller, its motor system, its sensory apparatus, etc. The classical engineering approach is a reductionist one in which all of these parts are considered separately. One of the main challenges that arise while designing a robot, however, is that the final robot behavior is the result of the interactions among all of its components. The behavioral and cognitive skills of robots are dynamic properties that unfold in time and arise from a large number of interactions between the agents' nervous system, body, and environment (Cangelosi et al. [2015]. The alternative perspective of Embodied Intelligence (Pfeifer and Bongard [2007]), instead, approaches the study of a robot's intelligent behavior by considering its relationships with the environment and the constraints posed by its body, perceptual and motor system, and brain.

An explanatory example of the advantages of adopting this viewpoint is the problem of grasping an object with a robotic hand. Using a dexterous hand with 20 degrees of freedom (Walker [2005]) or 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 perspective. The control of the first hand is a real challenge whether the body of the second one can passively exploit the interactions with the environment to simplify the controller's job. We can say that the second hand has a higher degree of morphological intelligence and can perform more morphological computation. When adopting this new approach, however, the design problem can not be decomposed into its simpler parts anymore and becomes very difficult to deal with. Evolutionary Robotics is a research field that draws inspiration from nature to solve this problem. In particular, the mechanism responsible for the creation of living beings, evolution, is applied to the creation of artificial beings. If both the morphologies and controllers of a robot are optimized, the evolutionary search may be able to find robots whose bodies exhibit a greater amount of morphological computation than robots designed by hand.

We study an Evolutionary Robotics system in which the morphologies of a population of robots and their controllers are evolved simultaneously. The best robots are chosen to produce new robotic offspring and the new bodies and brains are formed from sexual or asexual reproduction starting from their parents. The new robots are then tested on a task and a fitness value is assigned to them based on their performance. The best individuals are selected to form the new population.

In the case of asexual reproduction, a robot is produced by the random variation of a single parent. For sexual reproduction, the body and brain of the parents need to be combined to produce offspring. Given the stochastic nature of these operations, we can not assume the newly produced body and brain to be a good match. Parents with well-matching bodies and brains might produce a body and brain for their offspring that do not form a good combination. The joint evolution of morphologies and controllers can lead to a mismatch between the body and brain.

This is a well-known problem that was originally noted in Eiben et al. [2013] and further expanded in Eiben and Hart [2020]. The solution that has been proposed is the introduction of learning within the evolutionary process. This results in the addition of a new phase in the robot's life cycle between "birth", when the robotic phenotype is constructed from its genotype, and "mature life" when the robot carries out its task and tries to reproduce. This new stage corresponds to the "infancy" of a robot when the newborn robot is optimizing its controller for its body. It is important to understand that a robot's fitness is not calculated until the end of this stage when its controller is fully optimized to its body morphology. This means that the robot is not eligible for reproduction until learning is not over. The introduction of learning is part of a three-stage system called The Triangle of Life (Eiben et al. [2013]) in which each robot goes through birth, infancy, and mature life. Some studies have already researched the workings of these systems and found that learning drastically reduces the costs needed for the joint optimization of morphologies and controllers by allowing the controllers to be optimized for their morphologies (Jie Luo [2022]).

The joint evolution of morphologies and controllers is a complex problem with many moving parts and the addition of learning makes the problem increasingly more difficult to study. Not many studies have investigated such systems, often focusing on the evolution of controllers in fixed morphologies. A more in-depth analysis of the effects of the factors driving the evolutionary process is missing. In this research, we focus on the effects of the reproduction mechanism on the robots' controllers.

The adaptive advantage of sexual reproduction is currently an unsolved problem in biology. Individuals that have survived to be in the reproductive age have already proven that their genome is effective in the current environment. Why should they risk mixing their successful genome with that of another individual? Moreover, a sexually reproducing organism only passes around the 50% of its genetic material to each offspring. Asexually reproducing organisms, on the other hand, have the possibility of passing the totality of their genome to their offspring.

Some theories regard sex as an efficient method of producing variations, enabling organisms to adapt to changing environments. Sex can generate genetic diversity through the combination of genetic material from two parents.

Sexual reproduction allows for the purging of harmful mutations and the mixing of beneficial mutations from different individuals. However, in nature sex is more costly than asexual reproduction which does not need to expend resources in choosing a mate. Indeed, asexual reproduction involves creating offspring that are more genetically similar to their parents. The level of genetic variation in asexual reproduction is typically lower than in sexual reproduction because there is no mixing of genetic material between two individuals. Mutations are typically limited to small-scale changes in the genotype, whereas recombination can lead to more big-scale rearrangements. Asexual reproduction has the advantage of keeping intact the combinations of genes that have proved to be successful in the environment that they were exposed to (Engelstädter [2008]. However, it can limit the ability of a population of individuals to adapt to changing conditions. Another thing to note is that, despite the great diversity of life on Earth natural reproduction mechanisms work exclusively with one or two parents (Eiben [2003]). The majority of the species reproduce asexually, showing the viability of asexual reproduction. However, sex is a dominant feature in most multicellular organisms suggesting that the fitness advantages of sexual reproduction must outweigh its costs.

In simulated evolution, abstract variants of sexual and asexual reproduction are implemented as search operators. Some evolutionary techniques, like evolutionary programming, have worked almost exclusively with mutation (asexual reproduction). Others, like genetic algorithms, evolution strategies, and genetic programming, use recombination and mutation (sexual reproduction). A variety of reproduction and mutation operators have been proposed in the literature, with some studies increasing the number of parents above 2. The answer to which operator yields the best results is still not clear and probably depends on the field of application and type of evolutionary system. In Evolutionary Robotics, the choice between the two reproduction methods will depend on the characteristics of the task to solve, including the environment. If the goal of the optimization is to optimize a particular trait or set of traits, asexual reproduction may be the better choice. On the contrary, if we aim to explore a wider range of solutions and adapt to changing conditions, sexual reproduction might be more advantageous.

To test these ideas, we set up a system in which modular robots can reproduce and create offspring inheriting the parents' morphologies and controllers. Thus, in our system both the morphologies and the controllers are evolvable. The controllers of the robots are also made learnable by implementing a differential evolution algorithm that optimizes them on their new body. We test our system on two tasks: panoramic rotation and point navigation. In the first one, the robots have to spin around as many degrees as possible. For the second task, some target points in the environment are provided to the robot that has to reach them. A fitness function for each task is designed as a performance measure to drive the evolutionary process. The selection of parents and survivors based on the fitness function drives evolution to produce well-suited bodies and brains. The body of every offspring is produced by recombination and mutation of the genotypes of its parents' bodies. For the inheritance of the brain from its parents we design two different mechanisms. The first one, dubbed sexual reproduction, generates the new brain by recombination and mutation starting from the parent's genotype. The second mechanism, asexual reproduction, produces the new brain by the sole mutation of the genotype of its best parent.

This setup allows performing experiments to yield scientific insights into how evolution and learning influence each other. Evolution over generations and learning within a lifetime have been long speculated to interact with each other in nontrivial ways. Some disfavored theories of evolution, like Lamarckian inheritance, hypothesize that the behaviors learned during a lifetime could be directly transmitted to an individual's offspring so that they would be available as instincts soon after birth (Weismann [1893]). Another conjecture, the Baldwin effect, posits that behaviors learned during the lifetime of early generations would gradually become instinctual (Baldwin et al. [2018]). This phenomenon would be explained by the high costs of learning producing a selection pressure that can speed up learning. For example, an animal that is unable to learn to walk early in life may be more likely to die. In this context, different reproduction methods might influence the ability of the offspring to inherit its parents' traits in different ways.

Our study aims to answer three main research questions:

Research Question 1 How do the costs and benefits of reproduction differ between asexual and sexual reproduction?

The answer to this question would indicate the differences that arise between two Evolutionary Robotics systems that utilize sexual and asexual reproduction. The results would suggest which of the two reproduction methods is preferable in terms of the costs of running the evolutionary process and the results obtained from it.

Research Question 2 How do the mechanisms of genetic variation and inheritance differ between asexual and sexual reproduction, and how do these differences shape the genetic diversity and evolution of populations?

Jie Luo [2022] and Miras et al. [2020a] investigate the effect that learning has on the morphologies produced in an Evolutionary Robotics system and find that, even if learning only modifies a robot's brain, its introduction drives evolution toward diverse morphologies and that, in a way, the brain shapes the body. The reproduction mechanism of the brain is another factor that could potentially drive the evolutionary search in different directions. Additionally, we look into the robots' ability to learn and how it changes for the effect of the reproduction mechanism. To quantify the learning potential of a robot we use the learning delta, the performance difference between the robot before and after the learning phase.

Research Question 3 Will reproduction affect the behavior of the robots?

To answer this question we will compare the behavior of the best robots produced by the evolutionary process while using sexual or asexual reproduction and investigate their differences.

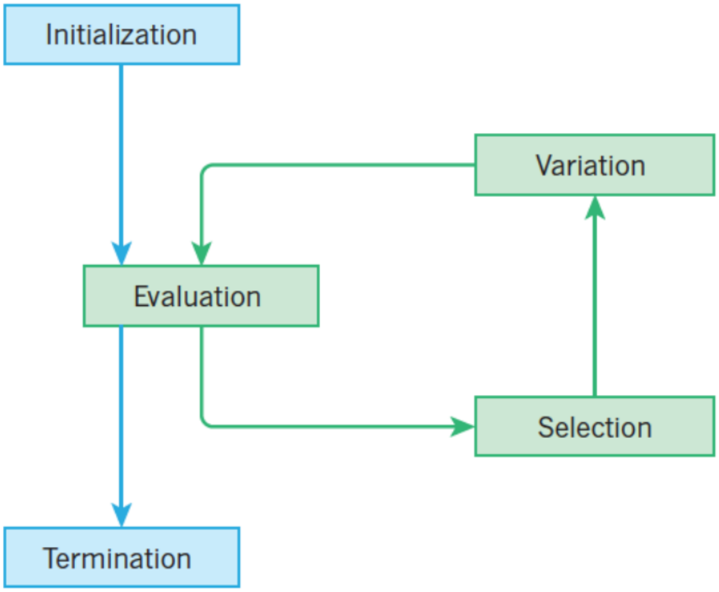

Figure 1: General schema of an Evolutionary Algorithm.

2 Related Works

2.1 Evolutionary Computing

The idea of mimicking biological evolution to solve problems in computing came with the advent of computers (Turing [2004]). Since the 1950s and 1960s, there have been several attempts to develop evolutionary-inspired algorithms that expanded into different branches: evolutionary programming, evolution strategies, genetic algorithms, and genetic programming. These branches merged in the 1990s under the name of Evolutionary Computing (EC). In the past years, EC has proven to be highly successful across a wide range of computational tasks including optimization, design, and modeling. From the perspective of the underlying substrate where this process takes place, EC represents a major transition of the evolutionary principles from 'wetware', the realm of biology, to 'software', the realm of computers (Eiben and Smith [2015]).

To express a problem using an evolutionary approach some elements have to be considered. Solutions to the problem are considered as individuals in a population and their quality is expressed using the notion of fitness. Analogous to natural evolution, evolutionary algorithms introduce a dichotomy between the genotype and the phenotype. At the higher level, phenotypes represent candidate solutions and have their fitness measured. At the lower level, genotypes are used to represent phenotypes in a form that can be manipulated to produce variations. To bridge the two levels, some kind of mapping is required. At each generation, a selection mechanism is used to select parents for the new offspring and to decide the composition of the next generation. To produce new individuals (offspring) from the selected parents, variation operators are employed at the genotypic level. Mutation operators randomly change some values from a single parent. Recombination operators, combine values from the genotype of two or more individuals. The entire process is controlled by an execution manager that initializes the first population and regulates the execution of the selection-variation cycles and the termination of the algorithm. It also manages the population size and other parameters affecting selection and variation. EAs are easily transferable from one application to another because only two components are problem dependent: the mapping between genotypes and phenotypes and the fitness function (Eiben and Smith [2015].

The Evolutionary Algorithm (EA) just described can be summarized as depicted in Figure 1. First, a population is initialized and its individuals are evaluated using a fitness function. The parents are then selected among the population using the results of the evaluation. The offspring are generated from the variation of the parents and a new population is created. The cycle continues until the termination criterion is met.

In the 1990s, two different research groups applied EAs to robot control systems (Cliff et al. [1993], Floreano and Mondada [1996], Harvey et al. [1997]). This marked a new transition from evolution-inspired software to hardware and the birth of a new field of study: Evolution of Things. In the year 2000, the publication of the book *Evolutionary* *robotics: The biology, intelligence, and technology of self-organizing machines* (Nolfi and Floreano [2000]) marked the beginning of Evolutionary Robotics as studied today.

2.1.1 Evolutionary Robotics

Evolution of Things is concerned with the application of evolutionary computation techniques to physical artifacts (Eiben [2014]). These artifacts can be passive, evolved by simulation, and assembled in a later stage, or they can be active, comprising sensors, actuators, and using a controller. Evolutionary Robotics (ER) is the field that studies active artifacts.

In designing a robot, many different subjects must be considered simultaneously: its morphology, sensory apparatus, motor system, control architecture, etc. (Siciliano et al. [2008]). One of the greatest challenges of robotics is that all of these systems interact and jointly determine the robot's behavior (Doncieux et al. [2015]). Engineering mostly follows the reductionist approach of considering all these aspects in isolation. 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. However, this design problem is far from trivial without decomposing it into smaller sub-problems. For this reason, drawing inspiration from nature can be helpful since living beings are good examples of systems endowed with embodied intelligence. The system responsible for their design, evolution, is therefore an attractive option for this alternative design methodology.

ER is a research area that applies EAs to design and optimize the body, the brain, or both for simulated and real autonomous robots (Nolfi and Floreano [2000], Doncieux et al. [2015]). It is an exciting area whose main rationale is that, as natural evolution has produced successful life forms for practically all possible environmental niches on Earth, it is plausible that artificial evolution can produce specialized robots for various environments, and tasks. This approach is not meant to replace the human design of robots for structured environments with known conditions. However, for complex, unstructured environments with unknown and possibly changing conditions, evolution offers great advantages (Lan et al. [2021a]).

Unlike classical engineering approaches, based on mathematics and physics, ER is based on a less understood biological mechanism. Experimental research on biological evolution is often slowed down by the fact that evolution requires many generations of big populations of individuals whose lives may last many decades. For this reason, most of the research focuses on organisms whose life cycle is short enough to allow laboratory experiments (Wiser et al. [2013]). ER offers an alternative synthetic approach where robots are the evolving entities that enable the experimental testing of hypotheses (Long [2012]). As John Maynard Smith, one of the fathers of modern theoretical biology, claims: "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 will have to look at artificial ones." (Maynard Smith [1992]). Considering embodied agents in realistic environments makes it possible to study complex interactions with other individuals and with the environment. ER has been already used to study some of the key issues in evolutionary biology, such as the evolution of cooperation, whether altruistic (Montanier and Bredeche [2011], Montanier and Bredeche [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 [2014a]), and collective swarming (Olson et al. [2013]).

2.1.2 Triangle of Life

The main foundation of this study lies in the Triangle of Life (ToL) framework. ToL is a generic model that describes the life cycle of a robot from generation (being conceived) to generation (conceiving new robots) (Eiben et al. [2013]). The main framework is depicted in Figure 2. The ToL framework is divided into three phases: birth, infancy, and mature life. The birth phase is the first stage of life and is described as the interval between the moment of activating a newly created genome and the moment when the robot organism encoded by this genome is completed. This is the period when morphogenesis is completed. Infancy is the stage in which the robot organism acquires the skills necessary for living in the given world. This is a fine-tuning phase where the robot learns to control its own body. It starts when the morphogenesis of the robot is completed and ends when it becomes capable of conceiving offspring. The final stage, maturity, starts when the organism becomes able of producing offspring and leads to a new Triangle for the newly conceived child.

2.2 Robot Design

A robot is a machine capable of carrying out a complex series of actions automatically. Its design comprises actuators that enable the movement of the robot, a controller to manage this movement, and optionally one or more sensors to receive input from the environment.

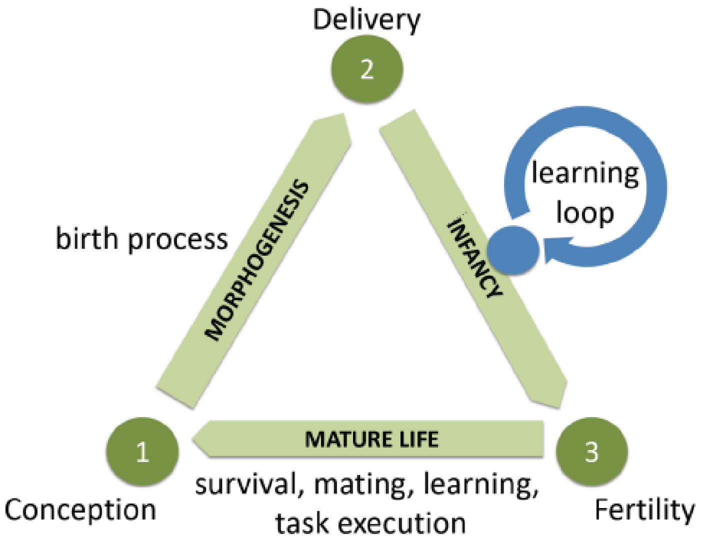

Figure 2: Framework for robot evolution: the Triangle of Life

2.2.1 Morphology Design

A variety of types of morphologies have been studied. Earlier work in ER focused on fixed morphologies such as salamanders (Ijspeert et al. [2007]), bipedal snake robots (Thakker et al. [2014], Matos and Santos [2014]), quadruped robots (Yosinski et al. [2011]), and a fish-like robot (Ayers and Crisman [1993], Webb [2002]).

Recently, new possibilities have been proposed to go beyond the traditional robots made of rigid arms, wheels, or legs. New types of robots exploit soft materials (Trivedi et al. [2008], Cheney et al. [2014a], Cheney et al. [2014b]) or swarms of simple robots (Bayindir and Şahin [2007]).

The possibility to evolve the morphology itself, however, sparked the research of modular robots that could be optimized in simulation and tested in real-world experiments (Zykov et al. [2007]). The first example of these robots is the 2000 GOLEM project (Lipson and Pollack [2000]) that reconstructed with bars and joints robots optimized in a simulation. In 2005, Zykov et al. [2005] used squared blocks made of two moving halves and a servo. Other projects employ swarms of robots able to combine themselves in a single body (Eiben et al. [2013]) or include the possibility of automatically assembling modules with the use of a robotic arm (Brodbeck et al. [2015]). The introduction of the RoboGen platform created a new approach to morphologically evolving robots with the use of 3D-printed modules from a simulated design and Arduino controllers (Auerbach et al. [2014a]). RoboGen is the main platform upon which many different studies have adapted their framework (Miras et al. [2018], Miras et al. [2020b], De Carlo et al. [2020a], Jelisavcic et al. [2017a]).

2.2.2 Controller Design

The controller is the brain of the robot, the software that decides what to do given the current situation according to programmed rules. Many different types of controllers can be used for robots from logic-based symbolic systems (Russell [2010]), to fuzzy logic (Saffiotti [1997]), behavior-based systems (Matarić [1998]), and boolean networks (Roli et al. [2011]). Evolutionary algorithms, thanks to their versatility, can be used with all of these systems to find the best parameters or the best architecture. However, in the context of ER, the complexity of the controller should not constrain evolution in any way. Using as little prior knowledge as possible makes it possible for ER to scale up to designs of unbounded complexity. For this and other reasons, artificial neural networks are currently the most used type of controller in ER. Feed-forward neural networks can approximate any continuous function to arbitrary accuracy given enough neurons (Cybenko [1989]). They can process various types of complex signals like images and sounds and have been used in robot control (Miller et al. [1995]). With recurrent connections, neural networks are also able to approximate any dynamical system (Funahashi and Nakamura [1993]). Additionally, since they are used in the field of neuroscience to model the brain, work in ER can build on research, for instance, on synaptic plasticity (Abbott and Nelson [2000]) and network theory (Bullmore and Sporns [2009]). Evolution can act on the parameters of the network, its architecture, or both.

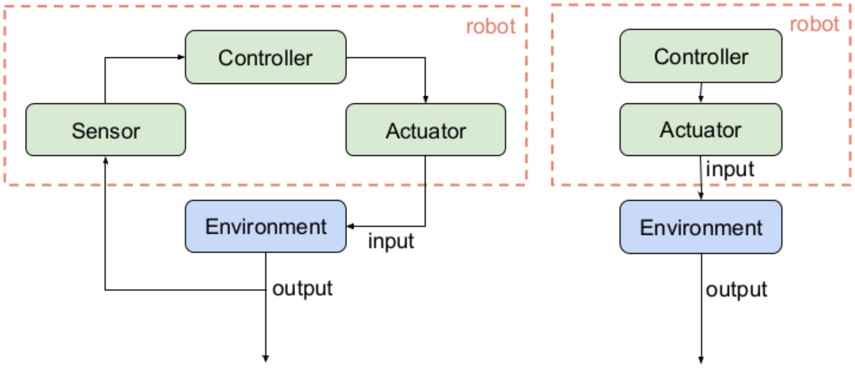

Figure 3: Closed-loop and open-loop systems

Open-Loop vs Closed-Loop Controllers can be divided into two groups: open-loop and closed-loop. Open-loop systems do not receive any feedback from the environment making their actions independent from the effect of previous actions. In a closed-loop system the opposite is true, establishing a feedback loop, as shown in Figure 3.

The advantage of using open-loop controllers stands in their simplicity and, for this reason, they are widely employed in ER. In these systems the parameters of the robot's actuators are changed by the controller, moving the corresponding wheel or joint. On the other hand, open-loop systems' lack of sensory feedback hinders their performance when dealing with unknown environments. In these environments, the sensory feedback used by closed-loop controllers might be very beneficial. This property of closed-loop controllers helps reduce the reality gap extending the applicability of the system beyond lab setup (Lan et al. [2021b], Lan et al. [2021a], Jie Luo [2022]).

Central Pattern Generators CPGs are biological neural circuits that produce rhythmic outputs in the absence of rhythmic input (Bucher et al. [2015]). This makes them able to activate a muscle or joint without the need for sensory input (Ijspeert [2008]). Standard CPGs are open-loop controllers that can be easily simulated in software and are, for this reason, widely popular in robotics. Another advantage of CPGs is that they can be used in distributed representations and are thus particularly useful when working with modular robots. For these reasons, much of the current work on ER systems employs these controllers (Lan et al. [2021a], Lan et al. [2021b], Miras et al. [2018], Miras et al. [2020b], Jie Luo [2022], De Carlo et al. [2020b]). Various studies also show the benefits of integrating sensory feedback in CPG-based controllers when performing locomotion in complex terrains (Fukuoka et al. [2003], Kimura et al. [1999], Kimura et al. [2007]).

2.3 Evolutionary Loop

The remaining component is the evolution itself and which parts of the system are going to be exposed to it.

2.3.1 Evolving Robots

ER is an intrinsically multi-disciplinary approach to robot design that consider the robot's morphology as a variable in the final behavior and therefore greatly expands the design space. For this reason, it opens up new ideas at the frontiers of different fields. While evolving controllers, a population of them goes through the loop of evaluation, selection, and variation. In this case, the topology and parameter of the robot's controller are the subjects of evolution.

A less common avenue of research is that of evolving morphologies. Since bodies, as their controllers, heavily influence the final behavior, evolving bodies is important to unlock the full potential of the evolutionary approach, especially for unknown and changing environments. Studies involving the evolution of robots' bodies include different approaches (Brodbeck et al. [2015], Sims [1994], Samuelsen et al. [2013], Veenstra et al. [2017]).

The joint evolution of controllers and morphologies is where the real potential of evolving machines can be unlocked (Lan et al. [2021a]). The first approaches to evolving virtual creatures were published in the 1990s (Sims [1994]) and later on with the GOLEM project in the year 2000 (Lipson and Pollack [2000]). More recently, studies have been performed using the RoboGen framework for joint evolution. Combinations of morphologies and controllers were used to optimize robots for a variety of tasks (De Carlo et al. [2020b], Miras et al. [2018], Miras et al. [2020b], Pagliuca and Nolfi [2021], Nygaard et al. [2017], Gupta et al. [2021]). Bongard [2011] studies the impact of changes in morphology on the behavior design process and finds that increasingly complex morphologies help design more robust behaviors.

The influence of the sensory apparatus can also be studied in ER. Hauert et al. [2009] designs efficient controllers that do not use previously required sensory information.

2.3.2 Behavior

Normally, in EC there is a three-step evaluation chain: genotype to phenotype to fitness. For robot evolution the chain is four-step: genotype to phenotype to behavior to fitness. In this four-step chain, the robot's morphology and controller form the phenotype. However, it could be argued that the phenotype should include the robot's final behavior too since the evaluation is ultimately performed on it. Furthermore, the behavior depends on many external factors, creating an unpredictable environment in which the robot is expected to perform (Eiben and Smith [2015]). In a robot, the behavior is influenced by the morphology and controller but in an ER system, the other way around is also true: the desired behavior is used to guide evolution and thus shape the robot controller and morphology (De Carlo et al. [2021]).

A variety of tasks has been used to evaluate robot performance in ER, either for the sake of studying artificial evolution or for the importance of the task in the robot's life cycle. Among the most studied tasks are Gait Learning (Jelisavcic et al. [2017b], Clune et al. [2009], Christensen et al. [2013]), Directed Locomotion (Lan et al. [2021b], Lan et al. [2018], Jie Luo [2022]) and Rotation (De Carlo et al. [2021], De Carlo et al. [2020b]). De Carlo et al. [2021] use more than one task to compute the fitness of the robots and find that multi-objective evolution explores more the space of morphological traits.

Kargar et al. [2021] introduce a set of behavioral descriptors for robot gaits. They also demonstrate how including them in the fitness function improves the quality of the gaits compared to the unsteady ones obtained with speed as the sole component of the selection pressure.

The majority of the research on ER, influenced by the view of evolution as an optimization algorithm, relies on fitness functions to drive the search process. Typically, the fitness value is expected to increase as the search process is led in the desired behavior. Another line of work has questioned this assumption and shown that this performance criterion can be misleading. Lehman and Stanley [2011] showed that using the novelty of a solution to drive the search process instead of its performance can lead to better results. This is a finding that emerged in multiple contexts (Lehman et al. [2008], Risi et al. [2009], Gomes et al. [2012], Gomes et al. [2013], Liapis et al. [2013]).

2.3.3 Environment

Another factor influencing the evolutionary process is the interaction of the robot phenotypes with the environment. Arguably, the fitness of a robot is determined by three factors, the body, the brain, and the environment Miras et al. [2020b]. In natural evolution, the environment largely determines the evolved life forms (Darwin [2004], Sapolsky [2017]). Miras and Eiben [2019] introduce some morphological descriptors and Miras et al. [2020b] study the effect of different environments on the evolution of robots and find morphological and behavioral differences in robots from different environments. Auerbach and Bongard [2014b] find that more complex environments tend to lead to the evolution of more complex body plans than those that evolve in simpler environments if increased complexity incurs a cost. Gupta et al. [2021] find a strong impact of the environment on evolved morphologies. Moreover, they train the controllers of robots with evolved morphologies on a suite of 8 new tasks and observe that morphologies evolved in more complex environments outperform the others. These results suggest that the morphologies evolved in more complex environments are more intelligent in the sense that they facilitate learning many new tasks both better and faster. A variety of environments were proposed in the literature, including flat terrains, sloped terrains (Miras et al. [2020b]), terrains with obstacles and randomly generated variable terrains(Gupta et al. [2021]), and different levels of gravity (Methenitis et al. [2015]).

Other methods have explored varying environmental conditions during the evolutionary process. This is necessary to select robots that are robust to environmental variations and can adapt to varying conditions on the fly (Carvalho and Nolfi [2022], Branke [2012], Risi and Stanley [2012], Cully et al. [2015]). Exposing robots to variable environmental conditions is also helpful to cross the reality gap (Jakobi et al. [1995], Koos et al. [2012]). Selecting challenging conditions for the evolving agents can be achieved by dividing the evolutionary process into successive phases of increasing complexity (Harvey et al. [1994], Gomez and Miikkulainen [1997]) or by competitive co-evolution, the evolution of agents that compete with other evolving agents (Nolfi and Floreano [1998], Rosin and Belew [1997]). Milano and Nolfi [2021] proposes a third method that enables evolutionary algorithms to select the environmental conditions that facilitate the evolution of effective solutions.

2.3.4 Genetic Encoding

There are two main classes of genetic encodings, direct and indirect encodings; the latter are also known as generative encodings (Miras [2021]). Direct encodings represent each phenotype component independently in the genotype, generative encodings allow the reuse of genotype portions that code for similar or identical phenotypes components. This property is incredibly important as it allows the reuse of solutions to parts of the whole optimization problem in different contexts. For instance, while only 30000 genes code all traits of the human body (Deloukas et al. [1998]), our brain by itself has trillions of neurons (Dellaert [1995]). Thanks to its reuse capacity, evolution can for example discover a particular limb only once and then repeat it multiple times in the body of a creature. Research on modularity and its relevance for evo-devo support this hypothesis(Bolker [2000], Kuratani [2009]).

Various studies compare the performance of robotic systems with different encodings (Collins et al. [2019], Komosiński and Rotaru-Varga [2001]). The most commonly used generative encodings to evolve robots are Compositional Pattern Producing Networks (CPPNs) (Stanley [2007a]) and L-Systems (Lindenmayer [1968]). It has been demonstrated that CPPNs can evolve complex patterns for a variety of tasks (Clune et al. [2009], Haasdijk et al. [2010], Clune and Lipson [2011], Auerbach and Bongard [2014a], Goff et al. [2021]). Miras [2021] report that there is a tendency for these two encodings to sample robots with certain traits more often than others. Robots produced by the CPPN often have a "spider" shape, and are slower but present a more stable gait. By contrast, robots produced by the L-System often have a "snake" shape, are much faster, and move with a more unstable gait.

2.3.5 Reproduction Mechanism

One of the most enduring puzzles in evolutionary biology is why sexual reproduction is so widespread. Individuals that have survived to reproductive age have genomes that have already proved themselves in the current environment, so why should they risk mixing their genes with those of another individual (Otto and Lenormand [2002])? Moreover, sexual reproduction is associated with several costs. First of all, in many species, it takes time and energy to secure a mate. Sexual reproduction is indeed usually slower and, while mating, individuals are not able to gather resources or avoid predators. Second, sexually produced offspring is more costly to produce. Sexual reproduction produces half the individuals per capita of asexual reproduction since its unit of reproduction is the couple. Sex should be an evolutionary dead-end, observed only rarely in nature. Contrary to these expectations, the vast majority of species reproduce sexually. This is a paradox that remains unexplained in the field of biology.

Evolutionary Computing has implemented both sexual and asexual operators as search operators. Several studies have investigated the advantages and disadvantages of mutation over crossover (Mahfoud [1994], Fogel and Atmar [1990], Fogel and Stayton [1994], Hordijk and Manderick [1995], Eshelman [1991], Spears [1993]). The problem of which mutation or crossover operators is preferable in certain circumstances is still open. Lim et al. [2017] reports that a balance between the two operators is the best choice to speed up the search process. Many multiparent reproduction mechanisms have also been introduced over the years (Mühlenbein [2005], Bersini [1992], Furuya [1993], Eiben et al. [1994]). A comprehensive overview of these operators is presented in Eiben [2003].

Evolutionary Robotics studies have used the two methods interchangeably. Sexual reproduction is the most commonly used mechanism for reproduction in the field and is generally applied to both morphologies and controllers (De Carlo et al. [2020a], Miras et al. [2018], Miras et al. [2020b] Jie Luo [2022], Jelisavcic et al. [2019]). Very few studies have used asexual reproduction in this context. Gupta et al. [2021] evolve the morphologies of modular robots with asexual reproduction but its controllers are randomly initialized at birth. To the limits of our knowledge, no studies have compared the effects of the two reproduction mechanisms.

2.3.6 Learning

In the Triangle of Life, infancy is the phase where the newly born robots learn to use their body. In the case of the joint evolution of the morphology and controller of a robot, this phase is particularly important, since the controller inherited by the parents might not be suited to the newly generated body. This potential mismatch has been originally noted in Eiben et al. [2013] and recently revisited in Eiben and Hart [2020]. A commonly proposed solution is the addition of learning before the fitness evaluation. Nygaard et al. [2017] demonstrated improvements in their ER system introducing a phase of sole controller evolution after the joint evolution of controller and morphology. Gupta et al. [2021] use evolution for the robot bodies and reinforcement learning to optimize their controllers. Goff et al. [2021] use an external controller archive and perform learning on these controllers. Finally, Miras et al. [2020a] study the effects of learning in morphologically evolvable modular robots and find that learning boosts the speed of the process, greatly reducing the number of generations required to reach a certain performance level. Even if the number of evaluations is much higher with learning, Miras et al. [2020a] argue that a learning trial is much cheaper than an evolutionary step (making a new robot), emphasizing the advantage of its introduction. Additionally, they provide evidence that lifetime learning influences the morphology of evolved robots.

A variety of learning strategies were employed to learn the brains of robots with fixed bodies (Schembri et al. [2007], Ruud et al. [2017], Luck et al. [2019], Jelisavcic et al. [2019], Lan et al. [2020], Schaff et al. [2019], Le Goff et al. [2020], van Diggelen et al. [2021], Nordmoen et al. [2021]). van Diggelen et al. [2021] perform a comparison between three learning algorithms on robots with fixed modular robots. The study finds RevDE (Tomczak et al. [2020]) and Bayesian Optimization (Lan et al. [2020]) to deliver the same solution quality with fewer evaluations than the evolutionary strategy algorithm NIPES (Le Goff et al. [2020]). Furthermore, they show that NIPES is less robust than the other two algorithms, with its performance varying highly between robot morphologies. Reinforcement Learning (RL) techniques have also been successfully used in the field of robotic control (Luck et al. [2019], Feng et al. [2022], Smith et al. [2022]). Gupta et al. [2021] use RL to learn the controller policies of modular evolving robots. They find that morphologies optimized in more complex environments are better and faster at learning many novel tasks and provide evidence of a morphological Baldwin effect (Baldwin et al. [2018], Mayley [1996]).

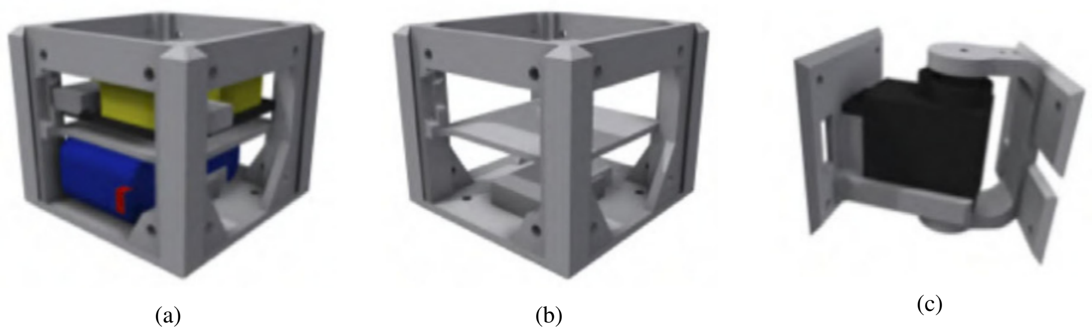

Figure 4: The modules of the RoboGen framework composing the body phenotypes. Core modules (a) contain the controller and a battery. Brick modules (b) compose the structure of the robot. Active hinges (c) make the robot able to move.

3 Methods

To investigate the differences between the sexual and asexual reproduction of a robot's brain we consider the case of the joint evolution of morphology and control of modular robots with the addition of controller learning.

3.1 Robot Morphology (Body)

3.1.1 Body Phenotype

The robots used for this study rely on the modular framework of the RoboGen platform, which was explicitly designed for studies on Evolutionary Robotics (Auerbach et al. [2014b]). RoboGen allows for real-world experiments by defining robots that can be built by assembling simple 3D-printed modules. We use a subset of RoboGen's component: all the robots are made of a core module and a combination of bricks and active hinges. The core module, a cuboid of dimensions $0.089 \times 0.089 \times 0.06$ meters shown in Figure 4a holds the circuits used for the controller of the machine and a battery for power and has four sockets for the connection of other modules on its lateral sides. Bricks, shown in Figure 4b, are cubic modules of dimensions $0.063 \times 0.063 \times 0.06$ meters that have four sockets on their lateral sides, which makes them able to connect to the other modules. Active Hinges, represented in Figure 4c add degrees of freedom (DoF) to the robot, and function as the actuators of the system, driven by the controller. The modules form a tree structure with the core module as the root. Child modules can be rotated by $90°$ making 3D morphologies possible.

3.1.2 Body Genotype

The bodies of the robots are encoded using a Compositional Pattern Producing Network (CPPN), introduced in (Stanley [2007b]). Like neural networks, a CPPN is a network of mathematical functions with weighted connections. Unlike neural networks, a CPPN can contain a variety of activation functions including *sine*, *cosine*, *Gaussian*, and *sigmoid*. The CPPNs used have four inputs and five outputs. The inputs are the x, y, and z coordinates of a component of the morphology and its distance to the core component in the tree structure. The output of the network is the probabilities of the component being a brick, a module, or an empty space and the probabilities to be rotated by $0°$ and $90°$.

The generation of the body's phenotype from the genotype starts by generating the core component at the origin. Then, moving outwards from the core component, a breadth-first search is performed on each socket, querying the CPPN network. The module type and rotation with the highest probability are chosen. The process stops when all the sockets have been evaluated or if ten modules have been created.

The coordinates of each module are a triple of integers of the form (p_x, p_y, p_z). The coordinates of the core component are $(0, 0, 0)$ and a module attached to its front socket will have coordinates $(1, 0, 0)$.

The recombination and mutation operators used for the body genotypes are the same as defined by MultiNEAT (https://github.com/MultiNEAT/MultiNEAT).

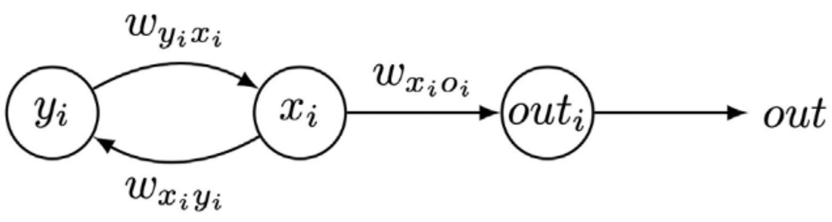

Figure 5: The structure of the CPG associated with the i_{th} joint. $w_{x_i y_i}$, $w_{y_i x_i}$, and $w_{x_i o_i}$ are the weights of the connections between the neurons and out is the activation value of the out_i neuron that controls the servo in a joint

3.2 Robot Controller (Brain)

3.2.1 Brain Phenotype

We use Central Pattern Generators (CPGs)-based controllers to drive the modular robots. Each joint of the robot has an associated CPG that is defined by three neurons: a x_i-neuron, a y_i-neuron, and an out_i-neuron. The recursive connection of the three neurons is shown in Figure 5. The change of the x_i and y_i neurons' states with respect to time is obtained by multiplying the activation value of the opposite neuron with the corresponding weight. To reduce the search space we set $w_{x_i y_i}$ to be equal to $-w_{y_i x_i}$ and call their absolute value w_i. The resulting activations of neurons x_i and y_i are periodic and bounded. The initial state of all x and y neurons is set to $\frac{\sqrt{2}}{2}$ because this leads to a sine wave with amplitude 1, which matches the limited rotating angle of the joints.

$$\begin{aligned} \dot{x}_i &= w_i y_i \\ \dot{y}_i &= -w_i x_i \end{aligned} \tag{1}$$

To enable more complex output patterns, connections between CPGs of neighboring joints are implemented. Two joints are said to be neighbors if their distance in the morphology tree is less than or equal to two. A robot and its network of CPGs are shown in Figure 6. Consider the i_{th} joint, and $\mathcal{N}_i$ the set of indices of the joints neighboring it, w_{ij} the weight of the connection between x_i and x_j. Again, w_{ij} is set to be $-w_{ji}$. The extended system of differential equations becomes:

$$\begin{aligned} \dot{x}_i &= w_i y_i + \sum_{j \in \mathcal{N}_i} w_{ji} x_j \\ \dot{y}_i &= -w_i x_i \end{aligned} \tag{2}$$

Because of this addition, x neurons are no longer bounded between $[-1, 1]$. For this reason, we use the hyperbolic tangent function (*tanh*) as the activation function of out_i-neurons.

$$out_{(i,t)}(x_{(i,t)}) = \frac{2}{1 + e^{-2x_{(i,t)}}} - 1 \tag{3}$$

3.2.2 Brain Genotype

For our brain genotype, we use a direct encoding of the CPG weights as an array of values. We have seen how every modular robot can be represented as a 3D grid in which the core module occupies the central position and each module's position is given by a triple of coordinates. When building the controller from our genotype, we use the coordinates of the joints in the grid to locate the corresponding CPG weight. To reduce the size of our genotype, instead of the 3D grid, we use a simplified 2D grid in which the third dimension is removed. For this reason, some joints might end up with the same coordinates and will be dealt with accordingly.

Since our robots have a maximum of 10 modules, every robot configuration we will be able to generate can be represented in a grid of 21×21. Each joint in a robot can occupy any position of the grid except the center. For this reason, the possible positions of a joint in our morphologies are exactly $(21 \cdot 21) - 1 = 440$. We can represent all the internal weights of every possible CPG in our morphologies as a 440-long array. When building the phenotype from this array, we can simply retrieve the corresponding weight starting from a joint's coordinates in the body grid.

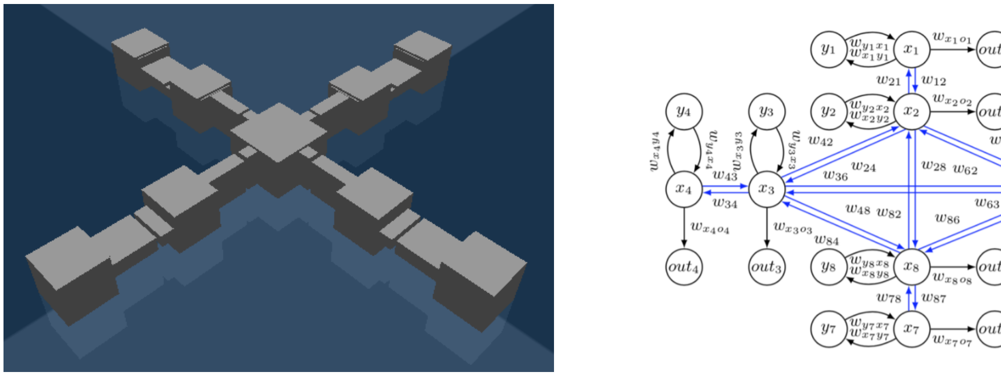

Figure 6: A robot and its corresponding network of Central Pattern Generators.

To represent the external connections between CPGs, we need to consider all the possible neighbors a joint can have. In the 2-dimensional grid, the number of cells in a distance-2 neighborhood for each position is represented by the Delannoy number[1] $D(2, 2) = 13$, including the central element. Each one of the neighbors can be identified using the relative position from the joint taken in consideration. Since our robots can assume a 3D position, we need to consider an additional connection for modules with the same 2D coordinates.

To conclude, for each of the 440 possible joints in the body grid, we need to store 1 internal weight for its CPG, 12 weights for external connections, and 1 weight for connections with CPGs at the same coordinate for a total of 14 weights. The genotype used to represent the robots' brains is an array of size 440×14. An example of brain generation from its genotype is shown in Figure 7.

It is important to notice that not all the elements of the genotype matrix are going to be used by each robot. This means that their brain's genotype can carry additional information that could be exploited by their children with different morphologies.

The recombination operator for the brain genotype is implemented as a uniform crossover where each gene is chosen from either parent with equal probability. The new genotype is generated by essentially flipping a coin for each element of the parents' genotype to decide whether or not it will be included in the offspring's genotype. In the uniform crossover operator, each gene is treated separately. The mutation operator applies a Gaussian mutation to each element of the genotype by adding a value, with a probability of 0.8, sampled from a Gaussian distribution with 0 mean and 0.5 standard deviation.

3.3 Evolution

The evolutionary process to which the robots are subjected follows the following steps. A population of N individuals is initialized. Each undergoes a learning process for its controller. The final fitness of the robot is the fitness obtained using the best controller found with during the learning process. The fittest individuals are selected as parents and used to generate offspring using sexual or asexual reproduction. Individuals from the original population and the offspring are then selected to form the next generation. This is the end of an iteration of the evolutionary loop. The pseudo-code of the evolutionary loop with sexual and asexual reproduction is provided in Algorithm 1 and 2. The separate steps will be further explained below.

3.3.1 Initialization

The initial population is composed of N individuals. The CPPN representing the genotype of the morphologies is initialized empty and goes through 10 mutations. The genotype of the controllers is randomly initialized by sampling from a Standard Normal distribution. The phenotype of each robot's morphology and controller is generated from the corresponding genotype as explained above.

3.3.2 Controller Learning

The algorithm used to learn the controllers is RevDE (Tomczak et al. [2020]), which was shown to work well with our modular robots (Van Diggelen et al. [2021]). RevDE is a differential evolution algorithm that in our system is used to

---

[1] (`https://en.wikipedia.org/wiki/Delannoy_number`)

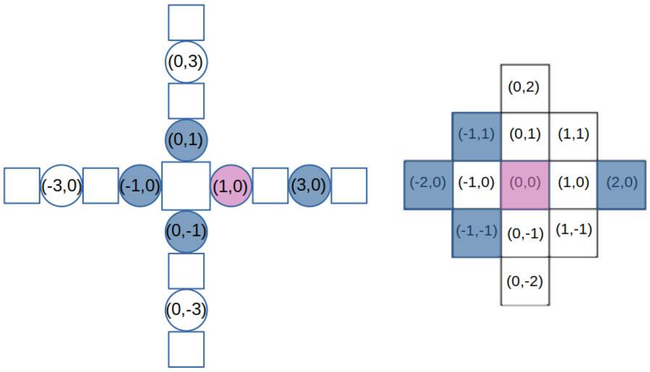

Figure 7: Generation of a brain from the corresponding genotype. The left image shows the schema of a spider robot with the coordinates of its joints in the 2D body grid. The left image is the distance 2 neighbor of the joint at (1,0). The coordinates reported in the neighborhood are relative to this joint. In the picture are highlighted in pink the joints we want to retrieve the CPG weights for and in blue its 2-distance neighbors.

Algorithm 1 Evolutionary Loop with Sexual Reproduction

```
INITIALIZE robot population (genotypes + phenotypes with body and brain)
EVALUATE each robot with learned brain (delivers a fitness value)
while not STOP-EVOLUTION do
    SELECT parents (based on fitness)
    RECOMBINE+MUTATE parents' bodies (delivers a new body genotype)
    RECOMBINE+MUTATE parents' brains (delivers a new brain genotype)
    CREATE offspring robot body (delivers a new body phenotype)
    CREATE offspring robot brain (delivers a new brain phenotype)
    INITIALIZE brain for the learning process (in the new body)
    while not STOP-LEARNING do
        ASSESS offspring (delivers a performance value)
        GENERATE new brain for offspring
    EVALUATE offspring with learned brain (delivers a fitness value)
    SELECT survivors / UPDATE population
```

Algorithm 2 Evolutionary Loop with Asexual Reproduction

```
INITIALIZE robot population (genotypes + phenotypes with body and brain)
EVALUATE each robot with learned brain (delivers a fitness value)
while not STOP-EVOLUTION do
    SELECT parents (based on fitness)
    RECOMBINE+MUTATE parents' bodies (delivers a new body genotype)
    MUTATE best parent's brain (delivers a new brain genotype)
    CREATE offspring robot body (delivers a new body phenotype)
    CREATE offspring robot brain (delivers a new brain phenotype)
    INITIALIZE brain for the learning process (in the new body)
    while not STOP-LEARNING do
        ASSESS offspring (delivers a performance value)
        GENERATE new brain for offspring
    EVALUATE offspring with learned brain (delivers a fitness value)
    SELECT survivors / UPDATE population
```

optimize the weights of the CPGs of our modular robots. The initial population of X weight vectors is created by using the inherited brain of the given robot. The inherited weight vector is altered by adding Gaussian noise to create the remaining $X - 1$ mutant vectors.

The algorithm works as follows:

1. Initialize a population of X samples (n-dimensional vectors)
2. Assess the performance of the X samples
3. Apply the reversible differential mutation operator and the uniform crossover operator to obtain N new samples
4. Assess the performance of the new samples
5. Select the best X samples
6. Repeat from step (2)
7. Terminate when the maximum number of iterations is reached

For the reversible differential mutation operator, three new candidates are generated by randomly picking a triplet from the population $(x_i, x_j, x_k) \in X$, then all three individuals are perturbed by adding a scaled difference in the following manner:

$$\begin{aligned} y_1 &= x_i + F(x_j - x_k) \\ y_2 &= x_j + F(x_k - y_1) \\ y_3 &= x_k + F(y_1 - y_2) \end{aligned} \tag{4}$$

where $F \in R_+$ is the scaling factor.

We can express Eq.4 as a linear transformation using matrix notation as follows:

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & F & -F \\ -F & 1 - F^2 & F + F^2 \\ F + F^2 & -F + F^2 + F^3 & 1 - 2F^2 - F^3 \end{bmatrix}}_{=R} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \tag{5}$$

To obtain the matrix R we need to plug y_1 into the second and third equations and y_2 into the last equation in Eq.4. As a result of this operation, we obtain $N = 3X$ new candidate solutions. The linear transformation R is reversible.

For the uniform crossover operation, Storn [1997] proposed to sample a binary mask $m \in \{0, 1\}^D$ according to the Bernoulli distribution with probability p shared across D dimensions, and calculate the final candidate according to the following formula:

$$v = m \odot y_n + (1 - m) \odot x_n \tag{6}$$

Following Pedersen [2010], to obtain stable exploration behavior the crossover probability p is fixed to a value of 0.9 and the scaling factor F is fixed to a value of 0.5.

3.3.3 Evaluation

We evaluate the robots on two different tasks: Panoramic Rotation and Point Navigation. Both tasks are carried out on a flat terrain. For each task, we must define a fitness function to serve as the guiding metric of our optimization problem.

Panoramic Rotation The panoramic rotation task consists in optimizing the robots to rotate around their vertical axis.

To solve this task, we can collect from the simulator the orientation of the robot sampled at 5 Hz during the evaluation. Since robot orientations are represented as quaternions in our simulation, we convert them to the equivalent triplet of vectors v_i, v_j and, v_k[2], represented in Figure 8. The conversion formulas are:

$$\begin{aligned} v_i &= [1 - 2(q_2^2 + q_3^2), 2(q_1 \cdot q_2 - q_3 \cdot q_0), 2(q_1 \cdot q_3 + q_2 \cdot q_0)] \\ v_j &= [2(q_1 \cdot q_2 + q_3 \cdot q_0, 1 - 2(q_1^2 + q_3^2, 2(q_2 \cdot q_3 - q_1 \cdot q_0)] \\ v_k &= [2(q_1 \cdot q_3 - q_2 \cdot q_0), 2(q_2 \cdot q_3 + q_1 \cdot q_0), 1 - 2(q_1^2 + q_2^2)] \end{aligned} \tag{7}$$

---

[2] (`https://en.wikipedia.org/wiki/Conversion_between_quaternions_and_Euler_angles`)

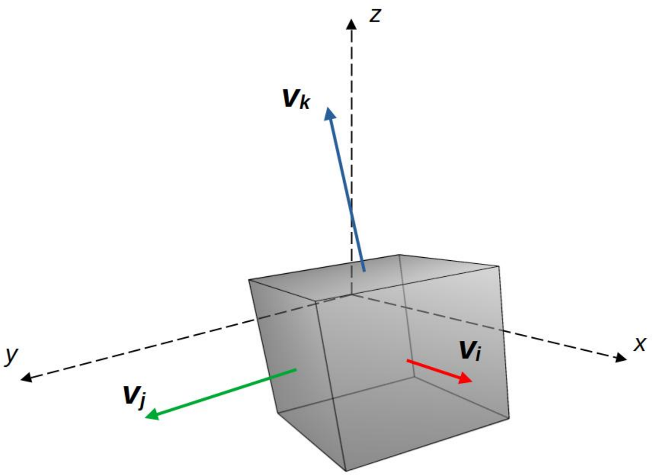

Figure 8: The orientation vectors obtained from the simulation.

where $q = (q_0, q_1, q_2, q_3)$ is the quaternion representing the orientation. v_i, v_j and, v_k are the vectors obtained by applying the rotation described by the quaternion to the base vectors (i, j, k).

The robots, due to their modularity, can assume a wide range of shapes and orientations. For each one of them, we need to identify the orientation vector whose rotation is going to be measured. At the start of the simulation, the orientation vector with the lowest x-axis component is identified. The rotation of this vector will be computed to measure the total rotation of the robot.

The angle between two vectors can be computed using their determinant and dot product. Indeed, the dot product of two vectors a, b is proportional to the angle between them and their determinant is proportional to the sine. We can compute the angle as:

$$\theta_i = atan2(det, dot) \tag{8}$$

where $det = a_x \cdot b_y - b_x \cdot a_y$ and $dot = a_x \cdot a_y + b_x \cdot b_y$.

Given the chosen vector at two consecutive timestamps T_{i-1}, T_i, the rotation between them can be computed using the method just described.

The final fitness function is the total rotation of the robot computed as the sum of the rotation of the orientation vector at each consecutive timestep.

$$F = \sum_{n=1}^{T} \theta_i \tag{9}$$

where T is the total number of timesteps in the evaluation.

For our experiments, we assign a positive sign to the counter-clockwise rotations and a negative one to the clockwise rotations.

Point Navigation The idea behind this task is for the robots to reach a sequence of N target points $P_1, ..., P_N$ in the given order. The points are defined as coordinates in a reference system where the origin corresponds to the starting point of the robot P_0. A target point is considered reached if the robot gets within 0.1 meters from it.

The fitness function for this task is designed:

- to maximize the number of targets reached
- to minimize the path followed by the robot to reach the targets

The data collected from the simulator is the following:

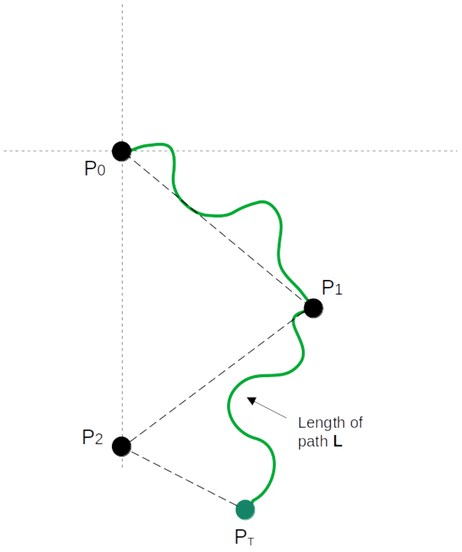

Figure 9: Illustration of the fitness function for the Point Navigation task. P_0 is the robot starting point and P_1 and P_2 are the target points. The green path is the one followed by the robot.

- The coordinates of the core component of the robot at the start of the simulation are approximately $O(0, 0)$;
- The coordinates of the robot, sampled during the simulation at 5 Hz, allowing us to plot and approximate the length of the followed path;
- The coordinates of the robot at the end of the simulation $P_T(x_T, y_T)$;
- The coordinates of the target points $P_1(x_1, y_1), ..., P_n(x_n, y_n)$.

Being k the number of target points reached by the robot at the end of the evaluation, and L the path followed, the fitness function is the following.

$$F = \sum_{i=1}^{k} dist(P_i, P_{i-1}) + (dist(P_{k+1}, P_k) - dist(P_T, P_{k+1})) - \omega \cdot L \tag{10}$$

The first term of the function is a sum of the distances between the target points the robot has reached. The second term is necessary when the robot has not reached all the targets and is the distance traveled toward the next one. The last term is used to penalize longer paths. ω is a constant scalar that is set to 0.1 in the experiments. An illustration of the fitness function with an example is shown in Figure 9.

Steering Policy To solve the point navigation task, the modular robots need to steer toward the target points. To make the robots steer, sensory feedback from the environment is combined with the activation values of the robot's CPGs. In this study, we use the coordinates provided by the simulation to mimic the use of a GPS sensor in real robots. We feed the robot the angle between its current position and the target position at time t, θ_t, where $\theta_t > 0$ means that the target is on the right and $\theta < 0$ means the target is on the left. When robots, need to turn right, joints on the right are slowed down, and vice versa. For modular robots, we need to scale the activation signal of the *out*-neuron of the corresponding CPG. A slow-down factor is introduced:

$$g(\theta_t) = \left(\frac{\pi - |\theta|}{\pi}\right)^n \tag{11}$$

The parameter n determines how strongly the joints are slowed down. In our experiments, we set $n = 7$ based on manual tuning. Joints on the left side of the robot are controlled using the following formula:

$$signal = \begin{cases} g(\theta_t) \cdot out & \text{if } \theta < 0 \\ out & \text{if } \theta \geq 0 \end{cases} \tag{12}$$

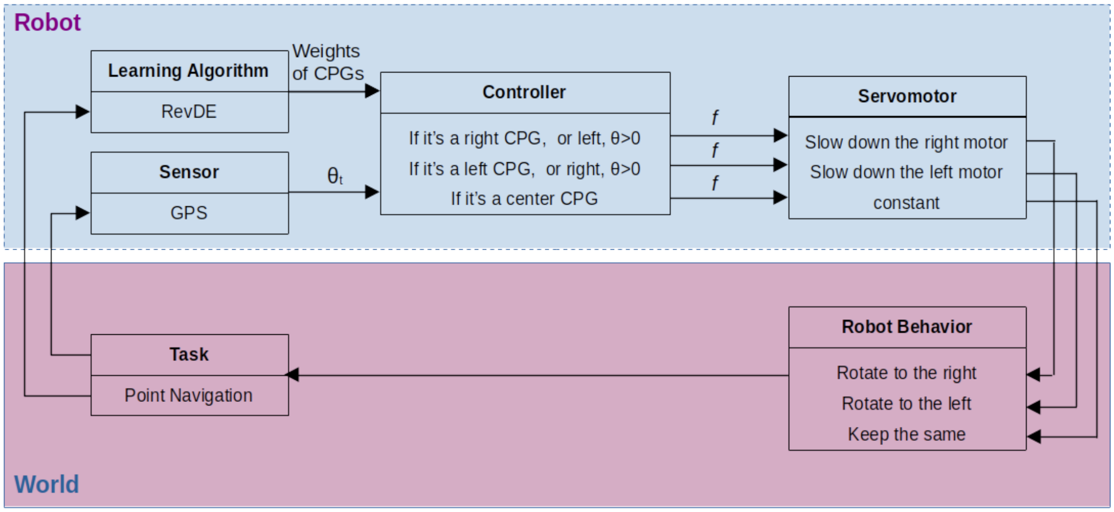

Figure 10: The overall architecture depicting how is steering applied to the joints. The error angle is computed using the robot's coordinates and the target point.

Analogously, for joints on the right:

$$signal = \begin{cases} out & \text{if } \theta < 0 \\ g(\theta_t) \cdot out & \text{if } \theta \geq 0 \end{cases} \tag{13}$$

where out is defined in Equation 3 The overall steering schema is shown in Figure 10

This simple policy will not lead to the correct steering behavior for every robot. The idea is that the emerging robots will be the ones able to use it successfully.

3.3.4 Reproduction

Parents are selected from the current generation using binary tournaments with replacement. We perform two tournaments in which two random potential parents each are selected. In each tournament the potential parents are compared, the one with the highest fitness wins the tournaments and becomes a parent.

The body of every new offspring is created through recombination and mutation of the genotypes of its parents. For the generation of the brain, instead, we use two different strategies which we refer to as asexual and sexual reproduction. The first strategy is called asexual because the genotype of the offspring is generated from only one parent. The genotype of the brain of the best performing parent is mutated before being inherited by its offspring. For sexual reproduction, instead, the child's brain is created through the recombination and mutation of its parents' brain genotypes.

3.3.5 Survivor Selection

For the selection of survivors, we use a variant of the $(\mu + \lambda)$ mechanism. Each generation is made up of μ individuals. The next generation is formed by the λ offspring and the best $\mu - \lambda$ parents.

4 Experimental Setup

The stochastic nature of evolutionary algorithms requires multiple runs under the same conditions and a sound statistical analysis (Bartz-Beielstein and Preuss [2007]). We perform 10 runs for each evaluation task and reproduction mechanism, for a total of 40 experiments. Each experiment consists of 30 generations with a population size of 50 individuals and 25 offspring. This means that a total of $50 + (25 \cdot (30 - 1)) = 775$ morphologies are generated, go through the learning loop and are evaluated. The algorithm for learning uses a population of 10 controllers for 10 generations, for a total of $(10 + 30 \cdot (10 - 1)) = 280$ performance assessments.

The fitness measures used to guide the evolutionary process are the same as the performance measure used in the learning loop. For this reason, we use the same test process for both. The tests for the task of point navigation use 40 seconds of evaluation time with two fixed target points at the coordinates of $(1, -1)$ and $(0, -2)$. The experiments for panoramic rotation, instead, use an evaluation time of 30 seconds.

The experiments are run in Revolve2 [3], a custom-built opens-source library for the optimization of modular robots (Hupkes et al. [2018]). All the robot evaluations are performed using the Mujoco simulation software (Todorov et al. [2012]). The code to run the experiments can be found at the following link: https://github.com/CarloLonghi/evolve-and-learn. All the experiments are run on a Linux computer with a 64 cores, 3GHz CPU, and 64GB of RAM, where they each take approximately 16 hours to finish, totaling 640 hours of computing time.

4.1 Performance Measures

The compare the two reproduction mechanisms we measure the efficacy of the optimization process by considering the trend of the mean and maximum fitness value averaged over the 10 different runs. We also use a special measure, the learning delta, which is the difference between the fitness of a robot before and after learning. The learning delta represents the learning potential of a robot morphology. Given a learning method and a learning budget (number of learning trials) which are the same for all generations, the learning delta shows how the morphology of a robot facilitates learning the parameters of the brain.

4.2 Morphological Descriptors

To study the evolution of the morphologies we use a set of quantitative descriptors.

Absolute Size The total number of modules in a robot body, comprising of bricks, joints, and core.

Branching The ratio between the total number of modules with an attachment on all four possible lateral sides and the maximum number of modules that could have had an attachment on all sides if they were attached differently.

Number of Bricks The number of bricks in the morphology.

Number of Hinges The number of active hinges in the morphology.

Relative Number of Limbs The number of extremities of a morphology relative to a practical limit. It is defined as:

$$L = \begin{cases} \frac{l}{l_{max}} & \text{if } l_{max} > 0 \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

with

$$l_{max} = \begin{cases} \frac{2 \cdot (m-6)}{3} + (m-6) \pmod 3 + 4 & \text{if } m \geq 6 \\ m - 1 & \text{otherwise} \end{cases} \tag{15}$$

where m is the total number of modules in the morphology, l is the number of modules with only one face attached to another module (except for the core component) and l_{max} is the maximum number of modules with one face attached that the morphology could have, if the modules were attached differently.

Proportion It represents the length-width ratio of the rectangular box containing the morphology. It is defined as:

$$P = \frac{p_s}{p_l} \tag{16}$$

where p_s is the shortest side of the morphology and p_l is the longest.

---

[3] `(https://github.com/ci-group/revolve2/)`

Symmetry The maximum between the symmetry of the 3 dimensions.

Width The width of the bounding box containing the morphology.

Height The height of the bounding box containing the morphology.

Depth The depth of the bounding box containing the morphology.

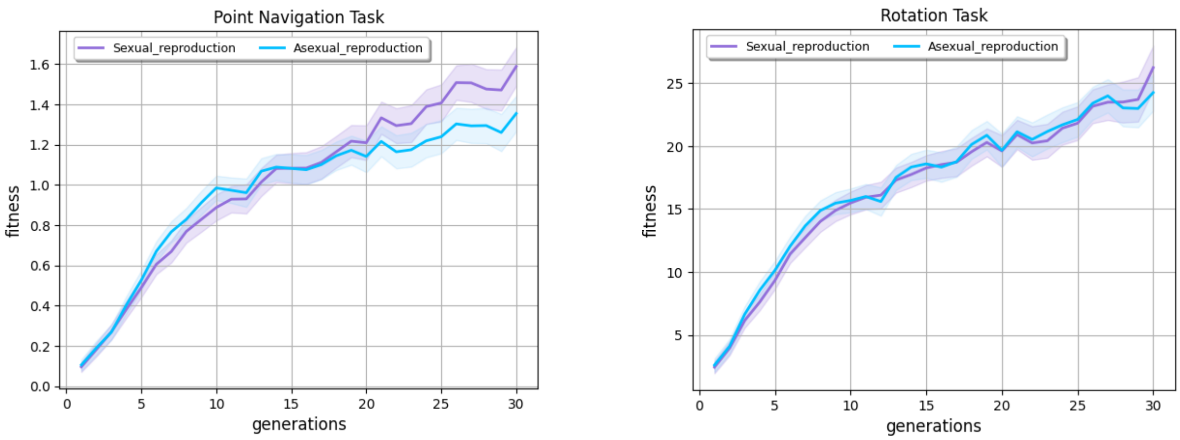

Figure 11: Mean and standard deviation of the fitness at each generation for the two tasks. Results are similar between reproduction methods but sexual reproduction is higher in the last generations.

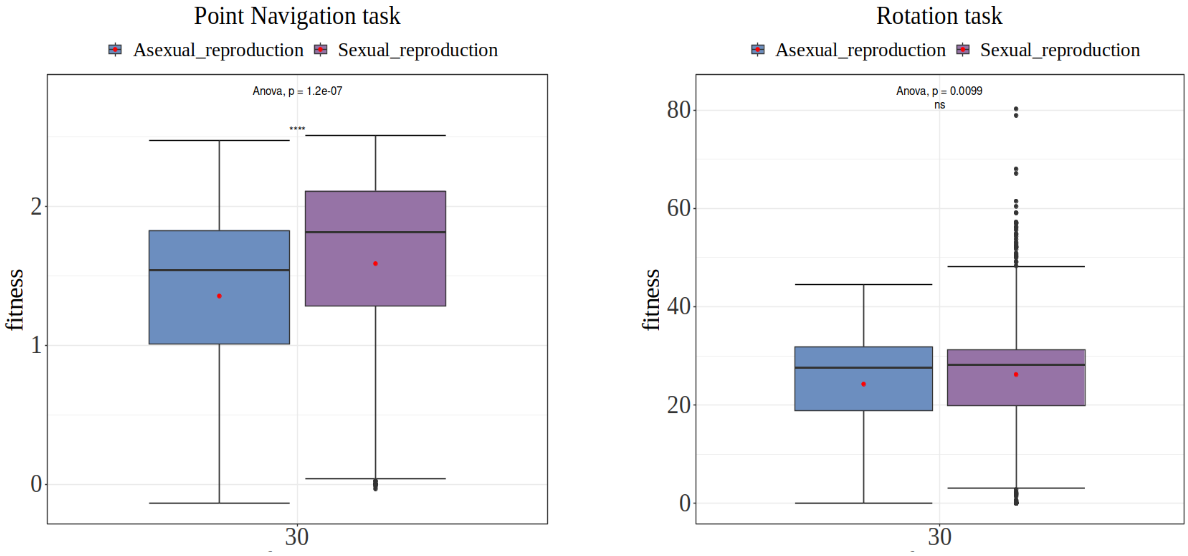

Figure 12: Box plot for the fitness values of the population at generation 30 for the two tasks. While for point navigation sexual reproduction has a better mean fitness, the difference is less noticeable for the rotation task. The best sexually reproducing robots for the rotation task, however, are far better than the ones produced by asexual reproduction.

5 Results

5.1 Performance

In Fig 11 are reported the average fitness of the population generated by the system at each generation on the two tasks. We can see that both reproduction methods can generate robots that can solve the two tasks successfully. Sexual reproduction achieves higher levels of mean fitness for the population of both tasks than asexual reproduction. For point navigation, sexual reproduction has a higher mean fitness value across generations and produces the robot with the highest fitness, 2.51. The highest fitness value obtained with asexual reproduction is 2.47. For the task of panoramic rotation the difference in terms of mean fitness is less pronounced but the best robots produced by sexual reproduction outperforms by huge margins the best robots produced by asexual reproduction. The sexually reproducing robot with the highest fitness value, 80.21, is almost twice as better as the best robot by asexual reproduction, whose fitness is only 44.52. This is highlighted in Fig 12.

5.2 Morphology

The morphological descriptors for the 10 experiments are reported in Fig 13 for the task of point navigation and Fig 14 for the task of panoramic rotation. The morphologies evolved for both tasks are big in size, having close to 10 modules on average, and are mostly made of hinges with very few bricks. For this reason, the bodies are mostly made of long limbs attached to the core module. In this setting, the higher branching values reported for the robots evolved for point navigation, indicate their higher number of limbs. This is also confirmed by their higher values of the relative number of limbs. Fig 16 and Fig 17 present the 5 best robots for both tasks and reproduction methods. All the robots showed but one, are made exclusively of hinges. The effect of branching and relative number of limbs is evident in these where the best robots for point navigation have 3 or 4 limbs made of hinges and those for rotation only have 2 or 3. Another difference between the morphologies generated for the two tasks is their symmetry. The robots of point navigation (Fig 13) have, on average, bodies that are more symmetrical than those of the rotation robots (Fig 14). This fact can be further validated by looking at the best robots in Fig 16 and Fig 17.

Another important distinction has to be made between the morphologies generated using sexual and asexual reproduction methods. Even if the reproduction method only affects the generation of the brains, the evolved morphologies assume different characteristics. It is evident from Fig 13 and 14 that, for both tasks, robots evolved using sexual reproduction have higher levels of branching, relative number of limbs, symmetry, and width. We present the fitness landscape plots using pairs of morphological descriptors as coordinates in Fig 15. The plots show the fitness landscape as a function of relative number of limbs over symmetry for both tasks and reproduction methods. For both tasks, the highest fitness robots from sexual reproduction are in the morphological space of higher symmetry and relative number of limbs. This means that the best robots from sexual reproduction are more symmetric than their asexual reproduction counterparts and their modules are arranged in a way that creates more limbs.

5.3 Learning Delta

Fig 18 shows the learning delta growing across generations for both tasks and reproduction methods. The learning delta grows very fast for the first 10 generations and then slows down but never stops increasing. This is an effect, already discovered in Miras et al. [2020a] and Jie Luo [2022] with different learning methods, different representations, and tasks. These results provide additional evidence that the introduction of learning drives the evolutionary process toward morphology with increasing learning potential.

The plot in Fig 18 shows also the difference in the learning delta between the two reproduction methods. Since the bodies of the robots are produced in the same way by sexual reproduction, the difference must be produced by the reproduction method of the brain. The learning delta not only shows the learning potential of a robot's morphology but also that of its brain. A reproduction method might produce robots more capable of learning complex behaviors.

For the point navigation task, sexual reproduction seems to produce bodies with higher learning potentials. The difference, however, is only noticeable in the last generations. In the rotation task, instead, asexual reproduction seems to have a higher learning delta earlier in the evolution but the situation is reversed later in the evolutionary process. Comparing side by side the data for the learning with the fitness values (Fig 11) we find an interesting phenomenon. In the point navigation task, the higher fitness values of the robots that use sexual reproduction are driven by their greater ability to learn and higher starting fitness (Fig 19). For the task of rotation, this is true only for the final generations. Earlier in the evolutionary process, the two reproduction methods obtain similar fitness values but the sexually reproducing robots exhibit a lower learning delta. This suggests that they need a lower amount of learning to reach the same fitness values. This is further exemplified by the fitness of the robots with the inherited brains before learning (Fig 19) where, for the first part of the evolution, sexually reproducing robots outcompete asexually reproducing ones.

5.4 Behavior

To get a better understanding of the robots' behavior, we visualize the trajectories and their density plots of the best performing robots for point navigation in the last generation across all runs (Fig 20). The plots show very similar trajectories being produced by both reproduction methods. However, the majority of the robots that use sexual reproduction reach the second target point whether only some of the robots produced using asexual reproduction can do the same.

We plot the vertical coordinate of the best robots from the last generation of the rotation task in Fig 21. We can see how, the best performing robots produced using sexual reproduction, keep a higher position on the vertical coordinate on average. This is mostly due to a very effective behavior which is composed of little jumps. On the other hand, the best robots of asexual reproduction adopt a simpler behavior by which they rotate while staying on the ground.

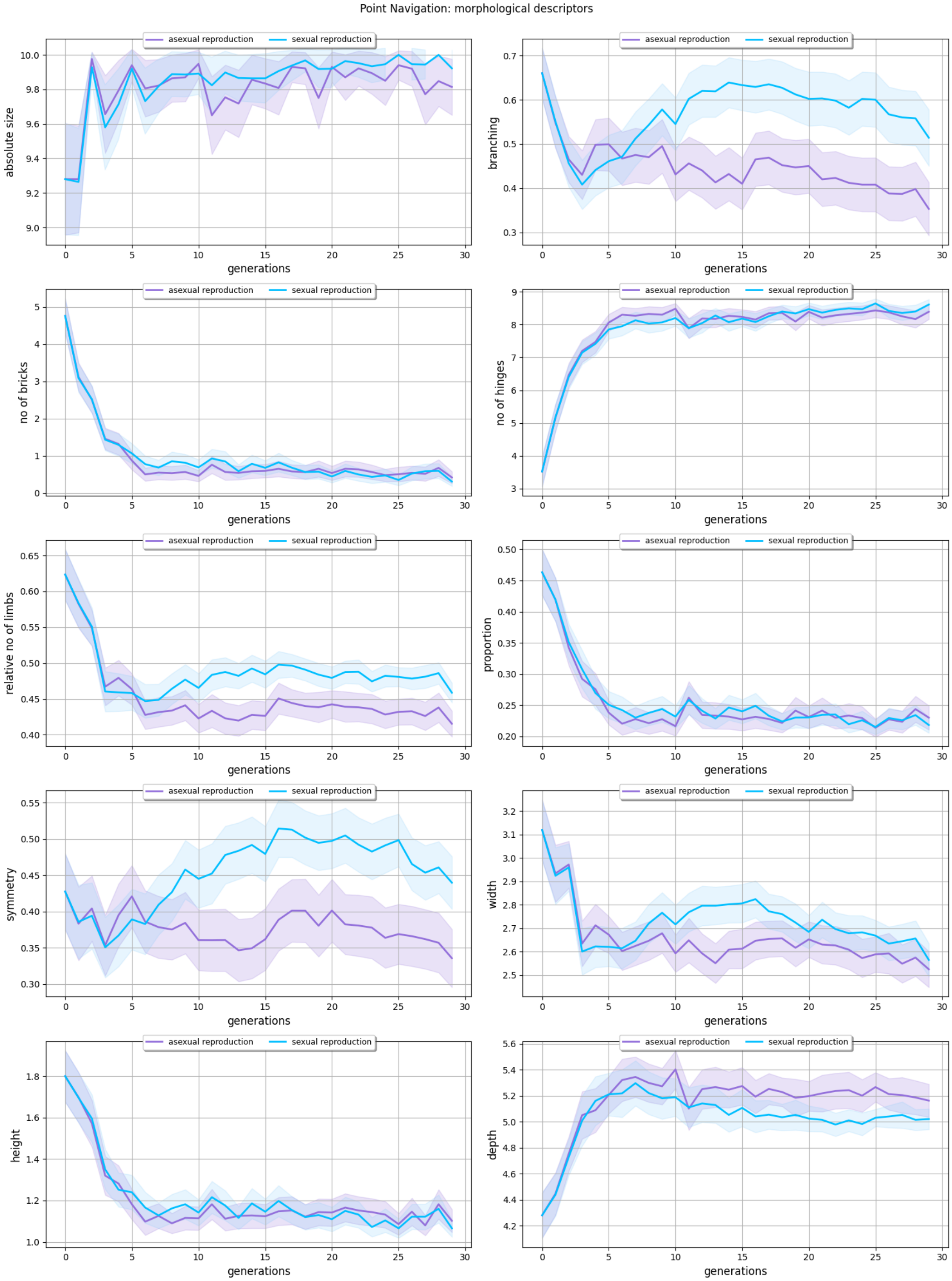

Figure 13: The 10 morphological descriptors for the bodies evolved for the point navigation task. The main difference between sexual and asexual reproduction is in the values of branching, relative number of limbs, symmetry, and width.

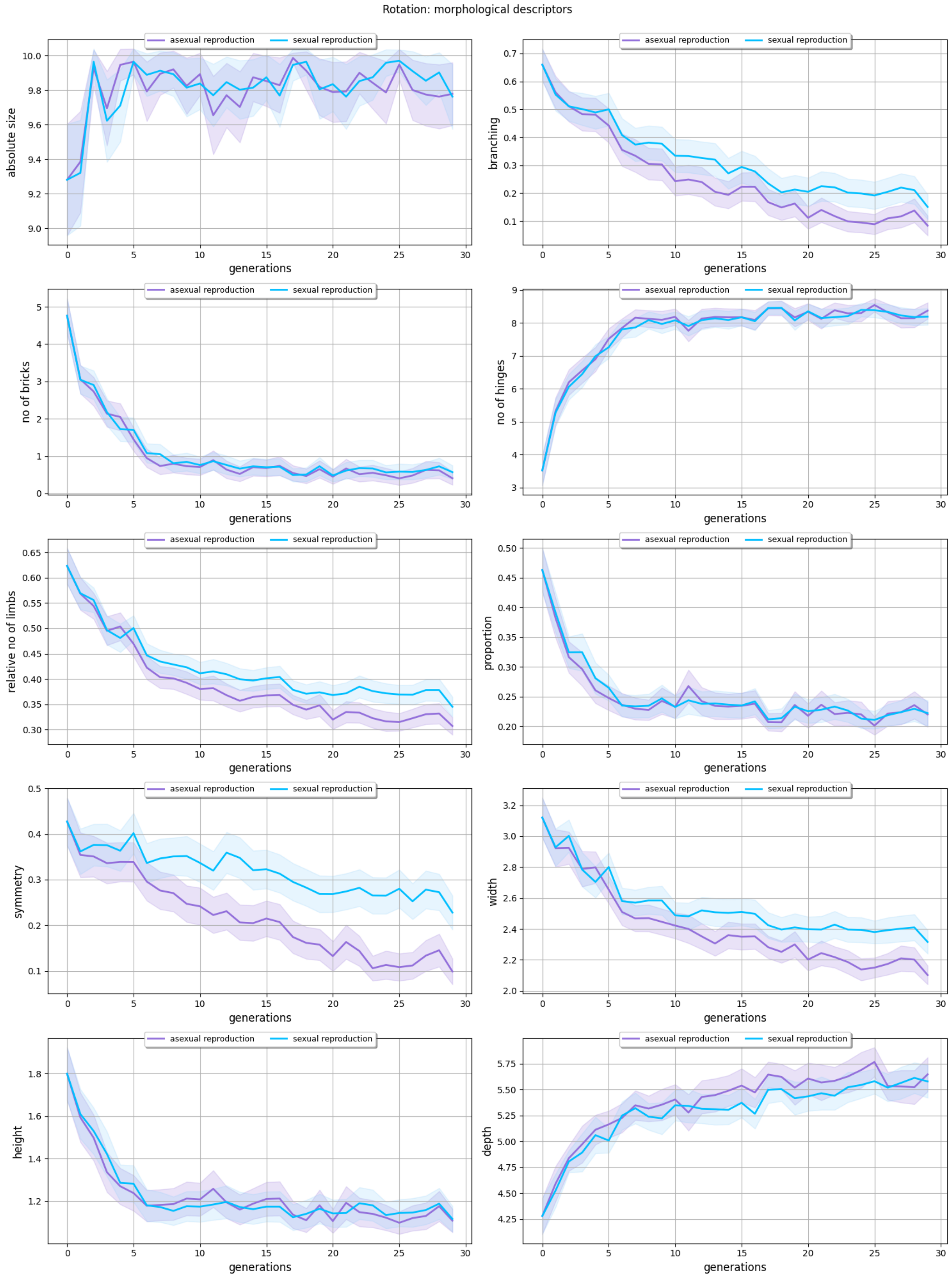

Figure 14: The 10 morphological descriptors for the bodies evolved for the panoramic rotation task. The main difference between sexual and asexual reproduction is in the values of branching, relative number of limbs, symmetry, and width.

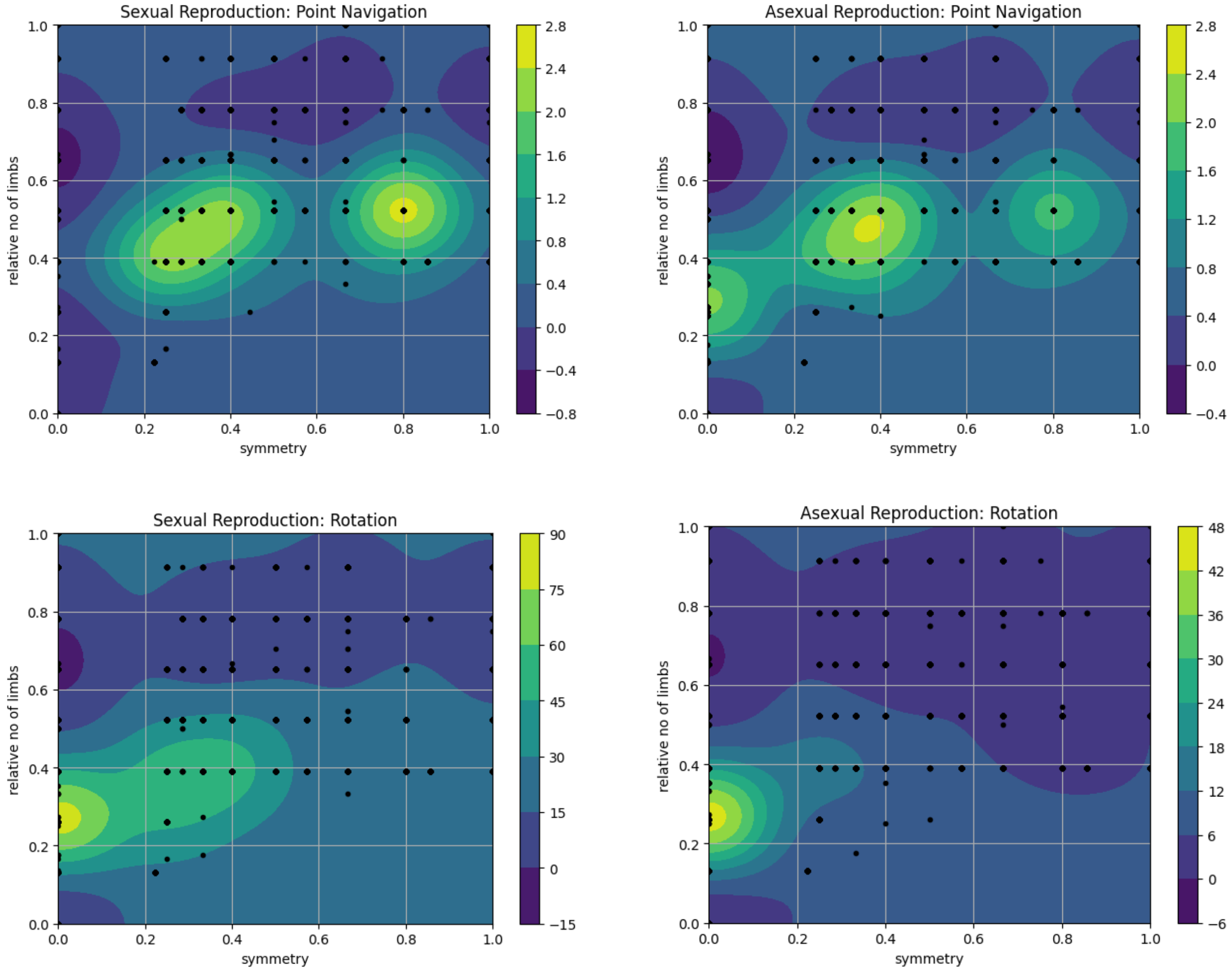

Figure 15: Fitness landscapes of the symmetry over relative number of limbs morphological descriptors for both tasks and reproduction mechanisms. Sexually reproducing robots are more symmetric and have more limbs.

6 Conclusion and Future Work

We evaluated our Evolutionary Robotics system using the sexual and asexual reproduction methods. Both methods manage to produce well performing robots on both tasks of point navigation and panoramic rotation. Robots that use the sexual reproduction mechanism, however, are better on average than those that use asexual reproduction. Moreover, sexual reproduction generates the robots with the highest fitness. This would answer our first research question. Our experiments suggest that, when optimizing robots to solve a task, sexual reproduction is the best choice.

Both reproduction methods drive the evolution process toward big robots composed of many active hinges. The best robots are made of long limbs attached to the core module. We find that the chosen brain reproduction method affects the characteristics of the resulting morphologies. Robots whose brains are evolved using sexual reproduction have more limbs and are more symmetric and wide. This is particularly interesting because the choice of the reproduction method only affects how the brains are reproduced. The results suggest that the way the brains of the offspring are generated produces changes in their bodies. To a certain extent, we are demonstrating how the brains can shape the bodies, a phenomenon already reported in Jie Luo [2022] and an inversion of the classic statement that the bodies shape the brain (Pfeifer and Bongard [2007]).

We also study the complex interaction between the evolutionary process and the introduction of learning. We find that the evolutionary process can generate robots with increasing learning potentials, a phenomenon already studied in Jie Luo [2022] and Gupta et al. [2021]. However, we notice a difference between robots produced by different reproduction methods. While, in the last generations of both tasks sexually reproducing robots exhibit a higher learning delta, for the first generations the learning delta of asexually reproducing robots is equal or even higher. In the rotation

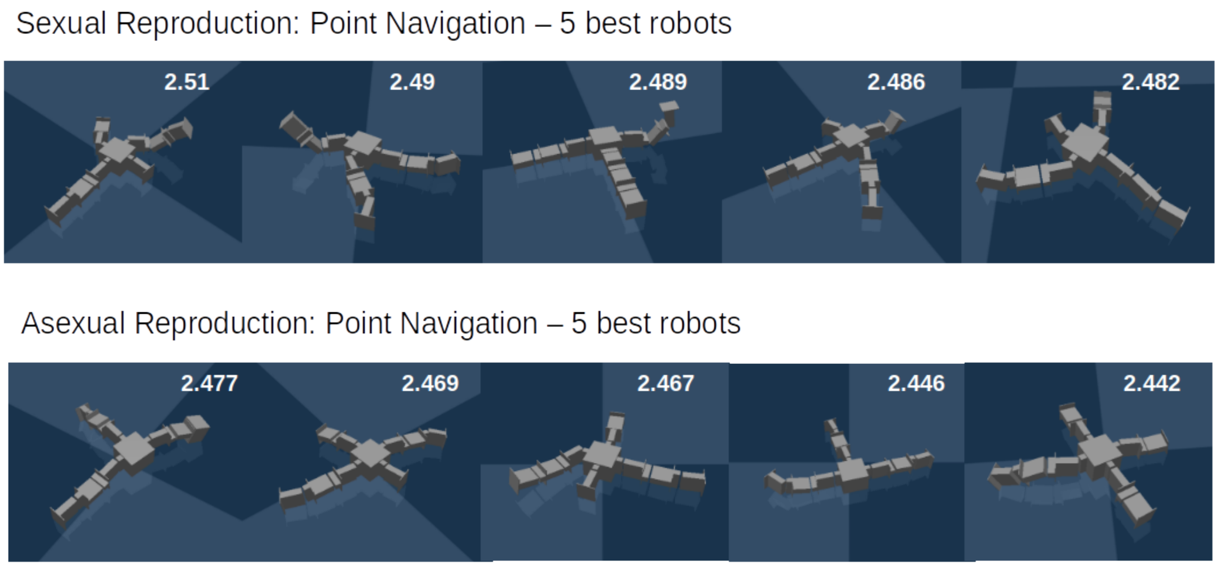

Figure 16: The 5 best robots produced for the point navigation task by sexual and asexual reproduction and their fitnesses. Sexually reproducing robots have higher fitness values.

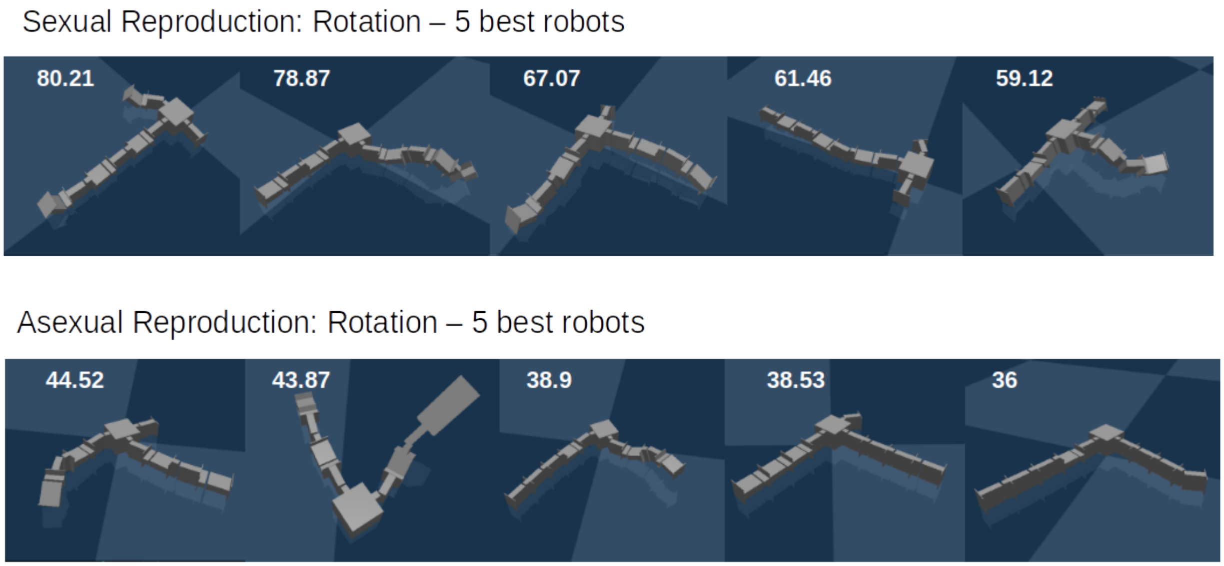

Figure 17: The 5 best robots produced for the panoramic rotation task by sexual and asexual reproduction and their fitnesses. Sexually reproducing robots have higher fitness values and are more diverse.

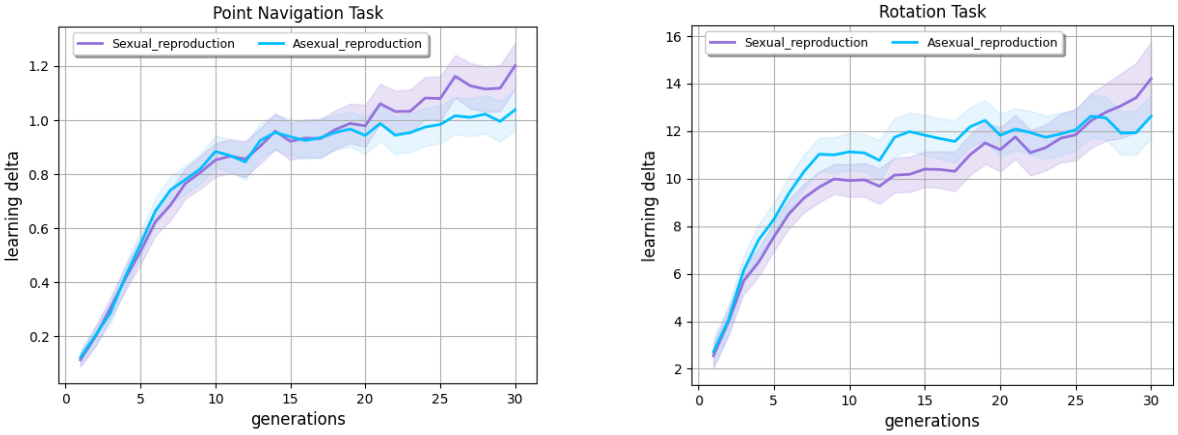

Figure 18: Learning delta of the fitness for each generation for both tasks and reproduction mechanisms. In the last generations, sexual reproduction produces robots with a higher learning delta. In the rotation task, the learning delta of the robots produced by asexual reproduction is higher in the first generations.

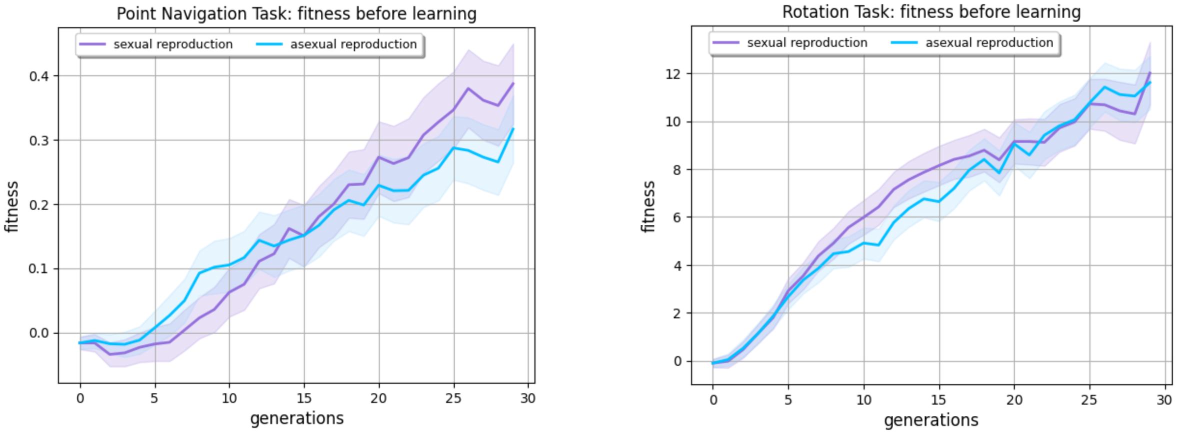

Figure 19: The fitness value of the robots before learning for both tasks and reproduction mechanisms. For the task of point navigation, the fitness of sexually reproducing robots with inherited brains is higher than that of the asexually reproducing ones only in the last generations. For the panoramic rotation task, the opposite is true.

task, the learning delta of asexually reproducing robots is higher than its sexual counterpart until the last 5 generations. Despite the better learning delta, sexually reproducing robots have a better fitness before learning which results in the two reproduction methods obtaining the same fitness values after learning. We conclude that asexually reproducing robots need to learn more to compensate for the low fitness obtained using their inherited brains whether sexually reproducing robots display better performance as soon as they are born. Sexual reproduction is therefore a more efficient mechanism, being able to better combine the controllers of two parents to produce brains more suited to the bodies of their offspring in the first generations and better learning robots in the later ones. The reproduction mechanism influences the ability of a robot's inherited brain to work out of the box and its capability to learn. These qualities, in turn, determine the performance of the robot and its probability of surviving. This chain of influence drives evolution to produce morphologies with higher degrees of morphological intelligence that make them better suited for the task.

To answer our third research question, we also highlight a difference in the behaviors of the robots that evolved using the two reproduction methods for both tasks. In point navigation, the majority of the best robots evolved using sexual reproduction reach the second target point while only a few of those asexually evolved do so. In the task of rotation, the difference in behavior is even more evident, with the sexually reproducing robots learning a jumping behavior that lets them rotate faster.

Possible future works will study the different reproduction methods in more challenging tasks and changing environments and test reproduction methods using multiple parents. Other future works will also allow for the possibility

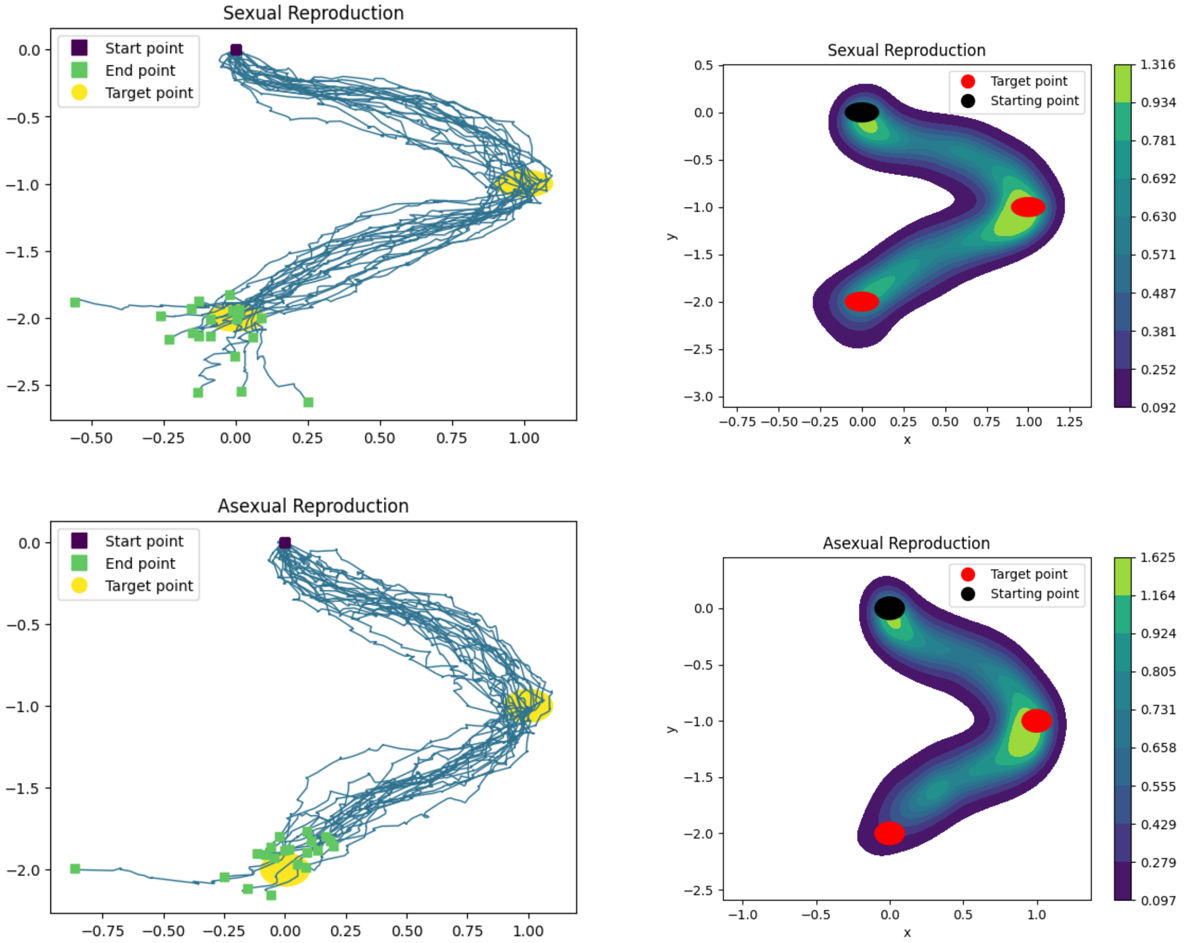

Figure 20: Trajectories of the best 20 robots from the last generation in the point navigation task. The majority of the robots from sexual reproduction reach the second target point. Using asexual reproduction, only a few of them do the same.

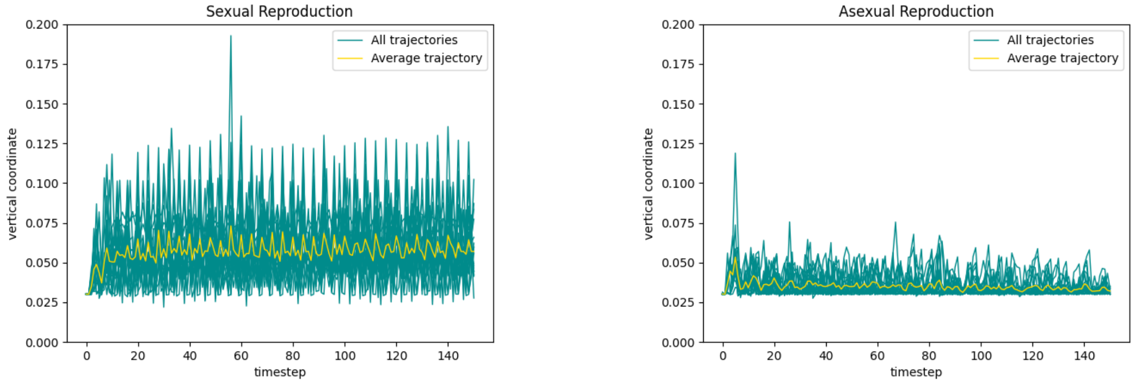

Figure 21: Vertical coordinate of the trajectories of the 20 best robots from the last generation in the panoramic rotation task. The robots from sexual reproduction exhibit a jumping behavior that helps them rotate faster.

of encoding the reproduction strategy in the genotype of the robots and finding the best one using evolution. The possibility of combining multiple tasks to evolve more generalist robots is also a promising research topic. Another line of future research will examine the possibility of implementing a form of Lamarckian evolution, where the learned brain features are mapped back to their genotypes, making them inheritable.

References

Angelo Cangelosi, Josh Bongard, Martin H Fischer, and Stefano Nolfi. Embodied intelligence. *Springer handbook of computational intelligence*, pages 697–714, 2015.

Rolf Pfeifer and Josh Bongard. *How the body shapes the way we think: a new view of intelligence*. MIT press, 2007.

R Walker. Shadow dextrous hand technical specification. *Shadow Robot Company*, 388, 2005.

Eric Brown, Nicholas Rodenberg, John Amend, Annan Mozeika, Erik Steltz, Mitchell R Zakin, Hod Lipson, and Heinrich M Jaeger. Universal robotic gripper based on the jamming of granular material. *Proceedings of the National Academy of Sciences*, 107(44):18809–18814, 2010.

A.E. Eiben, N. Bredeche, M. Hoogendoorn, J. Stradner, J. Timmis, A.M. Tyrrell, and A. Winfield. The triangle of life: Evolving robots in real-time and real-space. *Advances in Artificial Life, ECAL 2013*, 09 2013. doi:10.7551/978-0-262-31709-2-ch157.

A. E. Eiben and Emma Hart. If it evolves it needs to learn. In *GECCO 2020 Companion - Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion*, pages 1383–1384, 2020. ISBN 9781450371278. doi:10.1145/3377929.3398151.

Jakub M. Tomczak Jacintha Ellers Agoston E. Eiben Jie Luo, Aart Stuurman. The Effects of Learning in Morphologically Evolving Robot Systems. *https://arxiv.org/pdf/2111.09851.pdf*, 2022.

Jan Engelstädter. Constraints on the evolution of asexual reproduction. *BioEssays*, 30(11-12):1138–1150, 2008.

Agoston E Eiben. Multiparent recombination in evolutionary computing. *Advances in evolutionary computing: theory and applications*, pages 175–192, 2003.

August Weismann. *The germ-plasm: a theory of heredity*. Scribner's, 1893.

James Mark Baldwin et al. A new factor in evolution. *Diacronia*, (7):1–13, 2018.

Karine Miras, Matteo De Carlo, Sayfeddine Akhatou, and A. E. Eiben. Evolving-controllers versus learning-controllers for morphologically evolvable robots. In *Applications of Evolutionary Computation*, pages 86–99, 04 2020a. doi:10.1007/978-3-030-43722-0_6.

Alan Turing. Intelligent machinery (1948). *B. Jack Copeland*, page 395, 2004.

Agoston E Eiben and Jim Smith. From evolutionary computation to the evolution of things. *Nature*, 521(7553):476–482, 2015.

Dave Cliff, Phil Husbands, and Inman Harvey. Explorations in evolutionary robotics. *Adaptive behavior*, 2(1):73–110, 1993.

Dario Floreano and Francesco Mondada. Evolution of plastic neurocontrollers for situated agents. In *Proc. of The Fourth International Conference on Simulation of Adaptive Behavior (SAB), From Animals to Animats*. ETH Zürich, 1996.

Inman Harvey, Phil Husbands, Dave Cliff, Adrian Thompson, and Nick Jakobi. Evolutionary robotics: the sussex approach. *Robotics and autonomous systems*, 20(2-4):205–224, 1997.

Stefano Nolfi and Dario Floreano. *Evolutionary robotics: The biology, intelligence, and technology of self-organizing machines*. MIT press, 2000.

Agoston Endre Eiben. In vivo veritas: Towards the evolution of things. In *International Conference on Parallel Problem Solving from Nature*, pages 24–39. Springer, 2014.

Bruno Siciliano, Oussama Khatib, and Torsten Kröger. *Springer handbook of robotics*, volume 200. Springer, 2008.

Stephane Doncieux, Nicolas Bredeche, Jean-Baptiste Mouret, and Agoston E Eiben. Evolutionary robotics: what, why, and where to. *Frontiers in Robotics and AI*, 2:4, 2015.

Gongjin Lan, Maarten van Hooft, Matteo De Carlo, Jakub M Tomczak, and AE Eiben. Learning locomotion skills in evolvable robots. *Neurocomputing*, 452:294–306, 2021a.

Michael J Wiser, Noah Ribeck, and Richard E Lenski. Long-term dynamics of adaptation in asexual populations. *Science*, 342(6164):1364–1367, 2013.

John Long. *Darwin's devices: what evolving robots can teach us about the history of life and the future of technology*. Basic Books (AZ), 2012.

John Maynard Smith. Byte-sized evolution. *Nature*, 355(6363):772–773, 1992.

Jean-Marc Montanier and Nicolas Bredeche. Surviving the tragedy of commons: emergence of altruism in a population of evolving autonomous agents. In *European Conference on Artificial Life*, 2011.

Jean-Marc Montanier and Nicolas Bredeche. Evolution of altruism and spatial dispersion: an artificial evolutionary ecology approach. In *European Conference on Artificial Life (ECAL-2013)*, pages 1–8, 2013.

Markus Waibel, Dario Floreano, and Laurent Keller. A quantitative test of hamilton's rule for the evolution of altruism. *PLoS biology*, 9(5):e1000615, 2011.

Michael Solomon, Terence Soule, and Robert B Heckendorn. A comparison of a communication strategies in cooperative learning. In *Proceedings of the 14th annual conference on Genetic and evolutionary computation*, pages 153–160, 2012.

Dario Floreano, Sara Mitri, Stéphane Magnenat, and Laurent Keller. Evolutionary conditions for the emergence of communication in robots. *Current biology*, 17(6):514–519, 2007.

Sara Mitri, Dario Floreano, and Laurent Keller. The evolution of information suppression in communicating robots with conflicting interests. *Proceedings of the National Academy of Sciences*, 106(37):15786–15790, 2009.

Steffen Wischmann, Dario Floreano, and Laurent Keller. Historical contingency affects signaling strategies and competitive abilities in evolving populations of simulated robots. *Proceedings of the National Academy of Sciences*, 109(3):864–868, 2012.

Josh Bongard. Morphological change in machines accelerates the evolution of robust behavior. *Proceedings of the National Academy of Sciences*, 108(4):1234–1239, 2011.

Joshua E Auerbach and Josh C Bongard. Environmental influence on the evolution of morphological complexity in machines. *PLoS computational biology*, 10(1):e1003399, 2014a.

Randal S Olson, Arend Hintze, Fred C Dyer, David B Knoester, and Christoph Adami. Predator confusion is sufficient to evolve swarming behaviour. *Journal of The Royal Society Interface*, 10(85):20130305, 2013.

Auke Jan Ijspeert, Alessandro Crespi, Dimitri Ryczko, and Jean-Marie Cabelguen. From swimming to walking with a salamander robot driven by a spinal cord model. *science*, 315(5817):1416–1420, 2007.

Rohan Thakker, Ajinkya Kamat, Sachin Bharambe, Shital Chiddarwar, and Kishor M Bhurchandi. Rebis-reconfigurable bipedal snake robot. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 309–314. IEEE, 2014.

Vitor Matos and Cristina P Santos. Towards goal-directed biped locomotion: Combining cpgs and motion primitives. *Robotics and Autonomous Systems*, 62(12):1669–1690, 2014.

Jason Yosinski, Jeff Clune, Diana Hidalgo, Sarah Nguyen, Juan Cristobal Zagal, Hod Lipson, et al. Evolving robot gaits in hardware: the hyperneat generative encoding vs. parameter optimization. In *ECAL*, pages 890–897, 2011.

Joseph Ayers and Jill Crisman. Lobster walking as a model for an omnidirectional robotic ambulation architecture. In *Proceedings of the workshop on" Locomotion Control in Legged Invertebrates" on Biological neural networks in invertebrate neuroethology and robotics*, pages 287–316, 1993.

Barbara Webb. Robots in invertebrate neuroscience. *Nature*, 417(6886):359–363, 2002.

Deepak Trivedi, Christopher D Rahn, William M Kier, and Ian D Walker. Soft robotics: Biological inspiration, state of the art, and future research. *Applied bionics and biomechanics*, 5(3):99–117, 2008.

Nicholas Cheney, Jeff Clune, and Hod Lipson. Evolved electrophysiological soft robots. In *Proceedings of the 14th International Conference on the Synthesis and Simulation of Living Systems, ALIFE 2014*, pages 222–229, 2014a. ISBN 9780262326216. doi:10.7551/978-0-262-32621-6-ch037.

Nick Cheney, Robert MacCurdy, Jeff Clune, and Hod Lipson. Unshackling evolution: evolving soft robots with multiple materials and a powerful generative encoding. *ACM SIGEVOlution*, 7(1):11–23, 2014b.

Levent Bayindir and Erol Şahin. A review of studies in swarm robotics. *Turkish Journal of Electrical Engineering and Computer Sciences*, 15(2):115–147, 2007.

Victor Zykov, Andrew Chan, and Hod Lipson. Molecubes: An open-source modular robotics kit. In *IROS-2007 Self-Reconfigurable Robotics Workshop*, pages 3–6. Citeseer, 2007.

Hod Lipson and Jordan B Pollack. Automatic design and manufacture of robotic lifeforms. *Nature*, 406(6799):974–978, 2000.

Victor Zykov, Efstathios Mytilinaios, Bryant Adams, and Hod Lipson. Self-reproducing machines. *Nature*, 435(7039): 163–164, 2005.

Luzius Brodbeck, Simon Hauser, and Fumiya Iida. Morphological evolution of physical robots through model-free phenotype development. *PloS one*, 10(6):e0128444, 2015.

Joshua E. Auerbach, Deniz Aydin, Andrea Maesani, Przemyslaw M. Kornatowski, Titus Cieslewski, Grégoire Heitz, Pradeep R. Fernando, Ilya Loshchilov, Ludovic Daler, and Dario Floreano. Robogen: Robot generation through artificial evolution. In *Proceedings of the 14th International Conference on the Synthesis and Simulation of Living Systems, ALIFE 2014*, pages 136–137, 2014a. ISBN 9780262326216. doi:10.7551/978-0-262-32621-6-ch022.

Karine Miras, Evert Haasdijk, Kyrre Glette, and A.E. Eiben. Effects of selection preferences on evolved robot morphologies and behaviors. In *Artificial Life Conference Proceedings*, pages 224–231, Cambridge, MA, 2018. MIT Press, MIT Press.

Karine Miras, Eliseo Ferrante, and Agoston E Eiben. Environmental influences on evolvable robots. *PloS one*, 15(5): e0233848, 2020b.

Matteo De Carlo, Daan Zeeuwe, Eliseo Ferrante, Gerben Meynen, Jacintha Ellers, and A.E. Eiben. Influences of artificial speciation on morphological robot evolution. In *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 2272–2279, 2020a.

Milan Jelisavcic, Matteo De Carlo, Elte Hupkes, Panagiotis Eustratiadis, Jakub Orlowski, Evert Haasdijk, Joshua E Auerbach, and Agoston E Eiben. Real-world evolution of robot morphologies: A proof of concept. *Artificial life*, 23 (2):206–235, 2017a.

Stuart J Russell. *Artificial intelligence a modern approach*. Pearson Education, Inc., 2010.

Alessandro Saffiotti. The uses of fuzzy logic in autonomous robot navigation. *Soft computing*, 1:180–197, 1997.

Maja J Matarić. Behavior-based robotics as a tool for synthesis of artificial behavior and analysis of natural behavior. *Trends in cognitive sciences*, 2(3):82–86, 1998.

Andrea Roli, Mattia Manfroni, Carlo Pinciroli, and Mauro Birattari. On the design of boolean network robots. In *Applications of Evolutionary Computation: EvoApplications 2011: EvoCOMPLEX, EvoGAMES, EvoIASP, EvoINTELLIGENCE, EvoNUM, and EvoSTOC, Torino, Italy, April 27-29, 2011, Proceedings, Part I*, pages 43–52. Springer Berlin Heidelberg, 2011.

George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

W Thomas Miller, Richard S Sutton, and Paul J Werbos. *Neural networks for control*. MIT press, 1995.

Ken-ichi Funahashi and Yuichi Nakamura. Approximation of dynamical systems by continuous time recurrent neural networks. *Neural networks*, 6(6):801–806, 1993.

Larry F Abbott and Sacha B Nelson. Synaptic plasticity: taming the beast. *Nature neuroscience*, 3(11):1178–1183, 2000.

Ed Bullmore and Olaf Sporns. Complex brain networks: graph theoretical analysis of structural and functional systems. *Nature reviews neuroscience*, 10(3):186–198, 2009.

Gongjin Lan, Matteo De Carlo, Fuda van Diggelen, Jakub M Tomczak, Diederik M Roijers, and Agoston E Eiben. Learning directed locomotion in modular robots with evolvable morphologies. *Applied Soft Computing*, 111:107688, 2021b.

Dirk Bucher, Gal Haspel, Jorge Golowasch, and Farzan Nadim. Central Pattern Generators. In *eLS*, pages 1–12, 11 2015. ISBN 9780470015902. doi:10.1002/9780470015902.a0000032.pub2.

Auke Jan Ijspeert. Central pattern generators for locomotion control in animals and robots: a review. *Neural networks*, 21(4):642–653, 2008.

Matteo De Carlo, Daan Zeeuwe, Eliseo Ferrante, Gerben Meynen, Jacintha Ellers, and AE Eiben. Robotic task affects the resulting morphology and behaviour in evolutionary robotics. In *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 2125–2131. IEEE, 2020b.

Yasuhiro Fukuoka, Hiroshi Kimura, and Avis H Cohen. Adaptive dynamic walking of a quadruped robot on irregular terrain based on biological concepts. *The International Journal of Robotics Research*, 22(3-4):187–202, 2003.

Hiroshi Kimura, Seiichi Akiyama, and Kazuaki Sakurama. Realization of dynamic walking and running of the quadruped using neural oscillator. *Autonomous robots*, 7(3):247–258, 1999.

Hiroshi Kimura, Yasuhiro Fukuoka, and Avis H Cohen. Adaptive dynamic walking of a quadruped robot on natural ground based on biological concepts. *The International Journal of Robotics Research*, 26(5):475–490, 2007.

Karl Sims. Evolving 3d morphology and behavior by competition. *Artificial life*, 1(4):353–372, 1994.

Eivind Samuelsen, Kyrre Glette, and Jim Torresen. A hox gene inspired generative approach to evolving robot morphology. In *Proceedings of the 15th annual conference on Genetic and evolutionary computation*, pages 751–758, 2013.

Frank Veenstra, Andres Faina, Sebastian Risi, and Kasper Stoy. Evolution and morphogenesis of simulated modular robots: a comparison between a direct and generative encoding. In *European Conference on the Applications of Evolutionary Computation*, pages 870–885. Springer, 2017.

Paolo Pagliuca and Stefano Nolfi. The dynamic of body and brain co-evolution. *https://arxiv.org/abs/2011.11440*, 2021.

Tønnes F Nygaard, Eivind Samuelsen, and Kyrre Glette. Overcoming initial convergence in multi-objective evolution of robot control and morphology using a two-phase approach. In *European Conference on the Applications of Evolutionary Computation*, pages 825–836, 2017.

Agrim Gupta, Silvio Savarese, Surya Ganguli, and Li Fei-Fei. Embodied Intelligence via Learning and Evolution. *Nature Communications*, 12, 2021. doi:10.1038/s41467-021-25874-z.

Sabine Hauert, Jean-Christophe Zufferey, and Dario Floreano. Evolved swarming without positioning information: an application in aerial communication relay. *Autonomous Robots*, 26:21–32, 2009.

Matteo De Carlo, Eliseo Ferrante, Jacintha Ellers, Gerben Meynen, and AE Eiben. The impact of different tasks on evolved robot morphologies. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 91–92, 2021.

Milan Jelisavcic, Evert Haasdijk, and AE Eiben. Acquiring moving skills in robots with evolvable morphologies: Recent results and outlook. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 1735–1741, 2017b.

Jeff Clune, Benjamin E. Beckmann, Charles Ofria, and Robert T. Pennock. Evolving coordinated quadruped gaits with the hyperneat generative encoding. In *2009 IEEE Congress on Evolutionary Computation*, pages 2764–2771, 2009. doi:10.1109/CEC.2009.4983289.

David Christensen, Jørgen Larsen, and Kasper Stoy. Fault-tolerant gait learning and morphology optimization of a polymorphic walking robot. *Evolving Systems*, 5, 03 2013. doi:10.1007/s12530-013-9088-3.

Gongjin Lan, Milan Jelisavcic, Diederik M Roijers, Evert Haasdijk, and Agoston E Eiben. Directed locomotion for modular robots with evolvable morphologies. In *International Conference on Parallel Problem Solving from Nature*, pages 476–487. Springer, 2018.

Babak H Kargar, Karine Miras, and AE Eiben. The effect of selecting for different behavioral traits on the evolved gaits of modular robots. In *ALIFE 2022: The 2022 Conference on Artificial Life*. MIT Press, 2021.

Joel Lehman and Kenneth O Stanley. Evolving a diversity of virtual creatures through novelty search and local competition. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, pages 211–218, 2011.

Joel Lehman, Kenneth O Stanley, et al. Exploiting open-endedness to solve problems through the search for novelty. In *ALIFE*, pages 329–336, 2008.

Sebastian Risi, Sandy D Vanderbleek, Charles E Hughes, and Kenneth O Stanley. How novelty search escapes the deceptive trap of learning to learn. In *Proceedings of the 11th Annual conference on Genetic and evolutionary computation*, pages 153–160, 2009.

Jorge Gomes, Paulo Urbano, and Anders Lyhne Christensen. Introducing novelty search in evolutionary swarm robotics. In *Swarm Intelligence: 8th International Conference, ANTS 2012, Brussels, Belgium, September 12-14, 2012. Proceedings 8*, pages 85–96. Springer, 2012.

Jorge Gomes, Paulo Urbano, and Anders Lyhne Christensen. Evolution of swarm robotics systems with novelty search. *Swarm Intelligence*, 7:115–144, 2013.

Antonios Liapis, Georgios N Yannakakis, and Julian Togelius. Enhancements to constrained novelty search: Two-population novelty search for generating game content. In *Proceedings of the 15th annual conference on Genetic and evolutionary computation*, pages 343–350, 2013.

Charles Darwin. *On the origin of species, 1859*. Routledge, 2004.

Robert M Sapolsky. *Behave: The biology of humans at our best and worst*. Penguin, 2017.

Karine Miras and Agoston E Eiben. Effects of environmental conditions on evolved robot morphologies and behavior. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 125–132, 2019.

Joshua E. Auerbach and Josh C. Bongard. Environmental Influence on the Evolution of Morphological Complexity in Machines. *PLoS Computational Biology*, 10(1), 2014b. ISSN 1553734X. doi:10.1371/journal.pcbi.1003399.

Georgios Methenitis, Daniel Hennes, Dario Izzo, and Arnoud Visser. Novelty search for soft robotic space exploration. In *Proceedings of the 2015 annual conference on Genetic and Evolutionary Computation*, pages 193–200, 2015.

Jonata Tyska Carvalho and Stefano Nolfi. The role of environmental variations in evolutionary robotics: Maximizing performance and robustness. *arXiv preprint arXiv:2208.02809*, 2022.

Jürgen Branke. *Evolutionary optimization in dynamic environments*, volume 3. Springer Science & Business Media, 2012.

Sebastian Risi and Kenneth O Stanley. A unified approach to evolving plasticity and neural geometry. In *The 2012 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2012.

Antoine Cully, Jeff Clune, Danesh Tarapore, and Jean-Baptiste Mouret. Robots that can adapt like animals. *Nature*, 521 (7553):503–507, 2015.

Nick Jakobi, Phil Husbands, and Inman Harvey. Noise and the reality gap: The use of simulation in evolutionary robotics. In *European Conference on Artificial Life*, pages 704–720. Springer, 1995.

Sylvain Koos, Jean-Baptiste Mouret, and Stéphane Doncieux. The transferability approach: Crossing the reality gap in evolutionary robotics. *IEEE Transactions on Evolutionary Computation*, 17(1):122–145, 2012.

Inman Harvey, Phil Husbands, Dave Cliff, et al. Seeing the light: Artificial evolution, real vision. *From animals to animats*, 3:392–401, 1994.

Faustino Gomez and Risto Miikkulainen. Incremental evolution of complex general behavior. *Adaptive Behavior*, 5 (3-4):317–342, 1997.

Stefano Nolfi and Dario Floreano. Coevolving predator and prey robots: Do “arms races” arise in artificial evolution? *Artificial life*, 4(4):311–335, 1998.

Christopher D Rosin and Richard K Belew. New methods for competitive coevolution. *Evolutionary computation*, 5(1): 1–29, 1997.

Nicola Milano and Stefano Nolfi. Automated curriculum learning for embodied agents a neuroevolutionary approach. *Scientific reports*, 11(1):1–14, 2021.

Karine Miras. Constrained by design: Influence of genetic encodings on evolved traits of robots. *Frontiers in Robotics and AI*, 8:672379, 2021.

P Deloukas, GD Schuler, G Gyapay, EM Beasley, C Soderlund, P Rodriguez-Tome, L Hui, TC Matise, KB McKusick, JS Beckmann, et al. A physical map of 30,000 human genes. *Science*, 282(5389):744–746, 1998.

Frank Dellaert. Toward a biologically defensible model of development. Master’s thesis, Case Western Reserve University, 1995.

Jessica A Bolker. Modularity in development and why it matters to evo-devo. *American Zoologist*, 40(5):770–776, 2000.

Shigeru Kuratani. Modularity, comparative embryology and evo-devo: developmental dissection of evolving body plans. *Developmental biology*, 332(1):61–69, 2009.

Jack Collins, Ben Cottier, and David Howard. Comparing direct and indirect representations for environment-specific robot component design. In *2019 IEEE Congress on Evolutionary Computation (CEC)*, pages 2705–2712. IEEE, 2019.

Maciej Komosiński and Adam Rotaru-Varga. Comparison of different genotype encodings for simulated three-dimensional agents. *Artificial Life*, 7(4):395–418, 2001.

Kenneth O. Stanley. Compositional pattern producing networks: A novel abstraction of development. *Genetic Programming and Evolvable Machines*, 8(2):131–162, 2007a. ISSN 13892576. doi:10.1007/s10710-007-9028-8.

Aristid Lindenmayer. Mathematical models for cellular interactions in development i. filaments with one-sided inputs. *Journal of theoretical biology*, 18(3):280–299, 1968.

Evert Haasdijk, Andrei Rusu, and A. Eiben. Hyperneat for locomotion control in modular robots. In *Evolvable Systems: From Biology to Hardware*, pages 169–180, 09 2010. ISBN 978-3-642-15322-8. doi:10.1007/978-3-642-15323-5_15.

Jeff Clune and Hod Lipson. Evolving 3D objects with a generative encoding inspired by developmental biology. *ACM SIGEVOlution*, 5(4):2–12, 2011. ISSN 1931-8499. doi:10.1145/2078245.2078246.

Leni K. Le Goff, Edgar Buchanan, Emma Hart, Agoston E. Eiben, Wei Li, Matteo De Carlo, Alan F. Winfield, Matthew F. Hale, Robert Woolley, Mike Angus, Jon Timmis, and Andy M. Tyrrell. Morpho-evolution with learning using a controller archive as an inheritance mechanism. *http://arxiv.org/abs/2104.04269*, 2021.

Sarah P Otto and Thomas Lenormand. Resolving the paradox of sex and recombination. *Nature Reviews Genetics*, 3(4): 252–261, 2002.

Samir W Mahfoud. Crossover interactions among niches. In *Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence*, pages 188–193. IEEE, 1994.

David B Fogel and J Wirt Atmar. Comparing genetic operators with gaussian mutations in simulated evolutionary processes using linear systems. *Biological Cybernetics*, 63(2):111–114, 1990.

David B Fogel and Lauren C Stayton. On the effectiveness of crossover in simulated evolutionary optimization. *BioSystems*, 32(3):171–182, 1994.

Wim Hordijk and Bernard Manderick. The usefulness of recombination. In *ECAL*, pages 908–919. Citeseer, 1995.

L Eshelman. On crossover as an evolutionarily viable strategy. In *Proceedings of the Fourth International Conference on Genetic Algorithms*, pages 61–68. Morgan Kaufmann Publishers San Francisco, 1991.

William M Spears. Crossover or mutation? In *Foundations of genetic algorithms*, volume 2, pages 221–237. Elsevier, 1993.

Siew Mooi Lim, Abu Bakar Md Sultan, Md Nasir Sulaiman, Aida Mustapha, and Kuan Yew Leong. Crossover and mutation operators of genetic algorithms. *International journal of machine learning and computing*, 7(1):9–12, 2017.

Heinz Mühlenbein. Parallel genetic algorithms, population genetics and combinatorial optimization. In *Parallelism, Learning, Evolution: Workshop on Evolutionary Models and Strategies Neubiberg, Germany, March 10–11, 1989 Workshop on Parallel Processing: Logic, Organization, and Technology—WOPPLOT 89 Wildbad Kreuth, Germany, July 24–28, 1989 Proceedings*, pages 398–406. Springer, 2005.

Hugues Bersini. In search of a good evolution-optimization crossover. In *2nd Conf. on Parallel Problem Solving from Nature, Brussels, Belgium*, pages 479–488, 1992.

Hiroshi Furuya. Genetic algorithms for placing actuators on space structures. In *Proceedings of the 5th International Conference on Genetic Algorithms*, pages 536–542, 1993.

Agoston E Eiben, Paul-Erik Raué, and Zsófia Ruttkay. Genetic algorithms with multi-parent recombination. In *PPSN*, volume 94, pages 78–87, 1994.

Milan Jelisavcic, Kyrre Glette, Evert Haasdijk, and A. E. Eiben. Lamarckian Evolution of Simulated Modular Robots. *Frontiers in Robotics and AI*, 6(February):1–15, 2019. ISSN 2296-9144. doi:10.3389/frobt.2019.00009.

Massimiliano Schembri, Marco Mirolli, and Gianluca Baldassarre. Evolution and learning in an intrinsically motivated reinforcement learning robot. In *Proceedings of the 9th European Conference on Advances in Artificial Life*, volume 4648, pages 294–303, 09 2007. ISBN 978-3-540-74912-7. doi:10.1007/978-3-540-74913-4_30.

Else Line Ruud, Eivind Samuelsen, and Kyrre Glette. Memetic robot control evolution and adaption to reality. *2016 IEEE Symposium Series on Computational Intelligence, SSCI 2016*, (December), 2017. doi:10.1109/SSCI.2016.7850169.

Kevin Sebastian Luck, Heni Ben Amor, and Roberto Calandra. Data-efficient Co-Adaptation of Morphology and Behaviour with Deep Reinforcement Learning. *http://arxiv.org/abs/1911.06832*, (November), 2019.

Gongjin Lan, Jakub M. Tomczak, Diederik M. Roijers, and A. E. Eiben. Time Efficiency in Optimization with a Bayesian-Evolutionary Algorithm. In *Swarm and Evolutionary Computation*, page 100970, 2 2020. doi:10.1016/j.swevo.2021.100970. URL `http://arxiv.org/abs/2005.04166`.

Charles Schaff, David Yunis, Ayan Chakrabarti, and Matthew R. Walter. Jointly learning to construct and control agents using deep reinforcement learning. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 9798–9805, 2019. ISSN 10504729. doi:10.1109/ICRA.2019.8793537.

Léni K. Le Goff, Edgar Buchanan, Emma Hart, Agoston E. Eiben, Wei Li, Matteo de Carlo, Matthew F. Hale, Mike Angus, Robert Woolley, Jon Timmis, Alan Winfield, and Andrew M. Tyrrell. Sample and time efficient policy learning with CMA-ES and Bayesian Optimisation. In *The 2020 Conference on Artificial Life*, number January, page 2020, 2020. doi:10.1162/isal_a_00299.

Fuda van Diggelen, E. Ferrante, and A. E. Eiben. Comparing lifetime learning methods for morphologically evolving robots. In *GECCO '21: Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 93–94, 07 2021. doi:10.1145/3449726.3459530.

Jørgen Nordmoen, Frank Veenstra, Kai Olav Ellefsen, and Kyrre Glette. Map-elites enables powerful stepping stones and diversity for modular robotics. *Frontiers in Robotics and AI*, 8, 2021. ISSN 2296-9144. doi:10.3389/frobt.2021.639173. URL `https://www.frontiersin.org/article/10.3389/frobt.2021.639173`.

Jakub M. Tomczak, Ewelina Weglarz-Tomczak, and Agoston E. Eiben. Differential Evolution with Reversible Linear Transformations. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion*, pages 205–206, 2020. ISBN 9781450371278. doi:10.1145/3377929.3389972.

Gilbert Feng, Hongbo Zhang, Zhongyu Li, Xue Bin Peng, Bhuvan Basireddy, Linzhu Yue, Zhitao Song, Lizhi Yang, Yunhui Liu, Koushil Sreenath, et al. Genloco: Generalized locomotion controllers for quadrupedal robots. *arXiv preprint arXiv:2209.05309*, 2022.

Laura Smith, J Chase Kew, Xue Bin Peng, Sehoon Ha, Jie Tan, and Sergey Levine. Legged robots that keep on learning: Fine-tuning locomotion policies in the real world. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 1593–1599. IEEE, 2022.

Giles Mayley. Landscapes, learning costs, and genetic assimilation. *Evolutionary Computation*, 4(3):213–234, 1996.

Joshua Auerbach, Deniz Aydin, Andrea Maesani, Przemyslaw Kornatowski, Titus Cieslewski, Grégoire Heitz, Pradeep Fernando, Ilya Loshchilov, Ludovic Daler, and Dario Floreano. Robogen: Robot generation through artificial evolution. In *ALIFE 14: The Fourteenth International Conference on the Synthesis and Simulation of Living Systems*, pages 136–137. MIT Press, 2014b.

Kenneth O Stanley. Compositional pattern producing networks: A novel abstraction of development. *Genetic programming and evolvable machines*, 8(2):131–162, 2007b.

Fuda Van Diggelen, Eliseo Ferrante, and AE Eiben. Comparing lifetime learning methods for morphologically evolving robots. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 93–94, 2021.

Rainer Martin Storn. Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces. In *Journal of Global Optimization*, pages 131–141, 1997. doi:10.1023/A.

Meh Pedersen. Good Parameters for Differential Evolution. *Evolution*, pages 1–10, 2010. URL `http://www.hvass-labs.org/people/magnus/publications/pedersen10good-de.pdf`.

Thomas Bartz-Beielstein and Mike Preuss. Experimental research in evolutionary computation. In *Proceedings of the 9th annual conference companion on genetic and evolutionary computation*, pages 3001–3020, 2007.

Elte Hupkes, Milan Jelisavcic, and A. E. Eiben. Revolve: A Versatile Simulator for Online Robot Evolution. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 10784 LNCS:687–702, 2018. ISSN 16113349. doi:10.1007/978-3-319-77538-8_46.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 5026–5033. IEEE, 2012.