

Premature convergence in morphology and control co-evolution: a study

Luis Eguiarte-Morett¹  and Wendy Aguilar² 

Adaptive Behavior
2024, Vol. 32(2) 137–165
© The Author(s) 2023



Article reuse guidelines:
sagepub.com/journals-permissions
DOI: [10.1177/10597123231198497](https://doi.org/10.1177/10597123231198497)
journals.sagepub.com/home/adb



Abstract

This article addresses the co-evolution of morphology and control in evolutionary robotics, focusing on the challenge of premature convergence and limited morphological diversity. We conduct a comparative analysis of state-of-the-art algorithms, focusing on QD (Quality-Diversity) algorithms, based on a well-defined methodology for benchmarking evolutionary algorithms. We introduce carefully chosen indicators to evaluate their performance in three core aspects: task performance, phenotype diversity, and genotype diversity. Our findings highlight MNSLC (Multi-BC NSLC), with the introduction of aligned novelty to NSLC (Novelty Search with Local Competition), as the most effective algorithm for diversity preservation (genotype and phenotype diversity), while maintaining a competitive level of exploitability (task performance). MAP-Elites, although exhibiting a well-balanced trade-off between exploitation and exploration, fall short in protecting morphological diversity. NSLC, while showing similar performance to MNSLC in terms of exploration, is the least performant in terms of exploitation, contrasting with QN (Fitness-Novelty MOEA), which exhibits much superior exploitation, but inferior exploration, highlighting the effects of local competition in skewing the balance toward exploration. Our study provides valuable insights into the advantages, disadvantages, and trade-offs of different algorithms in co-evolving morphology and control.

Keywords

Evolutionary robotics, morphology and control co-evolution, premature convergence, morphological diversity, comparative analysis

Handling Editor: Geoff Nitschke, University of Cape Town, South Africa

1. Introduction

The automated design and construction of adaptive and autonomous machines, such as autonomous robots, remains a challenging problem that has yet to be fully solved. Currently, the only known process capable of producing machines that are fully adaptive and autonomous is biological evolution (Doncieux et al., 2015).

In the field of evolutionary robotics, evolutionary algorithms (which are part of the broader class of population-based metaheuristic algorithms) are employed to optimize various aspects of autonomous robots. These algorithms apply the principles of selection, variation, and inheritance inspired by natural evolution to the task of robot design, treating robots as artificial autonomous organisms capable of developing their own skills through interactions with the environment, without human intervention. When designing robots of this kind, multiple factors need to be taken into account concurrently, such as their morphology, sensors, actuators, control architecture,

and control policies, all of which interact to determine their behavior.

Two of the most important aspects that can be optimized are morphology (body) and control (brain). Morphology refers to the physical shape and structure of the robot, while control or control policy refers to the strategy for controlling the behavior that allows the robot to perform a certain task. The co-evolution of morphology and control, where these

¹Posgrado en Ciencia e Ingeniería de la Computación, Universidad Nacional Autónoma de México, Mexico City, Mexico

²Department of Computer Science, Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas, Universidad Nacional Autónoma de México, Mexico City, Mexico

Corresponding author:

Wendy Aguilar, Department of Computer Science, Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas, Universidad Nacional Autónoma de México, Circuito escolar, Ciudad Universitaria, Mexico City 04510, Mexico.

Email: weam@turing.iimas.unam.mx

two aspects of the robot mutually influence each other, is a key concept in evolutionary robotics. The morphology and control of a robot are deeply interdependent, and any changes in one have the potential to significantly influence the other. That's why both aspects must be considered as part of a whole in order to exploit existing interdependencies, which is quite different from the traditional approach to robotics that tends to be reductionist, aiming to keep the design process as modular as possible by focusing on one aspect of the robot at a time while ignoring the rest (Doncieux et al., 2015). The idea here is that the coupled dynamics between the body, the brain, and the environment of an agent result in the emergence of intelligent behavior, which is known as embodied (artificial) intelligence (Auerbach & Bongard, 2010). It's worth mentioning that since modifying the controllers and morphology of physical robots in this optimization process would be not only impractical but also costly, evolutionary robotics often relies on simulations using virtual creatures to perform this process, and subsequently manufactures physical robots based on the optimized designs.

An advantage associated with the use of evolutionary algorithms is their ability to be used as optimizers for any aspect of the robot, having only at their disposal a costly evaluation function, without any gradient information for it, while a disadvantage is the possibility of the algorithm converging to a local optimum (premature convergence) or not converging at all (Lipson et al., 2016). This is a widely recognized issue, commonly observed in evolutionary computing, and it is inherited in this domain due to the highly intricate nature of the problem, resulting in a rugged fitness landscape (Lipson et al., 2016). The prevalent issue observed in co-evolving morphology and control in virtual creatures or robots is that the evolution of morphology stagnates early on, resulting in a subset of very similar morphologies (Joachimczak et al., 2016; Kriegman et al., 2018; Lipson et al., 2016). This means that the potential benefits of co-evolving both aspects are lost, and optimization occurs only in the controllers. It has been found that this loss of morphological diversity is caused by the fact that morphological changes from one generation to another disrupt the optimized controllers, leading to individuals with recent morphological changes being discarded by selection pressure aimed at maximizing fitness in the chosen task (such as walking, swimming, goal-following, obstacle avoidance, etc.).

Some promising directions pursued previously include the imposition of mechanisms to protect morphological diversity, such as temporarily reducing selection pressure on recent morphological changes (Cheney et al., 2018), using Quality-Diversity (QD) algorithms such as MAP-Elites, defining a set of desired morphological characteristics in which quality is sought within the diversity itself (Nordmoen et al., 2020), and co-evolving control and

morphology together in a first phase, followed by evolving only the controller in a second phase to adapt to recent changes in morphology (Nygaard et al., 2017). All these proposals provide promising contributions and results, however, many of them often measure the benefits only against using a fitness evaluation function in a traditional evolutionary algorithm as a reference, and little has been done to compare various approaches that explicitly aim to mitigate the problem of premature convergence of morphology with respect to control, under the same framework, with different indicators and a well-defined benchmarking methodology for optimization algorithms.

In this article, we compare some of the most representative algorithms that have been used in the state of the art to mitigate the premature convergence problem of morphology with respect to control in evolutionary robotics. Our approach is experimental in nature, as we rigorously assess these algorithms under a novel, unified statistical framework, with respect to a series of relevant indicators. In particular, we aim to address the following research questions:

- How does an evolutionary algorithm that lacks any type of morphological diversity protection compare to other algorithms that do have it, in terms of the phenomenon of premature convergence of morphology with respect to control?
- Does the problem of premature convergence affect different state-of-the-art algorithms in the same way?
- How does the addition of morphological novelty as a secondary objective affect the fitness of solutions and the phenomenon of premature convergence of morphology with respect to control?
- How does the introduction of global selection pressure affect a QD algorithm? Does it still maintain the desirable characteristics of QD algorithms? How does it compare to a multi-BC QD algorithm in terms of the phenomenon of premature convergence and the fitness of solutions?
- How does the use of local competition instead of global fitness affect the fitness of solutions and the phenomenon of premature convergence of morphology with respect to control?
- How does the use of genetic diversity instead of crowding distance affect the solution space?
- Can adding a novelty objective based on a quality-driven behavior characterizer mitigate the issue of slow progress in QD algorithms when the diversity of interest (morphological) is unaligned with quality? Do any desirable characteristics of QD algorithms get lost?

The article is organized as follows. Firstly, in Section 2, we describe the main state-of-the-art strategies based on

diversity preservation that have been proposed to address the issue of premature convergence in the co-evolution of morphology and control. Next, in Section 3, we provide a detailed account of the experiments conducted to systematically analyze the aforementioned strategies. Subsequently, in Section 4, we introduce a series of performance indicators, which were used to compare the results obtained from the experiments presented in Section 5. Moving forward, in Section 6, we present a methodology based on voting theory to select the optimal algorithm, with a focus on fairness and the careful consideration of multiple pertinent factors. Lastly, in Section 7, we present the conclusions.

2. Strategies for mitigating premature convergence in morphology and control co-evolution

The convergence of evolutionary algorithms typically occurs when the population reaches a single solution or a set of very similar solutions. This is reflected as a loss of diversity in the solutions and causes the population to stop progressing.

The loss of diversity often happens when individuals in the population converge around local or global optima. When an evolutionary run has converged to local optima, it is said to be trapped in local optima, or in other words, it has experienced premature convergence. This problem can be attributed to a bias toward exploitation in the trade-off between exploration and exploitation. Such bias becomes more or less relevant depending on the problem, with increased relevance in problems with multiple local optima or highly rugged fitness functions, such as those encountered in the co-evolution of morphology and control. It is crucial to have strategies that maintain good exploration throughout the evolutionary run when co-evolving morphology and control in virtual creatures and robots.

In this section, we describe the main state-of-the-art techniques used to preserve diversity, serving as strategies to mitigate the premature convergence problem.

2.1. Fitness sharing

This diversity preservation technique involves scaling an individual's fitness inversely proportional to its similarity with respect to the other individuals in the population so that the fitness function is modified to penalize those individuals that are similar. In [Mouret & Doncieux, \(2009\)](#), similarity is defined as

$$\phi(\delta(i,j)) = \begin{cases} 1 - \frac{\delta(i,j)}{\sigma_{sh}} & \text{if } \delta(i,j) < \sigma_{sh} \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

Thus, the modified fitness of an individual i is defined as

$$f'_i = \frac{f_i}{c_i} \quad (2)$$

$$c_i = \sum_{j=1}^n \phi(\delta(i,j)), \quad (3)$$

where c_i is known as niche counting, σ_{sh} is a parameter defined by the user known as niche radius, and δ is a distance metric between solutions.

2.2. Speciation

Another approach to preserving diversity in evolutionary algorithms is to divide the population into species based on the similarity between individuals, such that:

- Each species has a representative genome that comes from the previous generation.
- It is decided whether an individual belongs to a certain species based on the niche radius σ_{sh} . Individual i belongs to the niche or species represented by the genome of individual j if $\delta(i,j) < \sigma_{sh}$.
- The genome of an individual is placed in the first niche where the previous condition is satisfied.
- If a genome does not belong to any niche, a new species represented by that genome is created.

Speciation can be carried out in combination with fitness sharing and crossing constraints, as happens in the NEAT algorithm ([Stanley & Miikkulainen, 2002](#)). However, it should be noted that it is not necessary to do so, and each one can be used as a separate technique.

2.3. Crossover operator restriction

By limiting the crossover operator, this technique allows only individuals within the same species to be crossed or, alternatively, permits the crossing of only those individuals belonging to different species.

2.4. Diversity as an objective

Using a multi-objective framework, it is possible to include a diversity metric as an additional objective to maximize. This diversity can be applied in both the variable and objective space and is typically problem-dependent.

Previous research has demonstrated that the best results are achieved by measuring the average distance d between an individual and the rest of the population. Consequently, for the purposes of this article, the diversity D of an individual x with respect to the population in generation n (P_n) is defined as

$$D(x) = \frac{1}{|P_n|} \sum_{y \in P_n} d(x, y). \quad (4)$$

In the context of the co-evolution of morphology and control of virtual creatures, the distance between individuals can be defined in various ways depending on the representation and/or subsequent phenotypic expression they have. The choice of distance metric can have a significant impact on the co-evolutionary process and the resulting solutions. As mentioned in [Mouret & Doncieux \(2009\)](#), if the genotype is represented as a neural network, calculating the distance between two genotypes is an extremely complex problem since computing the distance between graphs is NP-Complete in itself. However, this problem can be reduced by choosing a vector $\mathbf{v}^{(i)}$ that expresses desirable characteristics of the network, subject to a notion of distance that is suitable for the problem at hand.

For example, in [Mouret \(2011\)](#), several algorithms are compared, including the integration of diversity and fitness under a multi-objective paradigm, in the task of evolving controllers (neural networks) for two-wheeled mobile robots to reach a goal in an obstacle environment.

The distance between two individuals can be defined in terms of what is called the behavior vector. For example, the behavior vector \mathbf{b} of an individual x can be defined as

$$\mathbf{b}(x) = \mathbf{p}_r(T), \quad (5)$$

where $\mathbf{p}_r(T)$ is the final position of the robot in a simulation of duration T . The distance between individual x and some other individual y is defined as the Euclidean distance between their behavior vectors.

In [Nordmoen et al. \(2020\)](#), both morphology and control of modular robots are evolved for the task of walking the farthest distance possible. The morphology is given by a direct encoding based on trees, where each node in the tree is a module and each edge is a connection between two modules. There are two types of modules, non-mobile and joints, each supporting up to 5 and 3 child nodes, respectively. The control system consists of all individual controllers of each joint, and each controller is defined by 5 parameters. In this case, it is desired to protect morphological diversity, so the behavior vector of an individual x in the space \mathcal{Q} of all robots that can be represented with the mentioned encoding is defined as

$$\mathbf{b}(x) = (m_x, j_x), \quad (6)$$

where m_x is the number of non-mobile modules and j_x is the number of mobile modules. The distance between two individuals x and y is defined as

$$d_b(x, y) = 1 - e^{|\mathbf{b}(x) - \mathbf{b}(y)|}. \quad (7)$$

The diversity is defined in the same way as in equation (4). It should be noted that the distance defined in this case is

a function $d_b: \mathbb{R}^2 \rightarrow [0, 1] \times [0, 1] \subset \mathbb{R}^2$, so the multi-objective formulation is defined as

$$\max_{x \in \mathcal{Q}} (\mathbf{F}(x)),$$

$$\text{where } \mathbf{F}(x) = \begin{bmatrix} f(x) \\ D_m(x) \\ D_j(x) \end{bmatrix}.$$

2.5. Divergent search

The ideal of evolutionary computation to replicate the complexity and sophistication achieved in nature and use it as an optimizer is undoubtedly an extremely ambitious goal. However, this ambition has led to the search for more complex encodings and algorithms. According to this ideal, it should be enough to use selection pressure correctly toward an arbitrarily ambitious objective, but the results, especially in evolutionary robotics, tell us otherwise ([Pugh et al., 2016](#)).

Despite strong criticism from purists in the fields of machine learning and optimization, labeling it as an ad hoc method ([Pugh et al., 2016](#)), a question stands out as motivation to continue exploring the alternative of artificial evolution: how has nature been able to generate intelligence, and if we look at our best efforts to create artificial intelligent systems with traditional optimization, despite being better than humans in some tasks, they often fall short in areas where human cognition excels?

Divergent search proposes that perhaps the idea of evolution has been applied in a somewhat incorrect way. Natural evolution is divergent, rather than convergent. This idea gives rise to a new vision: evolution as a machine for diversification instead of optimization. This new perspective suggests that the problem lies in the abstraction of evolution that has been used all this time. If we look at natural evolution, it is clear that there is no optimization involved, and there is no clear objective except for a notion of reward for individuals that are not only better but also different (evolutionary niches: some organisms that are sufficiently different from their predecessors may enjoy reduced competition, making them more likely to survive).

2.6. Novelty search

It is a radicalization of the previous idea, completely disregarding optimization toward a particular goal (e.g., the fastest robot, the most efficient, the best at solving mazes), and directing the search solely through a notion of novelty as a consequence of diversity ([Pugh et al., 2016](#)). In Novelty Search, no fitness function based on a specific goal is optimized to explore the search space. Instead, a novelty

metric of the individual is rewarded, allowing us to know how far it is from the rest of the population (either in the search or solution space) (Lehman & Stanley, 2011a). A novelty metric that allows us to measure the dispersion at any point in the space in question is

$$N(x) = \frac{1}{k} \sum_{i=0}^k d(x, \mu_i). \quad (8)$$

This refers to the average distance of the individual x to its k nearest neighbors. Here, $d(x, \mu_i)$ represents a certain distance metric dependent on the domain. The greater the average distance to its k nearest neighbors, the more dispersed the point x is located in a particular region (Lehman & Stanley, 2011a).

Regarding the implementation, to direct the search process toward more dispersed regions, a novelty archive $\Lambda \subset \mathcal{Q}$ of previously produced solutions that meet a minimum novelty threshold is usually incorporated. The novelty of each individual $x \in P_j$ is evaluated by choosing its k nearest neighbors \mathbb{I} from the solutions in the population of the j -th generation $P_j \subset \mathcal{Q}$ and Λ (Omelianenko, 2019), that is, $\mathbb{I} = \{\mu_0 = x, \mu_1, \dots, \mu_i, \dots, \mu_k\} = \text{KNN}(x, P_j \cup \Lambda, d, k)$.

Novelty search has been successful in some domains, especially those in which the fitness evaluation space is considered “deceptive” (problems where objective-based evaluations fail), as objective-based evaluation is susceptible to getting trapped in local minima in which the evolutionary process converges to a single solution, losing diversity. An example of a problem where NS has been successfully applied is in the production of agents capable of navigating mazes. In particular, navigation in an old city whose map is unknown can be thought of as a deceptive problem, given that they often have an irregular path structure (Omelianenko, 2019). Novelty search has demonstrated that the evolutionary ability for diversification can be harnessed to find solutions close to global optima. In contrast to the traditional notion in evolutionary computation of survival of the fittest, novelty search emphasizes survival of the most novel (Lehman & Stanley, 2011a).

Similarly to adding diversity as an additional objective, novelty can also be incorporated as an extra objective. This was demonstrated by Mouret (Mouret, 2011). In this study, the multi-objective regimes of diversity+fitness and novelty+fitness were compared against novelty and fitness as single objectives respectively in evolving neural network controllers for mobile robots in the task of solving a maze. The best results were obtained using novelty+fitness, although only marginally better than diversity+fitness.

2.7. Quality-diversity

The problem with the Novelty Search approach is that it uses diversity as a means to achieve the usual end of finding

a global minimum (Pugh et al., 2016), that is, as an optimizer. Moreover, the idea of completely eliminating objectives means that there is no notion of good and bad solutions, which seems to be crucial in most applications. In contrast to the search for global optima, Quality-Diversity (QD) methods seek to emulate the way evolution works in nature, so that in a single run, diverse ways of solving the same problem are found, all equally viable. The objective of QD algorithms is to find a maximally diverse collection of individuals with respect to the space of possible solutions, where each member is maximally fit in its own evolutionary niche (Pugh et al., 2016). QD algorithms balance the search for diversity, which finds many evolutionary niches, with local searches for quality within each niche. Within a niche, competition is focused on individuals, thus producing the best possible individuals locally within the niche. This type of search is capable of finding diverse solutions while improving existing ones (Pugh et al., 2016).

In QD, the diversity measurement of an individual x and its position in the space of features or behaviors \mathcal{B} is given through its characterization in that space with a feature vector $\mathbf{b}(x)$, which represents an abstraction of the actions and/or characteristics of the phenotype of x in the context of the problem being solved. It is worth mentioning that this abstraction is equivalent to that of the behavior vector used in Mouret & Doncieux (2009). Other names for this in the QD literature are: Behavior Space and Behavior Characterization (BC) (Pugh et al., 2016), and Descriptor Space and Behavioral Descriptor (BD) (Chatzilygeroudis et al., 2020; Cully & Demiris, 2018). The choice of the features of interest is up to the user, but it should seek to be representative of the problem and how potential solutions solve it (Chatzilygeroudis et al., 2020). The idea of being able to represent solutions through a feature vector is similar to a dimensionality reduction in machine learning, in the sense of re-expressing data from a high-dimensional and difficult-to-handle space to a lower-dimensional subspace that is easier to handle. The big difference is that in this case, it is a model-free projection where the dimensions and characteristics represented in them are chosen manually, while in machine learning and deep learning, algorithms such as the well-known principal component analysis are used, which perform a linear projection from a high-dimensional space to a lower-dimensional one, or deep neural networks, which can act as feature selectors in addition to learning models. In certain models of deep learning, such as autoencoders, a pair of neural networks are used and trained to compress data from a high-dimensional feature space to a low-dimensional latent space (encoder network), and to decompress from the latent space to the original feature space (decoder network). The encoder component can be seen as a non-linear generalization of PCA, with the latent space as an approximation of the principal components (Gaier et al., 2020).

The general problem to be optimized in QD algorithms is to find, for each point $\mathbf{b} \in \mathfrak{B}$, the parameters x with the highest fitness value (Chatzilygeroudis et al., 2020), that is

$$\begin{aligned} x^* &= \arg \max F(x) \\ \text{s.t. } \mathbf{b} &= \mathbf{b}(x) \\ \forall \mathbf{b} &\in \mathfrak{B}. \end{aligned} \quad (9)$$

In general, QD algorithms work by sampling with a certain granularity \mathfrak{B} , and for each sampled region, obtaining the best solutions. In these algorithms, diversity has a higher priority than quality, and therefore, a greater number of regions in the space are visited, even if they may have less potential in terms of quality.

In QD, the feature space \mathfrak{B} is divided into t niches, $\mathbf{N} = N_1, N_2, \dots, N_t$. Where \mathbf{N} must cover \mathfrak{B} . Each point \mathbf{b}_j belongs to a niche N_i , according to a proximity rule. The objective of QD algorithms is to maximize a quality metric Q with respect to each niche (Pugh et al., 2016).

The promise of QD algorithms is that, in addition to being competent in performing tasks traditionally oriented toward optimization, they can surpass what is possible with traditional optimization algorithms (Pugh et al., 2016). These types of algorithms have also been called “illumination algorithms” because they reveal (or “illuminate”) the best achievable performance in each region of the phenotype space.

It should be noted that these types of algorithms are different from simply using a multi-objective framework and adding novelty or diversity as an additional objective. The distinction lies in the locality of the quality search in QD. Unlike the purely multi-objective case, where the notion of quality of an individual x is global, in QD, the notion of quality is local to the niche N_i , where the phenotype of x is according to $\mathbf{b}(x)$, and therefore relative to the quality of the phenotypes found so far in N_i .

2.8. Novelty Search with Local Competition

The NSLC algorithm aims to use a multi-objective framework, specifically with two objectives: quality and novelty. However, the approach to multi-objective optimization is somewhat special in this case. As explained, in QD algorithms, global optimization is not sought, as would happen in a traditional multi-objective approach. Instead, the goal is to find niches that are as diverse as possible through the novelty objective exposed by Novelty Search, and within each niche obtain high-quality solutions through the concept of “local competition,” which consists of evaluating the individual according to the number of individuals that it outperforms in terms of fitness within its own niche, that is, it locally looks for the best within the

same niche. There is no promotion of a concept of “best” globally, and this algorithm does not focus on inherently better niches.

In the work of Lehman and Stanley (Lehman & Stanley, 2011b), where this algorithm was first proposed and applied to morphology and control co-evolution, Pareto-based non-dominated sorting is used with two objectives: movement quality evaluated as a certain distance traveled (in this case the Euclidean distance) by an individual in a given simulation time, and novelty evaluated as a subroutine through Novelty Search in a three-dimensional morphological space (in this case, these morphological characteristics make up a behavioral characterization) composed of height, mass, and number of active joints (actuators). The novelty metric is given by the average Euclidean distance to the k nearest neighbors (in the same way as in pure NS), and the quality objective is given by the proportion of those same k nearest neighbors that are surpassed with respect to the movement quality metric. In NSLC, niches emerge when calculating the k nearest neighbors to the individual, so it can be said that the quality of a certain individual is calculated locally to the niche as the proportion of individuals in the niche that are surpassed by it.

The niche N of an individual x is defined as the k nearest neighbors to the individual in the \mathfrak{B} space according to a distance measured with respect to \mathbf{b} : $d_{\mathbf{b}}(\cdot, \cdot)$, that is

$$\begin{aligned} N_x &= \{x = \mu_0, \mu_1, \dots, \mu_i, \dots, \mu_k\}, \\ &= KNN(x, P_j \cup \Lambda, d_{\mathbf{b}}, k). \end{aligned} \quad (10)$$

NSLC can be formulated as a two-objective multi-objective problem, as follows

$$\begin{aligned} \max_{x \in Q} \left\{ \mathbf{F}(x) = \begin{bmatrix} Q(x) \\ N(x) \end{bmatrix} \right\}, \\ Q(x) = k - \|\{\mu_i \forall i \in \{0, \dots, k\} \mid \mu_i \in N_x, \\ x \in Q \ \& \ F(x) < F(\mu_i)\|\|, \\ N(x) = \frac{1}{k} \sum_{i=0}^k d_{\mathbf{b}}(x, \mu_i). \end{aligned} \quad (11)$$

In Lehman & Stanley (2011b), a substantial trade-off was found: despite obtaining more diverse results in terms of morphology, the fittest individuals obtained by NSLC were slightly worse than the fittest individuals from a traditional evolutionary algorithm. One possible hypothesis proposed is that unlike what happens in the natural world, in artificial evolution, populations with fixed sizes are used, and this limits the ability of each niche to explore for better individuals since exploring new niches necessarily involves redirecting resources away from niches that have already been discovered. This is another example of the complexity ceiling that the field has encountered.

2.9. MAP-Elites

Unlike NSLC, where niches emerge as part of the NS search process and the novelty archive it maintains, in the MAP-Elites (Multi-dimensional Archive of Phenotypic Elites) algorithm, the behavioral space is explicitly subdivided into niches (bins/containers) created by discretizing \mathcal{B} (Pugh et al., 2016). In this case, each bin stores the fittest individual (elite) produced so far, and unlike NSLC, the concept of population is discarded in each generation in favor of the set of elites discovered in each bin (Mouret & Clune, 2015). In this sense, it is not evolving a population, but the archive of elites (Cully & Demiris, 2018). The search process in MAP-Elites involves selecting an elite from some niche with uniform probability in each generation, and applying some variation operator (mutation or crossover) to the genotype. Afterward, the phenotype is obtained along with its corresponding feature vector, and evaluated. Then, based on each value of the phenotype's feature vector, it is determined which niche it belongs to. Finally, the fitness of the elite in the niche to which the new individual belongs is compared. If the fitness of the new individual is greater, it is mapped to the niche as the new elite individual (Mouret & Clune, 2015). As is often the case, this process is repeated until some problem-dependent stopping criterion is met. This process can be massively parallelized (Mouret & Clune, 2015) if a pessimistic locking scheme is considered, where each process blocks the choice of the bin, the mutation, and the potential substitution in the new bin. Batch processing can also be considered, where parallelism is implicit, similar to that found in population-based evolutionary algorithms.

It is noteworthy that in Chatzilygeroudis et al. (2020 and Cully & Demiris (2018), the authors propose a framework for the generalization of QD algorithms. This framework abstracts the main components common to all QD algorithms and provides an example of how it can be used to generate new QD algorithms. Additionally, the authors distinguish between two main types of archives present in QD algorithms: the NSLC archive, referred to as an unstructured archive or simply an archive, and the MAP-Elites archive, referred to as a "grid."

2.10. Multi-BC QD

One important aspect of a suitable feature space is its alignment with the concept of quality. Alignment in the context of a BC is defined as the degree to which finding diversity also leads to finding higher fitness (Pugh et al., 2016). It has been found that seeking diversity using a BC with higher alignment results in better outcomes simply because the search for diverse solutions also leads to better quality solutions, thereby mitigating the problem of premature convergence. Two examples where the effects of having aligned and unaligned BCs can be distinguished are:

- In the domain of maze-solving agents, the goal is to find the topology and weights of a neural network that controls the agent through inputs provided by proximity sensors (for obstacle detection) and navigation sensors (that inform the robot the direction of the goal) and outputs that control linear and angular acceleration. An associated BC simply based on the final position of the agent in the maze will be highly aligned, since an agent that continuously finds new positions will eventually reach the final position in the maze (Pugh et al., 2016).
- In the co-evolution of morphology and control of virtual creatures, the search for novel morphologies in many cases does not lead to better walking cycles (Lehman & Stanley, 2011b) (at least in the short term and without diversity protection (Cheney et al., 2018), and without introducing development (Kriegman et al., 2018)). However, it has been widely pursued to address the problem of premature convergence of morphology with respect to control, since premature convergence is often a consequence of a loss of diversity.

The degree of alignment is extremely difficult to measure a priori; however, it can be anticipated by considering the following two properties:

1. Each behavior is associated with a reduced range of fitness values.
2. The maximum fitness value in adjacent regions of the BC space is correlated.

In Pugh et al. (2016), it is suggested that when diversity is sought using unaligned BCs, QD algorithms fail to obtain optimal solutions, and in more complex problems, they may completely fail to make any progress toward even viable solutions. This is why they propose the use of new variants of MAP-Elites and NSLC algorithms capable of handling multiple BCs. The following are the most representative variants.

2.10.1. Multi-BC Novelty Search with Local Competition. This extension to the NSLC algorithm incorporates a third objective, a novelty metric aligned with quality, and can be especially useful in domains where the novelty used in NSLC is unaligned with quality. This second BC would incorporate desirable features that, according to the decision maker's judgment, could represent a potential alignment with quality. This leads to potential variations with more than two BCs, multiple notions of local competition, etc., using a multi-objective evolutionary algorithm capable of dealing with the curse of dimensionality. It should be noted that with each BC introduced, a new novelty archive will have to be introduced, along with the distance defined for that BC in order to calculate novelty.

2.10.2. Multi-BC Map-Elites (ME-ME). Similarly to MNSLC, this algorithm introduces a new BC aligned with quality, hence a new grid. In each generation, an equal number of parents are selected from each grid, and their children are mapped to both archives independently (a child may be saved in both, one, or neither archive).

3. Experiments

In this section, we present the experimental setup of this study, while following a framework as general as possible for conducting experiments with virtual creatures and evolutionary robotics.

3.1. Representation

We chose to adopt Lipson et al.'s approach (Lipson et al., 2016) as the basis for the morphology and control representation in our study. This approach employs a diploid genotype that encodes both morphology and control using two CPPN (Compositional Pattern Producing Network) neural networks (Stanley, 2007). These networks are structured as weighted, connected, acyclic graphs, where each node or neuron corresponds to a perceptron (McCulloch & Pitts, 1943).

The use of two different CPPNs allows for variations in one aspect without affecting the other, direct measurement of genetic diversity and complexity of the neural networks of each aspect, and proposing relevant indicators to detect and measure the phenomenon of premature convergence of morphology relative to control. Additionally, as already observed in the literature (Cheney et al., 2014), the use of CPPNs as an indirect encoding tends to lead to better evolvability, as they are capable of producing morphologies and controllers with regular patterns.

For the CPPNs used, we have a region $\Omega \subset \mathbb{R}^3$ which refers to an arbitrarily bounded three-dimensional design space, which in our case is

$$\Omega = \begin{cases} x & \in [0, 1] \\ y & \in [0, 1] \\ z & \in [0, 1]. \end{cases} \quad (12)$$

3.1.1. Morphology. For the experiments conducted in this article, we chose to evolve robots with soft voxel-based morphologies (softbots). We chose these morphologies for their promise in terms of versatility, ease of evolution, and potential to be transferred from virtual to physical worlds. We used morphologies with two types of voxels, active and passive. The active voxels act as muscles or actuators, each performing expansions and contractions at the same frequency but with different phase offsets, allowing for movements similar to the propagation of a wave. A valid

morphology is created in a discrete design space with a fixed resolution of $5 \times 5 \times 5$, and is composed of a combination of any number of contiguous voxels.

The neural network $CPPN_m$ has a fixed number of inputs and outputs. It has four inputs (x, y, z, r) , corresponding to the three Euclidean coordinates of the three-dimensional design space, plus a radial coordinate (with the hope of encouraging the appearance of radial patterns). Similarly, it has two outputs, $(p, m) \in \mathbb{R}^2$, where p indicates the presence or absence of the voxel in the morphology, and m indicates the material (active or passive). The way in which a softbot s morphology is “grown” from its CPPN is by making individual queries to the neural network for each element of the design space. If $p \in (-\infty, 0)$, the voxel remains empty, and if $p \in [0, \infty)$, the voxel belongs to the morphology and is either passive if $m \in (-\infty, 0)$ or active if $m \in [0, \infty)$.

We can say that the CPPNs in these experiments are functions of (x, y, z) , and not r , by considering as always having an input neuron with an activation function $r(x, y, z) = \sqrt{x^2 + y^2 + z^2}$.

Henceforth, the evaluation of $CPPN_m$ can be expressed as a function $\vec{m}_s: \Omega \rightarrow \mathbb{R}^2$

$$\vec{m}_s(x, y, z) = CPPN_m(x, y, z) = \begin{bmatrix} p(x, y, z) \\ m(x, y, z) \end{bmatrix}. \quad (13)$$

Which furthermore, gives place to the expression of the morphology as

$$M_s(x, y, z) = \begin{cases} \text{Voxel}_{nil} & \text{if } p \in (-\infty, 0) \\ \text{Voxel}_p & \text{if } p \in [0, \infty) \& m \in (-\infty, 0) \\ \text{Voxel}_a & \text{if } p \in [0, \infty) \& m \in [0, \infty) \end{cases} \quad (14)$$

$$(x, y, z) \in \Omega.$$

3.1.2. Control. The control system is composed of all the active voxels within the morphology, which is defined as

$$\mathcal{A}(s) = \{M_s(x, y, z) | (x, y, z) \in \Omega \& M_s(x, y, z) = \text{Voxel}_a\} \quad (15)$$

Once the morphology is obtained, all the control system's voxels are taken, and given the coordinates of each one $\Omega_A = \{(x, y, z) | (x, y, z) \in \Omega \& M_s(x, y, z) = \text{Voxel}_a\}$, a second network $CPPN_c$ corresponding to the control is queried. The evaluation $CPPN_c(x, y, z)$ yields the control parameters for each active voxel.

In $CPPN_c$, the inputs are the same as in $CPPN_m$, and the outputs are also two, $(\phi, \tau) \in [0, 2\pi] \times [-1, 1]$, where ϕ corresponds to the phase shift that the actuator voxel has with respect to a global frequency f_g , and τ corresponds to the period. For the case of ϕ , it can take scaled values of $[0, 2\pi]$ radians, which is achieved by always connecting the output u to a neuron with a logistic activation function scaled by 2π

$$\phi(u(x, y, z)) = \frac{2\pi}{1 + e^{-u(x, y, z)}}. \quad (16)$$

The period τ is used to obtain the global frequency f_g , averaging the frequency obtained for each active voxel $f(x, y, z) = 1/\tau(x, y, z)$, $(x, y, z) \in \Omega_A$

$$f_g = \frac{10}{\|\Omega_A\| \sum_{(x, y, z) \in \Omega_A} f(x, y, z) + 1.5}. \quad (17)$$

The value of τ is obtained by connecting the output v of the neural network to a logistic activation function biased in the interval $[-1, 1]$, where $v(x, y, z) : \Omega_A \rightarrow \mathbb{R}$. The evaluation of τ is defined as follows

$$\tau(v(x, y, z)) = \frac{2}{1 + e^{-v(x, y, z)}} - 1. \quad (18)$$

Moreover, the calculation of f_g has discontinuities at $1/\|\Omega_A\| \sum_{(x, y, z) \in \Omega_A} f(x, y, z) = -2/3$ and $\tau(v(x, y, z)) = 0$.

Therefore, all softbots whose f_g calculation leads to numerical errors of divisions by zero will be evaluated with a fitness of zero.

Henceforth, the evaluation of $CPPN_c$ can be expressed as a function $\vec{c}_s : \Omega_A \rightarrow \{[0, 2\pi] \times [-1, 1]\}$

$$\vec{c}_s(x, y, z) = CPPN_c(x, y, z) = \begin{bmatrix} \phi(x, y, z) \\ \tau(x, y, z) \end{bmatrix}. \quad (19)$$

Which gives place to the expression of the controller's parameters as

$$C_s(x, y, z) = \begin{bmatrix} f = f_g \\ \phi = \phi(x, y, z) \end{bmatrix}, (x, y, z) \in \Omega_A. \quad (20)$$

The control system is like a mass-spring-damper system, where masses correspond to the centroids of voxels (nodes), and springs to the edges formed from the Delaunay triangulation, thereby the resting length L_0 is determined. Actuation is achieved by altering the resting length according to the following signal

$$L(t) = \left[1 + A \sin\left(\frac{2\pi t}{\tau_1} + \phi_1\right) + A \sin\left(\frac{2\pi t}{\tau_2} + \phi_2\right) \right] \cdot L_0.$$

where τ_1 , τ_2 and ϕ_1 , ϕ_2 are the periods and phases associated with the centroids at the end of the spring. In other words, modifying the resting length serves as an input signal to the system. The controllers are open-loop, as they do not take into account any error signals or sensor feedback.

3.1.3. Softbot definition. Per the definitions for morphology and control presented in Sections 3.1.1 and 3.1.2, it is possible to define a softbot s , such as the ones evolved in these experiments, based on its morphology (Equation (13))

and control (Equation (19)) genotypes, as well as its morphology (Equation (14)) and control (Equation (20)) phenotypes.

The genotype G_s of a softbot s is defined as

$$G_s = (CPPN_m, CPPN_c, \vec{m}_s(x, y, z), \vec{c}_s(x, y, z)). \quad (21)$$

And the phenotype P_s of s , is defined as

$$P_s = (M_s(x, y, z), C_s(x, y, z)). \quad (22)$$

With which, a softbot s is defined as

$$s = (G_s, P_s). \quad (23)$$

3.2. Task

The task chosen for all experiments in this work was locomotion in a straight line, undoubtedly the most common task in the literature on morphology and control co-evolution. However, it is not the only one, and in practice, locomotion toward an objective (Lan et al., 2018; Lan et al., 2021; Sims, 1994;), locomotion with obstacles and in different environments (Stensby et al., 2021), moving in confined spaces (Cheney et al., 2015), or competing to seize an object (Miconi & Channon, 2006; Sims, 1994) may be more of interest. Nonetheless, the chosen task is comparatively simple and reduced, and the interest of this research is not to explore different environments or tasks (although they are equally valid research avenues), but mainly to investigate the incidence of premature convergence of morphology with respect to control, and to benchmark different algorithms with mechanisms for protecting morphological diversity according to relevant indicators.

3.3. Simulated environment

The softbots were tested on a flat surface within the Voxelyze physics engine (Hiller & Lipson, 2014), which was accelerated by GPU using CUDA, and referred to as Voxcraft (Liu et al., 2020).

3.4. Algorithms and parameters

Firstly, we shall define the unrestricted softbot search space \mathbb{S}' . The design space, per its resolution, is composed of $5 \times 5 \times 5$ voxels, and hence has a volume of $5^3 = 125$ voxels. The space \mathbb{S}' is composed of all softbots which can be expressed in that design space. In order to be able to define the actual search space which was explored in the experiments \mathbb{S} , a restriction to \mathbb{S}' was put in place in order to avoid trivial morphologies.

With such restrictions, \mathbb{S} is defined as

$$\begin{aligned} \mathbb{S} &= \{s | s \in \mathbb{S}' \text{ y} \\ 7 \leq M_a(s) + M_p(s) \leq 112 \\ \lceil 0.1 \cdot (M_a(s) + M_p(s)) \rceil &\leq M_a(s) \\ \leq \lfloor 0.9 \cdot (M_a(s) + M_p(s)) \rfloor \}, \end{aligned} \quad (24)$$

where

$$\begin{aligned} M_a(s) &= \|\mathcal{A}(s)\|, \\ M_p(s) &= \|\mathcal{P}(s)\|, \\ M_p(s), M_a(s) : \mathbb{S}' &\rightarrow [0, 125] \subset \mathbb{N}, \\ \mathbb{S} &\subset \mathbb{S}'. \end{aligned} \quad (25)$$

Functions $\mathcal{P}(s)$ and $\mathcal{A}(s)$, are defined as

$$\begin{aligned} \mathcal{A}(s) &= \{s.P_s.M_s(x, y, z) | (x, y, z) \in \Omega \\ &\quad \& s.P_s.M_s(x, y, z) = \text{Voxel}_a\} \\ \mathcal{P}(s) &= \{s.P_s.M_s(x, y, z) | (x, y, z) \in \Omega \\ &\quad \& s.P_s.M_s(x, y, z) = \text{Voxel}_p\}. \end{aligned} \quad (26)$$

The experiments were conducted using five different algorithms: (i) Single-Objective Evolutionary Algorithm (SO), (ii) Fitness-Novelty MOEA (QN), (iii) Novelty Search with Local Competition (NSLC), (iv) Multi-dimensional Archive of Phenotypic Elites (MAP-Elites), and (v) Multi-BC NSLC (MNSLC), with 20 runs of each algorithm to obtain statistically significant results.

To identify the parameters with which the algorithms had to be run, especially those with diversity protection mechanisms, runs were performed on benchmark problems with multimodal functions instead of directly on the morphology and control co-evolution problem due to its high computational cost. It was of particular interest to find a set of parameters that balanced the fact that a higher number of individuals per generation would take advantage of GPU simulations. The QN, NSLC, and MAP-Elites algorithms were run on the Rastrigin, Ackley, Eggholder, and the fourth Schaffer Function benchmark problems, paying special attention to the parameters used for the latter, as the domain for which it is defined ($x_1, x_2 \in [-100, 100]$) is similar in size to the misaligned BC used for the number of passive and active voxels $M_p, M_a \in [0, 125]$. Once the parameters for these functions were identified, some test runs were done with fewer generations (200) to further refine the obtained parameters. By doing all this, the following parameters were established for all experiments:

- *population_size*: 100
- *generations*: 1000
- *design_space_resolution*: [5,5,5]
- *k* (number of nearest neighbors considered to calculate novelty between individuals in the archive and the current population): 200
- *novelty_threshold* (initial novelty threshold value): 12.

- *novelty_floor* (the minimum value that the novelty threshold can take): 1.
- *max_archive_size* (the maximum archive size): 1500.
- *min_archive_size* (the minimum archive size): 1.
- *grid_dimensions*: [25, 25].
- *Mutation*: $p_m = 1/2$. Both neural networks, $CPPN_m$ and $CPPN_c$ are subject to a mutation with probability p_m . A random selection is made from six different types of mutations:
 - Node Addition—a connection (u, v) of the CPPN is randomly selected, with a weight $w_o = w(u, v)$. A new node q is added between u and v , creating the path $u \mapsto q \mapsto v$, where the connections (u, q) and (q, v) have weights $w(u, q) = 1$ and $w(q, v) = w_o$. A random activation function is chosen.
 - Node Removal—a hidden node connected to another node u is randomly selected and removed, while ensuring that the connectivity of any path passing through u is maintained.
 - Connection Addition—a pair of nodes u and v is randomly selected from the network, such that the edge (u, v) does not already exist in the network, both are hidden nodes, and $u \neq v$. Once the node pair is chosen, the connection (u, v) is created, and a random weight is assigned, either $w(u, v) = -0.1$ or $w(u, v) = 0.1$.
 - Connection Removal—a connection is randomly selected and removed.
 - Activation Function Change—a hidden or output node is randomly selected, and its activation function is changed to a different one from its original function.
 - Weight Mutation—a connection (u, v) is randomly selected, and its original weight $w_o = w(u, v)$ is modified based on a normal distribution, resulting in a new weight $w(u, v) = \max(-1, \min(1, \mathcal{N}(w_o, 0.25)))$.

Two archives and two grids are used in the experiment. The two archives are employed to measure aligned and unaligned novelty. The two fitness and aligned novelty grids are used to assess coverage and QD-scores. In the case of the fitness grid, MAP-Elites also employs it for the search process instead of a population.

3.5. Implementation

PymOO (Blank & Deb, 2020), a multi-objective evolutionary algorithms framework implemented in Python, was used due to its strong object-oriented approach, modularity, and extensibility, allowing for clean, cohesive, and simple

integration of the specific implementations of Novelty Search, Novelty Search with Local Competition, and MAP-Elites into a generic evolutionary algorithm, as well as the variations of these algorithms, and the evaluation function based on Voxcraft simulations.

3.5.1. Single-Objective Evolutionary Algorithm (SO). We took as a baseline the simple single-objective evolutionary algorithm ($\mu + \lambda$). The problem to be solved was

$$\max_{s \in \mathbb{S}} (F(s) = \|\vec{r}_s(t = T)\|), \quad (27)$$

where $F : \mathbb{S} \rightarrow \mathbb{R}$

This algorithm does not provide any kind of protection for morphological diversity and serves as a starting point to identify the phenomenon of premature convergence of morphology with respect to control and compare it with other algorithms that do provide protection for morphological diversity.

3.5.2. Fitness-Novelty MOEA (QN). In this experiment, the problem to be solved was

$$\max_{s \in \mathbb{S}} (\mathbf{F}(s)),$$

$$\text{where } \mathbf{F}(s) = \begin{bmatrix} F(s) \\ N_u(s) \end{bmatrix} \quad (28)$$

and $\mathbf{F} : \mathbb{S} \rightarrow \mathbb{R}^2$.

The NSGA-II implementation from the PyMOO framework was used.

We aimed to answer the following questions using this algorithm:

- How does adding morphological novelty as a second objective affect the phenomenon of premature convergence of morphology with respect to control?
- How does adding morphological novelty as a second objective affect the fitness of the solutions?
- How does this multi-objective approach that uses global fitness compare to its QD counterparts that consider local fitness (NSLC and MNSLC) in terms of the phenomenon of premature convergence and the fitness of the solutions?

3.5.3. Novelty search with local competition. This is the same algorithm as the one described in [Section 2.8](#), the problem to be solved was

$$\max_{s \in \mathbb{S}} \left\{ \mathbf{F}(s) = \begin{bmatrix} Q(s) \\ N_u(s) \end{bmatrix} \right\}, \quad (29)$$

where $\mathbf{F} : \mathbb{S} \rightarrow \mathbb{N} \times \mathbb{R}$.

The NSGA-II implementation from the PyMOO framework was used, replacing the crowding distance

with the genetic diversity calculation described in equation (54).

The purpose of including this algorithm in the experiments is to provide a response to the following questions:

- How does using local competition instead of global fitness affect the phenomenon of premature convergence of morphology compared to the control?
- How does using local competition instead of global fitness affect the fitness of solutions?
- How does using genetic diversity instead of crowding distance affect the results?
- How does it compare to a QD algorithm with fitness-based selection pressure (ME) and another Multi-BC algorithm in terms of the phenomenon of premature convergence and the fitness of solutions?

3.5.4. Multi-dimensional Archive of Phenotypic Elites (MAP-Elites). MAP-Elites was implemented using the generic genetic algorithm proposed in [Eiben and Smith \(2003\)](#), taking into account the following considerations:

1. **Algorithm Initialization:** The initialization of MAP-Elites relies on the discretization of the feature space into bins or niches and the initialization of a number of individuals, denoted as P , which can be considered as the initial population. Each individual is placed in its corresponding bin based on $\mathbf{b}(x)$.
 - **Discretization of \mathfrak{B} :** Given a feature space \mathfrak{B} of dimension n , defined as

$$\mathfrak{B} = \{\mathbf{b} \in \mathbb{R}^n \mid l_i \leq \mathbf{b}_i \leq u_i \forall i \in \{1, \dots, n\}\}, \quad (30)$$

where l_i and u_i are the lower and upper limits of the i -th feature, defining the possible values for each feature. The discretization b_i of a certain feature \mathbf{b}_i , with a certain granularity or number of points q_i , is given by

$$b_i = \{l_i + k_i \Delta \mathbf{b}_i \mid k_i \in \{0, \dots, q_i - 1\}\}, \quad (31)$$

$$\text{where } \Delta \mathbf{b}_i = \frac{u_i - l_i}{q_i - 1}.$$

The discretization B of the space \mathfrak{B} is defined as follows

$$B = b_1 \times b_2 \cdots \times b_i \cdots \times b_n$$

- **Population initialization and placement of individuals** in the discretized space B is done as follows: A random sample of individuals of size P is taken. P genotypes are initialized randomly, expressed as phenotypes, and evaluated. In order to place an individual x in the discretization B , the feature vector $\mathbf{b}(x)$ is taken into consideration. The bin in which x should be placed is determined based on the range of

values to which it belongs in each characteristic. The number of bins or subdivisions per characteristic is determined by $s_i = q_i - 1$. Therefore, we have

$$\Delta \mathbf{b}_i = \frac{u_i - l_i}{s_i}. \quad (32)$$

If we want the index of the bin where an individual x falls in the i -th dimension, given the value of $\mathbf{b}(x)_i$, we can simply solve for k_i in the resulting expression obtained by substituting equation (32) into equation (31). Considering that the bin indices start at zero, the expression that determines the index of the bin in the i -th dimension where the individual x is located, given its feature vector $\mathbf{b}(x)$, is as follows

$$J_i = \min \left(\left\lceil \frac{(\mathbf{b}(x)_i - l_i)s_i}{u_i - l_i} \right\rceil, s_i - 1 \right). \quad (33)$$

With this, it is possible to convert a real-valued feature vector into an integer-valued feature vector, where each entry determines the bin or stencil it belongs to, based on the value of the corresponding feature. In other words

$$(b_1, b_2, \dots, b_i, \dots, b_n) \rightarrow (J_0, J_1, \dots, J_i, \dots, J_n).$$

With this mapping, we can obtain the index of the bin to which the individual x belongs by

$$I_{\mathbf{b}(x)} = J_0 + \sum_{j=1}^n \left(J_j \prod_{i=0}^{j-1} s_i \right). \quad (34)$$

2. Survival: Each individual in a population of offspring of size P is evaluated and placed in a bin based on its feature vector \mathbf{b} . If the offspring is fitter than the elite of the bin or the bin is empty, the offspring becomes the new elite.
3. Selection: In the original algorithm, each individual in a parent population of size P is chosen with uniform probability from the elites present in the grid. In these experiments, we made the modification described in (Cully & Demiris, 2018) to select parents proportionally to their fitness.

The modification in the selection process was made in order to investigate the following research questions:

- How does the introduction of global selection pressure affect a QD algorithm like MAP-Elites?
- Does it still maintain the desirable characteristics of QD algorithms?
- How does it compare to a Multi-BC QD algorithm?

3.5.5. Multi-BC NSLC (MNSLC). It's the same algorithm described in section 2.10.1. The problem being addressed is the following

$$\max_{s \in \mathbb{S}} \left\{ \mathbf{F}(s) = \begin{bmatrix} Q(s) \\ N_u(s) \\ N_a(s) \end{bmatrix} \right\}, \quad (35)$$

where $\mathbf{F}: \mathbb{S} \rightarrow \mathbb{N} \times \mathbb{R}^2$.

The behavior or aligned novelty function $N_a(s)$ is described in detail later in Section 4.1.2. With the introduction of this algorithm to the experiments, we ask the following two research questions:

- Can the addition of a third novelty objective to NSLC in a BC that is assumed to be aligned with quality mitigate the problem of slow progress in QD algorithms when the diversity of interest (morphological) is misaligned with quality?
- Are any desirable characteristics of QD algorithms lost, especially those related to premature convergence of morphology with respect to control?

4. Performance indicators

In this section, we provide a detailed account of the proposed indicators that were used to compare experiments involving the co-evolution of morphology and control in virtual creatures, with particular attention to the issue of premature convergence of morphology relative to control.

Originally, 31 indicators were introduced, grouped into three relevant aspects: phenotypic diversity, genotypic diversity, and task performance. However, upon conducting linear regressions on the entire distribution of experimental data for each pair of indicators and obtaining correlation coefficients, it was decided to discard any indicator from any pair that had a correlation coefficient of $r \geq .85$ in order to reduce the indicators to only those that provided the most information. This led to a reduction of 11 indicators, which are presented below.

4.1. Task performance indicators

4.1.1. Fitness. The fitness function for the walking pattern learning task was the distance traveled by the softbot in a simulation of $T = 5$ [s], $\|\vec{r}_s(t = T)\|$, where $\|\cdot\|$ refers to the L_2 norm or Euclidean distance.

4.1.2. Behavioral or aligned novelty. A notion of diversity that appears to be quite aligned with fitness or quality is defined based on the three-dimensional simulation space in which the softbots perform, since it would be precisely measured in the feature space (BC) where the robots' fitness is measured.

Drawing direct inspiration from (Mouret, 2011), we define the behavioral distance between two softbots s_1 and

s_2 as the L_2 norm of the vector formed by their final positions after a simulation of duration T , that is

$$d_{\vec{r}}(s_1, s_2) = \|\vec{r}_{s_1}(T) - \vec{r}_{s_2}(T)\|, d_{\vec{r}} : \mathbb{S}^2 \rightarrow \mathbb{R}^{(+)} \quad (36)$$

Using the aforementioned distance, it is possible to define behavior novelty, or alternatively, aligned novelty (given that it is defined based on the distance over an aligned BC). To calculate the aligned novelty N_a of a softbot s an aligned novelty archive is required. By using equation (36) and substituting it into 8, we obtain

$$N_a(s) = \frac{1}{k} \sum_{i=0}^k d_{\vec{r}}(s, \mu_i),$$

$$N_a : \mathbb{S} \rightarrow \mathbb{R}^{(+)}. \quad (37)$$

4.1.3. QD indicators. In general, the primary metric for QD algorithms is the QD indicator. This indicator uses a grid, and regardless of the quality metric (Q) used in the selection criteria for elites in each bin, it is summed for all bins, resulting in an indicator that can inform the total or accumulated local quality of the solutions found so far or the quality of the diversity found so far.

Let $Q(x_i)$ be the quality of individual x_i (or any attribute of x_i) which has the highest quality found so far in bin i of the grid G , where $G(i) = x_i$. The QD indicator is defined as

$$QD - score = \sum_{i=0}^{|G|-1} Q(x_i = G(i)) \quad (38)$$

$$x_i \in Q.$$

It is worth mentioning that this kind of indicator, along with coverage (which will be discussed later), is always computed with respect to an archive (grid) since they are derived from the MAP-Elites algorithm. This is different from all the indicators mentioned earlier, which were based on the population. It is important to keep in mind that in MAP-Elites, the primary structure being evolved is the grid, not the population as in other algorithms.

We propose two QD indicators: QD_{ff} indicator (Fitness Grid/Fitness) and QD_{anan} indicator (Aligned Novelty Grid/Aligned Novelty).

For the indicator QD_{ff} , a grid is introduced where the main criterion for retaining the elite in each bin is fitness. In each evaluated generation, a MAP-Elites selection cycle is performed, and quality is obtained by substituting $Q(x) = F(x)$ in (38)

$$QD_{ff} = \sum_{i=0}^{|G_f|-1} F(s_i = G_f(i)) \quad (39)$$

$$s_i \in \mathbb{S}.$$

On the other hand, for the indicator QD_{anan} , a grid based on the proposed aligned behavioral characterization (BC) is

introduced, where the selection criterion for each MAP-Elites cycle was the aligned novelty N_a instead of fitness F . Similar to the previous indicator, the notion of quality for each bin, in this case, aligned novelty, are summed up to give a metric that tells us the quality found in terms of morphological diversity in the solutions of an algorithm, where quality is understood as a notion of diversity supposedly aligned with quality

$$QD_{an} = \sum_{i=0}^{|G_{an}|-1} N_a(s_i = G_{an}(i)) \quad (40)$$

$$s_i \in \mathbb{S}.$$

It should be noted that the aligned novelty in this grid is calculated not only with respect to the members of the grid and the current population but also with respect to the members of the aligned novelty archive. This is because the archive maintains a history of the solutions generated throughout the evolutionary run, not just the elites. If only the elites are considered against the current population, members of the current generation could be mistakenly considered more novel than they actually are, and members of the grid could be considered less novel, or vice versa.

4.2. Phenotypic diversity indicators

4.2.1. Morphological diversity. If we want to introduce diversity in the co-evolution of morphology and control of virtual creatures, we need a notion of distance, which can vary greatly. Furthermore, if we decide to measure distances between the neural networks that encode morphology (graphs), and we use a graph-defined distance, such as an edit distance, we would find that such a distance for directed and cyclic graphs is an NP-Complete problem in itself. However, the notion of vector and feature space (BC) can be introduced by making a model-free dimensionality reduction. We took inspiration from Nordmoen et al. (2020) to define the morphological feature vector of a softbot s as

$$\mathbf{b}_m(s) = (M_p(s), M_a(s)), \quad (41)$$

$$\mathbf{b}_m : \mathbb{S} \rightarrow \mathbb{N}^2.$$

where M_p and M_a are the number of passive and active voxels, respectively, in the morphology of the softbot s . With this, the morphological distance between two softbots s_1 and s_2 is defined as

$$d_m(s_1, s_2) = \|\mathbf{b}_m(s_1) - \mathbf{b}_m(s_2)\|, \quad (42)$$

$$\mathbf{b}_m : \mathbb{S}^2 \rightarrow \mathbb{R}^{(+)}. \quad (42)$$

Thus, the morphological diversity of a softbot s with respect to the population in generation n (P_n) can be defined by substituting equation (42) into equation (4)

$$D_m(s) = \frac{1}{|P_n|} \sum_{s' \in P_n} d_m(s, s'), \quad (43)$$

$$D_m : \mathbb{S}^2 \rightarrow \mathbb{R}^{(+)}$$

4.2.2. Morphological or unaligned novelty. In morphology and control co-evolution, the search for novel morphologies often does not lead to better walking patterns. Therefore, the BC defined earlier, on which the morphological distance was defined, is a misaligned BC. Similarly, if we define novelty using this distance, we will have a morphological or unaligned novelty. To calculate this unaligned novelty N_u of a softbot s , we use equation (42) and substitute it into equation (8)

$$N_u(s) = \frac{1}{k} \sum_{i=0}^k d_m(s, \mu_i), \quad (44)$$

$$N_u : \mathbb{S} \rightarrow \mathbb{R}^{(+)}$$

4.2.3. QD_{fun} indicator (Fitness Grid/Unaligned Novelty). The QD_{fun} indicator quantifies the accumulated unaligned novelty present in the fitness grid. It is computed by setting $Q_i(x) = N_u(x)$ in equation (38), and is evaluated with respect to the fitness grid, population, and the archive of unaligned novelties

$$QD_{fun} = \sum_{i=0}^{|G_f|-1} N_u(s_i = G_f(i)), \quad (45)$$

$$s_i \in \mathbb{S}.$$

4.2.4. Coverage. This metric is defined as the ratio between the number of discovered cells and the total number of cells in the grid. It can be expressed as

$$C = \frac{|\{G(i) \forall i \in \{0, \dots, |G| - 1\} | G(i) \neq nil\}|}{|G|}. \quad (46)$$

It is worth noting that for a given algorithm, any number of grids defined in the same BC will always have the same coverage, as it quantifies the proportion of the BC space that has been visited at least once, regardless of the quality metric being used to update the grid.

4.3. Genetic diversity indicators

These indicators arise in the implementation of NSLC (Novelty Search with Local Competition), as there is a need to define a more rigorous diversity measure among

genotypes than what is originally defined in the original paper of the algorithm (Lehman & Stanley, 2011b).

This indicator is derived from the implementation of NSLC, as there is a need to define a more rigorous diversity among genotypes compared to the original paper of the algorithm (Lehman & Stanley, 2011b). In that work, it is stated that there is a need to replace the crowding distance of NSGA-II as a mechanism to promote diversity along Pareto fronts with a genotype-based diversity indicator. The authors argue that a diversity defined in the space of local competition and novelty objectives may not necessarily reflect the diversity of interest (in this case, the diversity of morphologies). For instance, it is possible to have two individuals with exactly the same local competition and novelty scores. However, since these scores are entirely relative to the k nearest individuals, these two individuals could be located in completely different regions of the morphology space (i.e., they could have distinct morphologies). However, the crowding distance would penalize these two individuals heavily, potentially discarding them. Lehman and Stanley (Lehman & Stanley, 2011b) then proceed to define a genetic diversity based on a distance metric that counts the nodes and edges of the graphs that represent the morphology and control of the evolving robots. For the genotype distance defined in this article, the fact that we are working with genotype representations based on neural networks was leveraged to define a genotype distance based on the outputs of the CPPNs that encode the morphology and control. Considering that a neural network is a universal function approximator (Cybenko, 1989), a CPPN with m input neurons and n output neurons can be seen as a function

$$\mathbf{F} : \mathbb{R}^m \rightarrow \mathbb{R}^n. \quad (47)$$

Thus, it is possible to define a metric or pseudo-metric that captures the notion of distance between functions of this form. Recalling the inner product between two scalar-valued functions of a scalar variable $f, g : \mathbb{R} \rightarrow \mathbb{R}$, within the interval $[a, b]$, defined as

$$\langle f | g \rangle_a^b = \int_a^b f(x)g(x)dx. \quad (48)$$

the distance between f and g can be seen as $d(f, g) = \|f - g\|$.

If the norm of the function f is defined in terms of the inner product as

$$\|f\| = \langle f | f \rangle^{\frac{1}{2}}. \quad (49)$$

then

$$\|f - g\| = \langle f - g | f - g \rangle^{\frac{1}{2}} = \sqrt{\int_a^b (f - g)(f - g)dx} = \sqrt{\int_a^b (f(x) - g(x))^2 dx}. \quad (50)$$

Extending to vector fields $\mathbf{F}, \mathbf{G} : \mathbb{R}^m \rightarrow \mathbb{R}^n$, we can define a distance between vector fields over a region of space $\Omega \in \mathbb{R}^m$ as

$$\|\mathbf{F} - \mathbf{G}\| = \langle \mathbf{F} - \mathbf{G} | \mathbf{F} - \mathbf{G} \rangle^{\frac{1}{2}} = \sqrt{\int_{\Omega} (\mathbf{F} - \mathbf{G})^T (\mathbf{F} - \mathbf{G}) d\Omega}. \quad (51)$$

For the CPPNs used in the experiments, the region Ω corresponds to the design space. For a design space with a resolution of $R = X \times Y \times Z$, we have that

$$\Delta\Omega_i = \frac{b_1 - a_1}{X} \frac{b_2 - a_2}{Y} \frac{b_3 - a_3}{Z}. \quad (52)$$

As a result, the previously defined distance can be expressed in discrete form as

$$d(\mathbf{F}, \mathbf{G}) \approx \sqrt{\sum_{i=0}^{R-1} (\mathbf{F}(x_i, y_i, z_i) - \mathbf{G}(x_i, y_i, z_i))^T (\mathbf{F}(x_i, y_i, z_i) - \mathbf{G}(x_i, y_i, z_i)) \Delta\Omega_i, (x_i, y_i, z_i) \in \Omega}. \quad (53)$$

4.3.1. Genetic diversity. If we use the morphology and control neural networks as the gene expression function \vec{g}_s for a softbot s , where $\vec{g}_s : \mathbb{R}^3 \rightarrow \mathbb{R}^4$, based on equation (53), we can define the genetic diversity of s with respect to the population in generation n (P_n) by substituting equation (53) into equation (4) as

$$D_g(s) = \frac{1}{|P_n|} \sum_{s' \in P_n} d(\vec{g}_s, \vec{g}_{s'}), \quad (54)$$

$$D_g : \mathbb{S}^2 \rightarrow \mathbb{R}^{(+)}.$$

4.3.2. Genetic diversity of control. If we now consider only the neural network controller as the gene expression function for the controller of a softbot s , denoted as $\vec{c}_s : \mathbb{R}^3 \rightarrow \mathbb{R}^2$, we can apply a similar approach as in equation (54)

$$D_{gc}(s) = \frac{1}{|P_n|} \sum_{s' \in P_n} d(\vec{c}_s, \vec{c}_{s'}), \quad (55)$$

$$D_{gc} : \mathbb{S}^2 \rightarrow \mathbb{R}^{(+)}.$$

4.3.3. Genetic diversity of morphology. If we now do the same but instead we take the morphology neural network of a softbot s , we will have that

$$D_{gm}(s) = \frac{1}{|P_n|} \sum_{s' \in P_n} d(\vec{m}_s, \vec{m}_{s'}), \quad (56)$$

$$D_{gm} : \mathbb{S}^2 \rightarrow \mathbb{R}^{(+)}.$$

5. Results

In this section, the results of the experiments are presented in the form of convergence plots, box plots, and frequency diagrams along with their approximate densities through KDE for each of the indicators mentioned in Section 4. For the convergence plots and KDE, the bootstrapping resampling method was employed, which involves randomly sampling with replacement. In this case, 1000 samples were taken per generation from the 20 runs. The convergence plots display the average of the runs using solid colors, while the confidence interval ($ci = \pm 95\%$) is represented by transparent bands. Additionally, other interesting results are presented, either from other indicators or other visualizations.

5.1. Task performance

5.1.1. Fitness. Figure 1(a) shows that NSLC exhibits a rather slow fitness growth, which is characteristic of QD algorithms with novelty defined in a misaligned BC, as mentioned in (Pugh et al., 2016). In contrast, MNSLC shows a clearly higher and more erratic growth, evident both visually in Figure 1(a) and through the large number of outliers in Figure 1(c). Furthermore, it seems that the supposedly aligned BC is indeed aligned, given the higher growth rate when compared to NSLC. ME, QN, and SO exhibit a similar convergence pattern in this indicator, with ME performing slightly better (as seen in Figure 1(a), the median in Figure 1(c), and the skewed density estimate in Figure 1(b), biased slightly to the right), followed by QN and SO. SO appears to have a slightly more stable behavior than QN, while QN seems to have a slightly faster growth rate than SO (based on Figures 1a and 1(b), with Figure 1(c) showing nearly identical medians for both). Looking ahead, it is expected that MNSLC will perform at least as well as NSLC in the indicators where a QD algorithm is expected to yield better results.

5.1.2. Behavioral novelty N_a . Figure 1(d)-(f), provide insights into the performance of the different experiments for this indicator. It is noteworthy that ME and QN, which employ algorithms with a selection pressure aimed at global fitness,

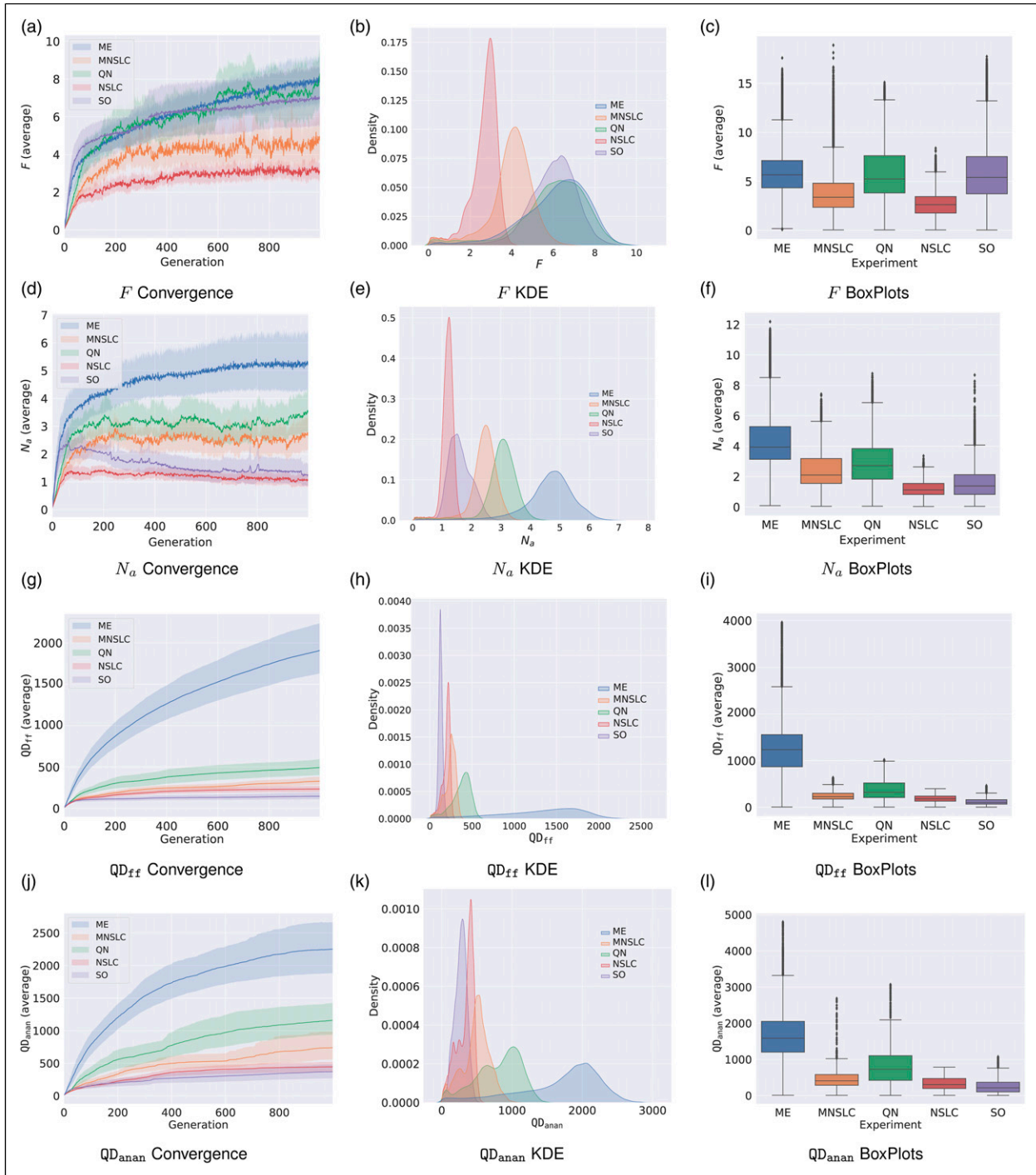


Figure 1. Convergence, KDE, and Box plots for task performance indicators: (a)-(c) Fitness F , (d)-(f) Behavioral novelty N_a , (g)-(i) Fitness Grid/Fitness QD_{ff} and (j)-(l) Aligned Novelty Grid/Aligned Novelty QD_{anan} .

outperform MNSLC, which explicitly maximizes aligned novelty as one of its objectives. This observation partly confirms that the proposed aligned novelty concept is indeed aligned, but there could also be a partial causal relationship with morphological diversity. If aligned novelty

was solely related to fitness, then SO would exhibit similar behavior to ME and QN, which is not the case. SO is similar to that of NSLC. Therefore, it seems that there is a component of aligned novelty that is related to fitness and another component that could be related to morphological

novelty. When seeking morphological novelty, changes in morphology over optimized controllers are equivalent to changes in the environment (Lipson et al., 2016), leading to erratic behaviors. These changes could result in trajectories of the softbots that exhibit higher aligned novelty but have lower fitness.

5.1.3. QD_{ff} indicator. For this indicator, it is expected that ME will have the best results by a significant margin, which is confirmed in Figure 1(g)-(i). The reason for this is because this indicator measures the total quality achieved in a grid over the misaligned BC, the MAP-Elites algorithm is using the same grid, with a selection pressure oriented toward quality, that is, seeking to maximize precisely this QD_{ff} . It should be noted that in the other four experiments, a similar ordering to that observed in Figure 1(d)-(f) was obtained, except that SO had the worst results, probably not due to a lack of quality in the solutions, but rather due to a lack of exploration of the misaligned BC. Conversely, NSLC had the second worst results, mainly due to the lack of quality of the solutions, not the lack of exploration. MNSLC seems to mitigate the lack of quality of NSLC solutions, but without achieving quality as high as QN, which has a better exploitative capability.

5.1.4. QD_{anan} indicator. For this indicator, some points similar to those observed in N_a and QD_{ff} can be noted. The high degree of alignment of N_a is evidenced again, first because as shown in Figures 1(j), 1(k), and 1(l), ME is the experiment with the best results, by a considerable margin, maximizing QD_{ff} and not QD_{anan} , and second because the remaining four experiments have the same ordering as the previous indicator, which also indicates: the bias toward exploitation of SO, the bias toward exploration of NSLC, the mitigation of MNSLC over the bias of NSLC, and in QN and ME, the effects of global exploitation on discovered diversity. Furthermore, the difference in scales and the greater growth exhibited particularly by QN and MNSLC, and to a lesser extent by NSLC and SO, compared to what is shown in Figure 1(g)-(i), makes it evident, at least to some extent, that aligned novelty considers not only the distance traveled but also the trajectory, and therefore indirectly the morphological diversity.

5.2. Phenotypic diversity

5.2.1. Morphological diversity. The impact of the loss of morphological diversity becomes evident when comparing algorithms where explicit diversity preservation is absent (SO) with those where such strategies are present (QN, ME, MNSLC, NSLC), as shown in Figure 2(a). The differences between these situations are clear and contrasting based on

this indicator. The information presented in Figure 2(a) reveals the following:

- The SO algorithm exhibits a decreasing trend that reaches a stable state at much lower values.
- In ME, a decreasing trend can be observed in the first few generations, which could be attributed to the fact that much of the morphological search space has not yet been explored in these early stages of optimization. However, this behavior ceases after a few generations, and a slow increase in morphological diversity can be observed, stabilizing at a value of approximately 35.
- A very rapid growth pattern can be observed in all algorithms that use NSGA-II and explicitly target N_u .
- In QN, a higher variance can be distinguished, as well as a decreasing pattern immediately after the initial period of rapid growth. This behavior is more pronounced than in NSLC and MNSLC, almost reaching the same value at which ME stabilizes.
- For the QD algorithms with a local selection pressure (NSLC and MNSLC), a slightly decreasing trend can be observed, which stops almost immediately and quickly stabilizes within a range of values.

In broad terms, the algorithms that exhibit the lowest loss of morphological diversity are the “purely” QD ones (NSLC and MNSLC), followed by QN, which integrates global quality rather than local quality and aligned novelty (or morphology) into a multi-objective framework. Next is ME, which does not include any objective for aligned novelty or explicit selection pressure toward it, but rather finds morphological diversity through a grid defined in the aligned behavior space and by maximizing QD_{ff} . In ME, the selection pressure toward global fitness becomes evident, with the loss of population-level diversity it represents compared to its “pure” QD counterparts (NSLC and MNSLC).

5.2.2. Unaligned novelty N_u . Regarding N_u , a similar trend to that observed in D_m is observed. As shown in Figure 2(d), MNSLC and NSLC exhibit the highest unaligned novelty, with a sharp decrease in the first generations followed by a stabilization in a rather slow, and approximately linear decrease. It is interesting to note that ME and QN converge to similar values, with ME staying slightly higher at the end of the evolutionary runs, especially considering that QN explicitly pursues the N_u objective, demonstrating the effect of using a global quality measure in contrast to its local version. As shown in Figure 2(e), and 2(f), QN also exhibits higher variance, possibly due to the conflicting objectives of global quality and unaligned novelty. In Figure 2(d), ME exhibits a more pronounced loss of novelty than the three previously mentioned experiments, once again highlighting

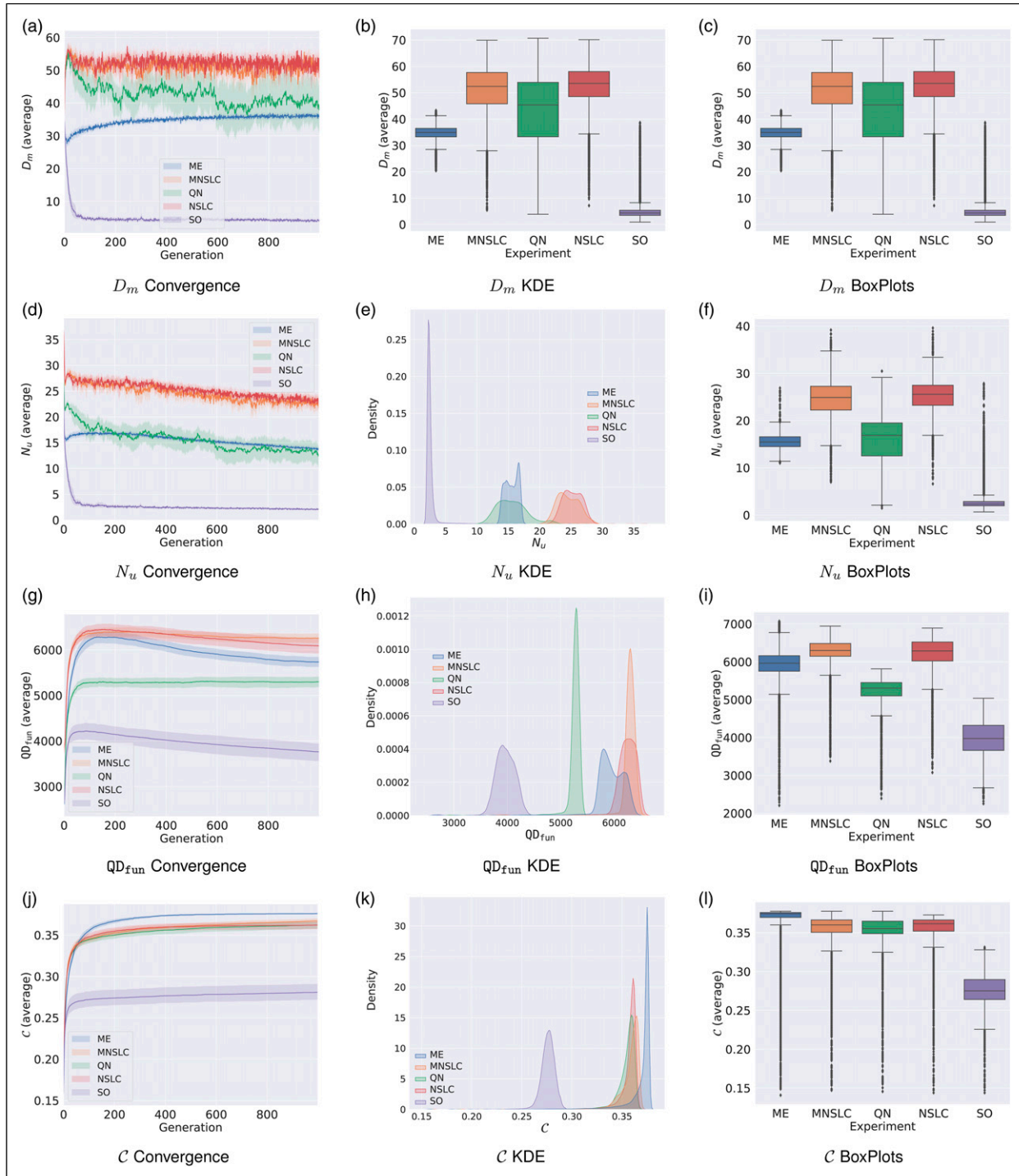


Figure 2. Convergence, KDE, and Box plots for phenotypic diversity indicators: (a)-(c) Morphological diversity D_m , (d)-(f) Unaligned novelty N_u , (g)-(i) Fitness Grid/Unaligned Novelty QD_{fun} , and (j)-(l) Coverage C .

the selection pressure toward global quality, in contrast to its “pure” QD counterparts, but then exhibits the same slow growth that resembles a logarithmic curve, followed by a more or less linear decrease. In Figure 2(f), it can be seen that ME exhibits a slightly lower median novelty than QN,

while also exhibiting a lower variance, and therefore less noisy behavior. Finally, SO exhibits similar behavior to what was observed in D_m , as shown in Figure 2(f), once again exhibiting outliers with rather high values, likely due to mutations that led individuals to be eliminated by the

purely exploitative selection operator, or because in the early generations, there are more novel individuals.

5.2.3. QD_{fun} indicator. Similarly to the other QD indicators examined previously, as can be seen in Figure 2(g), there is a growth pattern similar to a logarithmic function. However, it appears that as the rate at which new areas of the BC space are discovered decreases, the morphological novelty of the elites in the grid also begins to decline. On the other hand, the similarity of the behavior of ME to NSLC and MNSLC can be explained by the fact that the QD indicators are computed relative to the grid, and ME evolves a grid and not a population. By maintaining solutions distributed in the morphological BC, it is also implicitly seeking diversity and therefore novelty, ultimately placing its behavior closer to “pure” QD algorithms than to QN. Also, by examining the behavior of SO in this indicator, it can be observed that even in an experiment without any type of diversity protection, by using a grid to store the solutions found, some novelty is found, resulting in a behavior relative to the other experiments that is visibly better for SO than the one observed in the population-based novelty N_u .

5.2.4. Coverage C . As can be seen in Figure 2(j)–(l), even when using a selection probability proportional to the global quality, ME exhibits higher coverage than MNSLC, NSLC, and QN. In other words, a stochastic selection of bins in a grid, even when biased in an exploitative manner, finds more regions of the feature space than a search for novelty. This would likely be even more noticeable if the selection was uniform, that is, if pure MAP-Elites had been used.

5.3. Genetic diversity indicators

5.3.1. Genetic diversity D_g . When observing the behavior of this indicator in Figure 3(a)–(c) and comparing it with other diversity indicators, there seems to be a certain relationship between genetic diversity as defined in this article, and phenotype diversity. Regarding this indicator, each experiment presents a characteristic and easily distinguishable behavior at first glance. As shown in Figure 3(a), no experiment exhibits sudden growth at the beginning of the evolutionary runs, that is, in all experiments, there is some degree of loss of genetic diversity, a phenomenon that is prevalent in the literature. With regard to Figure 3(a), it can also be observed that:

- Starting with SO, a rapid loss of genetic diversity is observed, closely aligned with all the previously mentioned population-level phenotypic diversity indicators (N_u and D_m).
- QN exhibits the second-largest loss of genetic diversity, while ME exhibits the third-largest. This is likely due to the use of a grid in the morphological

feature space, which is closely related to the behavior found in population-level phenotypic diversity indicators. Once again, it appears that the introduction of selection pressure proportional to global quality separates ME from the other QD algorithms.

- In the case of NSLC and MNSLC, a slightly different behavior is observed, with one significantly outperforming the other. In this case, MNSLC outperforms NSLC, which can perhaps be explained by the fact that aligned novelty search leads to greater diversity in the controllers. Additionally, a slightly increasing behavior can be observed after the initial loss of diversity.

It is noteworthy that the two experiments that exhibit local competition (or quality) are also the ones that exhibit the highest genetic diversity. It also seems that QD algorithms (ME, NSLC, and MNSLC) are the ones that exhibit the highest genetic diversity. Furthermore, it appears that NSLC and MNSLC, which represent more purely the characteristics of QD algorithms, such as division into niches (unstructured in this case) in the feature space and the notion of local quality, are the ones that exhibit the highest genetic diversity. On the other hand, ME, by favoring a selection pressure proportional to global quality instead of a uniform selection, loses genetic diversity, as was the case with N_u and D_m .

5.3.2. Genetic diversity of control D_{gc} . As shown in Figure 3(d)–(f), this indicator follows the same ranking as D_g . In this case, it can be observed that aligned novelty may be caused (at least partially) by greater diversity in behavior, and in turn, by greater diversity in controllers. Furthermore, the two experiments that exhibit local competition also exhibit greater diversity in controllers, as can be contrasted with the behavior of QN in Figure 3(d), which also targets morphological novelty as a secondary objective. This could suggest that maximizing the score of local competition increases the genetic diversity of controllers. Taking this further, in the case of ME, it could be argued that QD algorithms exhibit greater genetic diversity of controllers, but once again, introducing the global quality proportional selection pressure erodes genetic diversity.

5.3.3. Genetic diversity of morphology D_{gm} . In Figure 3(g), it can be observed that SO shows a much faster-decreasing behavior than its counterparts with diversity protection. Interestingly, in that same figure, the behaviors of NSLC and QN are so similar. It seems that morphological genetic diversity ignores local competition, unlike D_{gc} . On the other hand, MNSLC, whose only difference with NSLC is the introduction of a third objective corresponding to aligned novelty, has the highest D_{gm} , which could be due to the fact

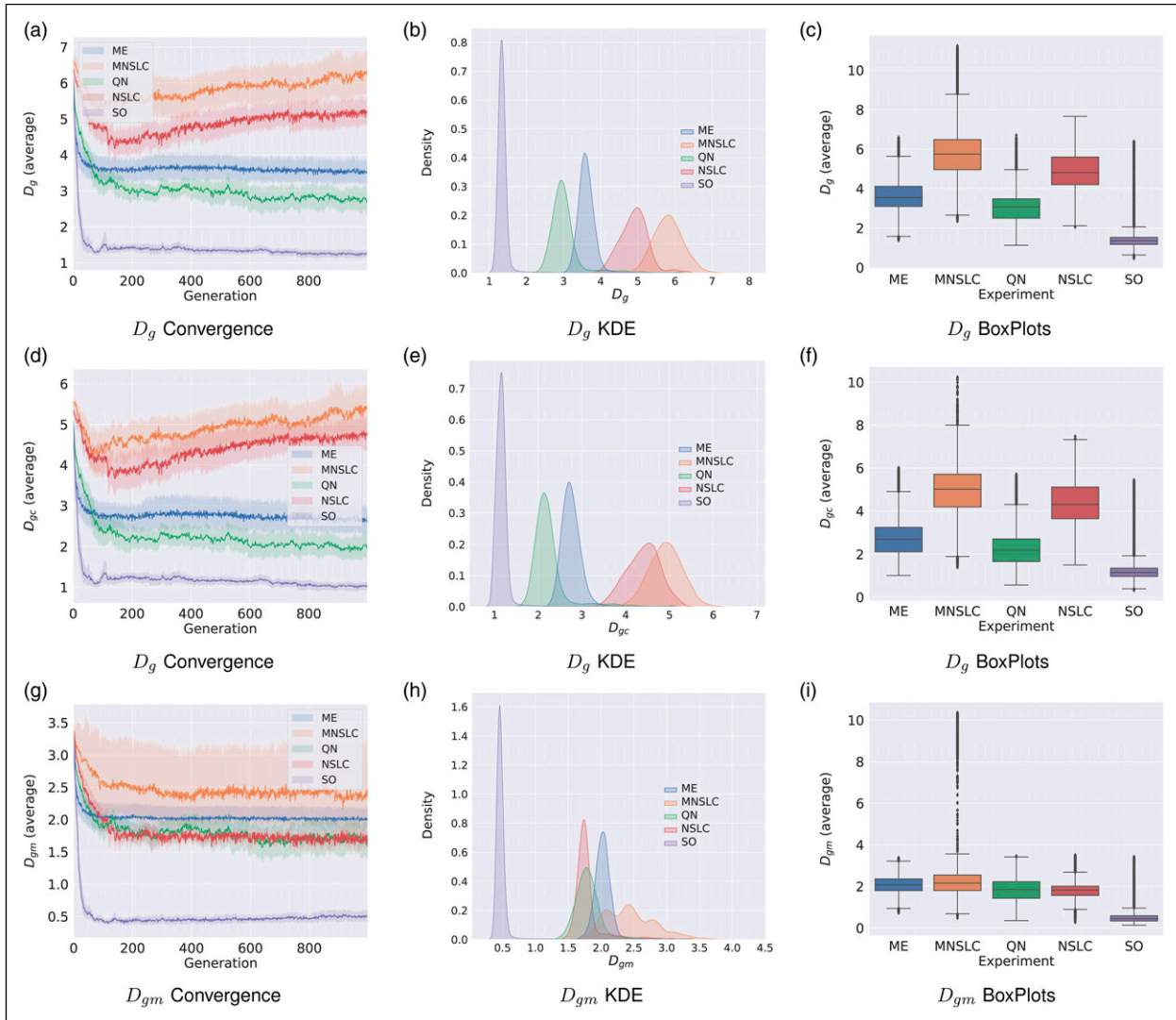


Figure 3. Convergence, KDE, and Box plots for genetic diversity indicators: (a)-(c) Genetic diversity D_g , (d)-(f) Genetic diversity of control D_{gc} , and (g)-(i) Genetic diversity of morphology D_{gm} .

that aligned novelty is partly caused by changes in morphologies associated with changes in trajectories.

5.4. Results of other indicators and visualizations

In this section, we present other indicators and visualizations in addition to those presented previously. Although they are not relevant for the selection of the winning algorithm, they do represent interesting results.

5.4.1. Complexity of the neural networks. Three basic but representative indicators of the complexity of the control and morphology neural networks are displayed for each experiment. These indicators include the number of nodes (Figures 4(a) and 4(b)), edges (Figures 4(c) and 4(d)), and total weight sum (Figures 4(e) and 4(f)).

It is interesting to note that the experiment that maintained the lowest complexity in both morphology and control neural networks was ME, while MNSLC and NSLC consistently obtained the largest neural networks and total weight sum. It could be desirable to find diverse and high-quality solutions with the least complex genotypes possible, something that seems to be achieved by ME.

5.4.2. The behavior space. Figure 5 shows the contour curves of the joint distribution of the softbots' positions at the end of the simulations, on the XY plane. This is a way to visually examine the effects of aligned novelty and fitness. For example, in the case of SO, it becomes evident why aligned novelty is so low, but fitness is high, due to the exploitative nature of the experiment, which leads it to only find a very small family of trajectories.

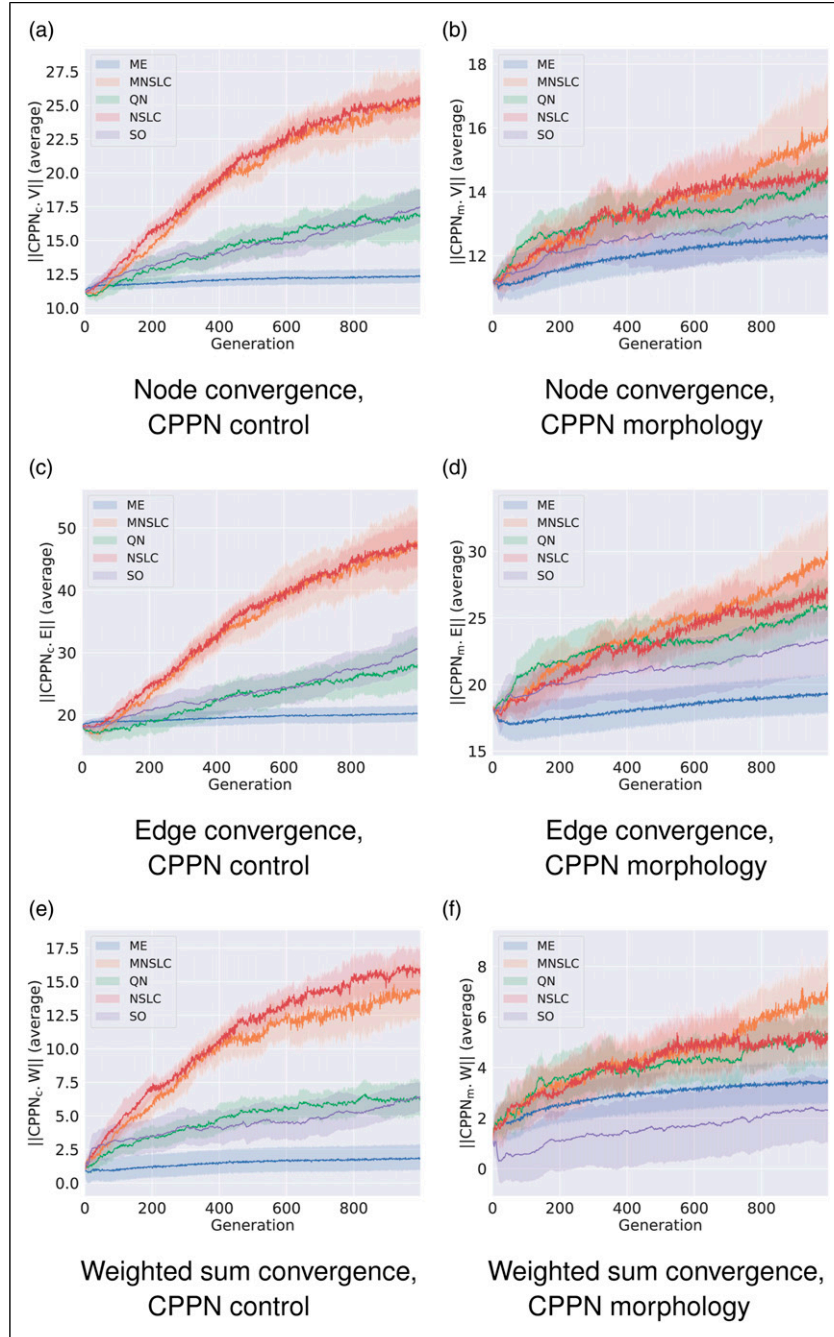


Figure 4. Complexity of CPPNs.

The case of ME makes evident the high result in both fitness and aligned novelty, given that it found high-quality solutions along different trajectories. QN presents a similar behavior to ME.

MNSLC presents a range of trajectories close to the origin, which highlights lower fitness than SO, ME, and QN, and lower aligned novelty than ME and QN.

NSLC presents a similar story to MNSLC, in terms of solutions closer to the origin, and also makes it clear that it

has lower aligned novelty than MNSLC, as it has a smaller range of trajectories.

6. Discussion

In this section, firstly we present a methodology based on voting theory in order to choose the best algorithm in the fairest possible way, taking into account multiple indicators and multiple algorithms. Subsequently, we discuss the

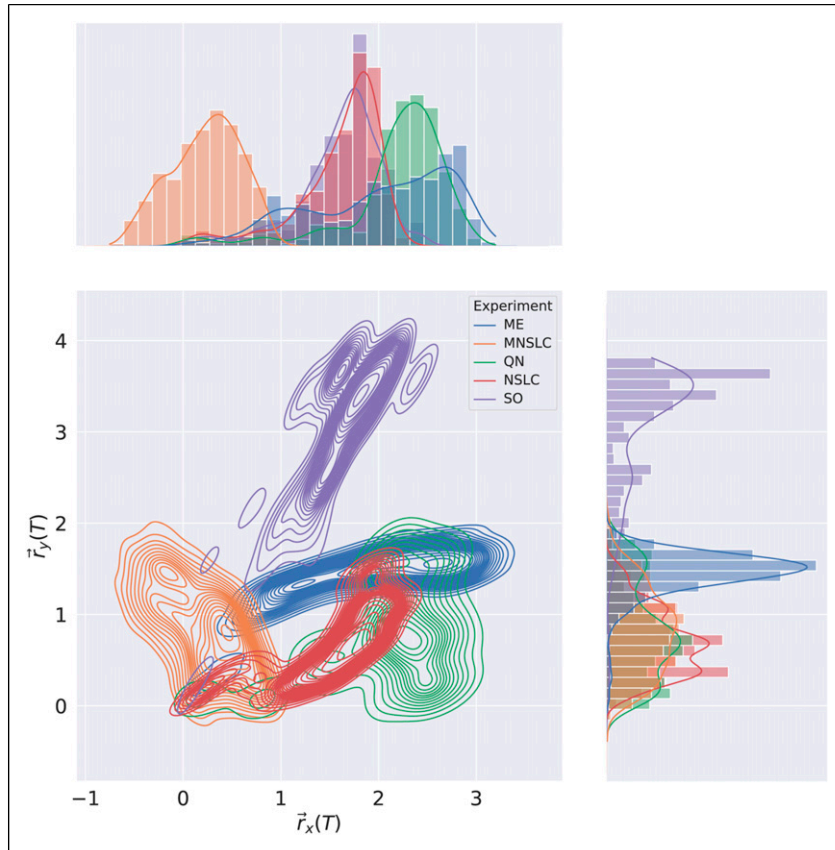


Figure 5. Joint behavior space KDE.

findings related to the application of this methodology to the results presented in Section 5. This methodology will be presented in its most general form, which takes into account several indicators, algorithms, and benchmark problems [Bartz-Beielstein et al. \(2020\)](#), but will be applied, for a single benchmark problem, multiple algorithms, and multiple indicators. By applying such methodology, to the main experiments and indicators, our ambition is to obtain in an exhaustive and multifactorial fashion, which algorithm presents the most desirable behavior for dealing with the challenge of premature convergence of morphology with respect to control, while compromising as little as possible other desirable aspects, especially of exploitative nature, leading to ultimately find solutions that are not only diverse morphologically but also functional.

For completeness, it is worth explaining that, based on the methodology for the analysis of optimization algorithms benchmarking, presented in [Bartz-Beielstein et al. \(2020\)](#), what was done in Section 5 corresponds to the first step, that is, exploratory data analysis. This section corresponds to confirmatory analysis, which consists of comparing each pair of algorithms relative to each pair of benchmark problems and indicators $(P_i, I_j) \in \mathbb{P} \times \mathbb{I}$ through hypothesis testing.

6.1. Hypothesis test

Since the data does not meet the assumptions for parametric statistics, namely, normality, we used the non-parametric Wilcoxon Rank-Sum test for conducting pairwise comparisons. Additionally, considering the high computational cost of the experiments, we avoided conducting the full set of experiments for each hypothesis ($5 \cdot 4 \cdot 11 = 220$), and instead, we only executed twenty trials per experiment. Therefore, the Wilcoxon Rank-Sum test was performed with Bonferroni correction. We choose $\alpha = .05/220 = .000227$ as the minimum p - value threshold to reject the null hypothesis.

6.2. Expressing the preferences

With a method in place to compare algorithms, we can arrange them in order based on this comparison. In the simplest case, we have 2 algorithms, 1 problem, and 1 indicator.

- Perform multiple independent runs
- Apply the chosen statistical test, with 3 possible outcomes

– $A_1 < A_2$. If $\mu_1 > \mu_2$; $p < \alpha$

– $A_2 < A_1$. If $\mu_2 > \mu_1$; $p < \alpha$

– $A_1 \sim A_2$. If $p \geq \alpha$

A matrix can be constructed to express the preferences of (P_1, I_1) for n candidates (algorithms) $\mathbb{A} = A_1, \dots, A_n$, where P_1 is a problem and I_1 is an indicator:

	A_1	\dots	A_n
A_1	–	\uparrow	\downarrow
\vdots	\downarrow	–	\downarrow
A_n	\uparrow	\leftrightarrow	–

The matrix can be read row-wise, from left to right as:

- $A_i < A_j = A_i \uparrow A_j$.
- $A_j < A_i = A_i \downarrow A_j$.
- $A_i \sim A_j = A_i \leftrightarrow A_j$

The table can be summarized into a count:

- +1 for the preferred algorithm.
- –1 for the algorithm that loses.
- 0 for indifference.

Using this counting method, we can order them from highest to lowest, and such ordering would express the preference for (P_1, I_1) . If we wanted to generalize for

$$\mathbb{P} = \{P_1, \dots, P_m\}$$

$$\mathbb{I} = \{I_1, \dots, I_l\}$$

then we would need to obtain the $\|\mathbb{P}\| \cdot (\|\mathbb{I}\| - 1)/2$ preferences of $\mathbb{P} \times \mathbb{I}$. In our case, we only have one problem, so we have one preference for each indicator.

The mean and standard deviation of all the indicators for each algorithm were calculated and are presented in

Table 1. By combining these results with the hypothesis tests, we determined the outcomes of the comparisons for each indicator. Next, using the results from all comparisons, we obtained the scores for each algorithm across all indicators (see **Table 2**). Finally, we determined the preferences for each indicator by tabulating the frequency of occurrences for each permutation (see **Table 3**).

6.3. Choosing the winner

To choose the winner, we applied the Borda count method, which is obtained as follows for each algorithm

$$CB(ME) = 5(5) + 4(1) + 3(3) + 2(2) + 1(0) = 42$$

$$CB(MNSLC) = 5(4) + 4(3) + 3(3) + 2(1) + 1(0) = 43$$

$$CB(NSLC) = 5(2) + 4(3) + 3(1) + 2(3) + 1(2) = 33$$

$$CB(QN) = 5(0) + 4(4) + 3(3) + 2(4) + 1(0) = 33$$

$$CB(SO) = 5(0) + 4(0) + 3(1) + 2(1) + 1(9) = 14$$

The Borda count results (see **Table 4**) indicate that the MNSLC algorithm is the winner, closely followed by ME. The order of the experiments based on this count is as follows: MNSLC, ME, NSLC, QN, SO.

Another way to choose the winner is through the Condorcet method, where:

- A matrix of $n \times n$ is created.
- It is filled with the times that a candidate A_i is preferred to A_j .
- A Condorcet winner is one who wins all of their one-on-one “matches.” That is, the candidate A_i who has been preferred over A_j more times than A_j has been preferred over $A_i \forall j; i \neq j$.

Table 1. Indicators by algorithm.

Indicator	ME	MNSLC	QN	NSLC	SO
N_a	4.61 (2.21)	2.41 (1.28)	3.00 (1.55)	1.19 (.52)	1.62 (1.04)
D_{gc}	2.76 (.86)	4.92 (1.19)	2.25 (.80)	4.39 (1.05)	1.20 (.49)
C	.37 (.02)	.36 (.02)	.35 (.02)	.36 (.02)	.28 (.02)
F	6.10 (2.81)	3.94 (2.53)	5.96 (3.20)	2.68 (1.21)	5.91 (3.07)
D_g	3.63 (.83)	5.84 (1.36)	3.04 (.86)	4.90 (.99)	1.41 (.56)
D_m	34.72 (2.71)	51.01 (8.68)	42.56 (15.20)	52.24 (7.70)	4.86 (3.20)
D_{gm}	2.03 (.47)	2.47 (1.61)	1.81 (.56)	1.80 (.41)	.50 (.33)
QD_{anan}	1678.72 (881.86)	493.25 (359.01)	803.88 (547.56)	335.00 (182.46)	269.88 (203.65)
QD_{ff}	1295.69 (705.72)	240.92 (101.44)	371.71 (217.51)	193.60 (82.90)	125.29 (76.88)
QD_{fun}	5929.06 (453.52)	6278.15 (333.42)	5254.00 (334.64)	6233.63 (418.72)	3967.39 (475.09)
N_u	15.54 (1.48)	24.58 (3.68)	15.65 (5.25)	25.25 (3.06)	2.68 (1.86)

Table 2. Scores by algorithm for each indicator.

Indicator	ME	MNSLC	NSLC	QN	SO
N_a	4	0	-4	2	-2
D_{gc}	0	4	2	-2	-4
C	4	2	0	-2	-4
F	4	-2	-4	1	1
D_g	0	4	2	-2	-4
D_m	-2	2	4	0	-4
D_{gm}	2	4	-2	0	-4
QD_{anan}	4	0	-2	2	-4
QD_{ff}	4	0	-2	2	-4
QD_{fun}	0	4	2	-2	-4
N_u	-2	2	4	0	-4

Table 3. Permutation incidence.

1	2	3	4	5	1
ME	MNSLC	NSLC	QN	SO	1
ME	QN	MNSLC	NSLC	SO	2
ME	QN	MNSLC	SO	NSLC	1
ME	QN	SO	MNSLC	NSLC	1
MNSLC	ME	QN	NSLC	SO	1
MNSLC	NSLC	ME	QN	SO	3
NSLC	MNSLC	QN	ME	SO	2

Table 4. Borda count.

Position	ME	MNSLC	NSLC	QN	SO
1	5	4	2	0	0
2	1	3	3	4	0
3	3	3	1	3	1
4	2	1	3	4	1
5	0	0	2	0	9
Borda count	42	43	33	33	14

Using the Condorcet method, we again find that the winner is MNSLC (see Table 5). However, as it is evident, the victory of MNSLC was marginal against ME, under both criteria. In terms of the Borda count, MNSLC scored 43, against the 42 of ME, with ME being in the first place five times, one more than MNSLC, which would have made ME the winner by the majority criterion. The difference was marked by the two more times that MNSLC was in the fourth place. With the Condorcet method, as mentioned before, the victory of MNSLC was also marginal, with six wins out of the 11 contests, compared to five of ME. If we examine Table 1, it becomes clear that the victories of MNSLC were almost exclusively focused on indicators that consider purely diversity (D_m , N_u , D_g , D_{gc} , D_{gm} , QD_{fun}), with the exception of coverage (C). This means that

Table 5. Condorcet method.

	ME	MNSLC	NSLC	QN	SO
ME	0	5	6	9	11
MNSLC	6	0	9	7	10
NSLC	5	2	0	6	9
QN	2	4	5	0	10
SO	0	1	2	0	0

MNSLC can find greater diversity, both morphological and control-related, but ME is superior in all criteria that consider global fitness and local quality, finding not only greater fitness (F), but also accumulated quality (QD_{ff}), aligned accumulated novelty (QD_{anan}), and aligned novelty (N_a).

Now, let's go back to the questions raised in Section 3.5.4. With the above analysis, we have answered how ME compares with a Multi-BC QD algorithm. To answer how the introduction of global pressure affects a QD algorithm like MAP-Elites and if it still maintains desirable characteristics of QD algorithms, we need to compare it against NSLC.

Returning to examine the Borda count in Table 4, we can observe that ME significantly outperforms NSLC. However, examining the Condorcet method in Table 5, we realize that ME only marginally outperforms NSLC, with six wins compared to NSLC's five, winning in almost the same indicators as MNSLC, with the addition of D_{gm} . This brings us to a similar conclusion as with ME and MNSLC. Considering the desirable characteristics of a QD algorithm, we find that ME was superior to both MNSLC and NSLC in terms of the quality found in diversity, but generally inferior in terms of the notions of diversity considered.

When considering the results of MNSLC versus NSLC, it seems that the victory was quite considerable in both Borda count and Condorcet method. Examining the matches one by one in Table 5, we can see that MNSLC won 9 out of the 11 matches against NSLC, while NSLC won the remaining ones. Furthermore, if we examine the two indicators in which NSLC won, we can observe that the only advantage NSLC has is that it achieves slightly more morphological diversity and novelty (D_m , N_u), specifically 1.23 and .67 on average, respectively, small but statistically significant differences. In both indicators, NSLC is in first place, followed immediately by MNSLC. Therefore, we can say that, although there is a loss of diversity and novelty relative to NSLC, and both indicators are key to premature convergence of morphology, this loss is minimal, albeit statistically significant. Similarly, taking into account the 9 indicators in which MNSLC was superior, we can observe that not only was MNSLC superior in all those related to fitness performance in the task, but it was also superior in two out of the four related to phenotype diversity and in all those related to genotype diversity. Moreover, the phenotypic diversity indicators in which it performed better were

those related to the fitness grid. Considering the observations regarding the comparison of MNSLC and NSLC, it seems that the addition of aligned novelty, firstly, and as already mentioned in the results, is indeed an appropriate notion of aligned novelty, and secondly, effectively mitigates the problem of slow fitness progress in QD algorithms, specifically in NSLC, without compromising performance in terms of morphological diversity.

When comparing the performance of QN with its “purely” QD counterparts (MNSLC and NSLC), we find that MNSLC outperforms QN both in Borda count and in the Condorcet method, while QN and NSLC have equal performance in Borda, and NSLC outperforms QN in Condorcet by one encounter. We can notice that the 4 victories of QN over MNSLC were exclusively in the indicators that consider global or local quality, and similarly, the 7 victories of MNSLC over QN were exclusively in the indicators that consider genotypic and phenotypic diversity. Regarding the 5 victories of QN over NSLC, we have a situation very similar to the previous one, except that a very marginal victory in genetic diversity of morphology (D_{gm}) is added to the QN victory count. Therefore, we can say that taking QN as a reference, the use of local competition leads to more diverse results in both genotype and phenotype, while the use of global fitness leads to results with higher global quality and higher quality on the diversity found.

We have already talked about the relative advantages and disadvantages among the different algorithms that present diversity protection, now we will talk about the absolute advantages and disadvantages that algorithms with diversity protection present compared to not presenting any type of diversity protection, using the simple genetic algorithm (SO) as a reference.

As far as the performance of QN compared to SO is concerned, the former outperforms the latter by more than double the number of Borda points, and QN wins all its encounters against SO, except for one, where there was a tie (the null hypothesis could not be rejected), which corresponds to the fitness, that is, the effect of adding morphological novelty as a second objective to the fitness of solutions to the virtual creature morphology and control optimization problem in the walking pattern task, when the first objective is global quality, is null. And while there is no effect on fitness, there are improvements in all other fitness-related indicators (QD_{ff} ; QD_{anan} , N_a), as well as all indicators related to genetic and phenotypic diversity. That is, the addition of morphological novelty not only does not affect global fitness, but it also improves related criteria such as aligned novelty, aligned novelty found in morphological diversity, and quality in morphological diversity. It also improves in all aspects related to diversity, including morphological diversity, and therefore, the robustness against the premature convergence phenomenon of morphology regarding control.

Comparing NSLC against SO, we found again an advantage of more than double the Borda count points, and NSLC wins 9 of its 11 encounters against SO. SO outperforms NSLC in the two population indicators related to fitness (F and N_a), and loses in any other indicator that incorporates some notion of diversity, from quality indicators in diversity (QD_{ff} ; QD_{anan}) to genotype and phenotype diversity indicators. Therefore, we can affirm that adding morphological novelty and changing the global fitness to local fitness leads to more diverse results in both genotype and phenotype, with higher quality over the discovered diversity, but with worse global fitness.

Moving on to MNSLC versus SO, we now have the best algorithm versus the worst one with respect to the Borda count ordering. This time, MNSLC wins 10 out of 11 encounters, while SO outperforms MNSLC only in one indicator, the global fitness, so it is noteworthy the effect of adding aligned novelty as a third objective, since MNSLC now outperforms SO in that indicator. With these observations, we can affirm that adding novelty in morphology and alignment as second and third objectives respectively, and changing global fitness for local fitness, leads to more diverse genotype and phenotype results, with higher quality over the found diversity, and with higher aligned novelty, but at the expense of worse global fitness.

Finally, comparing ME and SO, we observe that ME ranks second by a narrow margin, with SO coming in last. In this case, ME outperforms SO in each and every indicator, indicating that adding a selection pressure proportional to global fitness to MAP-Elites results in an algorithm that not only finds solutions that are more diverse and of higher quality in terms of diversity but also solutions that are globally fitter.

NSLC exhibits the worst progress in terms of fitness, which is consistent with the findings in the literature (Pugh et al., 2016). However, it is difficult to determine with this indicator alone whether it is only slow progress or a complete stall. On the other hand, MNSLC manages to outperform NSLC, but it appears that, when examining the last generations in the convergence graph, it would not grow much in subsequent generations after 1000. In the other algorithms, we can see that there is still a growing trend in those last generations, with ME presenting the clearest trend. However, even SO shows signs that it will continue to grow beyond generation 1000. Therefore, it is possible that if the experiments had been allowed to run for more generations, the convergence of SO to a final fitness value would have been observed.

As mentioned several times, behavioral novelty is indeed aligned with fitness; however, it is possible to obtain aligned novelty with high fitness through a not-so-novel trajectory, if the creature covers a greater distance than any previous solutions within its own niche. Conversely, aligned novelty can also be low if the creature does not achieve such

improvement. Another way to achieve aligned novelty is through novel trajectories, which are not necessarily associated with high fitness values and may be associated with changes in morphology that lead to erratic behaviors. Therefore, the reason why ME and QN outperform MNSLC in aligned novelty may be because both algorithms are capable of finding solutions that are both morphologically novel and have higher global fitness than MNSLC, covering both scenarios in which high aligned novelty can occur, that is, through diverse trajectories and consistent progress in global fitness. One way to examine this would be to explore the angles of the trajectories of each individual and the frequency of times per generation that the individual with maximum fitness was surpassed. This last metric could also be related to progress relative to fitness, and therefore to convergence to the final fitness value. Thus, it could also be tested whether higher aligned novelty is related to greater progress in terms of fitness.

Regarding the main findings that the examination of morphological diversity provides, it can be mentioned that there is undoubtedly a clear effect in the introduction of diversity protection mechanisms, which allows for much more diverse populations (i.e., with higher values of average morphological distance), and that not only is there a slower loss of morphological diversity, but also that morphological diversity is maintained, thus avoiding premature convergence of morphology with respect to the control. Regarding unaligned novelty, something very similar can be mentioned, except that in this case, it is not maintained for any algorithm, but rather, at best decreases slowly. It should be noted that this indicator is closely related to the parameters chosen for the archive, whereas at least the test problems, plots with logarithmic shape (with small values of k) or plots that reach a maximum value only to immediately start decreasing (when the archive had reached its maximum size) could also be found.

The coverage provides information about the ability of each algorithm to explore different morphologies, and in consistence with all indicators that incorporate some type of diversity, SO is found to perform much worse than the other experiments. ME outperforms all other experiments in this regard, despite having a probability of bin selection proportional to fitness, however, a uniform selection probability would likely achieve greater coverage.

Similarly, a loss of genetic diversity was found to varying degrees in all algorithms, with SO showing the greatest loss of genetic diversity, consistent with diversity indicators of the phenotype at the population level. The experiments that showed the least loss of genetic diversity were the QD algorithms. Just as in novelty and morphological diversity, ME had the worst performance of the three QD algorithms. These indicators are population-based, so it makes sense to argue that when introducing a parent selection proportional to global quality, rather than uniform, certain individuals

have a greater probability of being selected as parents of the next generation, sharing characteristics that make them fitter, and therefore similar to one another, which in turn results in less diversity and novelty.

It is worth noting that both MNSLC and NSLC, after performing the non-dominated sorting of NSGA-II, choose the survivors of the last front with a ranking based on genetic diversity, instead of crowding distance, which can also play a role not only in genetic diversity, but also in novelty and morphological diversity. If we compare with QN, we can argue something similar about how global selection pressure affects diversity, and the possible role that genetic diversity may play as a cutoff criterion in the last front of NSGA-II. It does not seem too risky to assume that the higher genetic diversity, morphological diversity, and morphological novelty presented by MNSLC and NSLC are due not only to local competition but also to the use of genetic diversity in NSGA-II; however, the necessary data collection was not carried out to determine this conclusively.

The genetic diversity of control presents the same ordering as genetic diversity and seems to cause a greater diversity of behavior, which in turn causes a higher aligned novelty (with MNSLC being the best in this indicator). Furthermore, the two experiments with local competition present greater diversity in the controllers and it could be argued that maximizing both local competition score and aligned novelty increases the genetic diversity of the controllers.

It is interesting to see that NSLC and QN are so similar in terms of morphological genetic diversity. It seems that morphological genetic diversity ignores local competition, and is only interested in morphological phenotypic diversity, and the increase in aligned novelty that it entails.

7. Conclusions

The co-evolution of morphology and control is a crucial concept in evolutionary robotics, where changes or adaptations in one aspect can influence and shape the development of the other, leading to mutual interaction between the two. A challenge in this process is the risk of premature convergence, where the evolution of morphology stagnates early on, resulting in a limited variety morphologies. This issue has been widely observed in studies involving virtual creatures or robots, as highlighted by [Lipson et al. \(2016\)](#), [Kriegman et al. \(2018\)](#), and [Joachimczak et al. \(2016\)](#). Previous research has investigated different strategies to tackle this problem. However, there has been a lack of comprehensive comparisons among these approaches using consistent evaluation criteria and benchmarking methodologies for optimization algorithms.

In this article, we presented a comparative analysis of several state-of-the-art algorithms proposed for mitigating

the issue of premature convergence of morphology with respect to control in evolutionary robotics. To evaluate the performance of these algorithms, we introduced a set of carefully chosen indicators that capture key aspects of the problem.

We concluded that MNSLC is the absolute winner, given that the majority of the indicators used for comparison aimed to measure diversity, whether it was at the genotype or phenotype level, and whether it was based on population or grid-based metrics. Despite ME exhibiting a more balanced trade-off between exploitation and exploration, MNSLC demonstrated a competitive preservation of diversity akin to NSLC. Additionally, MNSLC maintained a competitive level of exploitative capacity that remained unaffected by the integration of diversity protection measures. Notably, its exploitative capabilities, upon examination of the task-based indicators, even surpassed those of SO. Therefore, ME is a worse algorithm than MNSLC in terms of protection of morphological diversity but probably represents a better option in practice for the co-evolution of morphology and control of virtual creatures due to its balanced trade-off between exploration and exploitation.

In the most general case, comparing ME against MNSLC and NSLC, we can reiterate the same conclusion: ME is a worse algorithm than its “pure” QD counterparts (without global quality) in terms of protecting morphological diversity.

Regarding NSLC and MNSLC, the introduction of behavioral-driven novelty as a third objective served indeed as an adequate notion of aligned novelty necessary for the effective application of the Multi-BC version of NSLC. Its addition helped alleviate the slow progress of NSLC without compromising its performance. Comparing it to QN, the use of local competition instead of global quality led to more diverse results, while the use of global quality led to results with higher overall quality and quality over diversity. Therefore, MNSLC and NSLC yielded more desirable outcomes than QN in terms of diversity protection.

Comparing the different algorithms that incorporate diversity preservation allows us to identify their relative advantages and disadvantages, providing insights into the trade-offs made by each algorithm. However, for a more objective assessment of what each algorithm gains and losses, and to better understand their absolute advantages and disadvantages, it is important to compare them against a baseline algorithm. In this case, the baseline algorithm is one without any form of diversity preservation (SO).

In the case of QN, when the primary objective was global quality, it was determined that the incorporation of morphological novelty as a secondary objective in the fitness function did not have any significant impact. Furthermore, QN improved related criteria such as aligned novelty, aligned novelty found in morphological diversity, quality in morphological diversity, and all aspects related to diversity

itself, including morphological diversity itself. Therefore, QN exhibited greater robustness.

The incorporation of morphological novelty and the shift from global to local fitness in NSLC demonstrated several advantages. It led to increased resilience against premature convergence of morphology in relation to control, as well as enhanced quality in terms of the discovered diversity. However, this came at the expense of a lower overall fitness level.

In relation to MNSLC, it was found that incorporating morphological novelty and aligned novelty as the second and third objectives, respectively, and shifting from global to local fitness, led to more robust outcomes. Additionally, it resulted in higher quality in terms of the discovered diversity and increased aligned novelty. However, there was a trade-off with lower overall fitness.

Lastly, concerning ME, incorporating a fitness-proportional selection pressure to MAP-Elites led to the resulting algorithm not only achieving more robust solutions against premature convergence of morphology with respect to control and higher quality in terms of diversity but also attaining solutions that are more globally fit.

Acknowledgements

Luis Eguiarte-Morett would like to express gratitude to CON-ACYT for the scholarship awarded.

Declaration of Conflicting Interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

ORCID iDs

Luis Eguiarte-Morett  <https://orcid.org/0009-0004-6108-7733>

Wendy Aguilar  <https://orcid.org/0000-0003-1867-8859>

References

- Auerbach, J. E., & Bongard, J. C. (2010). Evolving cppns to grow three-dimensional physical structures. In: *Proceedings of the 12th annual conference on genetic and evolutionary computation* (pp. 627–634). GECCO '10. Association for Computing Machinery.
- Bartz-Beielstein, T., Doerr, C., Bossek, J., Chandrasekaran, S., Eftimov, T., Fischbach, A., Kerschke, P., López-Ibáñez, M., Malan, K. M., Moore, J. H., Naujoks, B., Orzechowski, P., Volz, V., Wagner, M., & Weise, T. (2020). *Benchmarking in optimization: Best practice and open issues*. ArXiv abs/2007. pp. 03488.

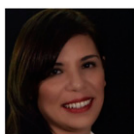
- Blank, J., & Deb, K. (2020). Pymoo: Multi-objective optimization in python. *IEEE Access*, 8, 89497–89509. <https://doi.org/10.1109/access.2020.2990567>
- Chatzilygeroudis, K., Cully, A., Vassiliades, V., & Mouret, J. B. (2020). *Quality-diversity optimization: A novel branch of stochastic optimization*. ArXiv abs/2012. pp. 04322.
- Cheney, N., Bongard, J., & Lipson, H. (2015). Evolving soft robots in tight spaces. In: *Proceedings of the 2015 annual Conference on Genetic and evolutionary computation, GECCO '15* (pp. 935–942). Association for Computing Machinery.
- 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(143), 20170937. <https://doi.org/10.1098/rsif.2017.0937>
- Cheney, N., MacCurdy, R., Clune, J., & Lipson, H. (2014). Unshackling evolution: Evolving soft robots with multiple materials and a powerful generative encoding. *ACM SIGEVOlution*, 7(1), 11–23. <https://doi.org/10.1145/2661735.2661737>
- Cully, A., & Demiris, Y. (2018). Quality and diversity optimization: A unifying modular framework. *IEEE Transactions on Evolutionary Computation*, 22(2), 245–259. <https://doi.org/10.1109/tevc.2017.2704781>
- Cybenko, G. V. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, 2(4), 303–314. <https://doi.org/10.1007/bf02551274>
- Doncieux, S., Bredeche, N., Mouret, J. B., & Eiben, A. E. G. (2015). Evolutionary robotics: What, why, and where to. *Frontiers in Robotics and AI*, 2, 4. <https://doi.org/10.3389/frobt.2015.00004>
- Eiben, A., & Smith, J. (2003). *Introduction To Evolutionary Computing* (vol. 45). Springer. <https://doi.org/10.1007/978-3-662-05094-1>
- Gaier, A., Asteroth, A., & Mouret, J. B. (2020). *Automating representation discovery with map-elites*. ArXiv abs/2003.04389.
- Hiller, J., & Lipson, H. (2014). Dynamic simulation of soft multimaterial 3d-printed objects. *Soft Robotics*, 1(1), 88–101. <https://doi.org/10.1089/soro.2013.0010>
- Joachimczak, M., Suzuki, R., & Arita, T. (2016). Artificial metamorphosis: Evolutionary design of transforming, soft-bodied robots. *Artificial Life*, 22(3), 271–298. https://doi.org/10.1162/ARTL_a_00207
- Kriegman, S., Cheney, N., & Bongard, J. (2018). Publisher correction: How morphological development can guide evolution. *Scientific Reports*, 8(1), 16057. <https://doi.org/10.1038/s41598-018-33706-2>
- Lan, G., De Carlo, M., van Diggelen, F., Tomczak, J. M., Roijers, D. M., & Eiben, A. (2021). Learning directed locomotion in modular robots with evolvable morphologies. *Applied Soft Computing*, 111, 107688. <https://doi.org/10.1016/j.asoc.2021.107688>
- Lan, G., Jelisavcic, M., Roijers, D. M., Haasdijk, E., & Eiben, A. E. (2018). Directed locomotion for modular robots with evolvable morphologies. In: *Parallel problem solving from nature* (pp. 476–487).
- Lehman, J., & Stanley, K. O. (2011a). Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary Computation*, 19(2), 189–223. https://doi.org/10.1162/EVCO_a_00025
- Lehman, J., & Stanley, K. O. (2011b). Evolving a diversity of virtual creatures through novelty search and local competition. In: *Proceedings of the 13th annual conference on Genetic and evolutionary computation* (pp. 211–218). GECCO '11. Association for Computing Machinery.
- Lipson, H., SunSpiral, V., Bongard, J., & Cheney, N. (2016). On the difficulty of co-optimizing morphology and control in evolved virtual creatures. *Artificial Life*, 226–233.
- Liu, S., Matthews, D., Kriegman, S., & Bongard, J. (2020). *Voxcraft-sim, a gpu-accelerated voxel-based physics engine*. <https://github.com/voxcraft/voxcraft-sim>
- Mcculloch, W., & Pitts, W. (1943). A logical calculus of ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5, 127–147.
- Miconi, T., & Channon, A. (2006). Analysing co-evolution among artificial 3d creatures. In E. G. Talbi, P. Liardet, P. Collet, E. Lutton, & M. Schoenauer (Eds), *Artificial evolution* (pp. 167–178). Springer Berlin Heidelberg.
- Mouret, J. B., (2011). Novelty-based multiobjectivization. In S. Doncieux, N. Bredèche, & J. B. Mouret (Eds), *New horizons in evolutionary robotics* (pp. 139–154). Springer Berlin Heidelberg.
- Mouret, J. B., & Clune, J. (2015). *Illuminating search spaces by mapping elites*. ArXiv abs/1504.04909.
- Mouret, J. B., & Doncieux, S. (2009). Overcoming the bootstrap problem in evolutionary robotics using behavioral diversity. In: *2009 IEEE congress on evolutionary computation*. pp. 1161–1168.
- Nordmoen, J., Veenstra, F., Ellefsen, K. O., & Glette, K. (2020). Quality and diversity in evolutionary modular robotics. In: *2020 IEEE symposium series on computational intelligence (SSCI)* (pp. 2109–2116).
- Nygaard, T., Samuelsen, E., & Glette, K. (2017). Overcoming initial convergence in multi-objective evolution of robot control and morphology using a two-phase approach. In: *Proceedings of the European conference on the applications of evolutionary computation* (pp. 825–836).
- Omelianenko, I. (2019). *Hands-on neuroevolution with Python*.
- Pugh, J., Soros, L., & Stanley, K. (2016). Quality diversity: A new frontier for evolutionary computation. *Frontiers in Robotics and AI*, 3. <https://doi.org/10.3389/frobt.2016.00040>
- Sims, K. (1994). Evolving 3d morphology and behavior by competition. *Artificial Life*, 1(4), 353–372. <https://doi.org/10.1162/artl.1994.1.4.353>

- Stanley, K. O. (2007). Compositional pattern producing networks: A novel abstraction of development. *Genetic Programming and Evolvable Machines*, 8(2), 131–162. <https://doi.org/10.1007/s10710-007-9028-8>
- Stanley, K. O., & Miikkulainen, R. (2002). Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10(2), 99–127. <https://doi.org/10.1162/106365602320169811>
- Stensby, E. H., Ellefsen, K. O., & Glette, K. (2021). Co-optimising robot morphology and controller in a simulated open-ended environment. In: *Applications of evolutionary computation* (pp. 34–49). Springer International Publishing.

About the Authors



Luis Eguiarte-Morett studied an M.Sc. in Computer Science, and B.Sc. in Computer Engineering at the Universidad Nacional Autónoma de México (UNAM). Currently, he works as a Senior Software Engineer in Thomson Reuters. His research interests include artificial life, video game AI, and computational creativity.



Wendy Aguilar studied a PhD, M.Sc., and B.Sc. in Computer Science at the Universidad Nacional Autónoma de México (UNAM). Currently, she works as an Assistant Professor at the Department of Computer Science of the Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas, UNAM. She is also a tutor and lecturer in the Master's Program in Computer Science at UNAM. Her research interests include pattern recognition, computational creativity, and artificial life.