Biodiversity in Evolved Voxel-based Soft Robots

Eric Medvet
DIA - Università di Trieste
Trieste, Italy
emedvet@units.it

Alberto Bartoli
DIA - Università di Trieste
Trieste, Italy
bartoli.alberto@units.it

Federico Pigozzi
DIA - Università di Trieste
Trieste, Italy
federico.pigozzi@phd.units.it

Marco Rochelli
DIA - Università di Trieste
Trieste, Italy
marco.rochelli@gmail.com

ABSTRACT

In many natural environments, there are different forms of living creatures that successfully accomplish the same task while being diverse in shape and behavior. This biodiversity is what made life capable of adapting to disrupting changes. Being able to reproduce biodiversity in non-biological agents, while still optimizing them for a particular task, might increase their applicability to scenarios where human response to unexpected changes is not possible.

In this work, we focus on Voxel-based Soft Robots (VSRs), a form of robots that grants great freedom in the design of both body and controller and is hence promising in terms of biodiversity. We use evolutionary computation for optimizing, at the same time, body and controller of VSRs for the task of locomotion. We investigate experimentally whether two key factors—evolutionary algorithm (EA) and representation—impact the emergence of biodiversity and if this occurs at the expense of effectiveness. We devise a way for measuring biodiversity, systematically characterizing the robots shape and behavior, and apply it to the VSRs evolved with three EAs and two representations.

The experimental results suggest that the representation matters more than the EA and that there is not a clear trade-off between diversity and effectiveness.

CCS CONCEPTS

• **Computing methodologies** → *Mobile agents*; *Continuous space search*; **Evolutionary robotics**; • **Computer systems organization** → **Evolutionary robotics**; • **Theory of computation** → **Evolutionary algorithms**;

KEYWORDS

Evolutionary robotics, Modularity, Neuroevolution, Representation, Diversity

ACM Reference Format:
Eric Medvet, Alberto Bartoli, Federico Pigozzi, and Marco Rochelli. 2021. Biodiversity in Evolved Voxel-based Soft Robots. In *Genetic and Evolutionary Computation Conference (GECCO '21), July 10–14, 2021, Online.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION AND RELATED WORK

One long-term goal for robotics is to build robots that are able to operate in hazardous and dynamic environments, without the need for human supervision [4]. Several challenges have to be tackled in order to achieve that goal, ranging from scalability in the design and building phases [16, 25], possibly with auto-fabrication, to effective mechanisms for robot adaptation to changes in the environment [7]. Adaptation can occur at different time-scales as, e.g., learning [11, 15] and development [35]; moreover, it can involve individual robots or an entire robotic ecosystem. One way to foster adaptation of a robotic ecosystem is to promote diversity: if there are multiple robots that are able to perform the same task while being different, those robots can, as a whole, adapt to dynamic environments more easily. Indeed, diversity of living creatures—*biodiversity*—is the way in which nature made life robust to disruptive changes and it is so valuable that it has to be preserved in order to protect life itself [49].

Beside producing diversity, nature also proved to be capable of improving life, i.e., to make living creatures more efficient, through evolution. Therefore, it is not surprising that researchers resorted to the paradigm of evolution also for optimizing robots, igniting the field of *evolutionary robotics* [36].

In spite of the good premises, actually obtaining diversity while evolving robots is not easy. First, the comprehension, measurement, and promotion of diversity are themselves challenges in the broader field of Evolutionary Computation (EC) [43], from which evolutionary robotics borrows the optimization techniques. The pursuit for diversity is indeed pushing the EC community toward novel paradigms that attribute greater and greater importance to diversity [8], possibly even greater than quality of evolved solutions [26].

Second, the complexity of robot-environment interplay makes it even harder to obtain useful diversity, i.e., diversity that does not affect effectiveness. Very recent works confirmed that evolution tends to end up with similar solutions for a given problem, e.g., locomotion; when diverse solution are stimulated by changing the environment, often they are less effective [33]. Somehow surprisingly, when the size of the search space is increased by allowing the concurrent optimization of morphology and controller of the robots, diversity seems to vanish even more easily, in particular for

GECCO '21, July 10–14, 2021, Online

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

the morphology [37]. Indeed, the recurrent convergence to some specific forms for the body occurs also in nature: for example, the crab is known to be a powerful attractor [23].

All to all, the characterization of biodiversity and a deep understanding of which factors favor or impede it are still open issues in evolutionary robotics [41].

In this work, we study the impact on biodiversity of evolved robots of two key components of EC: the evolutionary algorithm (EA) and the solution representation. We consider the scenario of *Voxel-based Soft Robots* (VSRs), a kind of robots composed of many soft blocks capable of basic sensing and actuation and arranged together in a structure forming the robot [20]. VSRs fit particularly well our research objective for two reasons. First, their intrinsic modularity [51] and the fact they are a form of soft robots [24] make VSRs a promising path toward autonomous robotics ecosystems. Second, they allow great freedom in their design: the shape [5], the controller [48], and even the sensory apparatus [12] can be optimized. Such a great freedom should, in principle, enable biodiversity.

We consider three EAs that differ in how they are supposed to deal with diversity, and two representations that, while allowing the concurrent evolution of VSRs morphology and controller, differ in their expressiveness and hence exhibit different potential for diversity.

We analyze diversity *systematically* for a very large number of VSRs (tens of millions), while still looking at diversity with the human eye, like we do when associating animals and plants with a specific *species*. For doing that, we rely on Machine Learning (ML) for automatically assigning species to VSRs. We learn a ML model for species classification that operates on descriptors extracted from simulations of VSRs that perform the task of locomotion. Then, we use the variability of predicted species as a measure of diversity of a population of VSRs, using the well established Simpson index [42].

A few other works exist that examined diversity in evolutionary robotics [1, 9, 15, 33, 40]. Our work shares with them the general methodology—we all vary some factors regarding the evolution and observe how some descriptors are affected—but differs in either the kind of robots that is considered or the factors that are varied. Most of the cited works focus mainly on the impact of the environment [1, 15, 33]. The study that is the most similar to ours is [9], that investigates whether a mechanism of artificial speciation can favor morphological diversity. We also devise an EA that employs speciation and, in this respect, both [9] and us were inspired by NEAT [45]. However, differently than the cited work, we (i) also consider diversity of behavior, (ii) work with a more expressive kind of robots, VSRs, and (iii) analyze the joint impact of representation and EA.

Our work is significant also because we show that it is possible to evolve both the morphology and the controller of robots, a task that is considered hard [27]. There are other works that tackled the same [21, 22, 37] or similar challenges [15], but on different robots. Interestingly, our results confirm some of the previous findings, as, e.g., that vibration is a frequently evolved behavior for achieving effective locomotion [22].

2 BACKGROUND: VOXEL-BASED SOFT ROBOTS

Voxel-based Soft Robots (VSRs) are a form of robots composed of several deformable cubes (called voxels) that perform actions by varying the volume of the voxels. VSRs have been first introduced in [20], along with a procedure for physically realizing them. In this study, we consider a 2-D variant of simulated VSRs, proposed in [31], that is particularly suitable for investigating optimization by means of evolutionary computation. In this section, we briefly describe the salient characteristics of VSRs that are relevant to our study: we refer the reader to [31, 32] for more details.

A VSR is defined by its *morphology* and its *controller*. The former describes how the voxels are arranged and, for each voxel, the sensors it is equipped with. The controller determines how the area of each voxel varies over the time, possibly based on the readings of the sensors of that voxel and other voxels. The ability of sensing both itself and the environment makes VSRs potentially more effective in performing a task as, e.g., locomotion [48].

2.1 Morphology

Voxels of a VSR are arranged in a 2-D grid: adjacent voxels are rigidly connected at their vertices. In the simulation, the voxel is modeled as an assembly of spring-dampers systems, masses, and distance constraints [31]; we set, for each voxel, the same values for the parameters of those components, which results in all the voxels having the same mechanical properties.

The area of a voxel changes based on the corresponding *control signal* imposed by the controller and on the external forces acting on the voxel. The control signal is a value in $[-1, 1]$, where -1 corresponds to maximum requested expansion and 1 corresponds to maximum requested contraction. Expansion and contraction are modeled in the simulation as instantaneous changes of the resting length of the springs in the spring-damper systems of the voxel.

Voxels can be equipped with zero or more sensors. Each *sensor* produces, at each time step, a *sensor reading* $\boldsymbol{s} \in \mathbb{R}^p$. We use three kinds of sensors and equip every voxel with one sensor of each kind. *Area* sensors (for which $p = 1$) perceive the ratio between the current area of the voxel and its rest area. *Touch* sensors ($p = 1$) perceive whether the voxel is touching the ground ($\boldsymbol{s} = 1$) or not ($\boldsymbol{s} = 0$). *Velocity* sensors ($p = 2$) perceive the velocity of the center of mass of the voxel along the x- and y- axes of the voxel itself (i.e., the axes rotate with the voxel).

2.2 Controller

The controller determines the value of the control signals for each voxel at each time step. We use the distributed neural controller proposed in [30]. It consists of a number of fully connected feedforward neural network (NNs), one for each voxel.

At each time step, each NN takes as input the local sensor readings and the $4n_{\text{signal}}$ values generated by the adjacent NNs (i.e., those of the adjacent voxels) at the previous time step. A zero-vector of the proper size is used as input for the NN of voxels at the boundaries of the VSR. Then, it outputs the local control signal and $4n_{\text{signal}}$ values that will be used by adjacent NNs at the next time step.

After some exploratory experiments and by taking into account the findings of [30], we set $n_{\text{signal}} = 1$ and use NNs with no inner layers and with tanh as activation function. Considering that the overall dimension of the sensor readings is 4, this results in each NN having 4+4+1 input neurons (the +1 being the bias) and 4+1 output neurons; hence, each network is described by $9 \times 5 = 45 = n_{\text{NN}}$ numerical parameters (the weights of the edges connecting the neurons of the two layers).

Despite its simplicity, this form of controller may result in interesting and variegate behaviors, since the interconnections between voxel NNs make the overall NN recurrent [30]: therefore, there is a further dynamics introduced by the recurrent NN that interacts with the dynamics induced by the mechanical model of the voxels.

3 MEASURING BIODIVERSITY

Biodiversity plays a crucial role in natural evolution and the sustenance of life on Earth. In this work, we take the inspiration from natural sciences and build our definition of biodiversity of VSRs on the concept of *species*. We define a way for determining the species of a VSR, based on its shape and behavior (in the task of locomotion, see Section 5), and then measure biodiversity as the variety of species in a population.

In order to have species that can be discerned by humans, as it happens for living creatures, we *define* species by human inspection. It is then necessary to associate any given VSR obtained in our experiments with one of those species. Performing such an association for every evolved VSR is essential in order to analyze and characterize biodiversity systematically. Since we evolve a large number of VSRs (in the order of several millions), species classification by human inspection is not feasible. For this reason, we use machine learning (ML) for automatically determining the species of a VSR, an approach that is used also for determining the species of plants [14] and animals [47].

In the following sections we describe all the steps for determining the species of a VSR and measuring the resulting biodiversity in populations.

3.1 Species based on human-inspection

Since in this study we deal with VSRs that can evolve both the morphology and the controller, we use shape and behavior for defining species. That is, given a VSR, we associate its shape with one of a few predefined classes, its behavior with one of a few predefined classes and define the species of the VSR as the pair of those classes.

For deciding how many classes to use for those classifications, we inspected a large number of videos of VSRs performing locomotion by randomly selecting individuals during the evolutions that we performed for this study (see Section 5).

Concerning the shape, we define three classes. *Blob* VSRs exhibit high eccentricity and few irregularities in their shape. *Legged* VSRs have a body with protrusions extruding as if they were limbs (and can, ideally, be used as "legs" to promote locomotion). *Other* robots are grouped together in a third class and do not fit any of the two descriptions above. Figure 1 shows a few samples of shapes of the three classes.

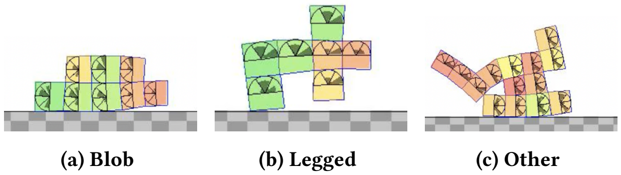


Figure 1: Frames extracted from the simulations of VSRs manually labeled to the three shape classes. The color of each voxel encodes the ratio between its current area and its rest area: red for < 1, yellow ≈ 1, green > 1; the circular sectors drawn at the center of each voxel indicates the current sensor readings.

Table 1: Distribution of the manually assigned labels for the shape and behavior classes on inspected simulations.

	Blob	Legged	Other	Total
Idle	732	465	747	1944
Galloping	331	252	476	1049
Crawling	170	469	576	1215
Vibrating	534	210	248	992
Total	1757	1396	2047	5200

Concerning the behavior, we define four classes. *Galloping* VSRs jump or hop, alternating between moments in which their body is in contact with the ground and moments in which it is not. *Crawling* VSRs always keep contact with the ground, contracting and expanding their body to move forward. *Vibrating* VSRs manifest a behavior similar to that of crawl, but move their body at a higher frequency that makes it hard to discern which parts of the body are touching the ground and which ones are not. Finally, *idle* VSRs do not show any movement useful for moving forward, i.e., either they stay steady or shake their limbs in such a disjointed manner that no clear behavior is discernible. We provide the video of a few VSRs exhibiting the behaviors of the four classes at https://youtu.be/my6c4L-b0eM.

3.2 Species classification

Having defined classes for shape and behavior as illustrated in the previous section, we need to synthesize two classifiers able to associate a previously unseen VSR with the corresponding classes automatically. To this end, we constructed a set of labeled examples, i.e., of VSRs associated with their corresponding species by a human operator. We inspected several simulation videos and assigned VSRs to shape and behavior classes until collecting 5200 labeled VSRs, distributed among classes as summarized in Table 1. We remark that the simulations are deterministic [31]: we hence need to simulate each VSR just once.

As in any ML application, two key design choices concern the features to extract for describing the observations (here, simulations of VSRs) and the learning technique. Since we aim at classifying the shape and the behavior separately, we define different features for the two classification tasks. We describe our choices in the next sections.

3.2.1 Shape features. Since the VSR body is an arrangement of voxels in a 2-D grid, we could extract the features concerning the shape directly from the grid. However, the grid is a static description of the VSRs and does not capture the robot as seen during its life. For this reason, we define the shape features based on the idea of the *dynamic pose* of the VSR. Intuitively, the dynamic pose can be regarded as a "long-exposure photograph" of the VSR during the simulation and we construct such a pose as follows.

Let S be a sequence of snapshots of a simulation of a VSR, where a *snapshot* is the description of every object in the simulation (namely, the ground and every voxel of the VSR) at a given time step. For each snapshot we determine the minimal bounding square $(x_0, y_0, x_0 + l, y_0 + l)$ around the VSR, that is, the smallest square parallel to the x-axis that completely contains the VSR. Then, we partition the minimal bounding square in 16×16 inner squares with side length $\frac{l}{16}$ and build a matrix $\boldsymbol{d} \in \{0, 1\}^{16 \times 16}$ where the element $d_{i,j}$ is 1 if and only if the corresponding inner square $(x_0 + (i - 1)\frac{l}{16}, y_0 + (j - 1)\frac{l}{16}, x_0 + i\frac{l}{16}, y_0 + j\frac{l}{16})$ is covered by the VSR for at least half of the area. Finally, we compute the dynamic pose as the element-wise mode of the matrices computed for the snapshots in S. Figure 2 shows a graphical representation of the dynamic pose of a few VSRs.

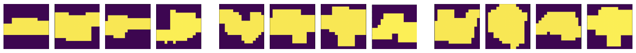
We use the 256 values of the dynamic pose as feature vector for the shape, obtaining $\boldsymbol{f}_{\text{shape}} \in \{0, 1\}^{256}$.

3.2.2 Behavior features. Since we deal with robots performing the task of locomotion, we define two groups of features that should capture the robot behavior while in locomotion, i.e., the *gait*, from two different points of view: the movement of the center of the VSR over the time and the way the VSR touches the ground while moving. We denote by $\boldsymbol{f}_{\text{center}}$ and $\boldsymbol{f}_{\text{footprints}}$ the two corresponding feature vectors, and by $\boldsymbol{f}_{\text{behavior}}$ their concatenation.

Center movement. Given a sequence S of snapshots, we extract the signals $x_c(k)$, $y_c(k)$ of the center of mass of the VSR from S. Then, we consider the signals $x'_c(k)$, $y'_c(k)$ of the first differences (i.e., $x'_c(k) = x_c(k) - x_c(k - 1)$) and compute their Fast Fourier Transform (FFT). Subsequently, we take the magnitude of the two FFTs, filter out the components corresponding to frequencies lower than $f_{\min}$ or greater than $f_{\max}$ (by taking into account the simulation time step Δt), and re-sample the remaining components in order to have n_{freq} components for each one of the two axes.

We use as feature vector the concatenation of the two resulting vector of magnitudes $\boldsymbol{f}_{\text{center}} = \left[\boldsymbol{f}_{\text{center},x}\ \boldsymbol{f}_{\text{center},y}\right] \in \mathbb{R}^{+2n_{\text{freq}}}$, with $\mathbb{R}^+ = [0, +\infty[$. After preliminary experiments and exploiting our expertise, we set $f_{\min} = 0\,\text{Hz}$, $f_{\max} = 10\,\text{Hz}$, and $n_{\text{freq}} = 100$.

Footprints. Concerning the features describing how the VSR touches the ground, we build a definition based on the concept of *footprint*. Given a snapshot, we consider the projection $[x_0, x_0 + l]$ of the minimal bounding square on the x-axis and we partition it in 8 equally sized segments. Then we build the footprint of the VSR in that snapshot as a binary sequence $\boldsymbol{m} \in \{0, 1\}^8$ where the element m_i is 1 if and only if the VSR is touching the ground for at least half of the corresponding segment $[x_0 + (i - 1)\frac{l}{8}, x_0 + i\frac{l}{8}]$.



(a) Blob **(b) Legged** **(c) Other**

Figure 2: Examples of prediction of the shape classifier, given as graphical representations of the dynamic pose $\boldsymbol{f}_{\text{shape}}$. For each class, poses of three VSRs are shown, from left to right: one with manually assigned label, one with correctly predicted label (a true positive), two with a wrongly predicted label (a false negative and a false positive).

Given a sequence S of snapshots, we extract some features from the most representative sequence of footprints, that we determine as follows. (1) We split S in a sequence $(S_1, S_2, \dots)$ of non-overlapping subsequences, each one corresponding to an interval of $\Delta t_{\text{footprint}}$ simulated time (we set $\Delta t_{\text{footprints}} = 0.5\,\text{s}$ after some preliminary experiments). (2) We build the sequence $M = \{\boldsymbol{m}_1, \boldsymbol{m}_2, \dots\}$ of footprints where each $\boldsymbol{m}_i$ is obtained as the element-wise mode of the footprints computed from snapshots in S_i. (3) We consider all the n-grams of footprints in M, with $2 \le n \le 5$, that occur at least twice and compute the overall duration of each n-gram, computed as the product between its number of occurrences and its duration. (4) We select as the *main footprint n-gram* $M^\star$ the n-gram with the greatest overall duration. (5) We compute the following descriptors for $M^\star$: duration $|M^\star|\Delta t_{\text{footprint}}$, average touch area $\frac{1}{|\star|}\frac{1}{8}\sum_{\boldsymbol{m} \in M^\star}\sum_{i=1}^{i=8} m_i$, number of occurrences, mode Δt_M of the intervals between subsequent occurrences of $M^\star$, rate of intervals that are equals to the mode.

We use as feature vector the five descriptors computed for the main footprint n-gram $M^\star$, i.e., $\boldsymbol{f}_{\text{footprint}} \in \mathbb{R}^{+5}$.

3.2.3 Learning technique. We rely on Random Forest [3] for building the classifiers for the shape and behavior classes based on $\boldsymbol{f}_{\text{shape}}$ and $\boldsymbol{f}_{\text{behavior}}$, respectively. We chose this supervised learning technique because it has been shown to be among the best general purpose classification techniques [13, 50].

For having an estimate of the accuracy of Random Forest on our two classification tasks, we performed a 5-fold cross validation assessment using the 5200 labeled simulations and obtained an average accuracy of 0.784 and 0.783 for shape and behavior classification, respectively. A trivial classifier (always predicting the most frequent class) taking into account the class imbalances obtained an accuracy of 0.394 and 0.374, respectively. Figure 2 shows a few examples of prediction of the shape class in the form of the corresponding dynamic poses.

3.3 Simpson index

Different measures of biodiversity have been used in the ecological literature. Among them, the Simpson index is one of the most commonly used [42]. Given a population of individuals that is partitioned based on species, this index is defined as $\lambda = \sum_{i=1}^{i=n} p_i^2$, where n is the number of species and p_i is the fraction of individuals of the i-th species.

Intuitively, the Simpson index measures the probability that two individuals picked at random with replacement belong to the same species. Since its semantics is the opposite of the one of biodiversity (i.e., $\lambda = 1$ for a population composed of a single species and it is < 1 for more diverse populations), in this study we use the inverse Simpson index (ISI) λ^{-1} (defined in $[1, +\infty[$): the greater the ISI, the more diverse the population.

4 EVOLUTION OF VSR MORPHOLOGY *AND* CONTROLLER

We want to investigate whether VSRs can be optimized for a given task, by means of EC, while maintaining biodiversity measured as described above. For fully exploiting the potential of expressing diverse solutions to a task that is intrinsic of VSRs, we need a way for evolving simultaneously the morphology and the controller.

We here propose two genotypic representations that encode a description of both the morphology and the controller of a VSR in a single numerical vector. The resulting optimization problem is hence a search in the numerical space $\mathbb{R}^p$, for which many techniques exist. We experiment with three EAs that fit this scenario, one of them being tailored to the specific goal of promoting the biodiversity. In the following sections, we describe the representations and the EAs.

It is worth to note that this is, to the best of our knowledge, the first approach to the simultaneous evolution of the morphology and the controller of VSRs that are able to sense the environment.

4.1 Representations

We define two representations for the morphology and the controller of the VSR in the form of a numerical vector $\boldsymbol{v} \in \mathbb{R}^p$. In both of them, a portion $\boldsymbol{v}_{\text{morph}}$ of $\boldsymbol{v}$ directly encodes a description of the morphology and the remaining, disjoint part $\boldsymbol{v}_{\text{ctrl}}$ describes the controller, i.e., $\boldsymbol{v} = \left[\boldsymbol{v}_{\text{morph}}\ \boldsymbol{v}_{\text{ctrl}}\right]$. The representations differ only in the latter.

Given a $\boldsymbol{v}_{\text{morph}} \in \mathbb{R}^{n^2_{\text{size}}}$, we build a morphology, i.e., a collection of adjacent voxels arranged in a 2-D grid, as follows. Let $n_{\text{size}} \times n_{\text{size}}$ be the size of a square grid enclosing the largest representable VSR morphology. First, we build a Boolean matrix $\boldsymbol{b} = \{\text{T}, \text{F}\}^{n_{\text{size}} \times n_{\text{size}}}$ where $b_{i,j}$ is set to true if and only if $v_k > 0$, with $k = i + (j - 1)n_{\text{size}}$. Then, we build the morphology by considering the largest connected part of $\boldsymbol{b}$ elements set to true and putting a voxel at each element of such set.

Concerning the controller part, and considered that we use the controller defined in Section 2.2, we propose two alternatives.

In the *heterogeneous* controller representation, that we denote by He, we assume that the NNs composing the controller (one NN for each voxel) can have different parameters. In order to favor the locality of the representation [38] and to make the controller representation agnostic with respect to the morphology representation, $\boldsymbol{v}_{\text{ctrl}}$ is the concatenation of the weights of the NNs of all the voxels, i.e., $\boldsymbol{v}_{\text{ctrl}} = \left[\boldsymbol{w}_{1,1}\ \dots\ \boldsymbol{w}_{n_{\text{size}},n_{\text{size}}}\right]$, where $\boldsymbol{w}_{i,j}$ is the vector of weights of the NN at (i, j) position in the enclosing grid. It follows that $\boldsymbol{v}_{\text{ctrl}} \in \mathbb{R}^{n^2_{\text{size}} \times n_{\text{NN}}}$. This representation is the same, for the controller part, as the one proposed in [30].

In the *homogeneous* controller representation, denoted by Ho, we assume that all the NNs have the same parameters $\boldsymbol{w}$. It follows that $\boldsymbol{v}_{\text{ctrl}} = \boldsymbol{w} \in \mathbb{R}^{n_{\text{NN}}}$.

It can be seen that the two controller representations differ in expressiveness and degeneracy [39], i.e., the degree to which different genotypes correspond to different phenotypes. The heterogeneous representation is the most expressive one, thus resulting in the largest search space. The homogeneous representation is the least expressive one: its search space is smaller and hence, in principle, easier to explore. However, it might be harder for the evolution to find the combination of genes that, when translated in the same NNs for each voxel, results in a VSR that exhibits the desired complex behavior.

We performed the experiments that we discuss below with $n_{\text{size}} = 5$, that resulted in $|\boldsymbol{v}|$ being $25 + 1125 = 1150$ and $25 + 45 = 70$, respectively for He and Ho representations. In an initial exploratory phase, we also experimented with $n_{\text{size}} = 10$, that resulted in findings similar to the one we found for $n_{\text{size}} = 5$, and with a radically different representation for the morphology. In the latter case, based on the observations of [5], we used the generative representation based on a mixture of bi-variate Gaussian distributions described in [32]: we observed unclear experimental findings that we plan to further investigate in the future.

4.2 Evolutionary algorithms

We use three EAs suitable for optimizing in the numerical space $\mathbb{R}^p$. Two of them are general purpose EAs, one is an EA that employs a form of speciation based on the concept of species defined in Section 3.1.

4.2.1 CMA-ES. The first EA is Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [18, 19], a state-of-the-art numerical optimizer that has recently shown to be effective also for the optimization of VSRs [12, 29]. CMA-ES iteratively optimizes the solution in the form of a multivariate normal distribution. At each iteration, it samples the distribution obtaining a population of solutions and then updates the parameters of the distribution by recomputing them based on the best half of the population. Non trivial heuristics are employed while updating the distribution—we refer the reader to [18] for more details.

In our experiments, we use the default parameters suggested in [17], namely the initial step size $\sigma = 0.5$ and the population size $\lambda = 4 + \lfloor 3 \log p \rfloor$, p being the dimension of the search space, i.e., 1150 or 70 depending on the representation. We set the initial vector of means by sampling uniformly the interval $[-1.0, 1.0]$ for each vector element. We let CMA-ES iterate until $n_{\text{evals}} = 30\,000$ fitness evaluations have been done.

CMA-ES is commonly considered a form of population-based optimization, since it is an improved form of evolutionary strategy. We remark, however, that the population in CMA-ES is indeed a realization of a multivariate normal distribution, i.e., all the individuals are "variations" of a single individual, the mean of the distribution. This observation is relevant in our settings, where we study the biodiversity of the evolved solutions.

4.2.2 Genetic algorithm. As second EA, we use a standard form of Genetic algorithm (GA). GA iteratively evolves a fixed-size population of n_{pop} individuals according to a μ+λ generational model [10], i.e., with overlapping: at each generation, the offspring and the parents are merged and the worst individuals are discarded. For building the offspring, we select individuals with tournament selection of size 5 and then we apply Gaussian mutation with $\sigma_{\text{mut}} = 0.35$, with probability $p_{\text{mut}} = 0.2$, or extended geometric crossover with probability $1 - p_{\text{mut}}$. With the latter, given two parents $\boldsymbol{v}_1, \boldsymbol{v}_2 \in \mathbb{R}^p$, the new individual is obtained as $\boldsymbol{v} = \boldsymbol{v}_1 + \boldsymbol{\alpha}(\boldsymbol{v}_2 - \boldsymbol{v}_1)$, where each element α_i of $\boldsymbol{\alpha}$ is chosen randomly with uniform probability in $[-1, 2]$.

As for CMA-ES, we build the initial population by sampling uniformly $[-1.0, 1.0]$ and iterate until $n_{\text{evals}} = 30\,000$ fitness evaluations have been done. Moreover, we set $n_{\text{pop}} = 100$ and $p_{\text{mut}} = 0.2$.

4.2.3 Speciated evolver. We designed this EA, that we denote by SE, specifically for this study. SE employs a form of speciation inspired by NEAT [45], the popular EA for evolving the topology and the weights of NNs. In NEAT, speciation was introduced with the purpose of protecting innovations introduced by modifications in the topology. In SE, we do not optimize the topology of the NNs composing the controller of the VSR; notwithstanding, speciation is, more broadly, a mechanism for promoting diversity in the population [43].

```
function evolve():
    $P \leftarrow$ initialize($n_{\text{pop}}$)
    while $\neg$shoulStop() do
        $P' \leftarrow \emptyset$
        $(P_1, \dots, P_k) \leftarrow$ kmeans($P$)
        $P' \leftarrow P' \cup \{$best($P$)$\}$
        foreach $i \in \{1, \dots, n\}$ do
            if $|P_i| \geq n_{\text{elite}}$ then
                $P' \leftarrow P' \cup \{$best($P_i$)$\}$
            end
        end
        $n'_{\text{pop}} \leftarrow n_{\text{pop}} - |P'|$
        $\boldsymbol{r} \leftarrow$ ranks(repr($P_1$), ..., repr($P_k$))
        foreach $i \in \{1, \dots, k\}$ do
            $c \leftarrow 0$
            while $c < n'_{\text{pop}} \alpha^{r_i} \frac{1}{\sum_{i=1}^{i=k} \alpha^{r_i}}$ do
                if $U(0, 1) \leq p_{\text{mut}}$ then
                    $\boldsymbol{v}$ = nth($P_i$, $c \bmod |P_i|$)
                    $P' \leftarrow P' \cup \{$mutate($\boldsymbol{v}$)$\}$
                    $c \leftarrow c + 1$
                else
                    $\boldsymbol{v}_1$ = nth($P_i$, $c \bmod |P_i|$)
                    $\boldsymbol{v}_2$ = nth($P_i$, $(c + 1) \bmod |P_i|$)
                    $P' \leftarrow P' \cup \{$crossover($\boldsymbol{v}_1, \boldsymbol{v}_2$)$\}$
                    $c \leftarrow c + 2$
                end
            end
        end
        $P \leftarrow P'$
    end
end
```

Algorithm 1: The algorithm of SE.

Similarly to GA, SE iteratively evolves a fixed-size population of n_{pop} individuals, as shown in Algorithm 1. At each iteration, individuals are partitioned in species according to a given criterion (described below) that also determines a single representative individual of each species. Then, the current best individual in the population and the best individual of every species larger than n_{elite} are moved in the offspring. The remaining individuals in the offspring are generated as follows. First, an offspring slot of size $n'_{\text{pop}} \alpha^{r_i} \frac{1}{\sum_{i=1}^{i=n} \alpha^{r_i}}$ is reserved to each species P_i depending on the rank r_i of the corresponding representative individual repr(P_i); $\alpha \in]0, 1]$ is a parameter of the algorithm—the closer to 1, the less the preference to fittest species. Then, the offspring slot is filled by applying Gaussian mutation or expanded geometric crossover (as in GA) to individuals of the corresponding species P_i.

As partitioning criterion, we explore three variants. All are based on the application of the k-means clustering technique [28] and select as representative individual the one closest to the centroid of the cluster. They differ in what we use for computing the distance among individuals.

In the first variant, that we denote by SE-g, we use the *genotype* $\boldsymbol{v}$, whose dimension is 1150 or 70 depending on the representation. In the second variant, denoted by SE-s, we use the vector $\boldsymbol{f}_{\text{shape}} \in \mathbb{R}^{256}$ of the *shape features* (see Section 3.2.1). Finally, in the third variant, denoted by SE-b, we use the vector $\boldsymbol{f}_{\text{behavior}} \in \mathbb{R}^{205}$ of the *behavior features* (see Section 3.2.2). In all cases, we compute the Euclidean distance after having properly normalized the vectors of the individuals of the current population.

In the experiments, we set $n_{\text{pop}} = 100$, $p_{\text{mut}} = 0.2$, as for GA, and $n_{\text{evals}} = 30\,000$, as for GA and CMA-ES. Moreover, we set $\alpha = 0.75$, $n_{\text{elite}} = 5$, and $k = 10$ (for k-means).

5 EXPERIMENTAL EVALUATION

We performed several experiments aimed at answering experimentally the following research questions:

RQ1 How do the representation and the EA impact the emergence of biodiversity?
RQ2 Is there a trade-off between effectiveness and biodiversity?

Moreover, since this is the first approach to the concurrent evolution of the morphology and controller of VSRs, we were also generally interested in understanding if effective solutions can be found in such a large design space.

For all the experiments, we considered the task of locomotion. The goal of the VSR is to travel as far as possible, in the positive x direction, along a flat surface and within a time interval of $t_{\text{final}} = 30$ s (simulated time). The fitness of the individual is the traveled distance, measured as $\Delta x_c = x_c(t_{\text{final}}) - x_c(0)$, where $x_c(t)$ is the x-position of the center of mass of the VSR at time t.

We used 2D-VSR-Sim [31] for the simulation of the VSRs, with a time step of $\Delta t = \frac{1}{60}$ s and all the other parameters set to default values. The code of the experiments is publicly available at https://github.com/pigozzif/VSRBiodiversity.

For each combination of EA and representation, we performed 10 evolutionary runs by varying the random seed of the EA—we remark that the simulations are instead deterministic. Overall, $10 \times 5 \times 2 \times 30\,000 = 3 \times 10^7$ VSRs were generated in our experiments: for each one of them we applied the two models for predicting the shape and behavior classes learned on the small subset of 5200 manually labeled VSRs, as described in Section 3.

5.1 Results

Table 2 summarizes the results of all our experiments. For each combination of EA and representation, the table shows the median and standard deviation, across the 10 evolutionary runs, of the key

Table 2: Median and standard deviation, across the 10 runs, of salient indexes computed on the population at the last iteration.

EA	Repr.	$\Delta x_c^\star$	ISI_shape	$\text{ISI}_\text{behav.}$	ISI
CMA-ES	He	89± 43	1.0±0.5	1.6±0.8	2.0±1.3
	Ho	119± 58	1.0±0.1	1.5±0.6	1.5±0.8
GA	He	139± 80	1.0±0.0	1.0±0.2	1.0±0.2
	Ho	117±120	1.0±0.0	1.0±0.0	1.0±0.0
SE-g	He	144± 62	2.3±0.4	2.4±0.6	4.6±1.1
	Ho	181± 64	2.1±0.5	2.4±0.6	4.7±1.1
SE-s	He	160± 60	2.0±0.5	2.3±0.3	4.1±1.0
	Ho	280± 50	2.0±0.3	2.3±0.6	4.6±1.1
SE-b	He	252± 85	1.9±0.5	2.5±0.7	3.9±1.5
	Ho	237± 98	2.0±0.4	2.2±0.4	3.7±1.6

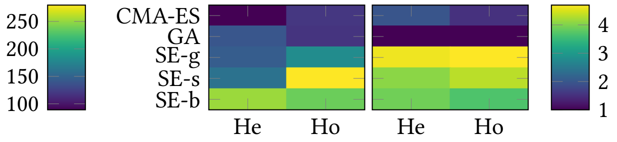


Figure 3: Heatmap of the median $\Delta x_c^\star$ (left) and ISI (right) computed on the population at the last iteration.

indexes measuring effectiveness in locomotion and biodiversity. The indexes are computed on the population at the last iteration of the EA. Concerning effectiveness, Table 2 shows the fitness $\Delta x_c^\star$ of the best individual measured in m—as a reference, a voxel side is 3 m long. Concerning biodiversity, the table shows the biodiversity ISI as defined in Section 3.3, and the biodiversity ISI_shape and $\text{ISI}_\text{behav.}$ computed considering only the shape and behavior class, respectively.

For easing the comparison, we report in Figure 3 the values of the two most important indexes ($\Delta x_c^\star$ and ISI). We performed the Mann-Whitney U test (after having verified the relevant hypotheses) between pairs of combinations for both $\Delta x_c^\star$ and ISI and found that the differences are statistically significant in the vast majority of cases.

It can be seen that all combinations of EA and representation are able to evolve effective VSRs, i.e., they score well in $\Delta x_c^\star$. In terms of $\Delta x_c^\star$, SE variants achieve the best results among the EAs, and Ho outperforms He with 3 on 5 EAs. The latter finding means that, despite the lower expressiveness of the Ho representation with respect to He, a "single NN" is capable of driving cooperatively an entire robot, when proper parameters are found. As it turns out from our experiments, finding these parameters is feasible with all the EAs, maybe because of the much lower dimension of the search space.

Figure 4 provides an overview of the distribution, across shape and behavior classes, of the evolved VSRs (considering the populations at the last iteration). Besides the number of VSRs for each class, the figure also shows the median $\Delta x_c^\star$ of individuals in the class. It can be seen that the relative majority of the evolved individuals belong to class Other/Idle. The most effective, i.e., fastest, VSRs are those of class Legged/Crawling, followed by Blob/Vibrating. The Crawling behavior is, in general, the most effective. Not surprisingly, Idle VSRs are not effective.

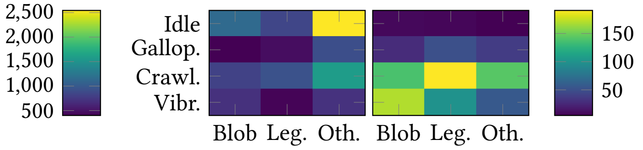
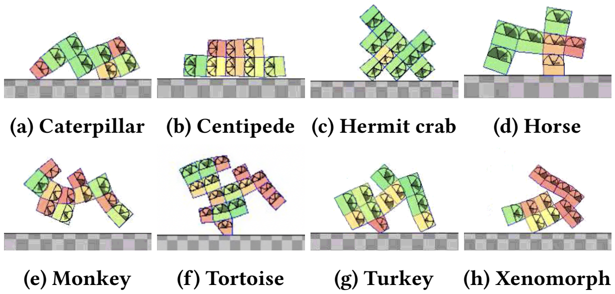


Figure 4: Heatmap of the number of VSRs (left) and their median $\Delta x_c^\star$ (right) computed on the population at the last iteration for each combination shape and behavior classes.


(a) Caterpillar **(b) Centipede** **(c) Hermit crab** **(d) Horse**

(e) Monkey **(f) Tortoise** **(g) Turkey** **(h) Xenomorph**

Figure 5: A subset of outperforming individuals, in terms of effectiveness and variety of shapes and behavior. A video can be found at https://youtu.be/P_fcCaTWC0o.

The results concerning Vibrating behavior call for some further remarks. If we attempted to physically realize those Vibrating VSRs, maybe using the approaches of [25, 46], they would likely not be as fast as in simulation—i.e., there would likely be a *reality gap* problem [34]. We think that Vibrating VSRs are evolved frequently for two reasons. First, energy consumption is not considered in our simulations, and hence inefficient behaviors are not discouraged [22]. Second, the recurrent nature of our neural controller, and in particular the voxel-to-voxel signal passing, likely favors the emergence of high frequency dynamics [30].

Beyond the overview given by the aggregate indexes, that we discuss more in detail below, we manually inspected a subset of the most effective VSRs and found that looked quite different. We showcase some of those VSRs in Figure 5; the corresponding video that can be found at https://youtu.be/P_fcCaTWC0o.

Those hand-picked VSRs strikingly mirror emergent patterns found in nature. For example, we named one individual "monkey" after running forward by alternating between its fore limb and its rear limb, while using an extrusion from its back as a sort of tail to balance. Another individual, "caterpillar", crawled ahead by pulsing its body. Others slithered like centipedes or trotted like equines, to name a few traits. In general, both primitive and complex morphologies emerged. Interestingly, evolution succeeded in adaptation also with bizarre and unusual solutions. In fact, some individuals covered long distances while possessing a shape that might have turned out an handicap. As a proof of concept, one individual (nicknamed the "tortoise") propelled forward using a pair of leg-like protrusions despite carrying on its back an uncomfortable hump (resembling a shell indeed) that might have hindered motion.

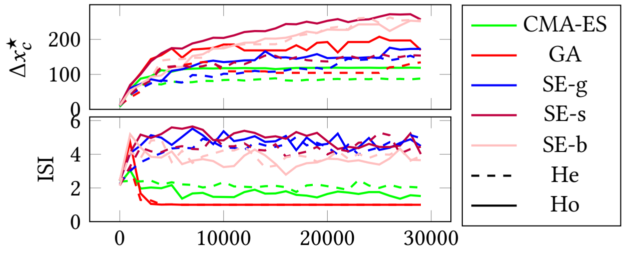


Figure 6: Median $\Delta x_c^\star$ (top) and ISI (bottom) during the evolution.

5.2 RQ1: impact of EA and representation on biodiversity

Figure 3 and Table 2 provide a clear view on the differences in biodiversity among EAs and representations. Among the EAs, the SE variants, that are designed to promote diversity, obtain much larger ISI than CMA-ES and GA. The differences are much less sharp for representations: interestingly, the much smaller search space induced by Ho does not result in a lower diversity with respect to He.

For deeper insights, we show in Figure 6 how effectiveness in locomotion and diversity vary during the evolution. Concerning the former, the top plot of the figure confirms the findings of Table 2 and also suggests that SE-s and SE-b, i.e., SE variants where speciation takes into account shape and behavior classes, might still improve $\Delta x_c^\star$ with a longer evolution.

Concerning diversity, the ISI plot of Figure 6 highlights the great differences among the EAs. While SE variants generally maintain a great diversity in the population, CMA-ES and GA do not. For CMA-ES, the finding is not surprising: as discussed in Section 4.2.1, this EA does not evolve a population of actually different individuals, but rather evolves one prototype individual by sampling its variants. We looked at the raw results of GA and found that the drastic drop in diversity is the result of the joint action of the crossover and the generational model (that employs overlapping): a good individual often mates with itself generating a duplicate, that rapidly fills the population. All to all, our GA turned out to operate with the wrong *exploration-exploitation trade-off*, a long-standing issue in EC [6]. We believe that this limitation might be addressed by employing some diversity promotion mechanism [44], maybe acting at different levels of the representation (genotype, phenotype, fitness) [2].

5.3 RQ2: effectiveness vs. diversity

Figure 7 shows the relation between effectiveness ($\Delta x_c^\star$) and biodiversity (ISI) in the form of a scatter plot with one point for each one of our evolutionary runs. The figure clearly shows that there is not, in our results, a trade-off between effectiveness and diversity: it is not true that greater effectiveness is obtained at the expense of diversity.

We dug in the results and found that when an evolution struggles in finding effective solutions, those solutions are often classified as Idle. On the contrary, effective solutions are associated with a more diverse combination of classes. It could be argued that this finding is a bias induced by the way we define biodiversity, i.e., based on perception of the shape and behavior from the human point of view—we are likely more able to spot differences in creatures that move in a somehow familiar way, rather than in creatures that fidget without an apparent purpose. However, the good effectiveness achieved by SE-s and SE-b EAs, whose working principle depends right on our definition of biodiversity, suggests that our choice is sound.

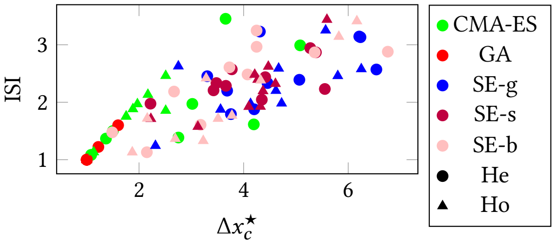


Figure 7: Scatter plot of ISI vs. $\Delta x_c^\star$, one point for each evolutionary run.

6 CONCLUDING REMARKS

We considered a subfield of evolutionary robotics that aims at optimizing Voxel-based Soft Robots (VSRs), a kind of modular soft robots composed by many simple soft blocks. We investigated whether evolutionary computation (EC) can be used for optimizing the morphology and the controller of VSRs in such a way that they effectively solve the task of locomotion in many different ways, i.e., if EC can achieve biodiversity of VSRs.

In particular, we focused on the impact of biodiversity of two key components of every EC approach, the evolutionary algorithm (EA) and the representation of the solutions. We experimented with three EAs, one of which designed purposely by us for favoring the emergence of biodiversity based on a speciation mechanism, and two representations.

We devised a way for measuring systematically the diversity of a population of evolved VSRs: we manually inspected a subset of VSRs and assigned them to a few classes for shape and behavior, defined based on human perception. Then, we learned a machine learning model for automatically classifying many millions of VSRs observed during the evolutions.

Our experimental results showed that the EA plays a more important role than representation in determining biodiversity. Moreover, they suggest that employing a diversity promotion mechanism that is based on a human-like notion of species can result in better effectiveness, as well as in larger biodiversity.

We believe that our work contributes to address one of the long-standing issues in evolutionary robotics, preservation of diversity, and provides further confirmations that bio-inspired methods can be effective also when dealing with artificial agents.

ACKNOWLEDGMENTS

F.P. was partially supported by a Google Faculty Research Award granted to E.M.. The experimental evaluation of this work has been done on CINECA HPC cluster within the CINECA-University of Trieste agreement.

REFERENCES

[1] Joshua E Auerbach and Josh C Bongard. 2014. Environmental influence on the evolution of morphological complexity in machines. *PLoS Comput Biol* 10, 1 (2014), e1003399.

[2] Alberto Bartoli, Andrea De Lorenzo, Eric Medvet, and Giovanni Squillero. 2019. Multi-level diversity promotion strategies for Grammar-guided Genetic Programming. *Applied Soft Computing* 83 (2019), 105599.

[3] Leo Breiman. 2001. Random forests. *Machine learning* 45, 1 (2001), 5–32.

[4] Edgar Buchanan, Léni K Le Goff, Wei Li, Emma Hart, Agoston E Eiben, Matteo De Carlo, Alan F Winfield, Matthew F Hale, Robert Woolley, Mike Angus, et al. 2020. Bootstrapping Artificial Evolution to Design Robots for Autonomous Fabrication. *Robotics* 9, 4 (2020), 106.

[5] Nick Cheney, Robert MacCurdy, Jeff Clune, and Hod Lipson. 2013. Unshackling evolution: evolving soft robots with multiple materials and a powerful generative encoding. In *Proceedings of the 15th annual conference on Genetic and evolutionary computation*. ACM, 167–174.

[6] Matej Črepinšek, Shih-Hsi Liu, and Marjan Mernik. 2013. Exploration and exploitation in evolutionary algorithms: A survey. *ACM computing surveys (CSUR)* 45, 3 (2013), 1–33.

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

[8] Antoine Cully and Yiannis Demiris. 2017. Quality and diversity optimization: A unifying modular framework. *IEEE Transactions on Evolutionary Computation* 22, 2 (2017), 245–259.

[9] Matteo De Carlo, Daan Zeeuwe, Eliseo Ferrante, Gerben Meynen, Jacintha Ellers, and AE Eiben. 2020. Influences of Artificial Speciation on Morphological Robot Evolution. In *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2272–2279.

[10] Kenneth A De Jong. 2006. *Evolutionary Computation: A Unified Approach*. MIT Press.

[11] AE Eiben and Emma Hart. 2020. If it evolves it needs to learn. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion*. 1383–1384.

[12] Andrea Ferigo, Giovanni Iacca, and Eric Medvet. 2021. Beyond Body Shape and Brain: Evolving the Sensory Apparatus of Voxel-based Soft Robots. In *International Conference on the Applications of Evolutionary Computation (Part of EvoStar)*. Springer.

[13] Manuel Fernández-Delgado, Eva Cernadas, Senén Barro, and Dinani Amorim. 2014. Do we need hundreds of classifiers to solve real world classification problems? *The journal of machine learning research* 15, 1 (2014), 3133–3181.

[14] Steven E Franklin and Oumer S Ahmed. 2018. Deciduous tree species classification using object-based analysis and machine learning with unmanned aerial vehicle multispectral data. *International Journal of Remote Sensing* 39, 15-16 (2018), 5236–5245.

[15] Agrim Gupta, Silvio Savarese, Surya Ganguli, and Li Fei-Fei. 2021. Embodied Intelligence via Learning and Evolution. *arXiv preprint arXiv:2102.02202* (2021).

[16] Matthew F Hale, Mike Angus, Edgar Buchanan, Wei Li, Robert Woolley, Léni K Le Goff, Matteo De Carlo, Jon Timmis, Alan F Winfield, Emma Hart, et al. 2020. Hardware design for autonomous robot evolution. In *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2140–2147.

[17] Nikolaus Hansen. 2006. The CMA evolution strategy: a comparing review. In *Towards a new evolutionary computation*. Springer, 75–102.

[18] Nikolaus Hansen. 2016. The CMA Evolution Strategy: A Tutorial. arXiv:cs.LG/1604.00772

[19] Nikolaus Hansen and Andreas Ostermeier. 2001. Completely Derandomized Self-Adaptation in Evolution Strategies. *Evolutionary Computation* 9, 2 (2001), 159–195. https://doi.org/10.1162/106365601750190398

[20] Jonathan Hiller and Hod Lipson. 2012. Automatic design and manufacture of soft robots. *IEEE Transactions on Robotics* 28, 2 (2012), 457–466.

[21] Gregory S Hornby, Jordan B Pollack, et al. 2001. Body-brain co-evolution using L-systems as a generative encoding. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*. 868–875.

[22] Michał Joachimczak, Reiji Suzuki, and Takaya Arita. 2016. Artificial Metamorphosis: Evolutionary Design of Transforming, Soft-Bodied Robots. *Artificial Life* 22, 3 (2016), 271–298. https://doi.org/10.1162/ARTL_a_00207 PMID: 27139940.

[23] Sara Kiley Watson. 2020. Why everything eventually becomes a crab. https://www.popsci.com/story/animals/why-everything-becomes-crab-meme-carcinization/

[24] Sangbae Kim, Cecilia Laschi, and Barry Trimmer. 2013. Soft robotics: a bioinspired evolution in robotics. *Trends in biotechnology* 31, 5 (2013), 287–294.

[25] Sam Kriegman, Douglas Blackiston, Michael Levin, and Josh Bongard. 2020. A scalable pipeline for designing reconfigurable organisms. *Proceedings of the National Academy of Sciences* 117, 4 (2020), 1853–1859.

[26] Joel Lehman and Kenneth O Stanley. 2008. Exploiting open-endedness to solve problems through the search for novelty.. In *ALIFE*. Citeseer, 329–336.

[27] Hod Lipson, Vytas Sunspiral, Josh Bongard, and Nicholas Cheney. 2016. On the difficulty of co-optimizing morphology and control in evolved virtual creatures. In *Artificial Life Conference Proceedings 13*. MIT Press, 226–233.

[28] Stuart Lloyd. 1982. Least squares quantization in PCM. *IEEE Transactions on Information Theory* 28, 2 (1982), 129–137.

[29] Eric Medvet and Alberto Bartoli. 2021. GraphEA: a Versatile Representation and Evolutionary Algorithm for Graphs. In *Workshop on Evolutionary and Population-based Optimization (Part of AIxIA)*.

[30] Eric Medvet, Alberto Bartoli, Andrea De Lorenzo, and Giulio Fidel. 2020. Evolution of distributed neural controllers for voxel-based soft robots. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*. 112–120.

[31] Eric Medvet, Alberto Bartoli, Andrea De Lorenzo, and Stefano Seriani. 2020. 2D-VSR-Sim: A simulation tool for the optimization of 2-D voxel-based soft robots. *SoftwareX* 12 (2020).

[32] Eric Medvet, Alberto Bartoli, Andrea De Lorenzo, and Stefano Seriani. 2020. Design, Validation, and Case Studies of 2D-VSR-Sim, an Optimization-friendly Simulator of 2-D Voxel-based Soft Robots. *arXiv* (2020), arXiv–2001.

[33] Karine Miras, Eliseo Ferrante, and AE Eiben. 2020. Environmental influences on evolvable robots. *PloS one* 15, 5 (2020), e0233848.

[34] Jean-Baptiste Mouret and Konstantinos Chatzilygeroudis. 2017. 20 years of reality gap: a few thoughts about simulators in evolutionary robotics. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. 1121–1124.

[35] M Naya-Varela, A Faina, and RJ Duro. 2021. Morphological Development in robotic learning: A survey. *IEEE Transactions on Cognitive and Developmental Systems* (2021).

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

[37] Paolo Pagliuca and Stefano Nolfi. 2020. The Dynamic of Body and Brain Co-Evolution. *arXiv preprint arXiv:2011.11440* (2020).

[38] Franz Rothlauf. 2006. Representations for genetic and evolutionary algorithms. In *Representations for Genetic and Evolutionary Algorithms*. Springer, 9–32.

[39] Franz Rothlauf and David E Goldberg. 2003. Redundant representations in evolutionary computation. *Evolutionary Computation* 11, 4 (2003), 381–415.

[40] Eivind Samuelsen and Kyrre Glette. 2014. Some distance measures for morphological diversification in generative evolutionary robotics. In *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation*. 721–728.

[41] Fernando Silva, Miguel Duarte, Luís Correia, Sancho Moura Oliveira, and Anders Lyhne Christensen. 2016. Open issues in evolutionary robotics. *Evolutionary computation* 24, 2 (2016), 205–236.

[42] Edward H Simpson. 1949. Measurement of diversity. *Nature* 163, 4148 (1949), 688–688.

[43] Giovanni Squillero and Alberto Tonda. 2016. Divergence of character and premature convergence: A survey of methodologies for promoting diversity in evolutionary optimization. *Information Sciences* 329 (2016), 782–799.

[44] Giovanni Squillero and Alberto Tonda. 2018. Promoting diversity in evolutionary optimization: why and how. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. 998–1016.

[45] Kenneth O Stanley and Risto Miikkulainen. 2002. Evolving neural networks through augmenting topologies. *Evolutionary computation* 10, 2 (2002), 99–127.

[46] Xin Sui, Hegao Cai, Dongyang Bie, Yu Zhang, Jie Zhao, and Yanhe Zhu. 2020. Automatic generation of locomotion patterns for soft modular reconfigurable robots. *Applied Sciences* 10, 1 (2020), 294.

[47] Michael A Tabak, Mohammad S Norouzzadeh, David W Wolfson, Steven J Sweeney, Kurt C VerCauteren, Nathan P Snow, Joseph M Halseth, Paul A Di Salvo, Jesse S Lewis, Michael D White, et al. 2019. Machine learning to classify animal species in camera trap images: Applications in ecology. *Methods in Ecology and Evolution* 10, 4 (2019), 585–590.

[48] Jacopo Talamini, Eric Medvet, Alberto Bartoli, and Andrea De Lorenzo. 2019. Evolutionary Synthesis of Sensing Controllers for Voxel-based Soft Robots. In *Artificial Life Conference Proceedings*. MIT Press, 574–581.

[49] David Tilman, Michael Clark, David R Williams, Kaitlin Kimmel, Stephen Polasky, and Craig Packer. 2017. Future threats to biodiversity and pathways to their prevention. *Nature* 546, 7656 (2017), 73–81.

[50] Michael Wainberg, Babak Alipanahi, and Brendan J Frey. 2016. Are random forests truly the best classifiers? *The Journal of Machine Learning Research* 17, 1 (2016), 3837–3841.

[51] Mark Yim, Wei-Min Shen, Behnam Salemi, Daniela Rus, Mark Moll, Hod Lipson, Eric Klavins, and Gregory S Chirikjian. 2007. Modular self-reconfigurable robot systems [grand challenges of robotics]. *IEEE Robotics & Automation Magazine* 14, 1 (2007), 43–52.