ROYAL SOCIETY
OPEN SCIENCE

rsos.royalsocietypublishing.org

Research

Article submitted to journal

Subject Areas:
Artificial Evolution, Robotics

Keywords:
evolution, individual learning, cultural learning

Author for correspondence:
Emma Hart
e-mail: e.hart@napier.ac.uk

Artificial evolution of robot bodies and control: on the interaction between evolution, individual and cultural learning

Emma Hart, Léni K. Le Goff

Edinburgh Napier University

We survey and reflect on evolutionary approaches to the joint optimisation of the body and control of a robot, in scenarios where a the goal is to find a design that maximises performance on a specified task. The review is grounded in a general framework for evolution which permits the interaction of evolution acting on a population with individual and cultural learning mechanisms. We discuss examples of variations of the general scheme of "evolution plus learning" from a broad range of robotic systems, and reflect on how the interaction of the two paradigms influences diversity, performance, and rate of improvement. Finally, we suggest a number of avenues for future work as result of the insights that arise from the review.

THE ROYAL SOCIETY
PUBLISHING

1. Introduction

Given the abundance of intelligent lifeforms existing on Earth arising as a product of evolution, it is unsurprising that the basic concepts of evolution have proved appealing to scientists for as long as they have had access to digital technologies. The fundamental ideas of evolution — that of population, variation, selection — have been "borrowed" to solve problems in diverse fields dating as far back as the 1950. For example, Turing described an evolutionary inspired method to solve control problems [1]; in the 1960s, a group of engineers in Germany invented the first 'Evolution Strategy'[1] in order to optimise the design parameters of a physical hinged plate and later of a dual-phase gas nozzle [2], while Holland's seminal book on genetic algorithms [3] was first published in 1975.

The 1990s onward saw a huge explosion in both the development of improved evolutionary algorithms (EAs), in the domains to which they were applied, and in theoretical understanding of their processes. EAs found success in solving many hard combinatorial optimisation problems in domains such as logistics and scheduling, as well as in continuous optimisation domains such as function or parameter optimisation. A natural follow-on from this was to apply artificial evolution to the field of robotics and Artificial Life (Alife) in order to evolve new designs and controllers, giving rise to the now vibrant field of Evolutionary Robotics. The field encompasses the use of artificial evolutionary algorithms to design morphology [4], control [5] or a combination of the two for both simulated and physical robots [6–9]. Furthermore, it has progressed to cover evolution with a wide range of materials, ranging from soft, flexible materials that can be easily deformed under pressure to provide adaptation [10], through modular approaches that evolve configurations of pre-existing components [11] to approaches that exploit the ability to rapidly print hard-plastic components in a diverse array of forms [12].

Embodied Artificial Evolution The move to applying EAs in robotics introduces an important conceptual shift with respect to previous work that applied EAs in the domains of combinatorial optimisation or function optimisation. There is a fundamental difference between evaluating a genome that encodes a string of numbers that (say) represent a solution to a function optimisation problem and evaluating a genome where the string of numbers represents the control parameters of a robot: in the latter case, the controller exists in a physical body that interacts with the environment[2]. This notion is captured by the now popular term "embodied intelligence", which describes the design and behaviours of physical objects situated in the real-world and was first introduced by Brooks in 1991 [13]. Pfiefer and Bongard's seminal text 'How the body shapes the way we think' [14] expanded on the idea that intelligent control is not only dependent on brain, but at the same time both constrained and enabled by the body. The text shows in great detail how intelligence might arise as a result of the interplay of morphology, materials, interaction with the environment, and control.

For the field of artificial evolution and robotics, Pfiefer and Bongard's work strongly suggests that it is insufficient for the computer-scientist to focus on only using evolution as tool to evolve neural control mechanisms for robots with fixed morphology — they must also consider morphological computation and study how artificial evolution might be used to realise this. *Morphological computation* refers to the fact that judicious use of material and/or design of body parts can enable certain processes to be performed by the body that otherwise would have to be performed by the brain [14]. Pfiefer and Bongard provide numerous examples from nature. For example consider a human leg: as muscles and tendons are elastic, the system is able to undertake small adaptive movements due to forces arising from impact with the ground, without any need for neural control. Another example can be found in leg coordination in insect walking: angle sensors in the joints provide global communication between the legs

---

[1] the modern version of this methods CMA-ES (Co-matrix adaption Evolution Strategy is to date one of the most commonly used optimisation methods in the domain of continuous optimisation

[2] the same is true whether in simulation or physical robots given that most modern simulations use sophisticated and realistic physics-engines to model environmental interactions

providing coordinated movement as a direct result of forces generated through interaction with the environment. Communication is mediated only by the environment, and not through any neural connections. Even if neural control is required, judicious placement of sensors on the body can provide correlated information that considerably simplifies neural control (the touch-sensors on the finger-tips that generate information when a hand closes on an object are an excellent example of this, described in detail in [14]

Co-evolution of morphology and control For the Evolutionary roboticist, this clearly points to the fact that to properly exploit the power of evolution to discover appropriate robots for a given task, one must consider the co-evolution of body and control, rather than simply use EAs to search for controllers for fixed (hand-designed) body-plans [15]. This has the obvious advantage of allowing evolution to discover for itself the appropriate balance between morphological and neural complexity in response to the particular environment and task under consideration.

In order to develop a system that can achieve this in practice, evolution should act on a population whose individuals have a genome that encodes the information required to generate both morphology and control. However, this notion raises a challenge for artificial evolutionary methods in the design of the genome encoding. Assuming no prior knowledge of what type of robot is required for a task, then a single population should contain a diverse set of genotypes that encode for morphologically diverse body-plans. Here, an artificial evolutionary system significantly diverges from its biological counterpart: in the latter, evolution takes places within distinct populations consisting of a single species. In contrast, in the artificial case, we essentially have multiple species inter-breeding within a single population: that is, if a species is defined by a particular set of morphological features, then we can have (for example) a single population that contains wheeled land-based robots, swimming robots and flying robots. A result of this is that offspring might bear little similarity to either parent. In such cases, an inherited neural controller — with inputs corresponding to sensory apparatus and outputs to actuator control — is likely to be incompatible with the new body [16,17]. The problem is exacerbated if the morphological space to which evolution is applied is rich.

One way of addressing the mis-match issue is to use an encoding for the controller that is *morphology-independent*, i.e. one in which the genome specifies a mechanism for *generating* the controller, rather than directly specifying it. These designs are referred to as generative encodings. However, even the controllers inherited via a morphology-independent mechanism are likely to require fine-tuning to specialise the controller to the nuances of the new body. This motivates the requirement for *individual-learning*.

If it evolves it needs to learn Eiben *et al* proposed an evolutionary framework for robotic evolution called the Triangle of Life [18]. The three sides of the triangle represent three phases: (1) morphogenesis in which are robot is created from a genome; (2) learning in which the new robot undergoes training to improve its inherited controller, perhaps following a syllabus of increasingly complex skills in a restricted or simplified environment; (3) life, i.e the robot is released into the intended real environment and its fitness is measured as its ability to accomplish a task.

Eiben and Hart [19] further expand on the role of the learning phase, arguing that it is in fact essential, particularly if evolving directly in hardware. In the extreme case where there is a complete mis-match between body and control between parents and offspring, the individual learning mechanism might be applied *tabula rasa*, i.e. learned from scratch [9]. On the other hand, using any representation that permits inheritance of either a suitable[3] controller or mechanism for generating a controller, the learning process can act as a form of adaptation over the course of an individual's lifetime. Having expended computational effort on applying a learning mechanism, one can then consider whether a Lamarkian [20] system might be used, in which the learned

---

[3] in a neural network controller for example, the term *suitable* would imply that the network has inputs and outputs that match the sensors and actuators on the offspring's new body-plan

controller (or controller generation mechanism) is written back to the genome. On the other hand, one can also study whether individual learning gives rise to a Baldwin effect [9], i.e. in analysing whether there is implicit selection for morphologies that are able to learn quickly.

Cultural Learning An alternative form of learning that might be considered in the context of Evolutionary Robotics is that of cultural learning [21]. First introduced in artificial evolution to improve problem-solving in the function optimisation domain [22], a cultural algorithm introduces a belief-space that captures useful knowledge learned across generations. An evolving population can interact with belief-space, influencing population based evolution. In turn, the belief space is continually updated as new knowledge is generated by the evolutionary process acting on the population [22]. This can 'bootstrap' individual learning processes by exploiting previously discovered controllers for example [17]. It should also be noted that an additional form of learning — that of *social* learning can also be applied in the context of ER but typically requires the use of a swarm in which one robot can learn from direct interaction with another robot in the swarm [23,24]. Here we restrict our discussion to the subset of evolutionary robotics which does not rely on a swarm, i.e. is focused towards emergence of a single robot appropriate for a specified task.

Overview of remainder of article The remainder of the paper provides a survey of the field of Evolutionary Robotics with respect to three axes identified above (i.e., evolution, individual learning and cultural learning). The review is restricted to evolutionary joint optimisation of body and controllers in the context of a particular task with a defined objective function that should be maximised. That is, we do no consider *open-ended* evolutionary systems in which there is no specific objective other than to survive and reproduce in an environment.

It first considers methods which can be used for the *joint-optimisation of body and control*, assessing their pros and cons. It then considers the role of *individual learning mechanisms* in. improving newly generated offspring, and the influence of the learning process on the main evolutionary process. Finally we evaluate the role of an additional form of learning that can loosely be described as *cultural learning*, that enables the learning process to exploit a structured knowledge-store that captures historical knowledge from multiple generations.

2. Frameworks for Joint Evolution of Body and Control

Figure 1 presents a general framework for evolution and learning that guides our discussion. It depicts four possible architectures in which robot bodies and controllers can be jointly optimised via processes which use either only evolutionary mechanisms (i.e.at the population level, fig. 1 **A**), those that augment evolution at the population level with individual learning (fig. 1 **B,C**), and finally those that also add cultural learning (fig. 1 **D**). The general framework encapsulates a dual-loop, in which evolution and learning processes can interact in various ways, with the relative balance in terms of computational effort between the two loops introducing an additional variable into the process. The four mechanisms are considered in turn below, followed by a general discussion.

(a) Joint optimisation via evolution only

We first discuss the process denoted **A** in figure 1 in which robots are designed only via an evolutionary process that acts on a population of genomes that define both body-plans and controllers, i.e there is a single evolutionary loop with no learning mechanism. This obviously necessitates that both morphology and controller of offspring *must* both be inherited from the parent robots. As we noted above, this introduces a constraint on the representation of the controller on the genome in that it must be capable of producing a controller that matches the new body, in terms of capturing input from sensors and outputs to the actuators. As mentioned, this

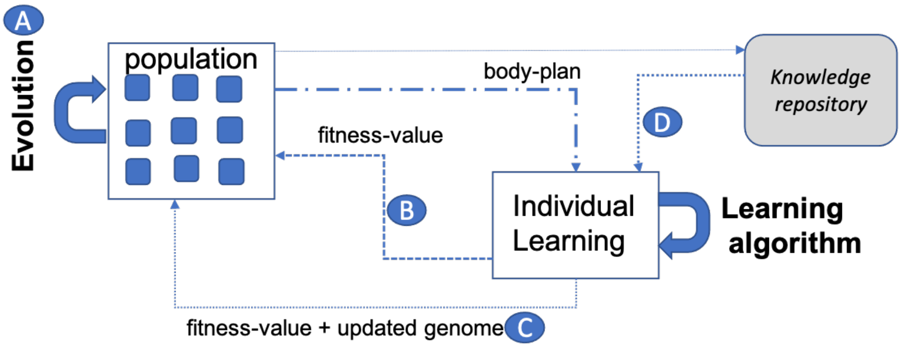

Figure 1. Four different processes for joint optimisation of body and control are shown.**A**: Robots are designed via an evolutionary process that acts on a population of genomes that define body-plans and controllers. **B**: Each new robot generated via evolution also refines its controller via individual learning over a lifetime: learning is either *tabula-rasa* or starts from the inherited controller. The evaluated fitness on a task/environment is returned to the evolutionary process **C**: as in B, however the learned controller is written back to the genome, i.e. follows a Lamarkian process. **D**: the learning algorithm is initialised from a knowledge repository that contains good controllers learned in previous generations, i.e represents a form of cultural evolution. The repository is filled with from information encoded in the population

.

often requires the use of generative representations that construct a controller from instructions encoded on the genome, rather than directly specify the control parameters.

Working in the field of modular robotics, Veenstra et al [25] evolve a blue-print that specifies both the body and controller of a modular robot, i.e. one that is built from a library of 'modules' that can connect together at multiple sites on each module. Their blue-print takes the form of a single directed tree that defines both body and control, therefore encapsulating its tightly-coupled nature. Each module contains its own controller in the form of a parameterised sine-wave function that controls its motors. Starting from a root node, each subsequent node of the tree specifies the type of module to be attached to the parent module, and the location it should be attached. As the control system is directly encapsulated within each module, the representation guarantees an appropriate control system.

They investigate multiple ways in which the tree can be encoded on the genome: (1) a direct encoding of the tree itself in which mutations can add/delete nodes or change the parameters of a node controller; (2) a parameterised L-system [26] which evolves a set of production rules specifying how a tree can be built; a neuro-evolution approach which evolves a neural network (a Compositional pattern producing network referred to as a CPPN [27], see below) that outputs values determining which (if any) module should be connected to each potential site on existing modules; (4) a cellular-encoding [28] which also produces a CPPN that is used to generate trees. In all cases, the controller parameters encapsulated with an module can be altered via mutation operators. Note that a CPPN is a special class of neural networks that makes use of a set of activation functions that output regular patterns. They were originally proposed as an abstraction of gene expression and embryonic development and have found particular favour with the evolutionary robotics community in terms of evolving morphology for their ability to produce repeated patterns within a body plan, for example, multiple limbs with identical form. Veenstra *et al* find that the direct encoding of the tree and the L-system produce the highest performing robots given a fixed evaluation budget, while the CPPN approach maintains the highest diversity within a population.

The approach of using a single encoding to represent both body and control is common in modular robot designs that use wave-like controllers to control servo-motors embedded in the components that are used to build the robot. For example, Brodbeck *et al* evolve robots that are constructed entirely in hardware with no simulation, using a 'mother robot' to automatically assemble offspring from a set of cubic active and passive modules. The genome contains n genes, one per module required. Each gene contains information about the module type to be used (active or passive), construction parameters and finally two parameters that specify the motor control of the module (the phase and amplitude of a sinusoidal controller). It is important to note however that the modular designs just discussed do not have sensing capabilities. Control is simplified in this situation as it is reduced to actuation of each module, without having to coordinate sensed observations from the environment with actuation.

While the above approaches uses the same encoding to specify both morphology and control, other authors separate the specification of each component into two separate encodings, which are represented on the same genome. Although this separation in some sense undermines the prevailing view that body and control are tightly coupled, the explicit separation isolates each component and therefore allows enables tailored changes to be made to either component. For example, Stensby *et al* [29] study a restricted morphological space in which the length of the joints of a bi-pedal walker can be adapted by evolution. In this case, some aspects of the morphology remain constant: each walker has the same number of limbs, connected in an identical manner. However, evolution can vary the length of each individual limb, thus introducing morphological change. One section of the genome encodes a vector specifying the width and height of each joint. A second vector encodes the weights of a fixed-size neural network that outputs the force to apply to the leg joints, thus forming a type of closed-loop control architecture. Specifically designed mutation-operators operate on each vector. Restricting the morphological adaptation to changing limb length considerably simplifies the task of representing controllers. The same fixed topology neural network controller can be applied to every new body-plan created: the task of evolution is therefore only to evolve the weights of the controller and inheritance is therefore straightforward.

Cheney *et al* consider joint optimisation in the context of evolving soft-robots composed of regular cubic-voxels that can be realised with either active material (i.e. analogous to muscles) or passive materials (analogous to tissue) in a line of work described in [30,31]. They opt for a dual-representation of body and controller, using two CPPNs encoded on the same genome. One CPPN provides the outputs required to define the morphology of a robot, while the second describes the controller, producing outputs that determine the actuation of each voxel.

In [30] the authors first demonstrated a now well-known problem that joint optimisation approaches can tend to lead to premature convergence of the morphology, and therefore restrict performance: this occurs as changes to morphology arising through evolution tend to lead to robots with controllers that are in the worst case unable to be applied to the new morphology, or at best, provide poor performance. In order to address this issue, they introduce the notion of 'morphological innovation protection' — the goal is to allow time for a controller to adapt to a new body via evolutionary mutations by temporarily reducing selection pressure on individuals that have new morphological features introduced via mutation. Their method removes robots from the population if they are 'dominated' by another robot with respect to two variables: their age and their fitness. This has the effect of protecting robots with poorly adapted controllers from being removed, allowing their controllers time to adapt to the new morphology. Their preliminary results show the method both prevents premature convergence of body-plans and results in more efficient robots with respect to the tasks that they are evolved for.

While all of the methods above are shown to be capable of evolving robots that exhibit high performance with respect to the tasks they are evolved for, it is clear that applying a learning mechanism over the lifetime of an individual has the potential to improve an inherited controller. Methods to implement individual learning are described in the next section.

(b) Joint Optimisation with Individual Learning

In these schemes, each time a new robot is generated via evolution it is given an opportunity to refine its controller by running an individual learning algorithm: this enables it to either improve its inherited controller over some fixed period of time, or in some cases, to learn a controller from scratch. In both cases, the learned fitness is associated with the robot at the end of learning process and used by the evolutionary process to drive selection (figure 1 B). Furthermore, depending on the representation used to encode the controller, the learned controller can be written back to the genome before further evolution takes place, i.e. a Larmarkian process can be followed (1 C). Examples of these approaches are now discussed.

(i) Individual Learning applied to inherited controllers

Jelasavic *et al* [20] tackle the problem of co-optimisation using a modular robotic framework in which the body-plan consists of an arrangement of pre-designed modules, and control is realised via a central pattern generator (CPG): first introduced by Ijspeert [32], a CPG is a form of neural circuitry that outputs cyclic patterns typically found in vertebrates. From the robotics perspective, such controllers are attractive in that they have few parameters and provide smooth control transitions. As in Cheney *et al*, the genome contains two separate components for encoding morphology and control. A tree representation (as in [25]) is used to encode morphology. The controller is encoded by a function that *generates* the weights of a CPG. The function that generates the CPG weights is a Compositional Pattern Producing Network (CPPN, as described above) that is itself evolved: the network is queried with the position of a source and target module, and outputs the weight of the connection from source to target and the bias and gain values required by the CPG. As the network can be queried with as many input/outputs pairs as required, it is completely morphology-agnostic.

To realise this system, the 'controller' part of the genome in fact contains a *population* of CPPNs: when an offspring is created, it inherits 50% of the CPPNs encoded on each parent. The learning algorithm (in this case an evolutionary algorithm) is then applied for a fixed number of iterations to improve this population. At the end of this process, the learned fitness assigned to the robot is that achieved by the best performing CPPN. Note that although the learning mechanism used in this case is evolutionary, in fact any suitable learner could be applied. This approach allows for both Darwinian and Lamarkian evolution. In the former case, the CPPNs inherited by offspring from their parents are those that the parents were born with. In the latter case, the inherited population is overwritten at the end of the learning process with the improved population of CPPNs. Their results show perhaps unsurprisingly that the Lamarkian approach considerably reduces the time required to learn, and that is particularly important when the available learning budget is small. More interestingly, they observe that most benefit is observed when there is high similarity between the *morphological* properties of the two parents: in this case, the CPPNs of each parents are already well-adapted to the specific morphology of the parents and therefore provide a suitable springboard for further adaptation.

A similar study is conducted by Miras *et al* [33]. Using a similar set up to that described above, they studied whether applying a learning mechanism to tune an inherited controller (without Larmarkianism) influenced the resulting performance and morphological properties of the evolved robots. Here they find that using the individual learning mechanism not only results in robots that perform better on the chosen task, but that it also produces significantly *larger* robots than those evolved without applying learning. Although the number of limbs in robots produced with/without learning was similar in each case, the number of modules was higher in the former case. Furthermore, they observe that the *learning delta* — the difference between the fitness of the inherited controller and that of the learned controller — increases over time. Thus, the system illustrates a Baldwin effect, i.e. that evolution selects for robots that are more capable of learning over time.

(ii) Individual Learning *tabula rasa*

The methods just described make use of indirect encodings of the controller (usually CPPNs) in order to address the problem of the mis-match between the structure of an inherited controller and the potentially modified morphology of a child robot. This elegantly solves the mis-match issue as the the CPPN can *generate* however many weights are required for an appropriate controller.

The method has been shown to have multiple advantages: as it generates the pattern of weights as a function of the geometry of the inputs and outputs of the domain it is tightly coupled to placement of actuators and sensors in the morphology; it can produce regular connectivity patterns which facilitate the emergence of movement such as walking gaits or swimming; it enables very large neural networks to be evolved through very compact encodings (analagous to DNA) [27]. However, on the downside, due to the indirect encoding, the fitness landscape can be discontinuous — small changes in the genotype can lead to major changes in the phenotype (and in fact the opposite, i.e. a major genotypic change can lead to a small phenotypic change). Several authors have also noted that the evolution progress using an indirect encoding is slower than that of a direct one, something that is of particularly concern if one is hoping to conduct fitness evaluations in hardware (see [34] for a detailed discussion on this).

To address these concerns, an alternative approach to individual learning is simply to create a neural controller with a suitable structure (i.e. appropriate inputs and outputs) once the new morphology of a child has been decoded from the genome, and then learn its weights from scratch using an individual learning algorithm that operates *directly* on a vector that explicitly represents the controller weights. This (a) removes the need to design a morphology-agnostic controller representation and (b) has the advantage that many types of learning algorithm can be applied to learn the controller (for example, reinforcement learning, evolution, stochastic gradient descent).

For example, Gupta et al [9] use a reinforcement learning algorithm to optimise the controllers of robots whose morphologies are evolved using an evolutionary algorithm. Robots are composed from articulated 3D rigid parts connected via motor actuated hinge joints. Each time a new morphology is produced via evolution, the RL algorithm is applied from scratch to optimise a policy to control the robot. Interestingly, they show that the coupled dynamics of evolution over generations and learning over a individual lifetime leads to evidence of a morphological Baldwin effect, demonstrated experimentally by a rapid reduction in the learning time required to achieve a pre-defined level of fitness over multiple generation. Hence, evolution selects for morphologies that learn faster, enabling behaviours learned by early ancestors to be expressed early in the lifetime of their descendants. They go on to suggest a mechanistic basis for both the Baldwin effect and the emergence of morphological intelligence, based on evidence that the coupled process tends to produce morphologies that are more physically stable and energy efficient, and can therefore facilitate learning and control.

Le Goff *et al* are motivated by the goal of applying artificial evolution and learning to physical populations of robots. Their model [4,8,17] envisages hybrid systems in which two populations of robots can exist simultaneously, one in simulation and another in hardware. Evolution can take place both *within* each population or *across* the two populations, that is, allowing mating between a physical mother and virtual father. Their goal is to find a learning algorithm that is as efficient as possible given that conducting experiments in hardware can be time-consuming. Furthermore, physical robots are subject to the effects of wear and tear that can degrade the robot performance and add noise to learning trials.

They propose an algorithm dubbed NIPES [35] for learning a controller from scratch once a morphology is known. The algorithm is a modification of the state-of-the-art optimiser CMA-ES proposed originally as an optimiser for continuous valued search-spaces. The algorithm is augmented with a novelty mechanism [36] that maintains diversity and is known to be effective particularly on deceptive navigation tasks such as maze-navigation where it is difficult to provide a suitable reward to drive learning. The algorithm also includes an adaptive restart mechanism that gradually increases population size, starting from a small population. Learning terminates if a behaviour that passes a fixed performance threshold is found. This aspect considerably improves efficiency by only increasing population size (and therefore trials required) if strictly necessary. A population of controllers are randomly initialised for each new morphology: NIPES is shown to be both high-performing and efficient, appropriately balancing exploration and exploitation when learning controllers in a morphological space that includes multiple sensors, wheeled actuators and joints. A comparison to another type of learner - a Bayesian Optimiser (BO) - shows that although both methods can achieve similar performance, the computational running time of the NIPES method is significantly shorter than BO. As noted above, this is therefore preferable in the context of physical robotics.

(c) Joint Optimisation with Individual and Cultural Learning

While the previous section has focused on individual learning mechanisms in which an individual robot improves its controller based only on information gathered during its own lifetime, we now turn our attention to mechanisms which enable the individual learning phase depicted in the Triangle of Life to be influenced by knowledge gathered over previous generations, stored in continually updated repository.

Cultural algorithms were first introduced into the field of artificial evolution in 1990s [21,37]. They are based on the notion that in advanced societies, *culture* accumulates in the form of knowledge-repositories that capture information acquired by multiple individuals over years of experience. If a new individual has access to this repository of information, it is able to learn things even when it has not experienced them directly. Cultural algorithms are prevalent in the field of swarm-robotics [23,24] in which robots learn to adapt to complex environments by learning from each other. However, here we restrict the discussion to methods in which the individual learning process of a single robot as depicted on the TOL can be influenced by external repositories of information built up over multiple generations. The repository effectively stores the results of previous individual learning trials that can be drawn upon by future generations.

Le Goff et al. [16,17] describe a method for bootstrapping the individual learning phase by drawing on such a repository, which can be used when learning by default is *tabula rasa* due to the difficulty of inheriting an appropriate controller (e.g. as in [9]). They consider that robots can be categorised by 'type' according to a coarse-grained definition of their morphology. Specifically, *type* is defined by a tuple (sensors, wheels, joints) which denotes the number of each component present in newly produced robots. An external repository stores the single best controller found through individual learning for each potential type. When a new robot is created via the evolutionary cycle, the learning phase is initiated by first selecting a controller from a robot with matching type from the repository (assuming one exists). A learning algorithm then attempts to improve this controller in the context of the new body. Note that the definition of type is deliberately high-level in order to make the process of finding a matching type more straightforward and therefore increase the chance of being able to bootstrap from the repository. It should be clear that robots of the same type may in fact have very different skeletons (and therefore sizes) and different layouts of sensors/actuators, and hence even an inherited controller may not perform well. Although finer-grained definitions of type that accounted for these differences could be introduced, they would reduce the probability of a match (a way to mitigate this would be to store several controllers per type in conjunction with an informed heuristic for selecting among them). The repository is updated whenever an improved controller is found for the type.

Le Goff *et al* demonstrate that the use of the repository results in a significant increase in performance over robots that learn from scratch [16,17]. However, they also observe a similar premature convergence of morphology as reported by Cheney *et al* [31]. Their finding further show that morphologies converge to only 13 different types (from a potential 1024 according to the definition). We return to this aspect in the discussion section.

Finally, although not strictly speaking an evolutionary approach, the work of Liao *et al* ([38]) is also worth mentioning due to the fact that they also employ a dual optimisation process to find the best morphology and controller for a walker micro-robot, optimising morphology and then control in separate loops making use of information learned across generations. Here a Bayesian Optimiser (BO) is used to learn a controller for each new morphology, taking advantage of previous information to initialise the optimiser, hence aligning with scheme D in figure 1. They use a form of BO known as contextual Bayesian optimization (cBO) [39] to optimise a Gaussian

Process (GP) model to define the controller policy. By encoding the morphologies as contexts, cBO takes advantage of the similarities between different morphologies and is therefore able to generalise to good polices for unseen designs faster. In essence, rather than learn a new GP model for each new robot evaluated, the models are shared such that learning for a robot b can start from a previously learned model for robot a if they have similar morphology. Conceptually therefore, this is very similar to the approach from Le Goff just described. However in this case, the repository consists of GP models rather than controller weights.

3. Discussion and future directions

We have described a number of ways in which artificial evolution can be used to jointly evolve the morphology and controller of robots. These processes are described in the context of a general framework that permits evolution and learning (both individual and cultural). A key aspect of the framework is the nature of the dual-loops of evolution and learning, and the manner in which they interact with each other. The pros and cons of this approach are discussed below.

Firstly, we note that the framework design enables the computational budget allocated to an experiment to be divided between evolution/learning in different ways depending on the situation. For example, if one wants to design physical robots in which a robot must be manufactured from the blueprint specified on its genome, then it should be clear that there is significant cost (in terms of time and material) associated with the production of the robot itself: for example, Hale *et al* [8,12] use 3D printing to produce robot skeletons followed by an automated process to attach wheels and sensors, a process which can take up to 6 hours per robot. An *evolutionary* trial of a new robot is therefore costly. On the other hand, once a robot is built, a *learning* trial is cheap. It might therefore be desirable to limit the number of cycles of evolution to reduce the number of robots that need to built in favour of increasing learning trials.

Secondly, we have presented work from several authors [9,33] that shows that adding an individual learning loop to the evolutionary cycle demonstrates the Baldwin effect, in that evolution rapidly selects for morphologies that learn quickly. A mechanistic basis is provided in some detail by Gupta [9]. However, we have also highlighted that the joint optimisation can lead to morphological stagnation, in that the population quickly converges to a fixed morphology [17,40,41]. As suggested by Cheney [41], an approach to tackling this is to ensure that new offspring that have accumulated significant morphological mutations are given time to re-adapt their controllers. This can be achieved by via the method of morphological innovation protection as proposed by Cheney; alternatively, in the context of the framework described in figure 1 this can be achieved by focusing effort on the learning loop and enabling sufficient time to adapt. Maintaining diversity within a population is of course generically challenging for most evolutionary dynamics. Gupta [9] proposes an alternative suggestion to address this by using an asynchronous steady-state evolution: rather than using the generational approach typical in most evolutionary robotics literature (in which the offspring population entirely replaces the parent population), in the asynchronous approach, small tournament based competitions are held within a population, where offspring only compete within the tournament, with winner directly replacing an older member of the existing population. Their results show that this method is able to deliver a highly diverse array of robots with respect to their morphological characteristics. The mini-tournaments referred to above play a part in delivering this diversity, helped also by the survival operator which is based on the age of the robot. This survival operator can be compared to the morphological innovation protection operator proposed by Cheney described earlier. Clearly there is further work to be done to address this issue.

This article has focused on methods for joint optimisation, covering examples in a variety of morphological spaces (e.g. modular robotics, soft robotics, 3d-printed skeletons with varied sensors and actuators) and types of controller (e.g. neural networks, central pattern generators or wave-forms). However we have largely ignored the role that the environment can play in influencing the evolution-learning cycle. This clearly adds an additional dimension in that is well recognised that environmental complexity fosters the evolution of morphological intelligence

[14]. Hence although we have described a general framework that elucidates the mechanisms by which evolution and learning can interact, the environment in which this occurs is crucial. Gupta [9] has shown that exposing robots to a variety of physical environments (e.g. varying the friction or angle of the surface over which the robots move) and the task (the objective to be optimised) during evolution with learning leads to robots that learn faster. This is also a fruitful avenue for future work to gain further insights in this direction.

An additional avenue for future work concerns the representation used to define the blueprint the specifies morphology and control. This can be considered along two dimensions: (1) whether a single encoding can specify both aspects or whether it is separated into two distinct parts, one for morphology and one for control and (2) whether a direct or generative encoding is used. With respect to the former, while a single encoding explicitly captures the notion of embodied intelligence results from the tight interconnection between morphology and control, separating the two provides a means for more fine-grained control in an artificial setting. In terms of representation, as described above, generative representations that construct either body and/or controller are inherently flexible and bypass issues associated with ensuring controllers have the appropriate form. On the other hand, when mapping from genotype to phenotype, small changes in genotype can lead to large changes in phenotype (and vice versa), resulting in a complex landscape for evolution to navigate.

Finally, we have restricted the discussion to the joint optimisation of morphology and control to realise the design of a robot optimised with respect to a specific goal. It should be noted that within the field of evolutionary robotics, there are other strands of study that are closely related. In the field of swarm-robotics, Thenius *et al* [23] propose a form of learning that they call 'local cultural adaptation' in which an individual robot exchanges information with those in a local sub-group in order to improve its control settings. Cully *et al* [42] have considered the role of learning in the context of adapting controllers in response to morphological change that occurs *during* a lifetime, while Walker *et al* [43] study a form of learning in which the *morphology* of a soft-robot can deliberately adapt during an individual lifetime (for example, growing an additional 'body-part'), mediated by environmental signals. The take-away message is that knowledge in a variety of forms obtained by multiple forms of learning mechanisms clearly plays a key role in augmenting evolution in the context of evolutionary robotics where robots have both form and behaviour, with many fruitful lines of research remaining open.

References

1. Turing A. 1950 Computing machinery and intelligence. *Mind* **59**, 433.
2. Rechenberg I. 1973 *Evolutionsstrategie – Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. PhD thesis Technical University of Berlin Reprinted by Frommann-Holzboog.
3. Holland JH et al.. 1992 *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press.
4. Buchanan E, Le Goff LK, Li W, Hart E, Eiben AE, De Carlo M, Winfield AF, Hale MF, Woolley R, Angus M et al.. 2020 Bootstrapping Artificial Evolution to Design Robots for Autonomous Fabrication. *Robotics* **9**, 106.
5. Doncieux S, Bredeche N, Mouret JB, Eiben AEG. 2015 Evolutionary robotics: what, why, and where to. *Frontiers in Robotics and AI* **2**, 4.
6. Lipson H, Pollack JB. 2000 Automatic design and manufacture of robotic lifeforms. *Nature* **406**, 974–978.
7. Brodbeck L, Hauser S, Iida F. 2015 Morphological evolution of physical robots through model-free phenotype development. *PloS one* **10**, e0128444.
8. Hale MF, Buchanan E, Winfield AF, Timmis J, Hart E, Eiben AE, Angus M, Veenstra F, Li W, Woolley R et al.. 2019 The ARE Robot Fabricator: How to (Re) produce Robots that Can Evolve in the Real World. In *The 2019 Conference on Artificial Life: A Hybrid of the European Conference on Artificial Life (ECAL) and the International Conference on the Synthesis and Simulation of Living Systems (ALIFE)* pp. 95–102. MIT Press.

9. Gupta A, Savarese S, Ganguli S, Fei-Fei L. 2021 Embodied Intelligence via Learning and Evolution. *arXiv preprint arXiv:2102.02202*.
10. Hiller J, Lipson H. 2011 Automatic design and manufacture of soft robots. *IEEE Transactions on Robotics* **28**, 457–466.
11. Jelisavcic M, De Carlo M, Hupkes E, Eustratiadis P, Orlowski J, Haasdijk E, Auerbach JE, Eiben AE. 2017 Real-world evolution of robot morphologies: A proof of concept. *Artificial life* **23**, 206–235.
12. Hale MF, Angus M, Buchanan E, Li W, Woolley R, Le Goff LK, De Carlo M, Timmis J, Winfield AF, Hart E et al.. 2020 Hardware design for autonomous robot evolution. In *2020 IEEE Symposium Series on Computational Intelligence (SSCI)* pp. 2140–2147. IEEE.
13. Brooks RA. 1991 Intelligence without Reason. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence - Volume 1* IJCAI'91 p. 569–595 San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
14. Pfeifer R, Bongard J. 2006 *How the body shapes the way we think: a new view of intelligence*. MIT press.
15. Bongard J. 2011 Morphological change in machines accelerates the evolution of robust behavior. *Proceedings of the National Academy of Sciences* **108**, 1234–1239.
16. Le Goff L, Hart E. 2021 On the challenges of jointly optimising robot morphology and control using a hierarchical optimisation scheme. In *2021 Genetic and Evolutionary Computation Conference Companion*. ACM.
17. Le Goff LK, Buchanan E, Hart E, Eiben AE, Li W, Carlo MD, Winfield AF, Hale MF, Woolley R, Angus M, Timmis J, Tyrrell AM. 2021 Morpho-evolution with learning using a controller archive as an inheritance mechanism. *arXiv:2104.04269*.
18. Eiben A, Bredeche N, Hoogendoorn M, Stradner J, Timmis J, Tyrrell A, Winfield A. 2013 The triangle of life: Evolving robots in real-time and real-space. In Lio P, Miglino O, Nicosia G, Nolfi S, Pavone M, editors, *Proc. of the 12th European Conference on the Synthesis and Simulation of Living Systems (ECAL 2013)* pp. 1056–1063. MIT Press.
19. Eiben A, Hart E. 2020 If it evolves it needs to learn. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion* pp. 1383–1384.
20. Jelisavcic M, Glette K, Haasdijk E, Eiben A. 2019 Lamarckian evolution of simulated modular robots. *Frontiers in Robotics and AI* **6**, 9.
21. Reynolds RG. 1999 Cultural algorithms: Theory and applications. In *New ideas in optimization* pp. 367–378.
22. Coello Coello CA, Becerra RL. 2004 Efficient evolutionary optimization through the use of a cultural algorithm. *Engineering Optimization* **36**, 219–236.
23. Thenius R, Moser D, Varughese JC, Kernbach S, Kuksin I, Kernbach O, Kuksina E, Mišković N, Bogdan S, Petrović T et al.. 2016 subCULTron-cultural development as a tool in underwater robotics. In *Artificial Life and Intelligent Agents Symposium* pp. 27–41. Springer.
24. Ali MZ, Daoud MI, Alazrai R, Reynolds RG. 2020 Evolving Emergent Team Strategies in Robotic Soccer using Enhanced Cultural Algorithms. *Cultural Algorithms: Tools to Model Complex Dynamic Social Systems* pp. 119–142.
25. Veenstra F, Glette K. 2020 How Different Encodings Affect Performance and Diversification when Evolving the Morphology and Control of 2D Virtual Creatures. In *Artificial Life Conference Proceedings* pp. 592–601. MIT Press.
26. Lindenmayer A, Jürgensen H. 1992 Grammars of development: discrete-state models for growth, differentiation, and gene expression in modular organisms. In *Lindenmayer Systems* pp. 3–21. Springer.
27. Stanley KO. 2007 Compositional pattern producing networks: A novel abstraction of development. *Genetic programming and evolvable machines* **8**, 131–162.
28. Gruau F. 1994 Neural network synthesis using cellular encoding and the genetic algorithm. .
29. Stensby EH, Ellefsen KO, Glette K. 2021 Co-optimising Robot Morphology and Controller in a Simulated Open-Ended Environment. In Castillo PA, Jiménez Laredo JL, editors, *Applications of Evolutionary Computation* pp. 34–49 Cham. Springer International Publishing.

30. Lipson H, Sunspiral V, Bongard J, Cheney N. 2016 On the difficulty of co-optimizing morphology and control in evolved virtual creatures. In *Artificial Life Conference Proceedings 13* pp. 226–233. MIT Press.
31. Cheney N, Bongard J, SunSpiral V, Lipson H. 2018 Scalable co-optimization of morphology and control in embodied machines. *Journal of The Royal Society Interface* **15**, 20170937.
32. Ijspeert AJ. 2008 Central pattern generators for locomotion control in animals and robots: a review. *Neural networks* **21**, 642–653.
33. Miras K, De Carlo M, Akhatou S, Eiben A. 2020 Evolving-Controllers Versus Learning-Controllers for Morphologically Evolvable Robots. In *International Conference on the Applications of Evolutionary Computation (Part of EvoStar)* pp. 86–99. Springer.
34. Le Goff LK, Hart E, Coninx A, Doncieux S. 2020a On pros and cons of evolving topologies with novelty search. In *Artificial Life Conference Proceedings* pp. 423–431. MIT Press.
35. Le Goff LK, Buchanan E, Hart E, Eiben AE, Li W, de Carlo M, Hale MF, Angus M, Woolley R, Timmis J et al.. 2020b Sample and time efficient policy learning with cma-es and bayesian optimisation. In *Artificial Life Conference Proceedings* pp. 432–440. MIT Press.
36. Lehman J, Stanley KO. 2011 Novelty search and the problem with objectives. In *Genetic programming theory and practice IX* pp. 37–56. Springer.
37. Reynolds RG. 1994 An introduction to cultural algorithms. In *Proceedings of the third annual conference on evolutionary programming* vol. 24 pp. 131–139. World Scientific.
38. Liao T, Wang G, Yang B, Lee R, Pister K, Levine S, Calandra R. 2019 Data-efficient learning of morphology and controller for a microrobot. In *2019 International Conference on Robotics and Automation (ICRA)* pp. 2488–2494. IEEE.
39. Metzen JH, Fabisch A, Hansen J. 2015 Bayesian optimization for contextual policy search. In *Proceedings of the Second Machine Learning in Planning and Control of Robot Motion Workshop*. IROS Hamburg.
40. Le Goff LK, Hart E. 2021 p. 1498–1502. In *On the Challenges of Jointly Optimising Robot Morphology and Control Using a Hierarchical Optimisation Scheme*, p. 1498–1502. New York, NY, USA: Association for Computing Machinery.
41. Cheney N, Bongard J, SunSpiral V, Lipson H. 2016 On the Difficulty of Co-Optimizing Morphology and Control in Evolved Virtual Creatures. In *Artificial Life Conference 2016* p. 226.
42. Cully A, Clune J, Tarapore D, Mouret JB. 2015 Robots that can adapt like animals. *Nature* **521**, 503–507.
43. Walker K, Hauser H, Risi S. 2021 Growing Simulated Robots with Environmental Feedback: an Eco-Evo-Devo Approach. .