Soft Robots Evolution

Stefano Leonardi Mat. 203777, Luca Zanella Mat. 207520, Enrico Zardini Mat. 207465

I. Introduction

This work aims at illustrating the approach adopted and the results obtained in soft robots evolution.

Soft robots are robots composed of materials similar to those living organisms are made of. In order to simulate them and their behaviour, *Voxelyze* physics engine and *VoxCAD* GUI [1] have been used. In particular, *Voxelyze* allows us to define materials with different properties and to use them to build robots, which are then simulated in an environment with customizable characteristics. The peculiarity lies in the simulation of the active materials corresponding to muscles: in fact, their contraction and expansion are determined by the variation of an environmental parameter, namely the temperature. The temperature is modelled as a sinusoidal function, characterised by customizable period and amplitude, whereas the active materials present non-zero Coefficient of linear Thermal Expansion (CTE). In this way, the temperature acts as a nerve impulse.

Voxelyze is a simulation library written in C++. In order to interact with it, *Evosoro* [2] has been used. Basically, *Evosoro* is a Python library built on the top of *Voxelyze* that allows to evolve and easily simulate soft robots: in fact, it provides an implementation of all the elements required to perform evolution, from the genotype -the Compositional Pattern Producing Network (CPPN) that defines the robot structure- to the mutation and selection operators, such as the Pareto selection.

Starting from this, four different scenarios have been explored:

- the evolution of new materials;
- the usage of a novelty-based approach in robots selection;
- the effect of evolving the controller, i.e. the signal determining the robot's movement;
- the evolution of robots able to overcome obstacles.

For each of them, Section II will describe the approach adopted, the problems encountered and the solutions applied, whereas Section III will present the results obtained. Finally, Section IV will conclude the work.

The source code is available at: https://github.com/ZarHenry96/Ev-SoftBots.

II. Methods

A. Materials Evolution

In [2], each robot can be made of four materials: green voxels, which expand and contract with a given frequency; red voxels, which are similar to green ones, but with counter-phase actuation; light blue voxels, which are passive and are not subject to actuation; blue voxels, which are similar to light blue ones, but are more resistant to deformation. In order to exploit the capabilities of Genetic Algorithms (GAs) in optimizing variables, an evolution of the physical properties each material is made of has been performed.

As mentioned in Section I, the genotype of each soft robot is represented by a CPPN which uniquely determines the morphology of the robot. The output layer of this network is composed of four nodes which determine the material which should be present in a given voxel. In order to evolve additional materials, four new output nodes have been introduced. The physical properties of each material that can be evolved by the GA are:

- Young modulus: the mechanical property which measures the stiffness of a solid material.
- Density: the measure of mass per unit of volume.
- CTE: the coefficient of thermal expansion.

At the beginning of the simulation, these new materials are initialized using the same physical properties of the already existing materials. Then, since these are real-valued properties, the materials of a new individual, generated through CPPN mutation by NEAT, are combined with the materials of another individual using arithmetic crossover with a crossover rate of 0.4. Additionally, these properties are mutated with a Gaussian distribution. The challenge is to find a proper set of constraints for each one of the physical property, otherwise, the robot might achieve a large fitness[1] value, exhibiting a strange behaviour which needs to be avoided. This problem has been addressed using the parameters suggested in [2].

Two types of objective functions have been optimized:

- robots which maximize the distance reached at the end of the simulation and minimize the number of old materials employed;
- robots which use the same criteria but also minimize the number of voxels.

B. Novelty-Based Selection

Instead of evaluating a candidate based on an objective function, novelty search evaluates new candidates based on their "*novelty*". An individual is said to be "*novel*" when it exhibits a behaviour different from the behaviours of the individuals in the same generation. In particular, the trajectory of each robot has been considered to determine its novelty. The challenge here is to find a good metric to evaluate the similarity between trajectories. In our work, to achieve this task, the angular metric for shape similarity (AMSS) [3] has been used. This similarity measure is based on vectors representing the shape of the data through time and compares different vector sequences using a variant of the cosine similarity.

During each simulation, the centroids of every soft robot are collected through time. In order to have more comparable

[1]The fitness is intended as the euclidean distance between the final and the initial position.

data, a roto-translation transformation is applied to the (x, y) coordinates of each centroid to align the motion of soft robots to the same initial direction. Then, between every two successive points, the displacement vector is obtained by computing their difference. Finally, the sequence of these displacement vectors forms the trajectory of the soft robot.

After the computation of the trajectories, the similarity between each robot's trajectory and all other robots' trajectories is evaluated. Since the population has a fixed size, only the individuals with the highest dissimilarity are selected to form the population of the next generation.

C. Controller Evolution

As explained in Section I, the muscles' contraction and expansion are simulated through the temperature variation in *Voxelyze*. Hence, the controller, i.e. the sinusoidal signal determining the robot's movement, is defined by three elements: the period of variation of the temperature; the temperature amplitude; the CTE of the muscles.

In practice, the basic genotype -the CPPN- has been extended with a controller, which is initialised using the default values in [2] and is evolved using a simple GA. Basically, once the new individuals have been generated through CPPN mutation by NEAT, a peculiar crossover is performed on a certain percentage of children: in particular, the controller resulting from the arithmetic mean of the controller parameters of two random individuals taken from the previous generation is associated to the mutated CPPN; in this way, the variability increases. At that point, a Gaussian mutation is applied to all new individuals, limiting the resulting values to specific ranges in order to avoid uncontrolled behaviours.

Moreover, in addition to the pre-defined Pareto selection, which is based on the concepts of Pareto dominance and Pareto level, a new selection function, namely the fitness tournament, has been developed: *tournament_size* random individuals are taken from the resulting population ($parents + children$) and the worst element in terms of fitness is deleted; this procedure is repeated until the population has returned to the original size.

D. Evolution versus Obstacles

Voxelyze allows users to define static structures of voxels through the so-called *Boundary Conditions*. These ones, if designed appropriately, can be exploited as obstacles in the simulations. Hence, the original idea was to perform competitive co-evolution of soft robots and environments. However, due to the lack of computational power, the long-time required to execute the simulations and the results obtained in the simplest scenario, the objective has turned into the evolution of a robot able to overcome a simple obstacle: a one voxel high wall.

Since the robots do not move all in the same direction, the obstacles should enclose them. In practice, they have been designed as square fences, which can also be assembled to build stairs or other types of obstacles. Another aspect that is worth mentioning is the fact that the pre-defined fitness metric is the euclidean distance between the final and the initial position. Hence, robots moving towards the corners would be preferred. In order to avoid this, new fitness metrics have been implemented in *Voxelyze*, namely:

- *MaxXYDist*, i.e. the maximum value between the distance covered w.r.t. the *X* axis and the distance covered w.r.t. the *Y* axis;
- the *L coefficient*, defined as the product between the previous metric and the speed at which that distance has been reached. In this way, the robots moving faster towards the obstacles are preferred.

Finally, since the robots evolved using the default controller values turned out not to have enough energy to overcome the obstacles, controller evolution has been exploited also in this scenario.

III. Results

This section presents the results obtained in the four scenarios. In particular, all simulations have been run using the parameters illustrated in Table I.

TABLE I
SIMULATION PARAMETERS VALUES

Parameter	Value
Individual size	$(6, 6, 6)$ $voxel$
Population size	15
Random individuals	1
Simulation time	5 $seconds$
Initialization time	1 $second$

A. Materials Evolution

The evolution of the materials has been performed optimizing two penalty regimes. All two require the soft robot to maximize the distance between the centre of mass of the initial and final position of the robot and to minimize the number of "*standard*" materials employed. Additionally, in the second treatment, the robots are penalized for their number of voxels. Since this is a multi-objective optimization problem, the Pareto selection technique provided by [2] has been used.

Through a qualitative analysis, the behaviours evolved by the best soft robots resemble the ones that have been observed in [2], with the difference that they are composed of materials with different physical properties. In both objectives, the robots exploit the coordination of contiguous regions of the same novel materials to achieve locomotion. While in the first treatment the best evolved soft robots leverage as many voxels as possible to move, in the second treatment soft robots succeed in achieving locomotion using a little number of materials.

In the former, the best-evolved individuals exploit a combination of a muscle with the highest/lowest CTE possible according to our constraints and a highly deformable tissue, both with a high density. In particular, the soft robot with the highest locomotion speed is formed by muscles with a CTE of -0.04 and a density of 586635 kg/m^3, and tissues with Young's modulus of 5 MegaPascal and a density of 1273400 kg/m^3. In the latter, several robots which fall to the floor and move in a similar way a fish would do out of the water

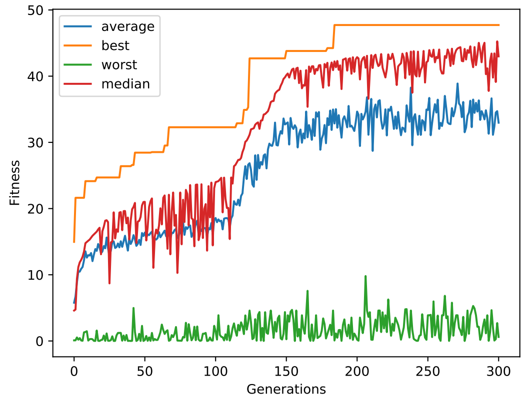

Fig. 1. A representation of the fitness trends, defined as the final euclidean distance, of the soft robots, evolved together with the materials they are composed of.

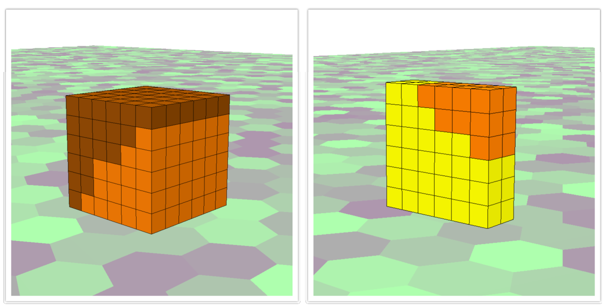

Fig. 2. Best soft robots obtained through materials evolution, without penalising the number of voxels (left) and by penalising the number of voxels (right), respectively.

have been observed. Since in this scenario robots with a high number of voxels are penalized, the best-evolved individuals mostly exploit muscles that expand and contract with a high CTE. In particular, the soft robot with the highest locomotion speed is formed by two types of muscles with a CTE of 0.04 and -0.04 respectively. The best soft robots evolved in both scenarios are depicted in Figure 2. The orange and yellow materials represent muscles (with opposite CTE) and the brown material represents a tissue.

The fitness trends of the first treatment, which has been simulated for 300 generations, are shown in Figure 1. Although a large number of generations is required to find soft robots which move faster than the best ones, it can be noticed how increasing the number of generations lead to robots which become increasingly better. Differently from the first objective, the second treatment has been simulated for 50 generations.

B. Novelty-Based Selection

Evolving the soft robots without the objective to improve the fitness, but only maximizing the differences between the trajectories of the individuals, turn out to exploit particular behaviours analysed over 200 generations:

- No constraints were applied to the number of voxels or the type: tissues/muscles, despite that the generated robots are mainly composed of muscles. A majority of tissues in a robot does not significantly change the trajectory, so the robot is not well evaluated.
- As expected the overall resulting fitness is not comparable w.r.t. the other two evolution techniques, it is much lower.
- An encountered problem is the repetition over the generations of the same voxels patterns. The selection of the best individuals is performed w.r.t. the current generation and not the previous ones, if a soft robot is identical to already appeared ones this is not penalized, resulting in a "*patterns loop*".

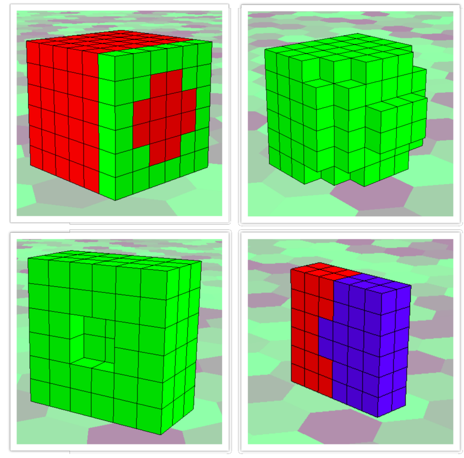

Fig. 3. Examples of behaviours evolved to be as different as possible.

Concluding, an overview of some samples evolved are shown in Figure 3.

C. Controller Evolution

In order to verify the effectiveness of controller evolution, three types of simulations have been employed:

- basic evolution of soft robots using what provided by [2], i.e. Pareto selection as the selection function, maximization of the final euclidean distance and minimization of the number of muscles as objectives;
- controller evolution with maximization of the final euclidean distance as objective and the fitness tournament described in Section II-C as selection mechanism;
- controller evolution using Pareto selection as selection function, maximization of the final euclidean distance and minimization of the number of muscles as objectives.

As regards the controllers use a crossover rate equal to 0.4.

The fitness -the final euclidean distance- of the best individual for each type of simulation over 40 generations is shown in Figure 4. Basically, the controller evolution outperforms the basic one, as expected. In particular, the best individuals obtained are characterised by lower period and higher temperature amplitude and CTE. Moreover, the fitness tournament achieves better results than the Pareto selection in terms of distance covered, which is reasonable since the latter also tries to minimize the number of muscles. In fact, by looking at the best individuals evolved -which are shown in Figure 5- it appears evident that the one obtained using Pareto selection

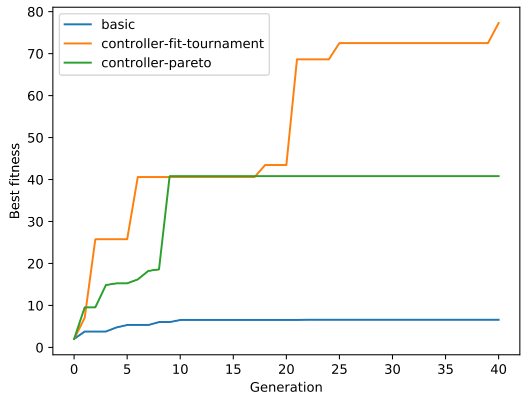

Fig. 4. Fitness of best individual in basic and controller evolution (fitness tournament / Pareto selection) throughout generations.

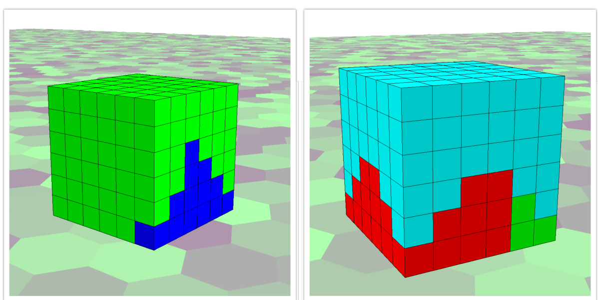

Fig. 5. Best soft robots obtained through controller evolution, using fitness tournament (*left*) and Pareto selection (*right*), respectively.

(on the right) has far fewer muscles. Nevertheless, from a qualitative point of view, the soft robot evolved using Pareto selection turns out to be better: indeed, its locomotion strategy resembles more one of a living organism.

D. Evolution versus Obstacles

As explained in Section II-D, the obstacles to face have been limited to a one-voxel high wall. Several attempts have been done in order to obtain a robot able to overcome it:

1) Evolving the robot using Pareto selection as selection mechanism, the maximization of *MaxXYDist* -or *L coefficient*- and the minimization of the number of muscles as objectives. These trials highlighted that the muscles with the default controller configuration do not have enough energy to overcome the wall;
2) Evolving the controller using the fit tournament as selection mechanism and the *MaxXYDist* -or the *L coefficient*- as fitness metric. In these cases, some robots succeeded in overcoming the wall; however, they were composed only of muscles and they were exhibiting very strange and unnatural behaviours;
3) Evolving the controller using Pareto selection as selection function and the maximization of *MaxXYDist*, the maximization of *L coefficient* and the minimization of the age of the solutions as objectives. In this way, both the robots covering major distances, the ones moving faster towards the wall and the younger ones are preserved. Some constraints on the robots composition have been introduced so as to avoid "*muscles-only*" robots: a valid soft robot has at least 10% non-muscles voxels and at least one bone.

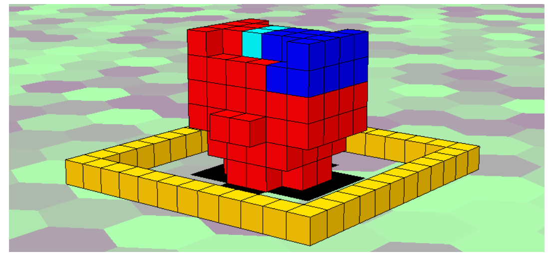

Fig. 6. Soft robot able to overcome the obstacle.

The last attempt has led to the evolution of a soft robot able to overcome the obstacle in 80 generations: that soft robot is the one shown in Figure 6. It exploits the bone part as a pivot to jump over the wall and successfully overcome it. In fact, as the first thing the robot rotates and the bones become its base.

IV. Conclusion

In this work, the functionalities provided by Evosoro[2] [2] have been extended through: the inclusion of a method for evolving the materials of a soft robot while evolving also its morphology; a selection method based no more on an objective function, but on the "*novelty*" of the trajectory exhibited by soft robots; the evolution of the signal controlling the robot's movements together with the robot's morphology; the introduction of an environment with simple obstacles to evolve complex robot behaviours and morphologies.

The experiments done show that, even though the dimensionality of the search space increases, the algorithm still succeeds in evolving soft robots with complex structures that learn to move in a little number of generations.

Contributions

Stefano Leonardi has worked on novelty-based selection, *Luca Zanella* has worked on materials evolution and contributed to novelty-based selection, *Enrico Zardini* has worked on controller evolution and evolution versus obstacles.

References

[1] Jonathan Hiller and Hod Lipson. Dynamic simulation of soft multimaterial 3d-printed objects. *Soft Robotics*, 1(1):88–101, 2014.

[2] Nick Cheney, Robert MacCurdy, Jeff Clune, and Hod Lipson. Unshackling evolution: Evolving soft robots with multiple materials and a powerful generative encoding. In *Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation*, GECCO '13, page 167–174, New York, NY, USA, 2013. Association for Computing Machinery.

[3] Tetsuya Nakamura, Keishi Taki, Hiroki Nomiya, Kazuhiro Seki, and Kuniaki Uehara. A shape-based similarity measure for time series data with ensemble learning. *Pattern Analysis and Applications*, 16(4):535–548, 2013.

[2] The Python library based on the Voxelyze physics engine which provides an interface for the simulation and design of soft robots.