Seeking Quality Diversity in Evolutionary Co-design of Morphology and Control of Soft Tensegrity Modular Robots

Enrico Zardini
University of Trento
Trento, Italy
enrico.zardini@unitn.it

Davide Zappetti
École Polytechnique Fédérale de
Lausanne, Switzerland
davide.zappetti@epfl.ch

Davide Zambrano
École Polytechnique Fédérale de
Lausanne, Switzerland
davide.zambrano@epfl.ch

Giovanni Iacca
University of Trento
Trento, Italy
giovanni.iacca@unitn.it

Dario Floreano
École Polytechnique Fédérale de
Lausanne, Switzerland
dario.floreano@epfl.ch

ABSTRACT

Designing optimal soft modular robots is difficult, due to non-trivial interactions between morphology and controller. Evolutionary algorithms (EAs), combined with physical simulators, represent a valid tool to overcome this issue. In this work, we investigate algorithmic solutions to improve the Quality Diversity of co-evolved designs of Tensegrity Soft Modular Robots (TSMRs) for two robotic tasks, namely goal reaching and squeezing trough a narrow passage. To this aim, we use three different EAs, i.e., MAP-Elites and two custom algorithms: one based on Viability Evolution (ViE) and NEAT (ViE-NEAT), the other named Double Map MAP-Elites (DM-ME) and devised to seek diversity while co-evolving robot morphologies and neural network (NN)-based controllers. In detail, DM-ME extends MAP-Elites in that it uses two distinct feature maps, referring to morphologies and controllers respectively, and integrates a mechanism to automatically define the NN-related feature descriptor. Considering the fitness, in the goal-reaching task ViE-NEAT outperforms MAP-Elites and results equivalent to DM-ME. Instead, when considering diversity in terms of "illumination" of the feature space, DM-ME outperforms the other two algorithms on both tasks, providing a richer pool of possible robotic designs, whereas ViE-NEAT shows comparable performance to MAP-Elites on goal reaching, although it does not exploit any map.

CCS CONCEPTS

• **Computing methodologies** → **Evolutionary robotics**; **Genetic algorithms**.

KEYWORDS

Soft Tensegrity Modular Robots, Co-evolution, NEAT, Viability Evolution, MAP-Elites, Quality Diversity

1 INTRODUCTION

Soft robots might be one of the key technologies of the future. Indeed, their robustness and adaptive morphology allow them to overcome situations and perform tasks in which traditional hard robots are of limited applicability [18]. For example, they can perform locomotion on rough terrains with risking damage [19], or squeeze through narrow passages [1]. Hence, they can be employed in the exploration of hard-to-access environments [9], as well as in the medical field [28]. Recently, researchers have applied the soft robotic paradigm to modular robotics [29], developing novel soft modular robots able to explore a larger morpho-functional space than their rigid counterparts while exhibiting robustness and mechanical adaptability [8, 18].

Designing soft modular robots is notably difficult due to the hard-to-model dynamics of soft materials and the non-trivial interactions between morphology and controller [12]. These difficulties, together with the lack of analytical methods, make evolutionary algorithms (EAs), coupled with physics simulators, one of the most promising design tools [10]. Indeed, not only can EAs discover unconventional high-performing solutions, but they can also provide a pool of possible designs by "illuminating" the design space [21]. This latter aspect is especially significant in soft modular robots, where the influence of the morphology and controller parameters on the robot performance is hard to predict, thus having a number of eligible candidates for the physical realization is desirable. We should remark however that, although other paradigms to design soft modular robots exist [11, 13, 15, 20, 26], only a few involve the use of EAs: notably and extensively the voxel-based one (VSRs) [2, 24] and the tensegrity soft modular robots (TSMRs) [27], the latter in a preliminary work with an open-loop controller [4] and in another work [6] where the Multi-dimensional Archive of Phenotypic Elites (MAP-Elites) [21] was applied to diversify only behavioral properties rather than morphological ones.

This work also focuses on TSMRs. More precisely, on the joint optimization of TSMR morphologies and NN-based controllers, as the optimization of either morphology only or controller only via EAs does not leverage the potential synergy between them. Specifically, we perform the co-optimization using three different approaches, namely MAP-Elites and two proposed custom EAs: 1) Viability Evolution (ViE) [16] coupled with Neuro-Evolution of Augmenting Topologies (NEAT) [25], denoted as *ViE-NEAT*, and 2) a method based on MAP-Elites, dubbed as *Double Map MAP-Elites* (DM-ME). The latter makes use of two maps, associated to an entity-related and a controller-related feature descriptor (FD), respectively, and includes a dimensionality reduction mechanism [3] for the automatic definition of the controller-related FD.

We experimentally compare the three approaches on two robotic tasks: goal reaching and squeezing through a narrow passage. Our results show that DM-ME outperforms the other two approaches at illuminating the search space on both tasks. In addition, although

ViE-NEAT does not keep any map, it achieves very similar performance to MAP-Elites on goal reaching (in terms of illumination).

The rest of this paper is structured as follows: Section 2 deals with the background and related work; Section 3 introduces the proposed methods, i.e., ViE-NEAT and DM-ME; Section 4 describes the experimental setup; Section 5 presents the numerical results. Finally, Section 6 provides the main conclusions of our study.

2 BACKGROUND

This section introduces the TSMRs employed in this work and the EAs related to the algorithms we have considered. It also introduces the mechanisms that have been integrated in DM-ME.

2.1 Tensegrity soft modular robots

The term *tensegrity* has been coined by the architect R. Buckminster Fuller [7] and denotes a structure that maintains its mechanical integrity through the pre-stretching of some elements constantly in tension (known as strings or cables) connected in a network with other elements constantly under compression (called rods or struts). Several kinds of tensegrity structures exist: this work focuses on the icosahedron tensegrity, which consists of 6 rods connected by 24 pre-stretched cables. This particular structure has been chosen due to its properties: the ability to deform in all directions; the symmetry; the small number of rods and cables; the possibility to carry a payload in the inner volume [27]. The last point is particularly important since it allows to equip the structure with a servomotor, which enables structural contractions, and different kinds of sensors.

The term *modular* denotes robots composed of a variable number of building blocks. In this work, the icosahedron tensegrity has been used as a module. In detail, the nodes of the icosahedron draft eight triangular faces, which are exploited to link the modules together by connecting the triangles' vertices. In the experiments, we take in consideration only linear chains of modules, both for simplicity and feasibility of a future hardware implementation. The appearance of a TSMR in the simulation framework that we have used (see Section 4.1) is shown in Figure 1.

The term *soft* refers to the fact that these robots are not rigid. Indeed, they are able to deform their structure by contracting and expanding their modules.

2.2 Neuro-evolution of augmenting topologies

NEAT [25] is a well-known neuro-evolutionary algorithm capable of evolving both topology and weights of artificial neural networks (NNs). It exploits an advanced genetic encoding based on the *historical marking* mechanism. In practice, each topological feature is marked with a number that allows to match homologous genes while performing crossover; in this way different genotypes can be aligned and crossed over. Moreover, the initialization of the population is addressed through a *complexification* process: the initial population consists of minimal networks, which are gradually made more complex as the search progresses. In this way the evolution will focus on the most promising parts of a very high-dimensional search space. Diversity and innovations are preserved through speciation: the population is split into species by means of a clustering method that groups individuals by similarity and the competition is limited to individuals within each species. It is also worth mentioning the mechanism known as species-elitism, which preserves a certain number of species (the best ones) from extinction due to stagnation, i.e., no improvement for long time. In this work, we used the NEAT implementation provided by `NEAT-Python`[1].

2.3 Diversity-driven evolutionary algorithms

Among the various EAs proposed in the literature for seeking diversity (either explicitly or implicitly) the following two have been taken into account here: Viability Evolution ("ViE") [16, 17], which is characterized by a good balance between exploitation and exploration, and MAP-Elites [21], which is focused on "illuminating" the search space.

ViE is an evolutionary paradigm based on the elimination of *unviable* individuals. In detail, ViE makes use of *viability boundaries* that discriminate between viable individuals, which satisfy specific requirements, and unviable ones, not satisfying them. The boundaries in question are dynamic: at the beginning they are set so as to include all the individuals belonging to the population; then, they are progressively shrunk either towards the optimal values or towards target values that correspond to some constraints. In this work we used ViE to evolve a population of TSMR morphologies; since there are no specific constraints in this case, the viability boundary represents a limit on the worst fitness.

MAP-Elites is a Quality Diversity (QD) algorithm, i.e., an algorithm that aims at generating a large collection (archive) of diverse and high-performing solutions (see [22, 23] for more details on QD). In particular, it aims at discovering the best-performing solution of each cell of a user-defined feature space, which is a lower dimensional space with respect to the original search space. In order to be added to the archive, a solution is first projected onto the feature space, producing a feature descriptor; if the corresponding cell is empty or contains an individual with worse performance, the solution is added to the archive (replacing the occupant, if present), otherwise it is discarded. In this work we used MAP-Elites to evolve morphology-controller pairs.

2.4 MAP-Elites with multiple maps

Multiple feature maps have already been employed in previous research, although for the evolution of *a single entity*. For example, in [23], two MAP-Elites archives are employed for the evolution of the controller of a wheeled robot in a maze navigation task. The two FDs used there are the endpoint of the simulation and the main orientation of the robot for each fifth of the simulation; thus, they are two behavioral properties of the evolved controller. Multiple archives are used also in [5] (although with a different QD algorithm), but even there the FDs represent only behavioral features. Instead, the DM-ME algorithm introduced here is devised for the evolution of *two entities*: a generic entity (in our case, a TSMR morphology) and its related controller (a NN). Hence, the FDs of the two archives refer to distinct objects.

2.5 Automatic feature descriptor definition

The selection of the FD is a critical factor for MAP-Elites. A mechanism to define it in an automatic way has been introduced in

[1] https://github.com/CodeReclaimers/neat-python

[3]. In practice, the FD is obtained by applying a Dimensionality Reduction (DR) algorithm, i.e., the Principal Component Analysis (PCA), on the sensory data extracted during the evaluation of the individual; the DR's output vector represents the FD that is used for the insertion in the archive. Nevertheless, the FD in question consists of continuous values, which is acceptable for an unstructured archive, such as the one used in [3], but not for MAP-Elites. Hence, an additional processing step to discretize the feature values is required. Basically, a first discretization to determine the cells boundaries is performed at the beginning of the evolution using the values retrieved by the PCA for the initial population. Then, the boundaries are recomputed every time the PCA is fitted to data or a value outside the current boundaries is returned. It is worth also mentioning that the PCA is scale-sensitive; hence, the sensory data are not directly provided as input to the PCA, but is standardized first by removing the mean and dividing by the standard deviation. As regards the *standardizer*, it is fitted to data every time the fitting is done for the PCA. In conclusion, the only information that is required is the size of the FD (i.e., the number of cells per dimension).

3 PROPOSED METHODS

We present now the algorithmic details of the proposed methods, i.e., ViE-NEAT and Double Map MAP-Elites (DM-ME).

3.1 ViE-NEAT

ViE-NEAT consists in the parallel evolution of two populations, a population of entities (morphologies) and one of NN-based controllers. The former is evolved using ViE, the latter using NEAT. The only interaction between the two populations happens at the evaluation time: in detail, at each generation the individuals belonging to the two populations are randomly paired for evaluation. In principle pairing should be one-to-one. However, since in the NEAT implementation used here the population is not fixed in size, it may happen that an individual from one population is paired with two individuals from the other, due to the bigger size of the other population at play. The pseudocode is shown in Algorithm 1.

3.2 Double Map MAP-Elites (DM-ME)

DM-ME deals with individuals represented by an <entity-NN> pair: the entity encoding depends on the domain of application, whereas NEAT's genetic encoding is used for NNs.

Algorithm 1 ViE-NEAT

```
1: procedure main():
2:     P_ViE ← initViEPopulation()                    ▷ entity pop.
3:     P_NEAT ← initNEATPopulation()                  ▷ NN pop.
4:     for t = 1 → T do
5:         evaluate(P_ViE, P_NEAT)                    ▷ random pairing used
6:         P_ViE ← runOneGeneration(P_ViE)
7:         P_NEAT ← runOneGeneration(P_NEAT)
```

The functioning of DM-ME is illustrated in Algorithm 2. At the beginning, the archives are initialized with a large number of randomly generated <entity-NN> pairs, exploiting the corresponding

Algorithm 2 Double Map Map-Elites (DM-ME)

```
1: procedure main(numInitialSolutions, archiveSize):
2:     A_E ← ∅                                        ▷ entity archive
3:     A_NN ← ∅                                       ▷ NN archive
4:     PCA ← initPCA(archiveSize)
5:     X ← generateRandomSolutions(numInitialSolutions)
6:     evaluate(X)                                    ▷ get fitness and sensory data
7:     addAllToMap(A_E, X)
8:     addToNNMap(A_NN, PCA, X)
9:     for t = 1 → T do
10:        X ← randomSelection(A_E, A_NN)
11:        X ← randomVariation(X)
12:        evaluate(X)
13:        addAllToMap(A_E, X)
14:        addToNNMap(A_NN, PCA, X)
15: procedure addAllToMap(A, X):
16:    for x ∈ X do
17:        addToMap(A, x)
18: procedure addToNNMap(A_NN, PCA, X):
19:    SD_X ← getSensoryData(X)
20:    if PCA.notFitted() then
21:        PCA.fit(SD_X)
22:    else if isFittingTime then
23:        SD_A ← getSensoryData(A_NN)
24:        PCA.fit(SD_A)
25:        reInsert(A_NN, PCA)
26:    PCA.transform(X, SD_X)                         ▷ compute NN FD
27:    if outOfBounds(X) then
28:        PCA.reComputeBounds(A_NN, X)
29:        reInsert(A_NN, PCA)
30:        PCA.transform(X, SD_X)
31:    addAllToMap(A_NN, X)
```

sensory data to fit the PCA. Then, until the desired number of generations has been reached, a batch of elites is sampled from the two maps, half from each of them. Next, the individuals are mutated as follows: with equal probability, either only the entity is mutated, only the NN is mutated, or both of them are mutated. The mutation of the entity is domain-dependent, whereas the NNs are mutated according to NEAT's mutation operator (crossover is not used). After mutation, the individuals are evaluated and added to the archives based on the respective FDs. In this regard, each of the two archives is an independent MAP-Elites map, and the `addToMap()` archive insertion procedure is handled accordingly: an individual (an <entity-NN> pair) is added to an archive if the corresponding cell is empty or occupied by a worse individual, hence, it may be added to one map, but not to the other, depending on the fitness of the currently stored individuals. The peculiarity lies in the fact that a FD related to the entity is used for the first map, whereas a FD related to the NN is employed for the second one. As shown in [23], the use of multiple adequate FDs (in the form of multiple maps for MAP-Elites) leads to better results, especially in difficult domains, since different perspectives are taken into account. As for our DM-ME, the two FDs employed are obtained in different ways. In particular, the entity is projected as usual on a

Table 1: Main parameters for all the experiments.

	Goal Reaching	Squeezing
# Runs	10	10
Run's budget	45000 evaluations	45000 evaluations
# Targets	4	2
Distance	45 cm	60 cm
Bearing	90° l, 45° l, 45° r, 90° r	5° l, 5° r
# Simulation seeds	2	2
Simulation time	40 s	40 s

user-defined feature space, in order to obtain the corresponding FD. Instead, for the NN, the mechanism described in Section 2.5 is used. More specifically, the PCA is fitted using the sensory data of the individuals contained in the NN archive with a frequency that decreases exponentially, as in [3]; obviously, at every fitting, it is necessary to re-compute the FDs of the individuals that are already in the archive and re-insert them, which is done also when the space is discretized between two PCA updates due to the presence of values outside the current cells boundaries.

4 EXPERIMENTAL SETUP

This section presents the details of our experimental setup.

4.1 Implementation details

Apart from NEAT, for which we used the existing `NEAT-Python` library, the other EAs involved in the experimentation were implemented from scratch in Python, and coupled with a custom TSMR simulation framework developed in C++, named Tensoft. The latter is based on the NASA Tensegrity Robotics Toolkit[2], a collection of tools for modeling and simulating tensegrity robots, built on the top of the Bullet Physics[3] engine. Our source code is available at: https://github.com/lis-epfl/Tensoft-G21.

4.2 Tasks

TSMRs have only recently become a subject of study, as demonstrated by [14]; hence, apart for locomotion [4], their capabilities are still largely unknown. Here, we investigate two tasks that are fairly more complex than plain locomotion: *goal reaching* and *squeezing*.

Goal reaching consists in having the robot moving towards a target placed somewhere in the environment. While this is a typical task in robotics research, modular soft structures have been barely taken into account in this regard. In practice, the objective consists in finding a controller able to drive the associated morphology towards different targets, based on the sensory information provided, i.e., distance and bearing to the target. In detail, distance and bearing are computed with respect to the position and the direction of the front face of the head module, respectively. An example of final situation of the task in Tensoft is shown in Figure 1 (left).

Squeezing consists in having the robot shrinking through a restricted aperture that is narrower than the robot size in terms of width and/or height. Due to the high degree of complexity, this task has not been addressed extensively in previous research. In this work, squeezing has been dealt with as an advanced form of goal reaching, by positioning the target beyond an aperture narrower than the maximum robot width. At the beginning of the simulation, the robot is enclosed by four walls, and it has to pass through the aperture to reach the target. In this case the sensory information provided includes distance and bearing to the target, distance and bearing with respect to the center of the entrance of the aperture (computed analogously to the ones referred to the target), and the presence of obstacles in front of the head module within a certain range (set to 10 cm in our experiments). An example of final situation of the task in Tensoft is shown in Figure 1 (right).

[2]https://github.com/NASA-Tensegrity-Robotics-Toolkit/NTRTsim
[3]https://github.com/bulletphysics/bullet3/archive/2.88.tar.gz

Table 2: Parameter setting employed for ViE-NEAT.

ViE	
Population size (init.)	48
# Mutants	48

NEAT	
Population size	54
Individual elitism	3
Species elitism	2
Compatibility threshold	2.85

Table 3: Parameter setting for MAP-Elites and DM-ME.

	MAP-Elites	DM-ME
# Initial solutions	1080	1080
Batch size	24	24
Archive(s) size	[9, 10, 9, 10]	[9, 10] [9,10]
PCA updates	[0, 50, 150, 350, 750, 1550]	
Trajectory sampling	1 s	1 s

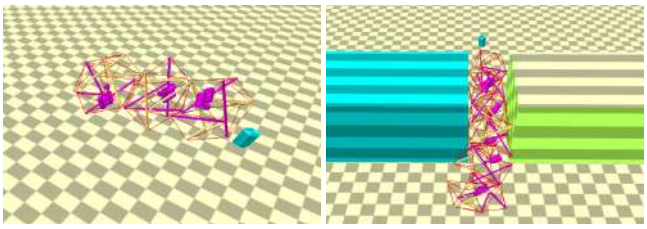


Figure 1: Example of final situation for goal reaching (left) and squeezing (right). The cyan cylinder depicts the target.

4.3 Encoding

The TSMR morphology encoding includes global properties, affecting the entire robot, and local ones, specific to each module. In detail, the global genes are the number of modules and the stiffness, which determines the degree of deformability of the modules. Instead, the local ones include only the number ρ, which identifies the triangular face that is connected to the next module in the chain. This influences not only the weight balance, but also how the robot moves since the servomotor acts always on the same pair of faces. The allowed mutations include: the addition of a new module in a random position; the deletion of a random module in the chain; the change of the robot stiffness; the change of the connection face to the next module (local mutation).

As concerns the controllers, the encoding and the allowed mutations are defined by NEAT. In particular, the TSMR controller has been implemented as a feed-forward NN that at each timestep of the simulation takes as input the sensory information provided for the considered task and produces as output the actuation parameters,

i.e., the frequency f and phase ϕ of a sinusoidal signal that controls contraction/expansion for all modules.

4.4 Algorithm and task configurations

Three different approaches have been applied to both the goal reaching and squeezing tasks:

- co-evolution of morphology, evolved through ViE, and controller, evolved through NEAT (i.e., using ViE-NEAT);
- evolution of morphology-controller pairs through MAP-Elites;
- evolution of morphology-controller pairs using DM-ME.

The main parameters for all the experiments, common to the three approaches, are reported for each task in Table 1. For a fair comparison, the budget, limited by the computational resources available, has been defined for all algorithms in terms of total number of solution evaluations.

Each solution is evaluated against different targets placed at different orientations and, since the simulations are noisy, for each target multiple simulations are run, with different seeds for the actuation noise. All the simulations contribute to the fitness of the individual, to be minimized, which is computed for both tasks as:

$$\overline{d_t} = \frac{1}{N_s}\sum_{s=1}^{N_s} d_{ts} \quad t \in \{1, \dots, N_t\}$$

$$f = \frac{1}{N_t}\sum_{t=1}^{N_t}\overline{d_t} + \frac{1}{2} \times (\max_{t=1\dots N_t}\overline{d_t} - \frac{1}{N_t}\sum_{t=1}^{N_t}\overline{d_t}) \tag{1}$$

where N_s is the number of simulation seeds, N_t is the number of targets, d_{ts} is the final distance of the individual from the t-th target using the s-th seed, and f is the resulting fitness.

In practice, first the average final distance across simulation seeds is calculated for each target. Then, the fitness of the individual is computed as the mean value of the average distances plus a penalty equal to half of the difference between the worst (maximum) and the mean average distance across targets. Actually, in the squeezing experiments d_{ts} represents the final distance with any potential bonus already deducted: if the robot succeeds in crossing the entrance of the aperture by at least 4 cm, a fixed bonus of 4 cm is deducted from the final distance. In this way, the individuals capable of entering the aperture are favored by evolution.

As concerns the squeezing experiments, we set the wall's width to 16.67 cm, its height to 12 cm, and the aperture width to 8 cm. In particular, the wall's width, which corresponds to the length of the aperture, is set to 1.5 times the length of a TSMR module. Instead, the width of the aperture is set to 72% of a module's width (note that the width of a module is equal to its length, i.e., about 11 cm).

A brief description of the approach-specific configurations, which are common to the two tasks, is provided in the following paragraphs. Indeed, only the number of controller input nodes is task-dependent (2 for goal reaching, 5 for squeezing, see Section 4.2).

ViE-NEAT. The parameter configuration of the ViE-NEAT approach is shown in Table 2. This configuration is characterized by elitism at both individual and species level, a one-to-one (on average) morphology-controller pairing for evaluation (a slightly larger NEAT population is required to compensate for the presence of elite individuals), and a relatively low threshold for speciation.

If an individual from one population is paired with two individuals from the other (due to the bigger size of the other population at play), the fitness of that individual is computed as:

$$f = \frac{1}{N_p}\sum_{p=1}^{N_p} f_p + \frac{1}{2} \times (\max_{p=1\dots N_p} f_p - \frac{1}{N_p}\sum_{p=1}^{N_p} f_p) \tag{2}$$

where N_p is the number of paired individuals belonging to the other population, f_p is the fitness obtained with the p-th paired individual, according to Eq. (1), and f is the resulting fitness. Since the controllers are not associated to a fixed morphology, the number of output nodes, which is morphology-dependent, is set to two times the maximum allowed number of modules, i.e., 20 (2×10), and only the required number of outputs is considered at the evaluation time (the same holds for the MAP-Elites and DM-ME experiments).

MAP-Elites and DM-ME. The configuration used for each of the two algorithms is reported in Table 3; the only difference lies in the archive(s) size. The mutation of individuals (morphology-controller pairs) in MAP-Elites is performed as in DM-ME (see Section 3).

As regards the feature descriptors, the FD used for MAP-Elites corresponds to the concatenation of the FDs employed in the two archives of DM-ME, i.e., the morphology-related and the controller-related FDs (see the archives size in Table 3). As a consequence, the archive filled by MAP-Elites has a higher number of cells than the ones explored by DM-ME (8100 cells vs 90 cells $\times$ 2 archives). Nevertheless, a comparison with equal number of cells would require using different FDs for the two algorithms, which would make the comparison unfair. Moreover, DM-ME has been specifically designed to reduce the total number of cells and thus to increase the selective pressure (with equal FDs) w.r.t. the original MAP-Elites.

In detail, the morphology-related FD is represented by the number of modules and the module stiffness; instead, the controller-related FD is provided by the PCA, exploiting the trajectory of the head module as sensory data. Since each morphology-controller pair is simulated with multiple targets, the sensory data vector are defined as the concatenation of the various trajectories: 320 elements in the case of goal reaching (4 targets $\times$ 40 s $\times$ 2 coordinates), and 160 elements in the case of squeezing (2 targets instead of 4). In addition, since a solution is simulated multiple times with different seeds, the average trajectory across seeds is taken as representative for each target. Actually, the size of the controller-related FD has been chosen so that the number of cells is the same for the morphology-related and the controller-related feature space.

5 RESULTS

In the following, we present the results achieved in the goal reaching and squeezing experiments. We must remark that the fitness trends, the heatmaps and the statistics shown here have been computed after applying the following transformation to the fitness values obtained throughout the evolution:

$$f_s = d_{init} - f \tag{3}$$

where d_{init} is the initial distance from the target (45 cm for goal reaching and 60 cm for squeezing, see Table 1), f is the fitness of the individual measured according to Eq. (1) or (2) during the evolutionary process, and f_s is the fitness used for plots and statistics

computation. In this way, the minimization problem is presented as a maximization one, to facilitate the analysis in terms of QD.

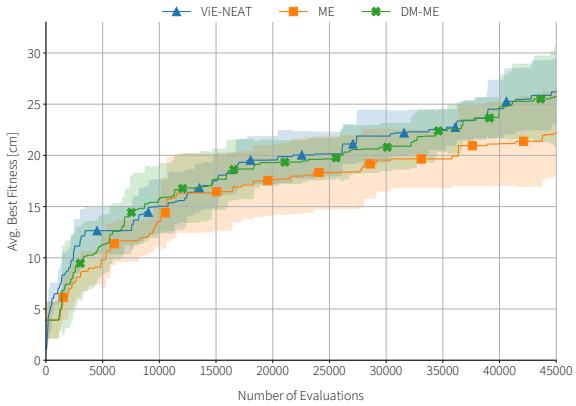


Figure 2: Best fitness (mean ± std. dev. across 10 runs) over evaluations for the goal reaching task. In particular, the trends shown are related to: the morphology population for ViE-NEAT; the one and only archive for MAP-Elites (ME); the morphology archive for DM-ME. The fitness values (to be maximized) shown here have been obtained through Eq. (3).

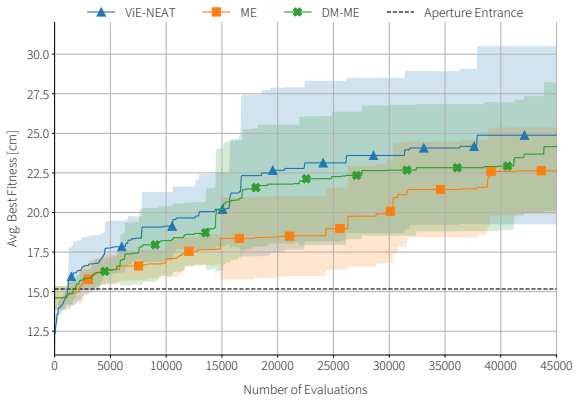


Figure 3: Best fitness (mean ± std. dev. across 10 runs) over evaluations for the squeezing task. In particular, the trends shown are related to: the morphology population for VIE-NEAT; the one and only archive for MAP-Elites (ME); the morphology archive for DM-ME. The fitness values (to be maximized) shown here have been obtained through Eq. (3).

All the experiments have been executed using a High Performance Computing (HPC) infrastructure. In detail, for each run, one node with 2 Intel Broadwell processors running at 2.6 GHz, with 14 cores each (28 cores in total), and 16/32 GB of RAM has been used.

Goal reaching. Figure 2 shows the best fitness trend (mean ± std. dev. across 10 runs) for ViE-NEAT, MAP-Elites and DM-ME. In detail, ViE-NEAT and DM-ME exhibit an almost equivalent performance on average, whereas MAP-Elites turns out to be worse. The descriptive statistics of the best fitness across runs are reported in Table 4 (up) for each algorithm. It can be seen that both ViE-NEAT and DM-ME obtain a higher mean best fitness value than MAP-Elites. It is also worth highlighting that the best individual across all the goal reaching experiments has been discovered by DM-ME[4]. Both the equivalence of ViE-NEAT and DM-ME and the superiority of ViE-NEAT with respect to MAP-Elites are statistically significant, see the results of the Wilcoxon rank-sum test in Table 4 (bottom).

Table 4: Descriptive statistics of the best fitness across 10 runs for the goal reaching task (up); Wilcoxon rank-sum test ($\alpha = 0.05$) applied to the distributions of best fitness (bottom).

Algorithm	Mean	Std	Min	Max	Median
ViE-NEAT	26.21	3.22	21.76	30.79	26.08
MAP-Elites	22.16	4.13	15.40	29.14	21.50
DM-ME	25.76	4.93	20.66	39.36	24.22

Algorithms		p-value
ViE-NEAT	MAP-Elites	0.034293
ViE-NEAT	DM-ME	0.545350
MAP-Elites	DM-ME	0.082099

Table 5: Descriptive statistics of the best fitness across 10 runs for the squeezing task (up); Wilcoxon rank-sum test ($\alpha = 0.05$) applied to the distributions of best fitness (bottom).

Algorithm	Mean	Std	Min	Max	Median
ViE-NEAT	24.88	5.63	18.81	35.94	22.75
MAP-Elites	22.63	2.74	18.60	26.05	22.63
DM-ME	24.16	4.07	18.99	31.00	23.49

Algorithms		p-value
ViE-NEAT	MAP-Elites	0.496292
ViE-NEAT	DM-ME	0.820596
MAP-Elites	DM-ME	0.496292

Table 6: Descriptive statistics of the QD score of the morphology archives generated in the goal reaching (up) and squeezing (bottom) experiments.

Algorithm	Mean	Std	Min	Max	Median
ViE-NEAT	535.61	59.21	461.81	641.67	524.60
MAP-Elites	552.27	35.41	519.24	642.73	544.26
DM-ME	656.08	35.05	589.36	707.55	655.60
Algorithm	Mean	Std	Min	Max	Median
ViE-NEAT	1196.26	60.05	1099.57	1291.28	1203.39
MAP-Elites	1293.25	15.30	1264.05	1314.11	1296.67
DM-ME	1322.52	14.60	1296.39	1341.12	1324.52

Table 7: Wilcoxon rank-sum test ($\alpha = 0.05$) applied to the QD score of the morphology archives generated in the goal reaching (GR) and squeezing (SQ) experiments.

Algorithms		p-value (GR)	p-value (SQ)
ViE-NEAT	MAP-Elites	0.325751	0.000881
ViE-NEAT	DM-ME	0.000881	0.000157
MAP-Elites	DM-ME	0.000507	0.002497

[4]The videos of the best individuals are available at https://tinyurl.com/ynaav7ek.

The table also shows that in this case DM-ME does not statistically outperform MAP-Elites.

The other evaluation metric we are interested in evaluating here is the *illumination* of the feature space(s). With reference to Figure 4 (left column, first three rows), which provides a heatmap representation of the archives, or archive projections, generated in five runs of the goal reaching experiments, it can be noted that DM-ME appears to perform better in terms of morphology archives produced (although the difference is not so pronounced), whereas ViE-NEAT shows almost equivalent performance to MAP-Elites. As regards the controller archives, it is worth remembering that each cell corresponds to a different trajectory according to the PCA and, since not all trajectories can lead to good results, it is reasonable to find also bad performing individuals inside them. In addition, the coverage of the controller archives is influenced by the re-discretization of the features space. Hence, it is reasonable to focus on the morphology archives. The difference in terms of illumination capability between the three algorithms is confirmed by the statistics computed on the QD score (i.e, the sum of the fitness of the individuals in the archive [22]) of the morphology archives. All the ten runs per algorithm have been taken into account for the computation of these statistics, see Table 6 (up). Both the superiority of DM-ME on the other two methods and the equivalence between ViE-NEAT and MAP-Elites in terms of QD score are statistically significant, see the results of the Wilcoxon rank-sum test in Table 7.

An inverse analysis has been also performed on MAP-Elites and DM-ME. The double archives generated by DM-ME have been projected onto single ones, and then compared with the single archives produced by MAP-Elites. This requires to re-simulate the individuals contained in the morphology archives of DM-ME in order to collect their trajectories and compute the controller-related FDs. Specifically, the PCA and the cell boundaries used for the projection (i.e., the final ones) are potentially different from the ones used for the evaluation of the considered individuals during the evolution. Hence, if the reconstructed single archives were projected back onto separate morphology-controller ones, the controller archives would be potentially different from the original ones. The results of the double-to-single projection show that most of the individuals contained in the controller archives generated by DM-ME tend to be concentrated in a small number of cells in the morphological feature space, typically with a medium-high number of modules and very low stiffness. This is reasonable, since those morphological properties allow a greater variety of mid-performing behaviors. Instead, as expected, the single archives generated by MAP-Elites show a higher coverage of the 4-dimensional feature space, especially in the region of medium-low stiffness. The heatmap representation of the single archives can be found in the Supplementary Material.

Concerning the execution time, MAP-Elites (154.3 ± 9.22 hours) and DM-ME (145.4 ± 10.76 hours) turn out to be considerably more expensive than ViE-NEAT (94.3 ± 13.15 hours). It is worth mentioning however that both MAP-Elites and DM-ME present an overhead due to the need for collecting the sensory data (trajectories), which is written by the simulator into temporary files.

Squeezing. Figure 3 shows the best fitness trend (mean ± std. dev. across 10 runs) for the three algorithms. The black dashed line represents the fitness limit corresponding to a robot entering the aperture in a single simulation. By looking at the plot, it turns out that the three algorithms discover well-performing solutions. In this case, the best individual has been discovered by ViE-NEAT, and again both ViE-NEAT and DM-ME obtain a higher mean best fitness value than MAP-Elites, see the descriptive statistics reported in Table 5 (up). On the other hand, the three algorithms result statistically equivalent, see the results of the Wilcoxon rank-sum test provided in Table 5 (bottom). Nevertheless, by looking at the behavior of the best individuals[5], it turns out that the best individual evolved by DM-ME is the only one, among the best individuals obtained by all algorithms, that is able to pass through the aperture and reach the target if the simulation is allowed to continue beyond the 40 s used in the evolutionary process. This also demonstrates that the task is achievable.

As concerns the illumination of the search space, Figure 4 (right column) provides a heatmap representation of the archives, or archive projections, generated in five runs of the squeezing experiments. As in the case of goal reaching, we focus the analysis on the morphology archives. In this case, some runs of ViE-NEAT are unable to fill all the cells of the grid. Except for that, the archives generated appear to be very similar, especially the ones produced by MAP-Elites and DM-ME. Nevertheless, by looking at the statistics of the QD score of the archives generated in the ten runs, reported in Table 6 (bottom), it turns out that DM-ME outperforms MAP-Elites, which in turn outperforms ViE-NEAT. Table 7 provides the statistical evidence based on the Wilcoxon rank-sum test.

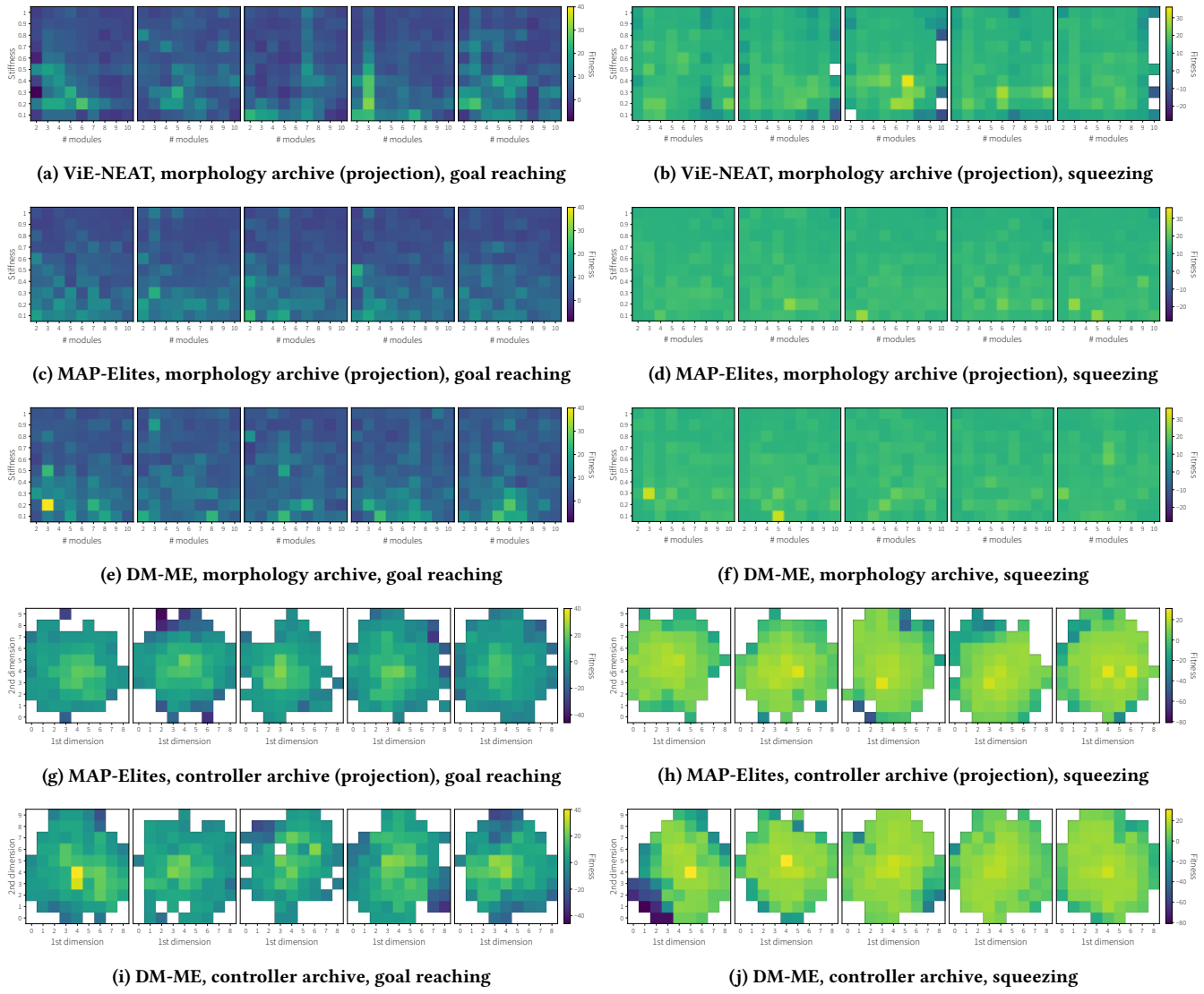
The heatmap representation of the single archives, the ones generated by MAP-Elites and the ones resulting from the projection of the double archives produced by DM-ME, can be found in the Supplementary Material. The observations made for goal reaching, as concerns this inverse analysis, hold also for squeezing.

As in goal reaching, the execution time of MAP-Elites (103.3±5.87 hours) and DM-ME (94.9 ± 12.23 hours) turns out to be higher than that of ViE-NEAT (70.9 ± 6.07 hours). The lower execution time with respect to goal reaching is mainly due to the number of targets considered for each individual, which in this case is 2 instead of 4 (see Table 1). However, a single simulation of the squeezing task tends to require a little more time than one of the goal reaching task, due to the interactions between the robot and the walls.

6 CONCLUSIONS

In this work we have addressed the joint optimization of morphology and controller of TSMRs. In detail, we have considered three different evolutionary approaches, i.e., MAP-Elites, ViE-NEAT, and DM-ME, with the last two being algorithms proposed here for co-evolving morphologies and controllers. In order to compare the three algorithms, we have conducted an experimental campaign on two robotic tasks: goal reaching and squeezing. As concerns goal reaching, ViE-NEAT outperforms MAP-Elites and results equivalent to DM-ME in terms of best fitness. As regards squeezing, the three algorithms achieve similar results in terms of quality of discovered solutions. Moreover, DM-ME outperforms the other two approaches in terms of illumination/exploration of the feature space (measured with QD score) in both tasks. The higher total fitness of the resulting archive implies a higher number of well-performing

[5]The videos of the best individuals are available at https://tinyurl.com/y3e9neej.



(a) ViE-NEAT, morphology archive (projection), goal reaching

(b) ViE-NEAT, morphology archive (projection), squeezing

(c) MAP-Elites, morphology archive (projection), goal reaching

(d) MAP-Elites, morphology archive (projection), squeezing

(e) DM-ME, morphology archive, goal reaching

(f) DM-ME, morphology archive, squeezing

(g) MAP-Elites, controller archive (projection), goal reaching

(h) MAP-Elites, controller archive (projection), squeezing

(i) DM-ME, controller archive, goal reaching

(j) DM-ME, controller archive, squeezing

Figure 4: Archives generated by five runs of each experiment configuration. The first row shows the archives generated by ViE-NEAT, obtained by projecting on the morphology feature space the individuals discovered throughout the evolution (the reconstruction of the controller archives would require modifying the algorithm since ViE-NEAT does not include the PCA). The second and third row show the morphology-related archives generated by MAP-Elites (i.e., the projection from the resulting single archive) and DM-ME, respectively. The fourth and fifth row display the corresponding controller-related archives. In all cases, the fitness values (to be maximized) shown here have been obtained through Eq. (3).

candidates for the physical realization. Notably, ViE-NEAT achieves very similar performance to MAP-Elites in terms of illumination of the search space in the goal reaching task, although it does not exploit any map. Future work includes testing other configurations of DM-ME and co-evolution, other robotic tasks, and more complex TSMR structures.

ACKNOWLEDGMENTS

We thank Daniele Bissoli for implementing the original framework on which this work was built.

REFERENCES

[1] Nick Cheney, Josh Bongard, and Hod Lipson. 2015. Evolving Soft Robots in Tight Spaces. In *Proceedings of the Genetic and Evolutionary Computation Conference*. ACM, New York, NY, USA, 935–942.

[2] Nick Cheney, Robert MacCurdy, Jeff Clune, and Hod Lipson. 2013. Unshackling evolution: evolving soft robots with multiple materials and a powerful generative encoding. In *Proceedings of the Genetic and Evolutionary Computation Conference*. ACM, New York, NY, USA, 167–174.

[3] Antoine Cully. 2019. Autonomous skill discovery with quality-diversity and unsupervised descriptors. In *Proceedings of the Genetic and Evolutionary Computation Conference*. ACM, New York, NY, USA, 81–89.

[4] D. Floreano D. Zappetti, J. M. Bejjani. 2021. Evolutionary co-design of morphology and control of soft tensegrity modular robots with programmable stiffness. arXiv:2101.11772 [cs.RO]

[5] Stephane Doncieux and Alexandre Coninx. 2018. Open-Ended Evolution with Multi-Containers QD. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion* (Kyoto, Japan) *(GECCO '18)*. Association for Computing Machinery, New York, NY, USA, 107–108. https://doi.org/10.1145/3205651.3205705

[6] K. Doney, A. Petridou, J. Karaul, A. Khan, G. Liu, and J. Rieffel. 2020. Behavioral Repertoires for Soft Tensegrity Robots. In *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, Canberra, Australia, 2265–2271. https://doi.org/10.1109/SSCI47803.2020.9308218

[7] Buckminster Fuller. 1961. Tensegrity. *Portfolio Artnews Annual* 4 (1961), 112–127.

[8] Jürg Germann, Andrea Maesani, Ramon Pericet-Camara, and Dario Floreano. 2014. Soft cells for programmable self-assembly of robotic modules. *Soft Robotics* 1, 4 (2014), 239–245.

[9] Ahmed Hallawa, Giovanni Iacca, Cagatay Sariman, Touhidur Rahman, Michael Cochez, and Gerd Ascheid. 2020. Morphological evolution for pipe inspection using Robot Operating System (ROS). *Materials and Manufacturing Processes* 35, 6 (2020), 714–724.

[10] David Howard, Agoston E Eiben, Danielle Frances Kennedy, Jean-Baptiste Mouret, Philip Valencia, and Dave Winkler. 2019. Evolving embodied intelligence from materials to machines. *Nature Machine Intelligence* 1, 1 (2019), 12–19.

[11] Toby Howison, Simon Hauser, Josie Hughes, and Fumiya Iida. 2020. Reality-assisted evolution of soft robots through large-scale physical experimentation: a review. arXiv:2009.13960 [cs.RO]

[12] Cecilia Laschi, Barbara Mazzolai, and Matteo Cianchetti. 2016. Soft robotics: Technologies and systems pushing the boundaries of robot abilities. *Science Robotics* 1, 1 (2016), 11 pages.

[13] Chiwon Lee, Myungjoon Kim, Yoon Jae Kim, Nhayoung Hong, Seungwan Ryu, H Jin Kim, and Sungwan Kim. 2017. Soft robot review. *International Journal of Control, Automation and Systems* 15, 1 (2017), 3–15.

[14] Hajun Lee, Yeonwoo Jang, Jun Kyu Choe, Suwoo Lee, Hyeonseo Song, Jin Pyo Lee, Nasreena Lone, and Jiyun Kim. 2020. 3D-printed programmable tensegrity for soft robotics. *Science Robotics* 5, 45 (2020), 11 pages.

[15] Jun-Young Lee, Woong-Bae Kim, Woo-Young Choi, and Kyu-Jin Cho. 2016. Soft robotic blocks: Introducing SoBL, a fast-build modularized design block. *IEEE Robotics & Automation Magazine* 23, 3 (2016), 30–41.

[16] Andrea Maesani, P. N. R. Fernando, and Dario Floreano. 2014. Artificial Evolution by Viability Rather than Competition. *PLoS ONE* 9 (2014), 1–12.

[17] Andrea Maesani, Giovanni Iacca, and Dario Floreano. 2016. Memetic Viability Evolution for Constrained Optimization. *IEEE Transactions on Evolutionary Computation* 20, 1 (2016), 125–144.

[18] Stefano Mintchev and Dario Floreano. 2016. Adaptive morphology: A design principle for multimodal and multifunctional robots. *IEEE Robotics & Automation Magazine* 23, 3 (2016), 42–54.

[19] Stefano Mintchev, Davide Zappetti, Jerome Willemin, and Dario Floreano. 2018. A soft robot for random exploration of terrestrial environments. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, Piscataway, NJ, USA, 7492–7497.

[20] Stephen A Morin, Sen Wai Kwok, Joshua Lessing, Jason Ting, Robert F Shepherd, Adam A Stokes, and George M Whitesides. 2014. Elastomeric tiles for the fabrication of inflatable structures. *Advanced Functional Materials* 24, 35 (2014), 5541–5549.

[21] Jean-Baptiste Mouret and Jeff Clune. 2015. Illuminating search spaces by mapping elites. arXiv:1504.04909 [cs.AI]

[22] Justin Pugh, L. Soros, Paul Szerlip, and Kenneth Stanley. 2015. Confronting the Challenge of Quality Diversity. In *Proceedings of the Genetic and Evolutionary Computation Conference*. ACM, New York, NY, USA, 967–974.

[23] Justin K. Pugh, Lisa B. Soros, and Kenneth O. Stanley. 2016. Quality Diversity: A New Frontier for Evolutionary Computation. *Frontiers in Robotics and AI* 3 (2016), 17 pages.

[24] Dylan Shah, Bilige Yang, Sam Kriegman, Michael Levin, Josh Bongard, and Rebecca Kramer-Bottiglio. 2020. Shape Changing Robots: Bioinspiration, Simulation, and Physical Realization. *Advanced Materials* 2002882 (2020), 12 pages.

[25] Kenneth O. Stanley and Risto Miikkulainen. 2002. Evolving Neural Networks through Augmenting Topologies. *Evolutionary Computation* 10 (2002), 99–127.

[26] Andrea Vergara, Yi-sheng Lau, Ricardo-Franco Mendoza-Garcia, and Juan Cristóbal Zagal. 2017. Soft modular robotic cubes: toward replicating morphogenetic movements of the embryo. *PloS one* 12, 1 (2017), e0169179.

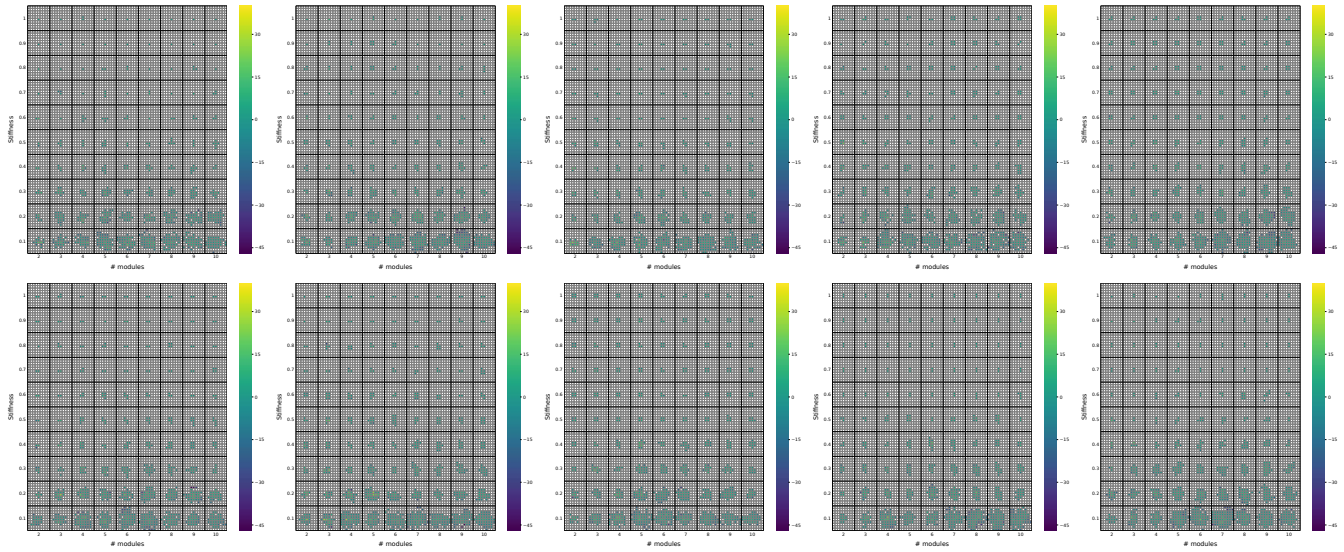
[27] Davide Zappetti, Stefano Mintchev, Jun Shintake, and Dario Floreano. 2017. Bio-inspired Tensegrity Soft Modular Robots. In *Proceedings of Conference Living Machines: Biomimetic and Biohybrid Systems*. Springer, Cham, Switzerland, 497–508.

[28] Boyu Zhang, Yingwei Fan, Penghui Yang, Tianle Cao, and Hongen Liao. 2019. Worm-like soft robot for complicated tubular environments. *Soft robotics* 6, 3 (2019), 399–413.

[29] Chao Zhang, Pingan Zhu, Yangqiao Lin, Zhongdong Jiao, and Jun Zou. 2020. Modular Soft Robotics: Modular Units, Connection Mechanisms, and Applications. *Advanced Intelligent Systems* 2, 6 (2020), 1900166.

SUPPLEMENTARY MATERIAL

MAP-Elites



Double Map MAP-Elites (DM-ME)

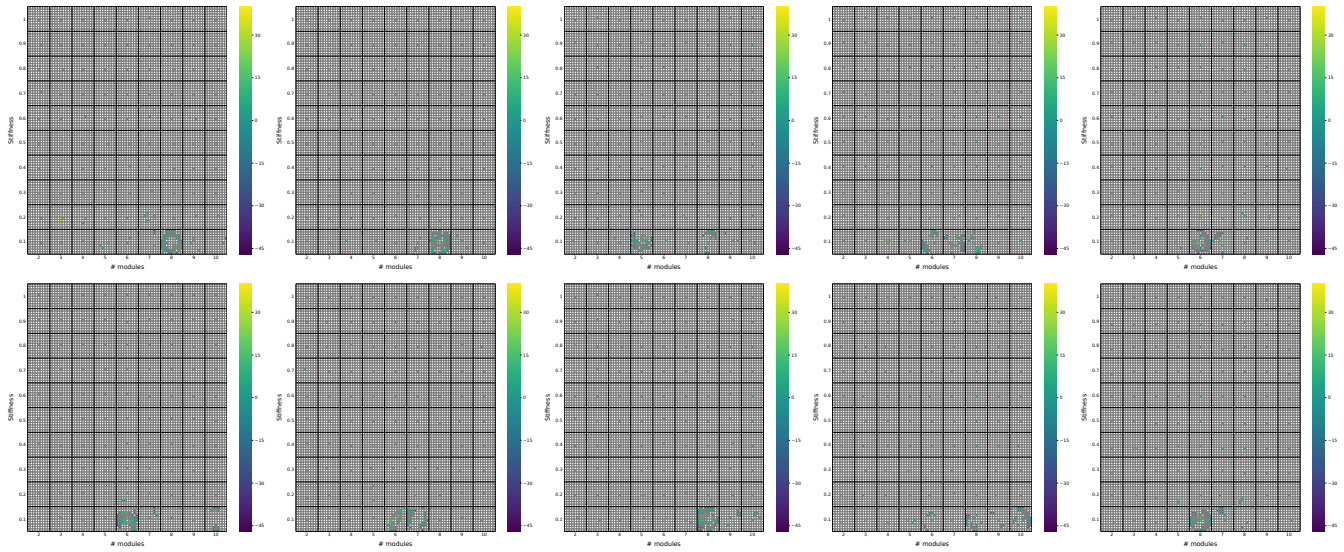


Figure 5: Single archives for the goal reaching task. In detail, the first two rows show the single archives generated by the 10 runs of MAP-Elites, whereas the other two show the projection of the double archives produced by Double Map MAP-Elites (DM-ME) onto a single one.

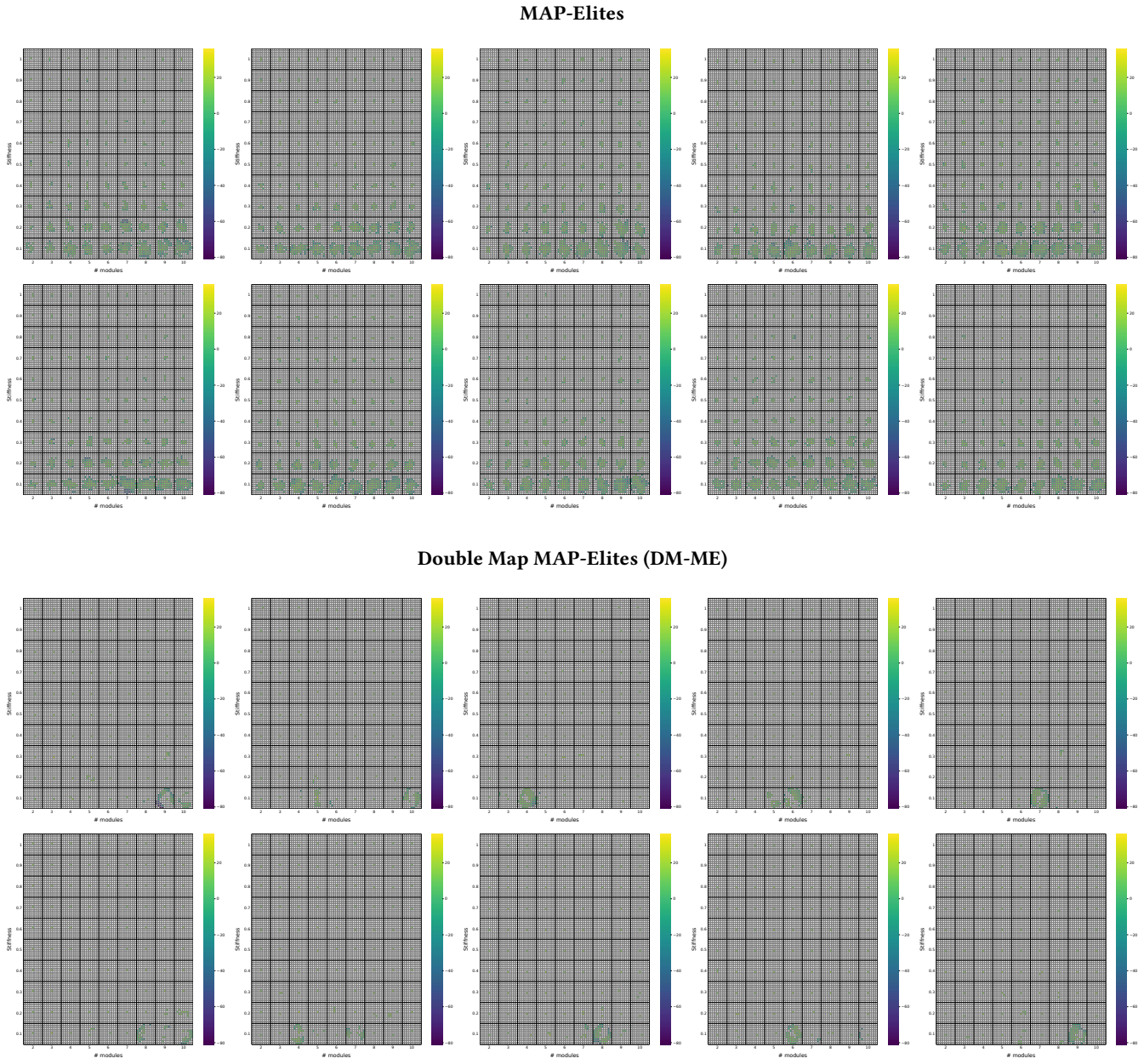


Figure 6: Single archives for the squeezing task. In detail, the first two rows show the single archives generated by the 10 runs of MAP-Elites, whereas the other two show the projection of the double archives produced by Double Map MAP-Elites (DM-ME) onto a single one.