On the challenges of jointly optimising robot morphology and control using a hierarchical optimisation scheme

Léni K. Le Goff and Emma Hart
{l.legoff2,e.hart}@napier.ac.uk
Edinburgh Napier University
Edinburgh, Scotland,UK

ABSTRACT

We investigate a hierarchical scheme for the joint optimisation of robot bodies and controllers in a complex morphological space. An evolutionary algorithm optimises body-plans while a separate learning algorithm is applied to each body generated to learn a controller. We investigate the interaction of these processes using a weak and then strong learning method. Results show that the weak learner leads to more body-plan diversity but that both learners cause premature convergence of body-plans to local optima. We conclude with suggestions as the framework might be adapted to address these issues in future.

KEYWORDS

learning, evolution, morphology, optimisation

ACM Reference Format:
Léni K. Le Goff and Emma Hart. 2021. On the challenges of jointly optimising robot morphology and control using a hierarchical optimisation scheme. In *2021 Genetic and Evolutionary Computation Conference Companion (GECCO '21 Companion), July 10–14, 2021, Lille, France.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3449726.3463156

1 INTRODUCTION

Automating the process of robot design is a long term goal of the field of evolutionary robotics. Many studies now focus on the joint optimisation of morphology and control, usually in simulation. Approaches that operate in a morphological search-space defined by pre-defined modules are particularly common [6, 7, 12, 14] while increasingly the idea is being applied in soft-robotics, using voxel based simulators [7, 8]. Recent work by Buchanan *et al* [1] extends the approach to a very complex morphological space that mixes some pre-defined modules (sensors, wheels, joints) with a 3d printed skeleton that can take any shape or form. Unlike the vast majority of previous work relating to joint body-control optimisation, the robots contain a variety of sensors, which introduces additional challenges regarding automate design of controllers: most previous approaches optimise controllers to produce actuation (e.g. through the use of central pattern generators) but do not attempt to link externally sensed information with actuation.

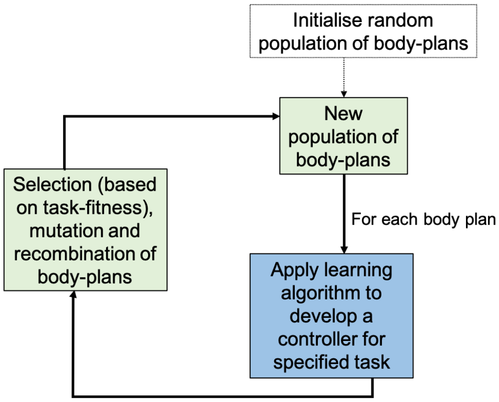
Initialise random population of body-plans
New population of body-plans
For each body plan
Apply learning algorithm to develop a controller for specified task
Selection (based on task-fitness), mutation and recombination of body-plans

Figure 1: Interaction between an evolutionary loop to evolve body-plans and learning-loop to learn controllers

While some authors have attempted to *simultaneously* optimise morphology and control, Cheney *et al* [3] show (in the soft-robotics domain) that this can lead to convergence of morphology before convergence of control, and a failure to adequately explore the space of morphology parameters. More strongly, they suggest that simultaneously changing morphology and control parameters can even be counter-productive, as a controller is specialised for some particular morphology [3]. Liao *et al* [10] propose that using a hierarchical approach is preferable, in which given a batch of morphologies, each robot independently learns a controller, and then the performance of these learned prototypes influences selection of the next generation of morphologies.

The general idea is illustrated in figure 1 and consists of two loops. The outer loop uses an evolutionary algorithm to evolve morphology, while the inner loop applies a learning algorithm to each new morphology to optimise a controller. The learning algorithm can itself be an evolutionary algorithm but alternative algorithms have also been used, e.g. Reinforcement Learning (RL) in [6], Bayesian Optimisation (BO) in [10]. The separation of the two loops brings an additional advantage compared to simultaneous optimisation when considering the co-design of robots that also need to be *fabricated*: while it is extremely costly (in terms of time and material) to generate a physical body, learning trials are relatively computationally cheap, hence it makes sense to be able to allocate more budget to learning controllers than bodies [10].

In this paper we investigate the interaction between morphology evolution and controller learning in the rich morphological space defined in [1] that includes a variety of sensors and actuators using the hierarchical method shown in figure 1. We evaluate two forms of

GECCO '21 Companion, July 10–14, 2021, Lille, France

ACM ISBN 978-1-4503-8351-6/21/07…$15.00
https://doi.org/10.1145/3449726.3463156

learning, used in conjunction with a single evolutionary approach to optimise morphology. Specifically we compare a 'strong' form of learning based on an evolution strategy (ES) with a weaker form that uniformly samples controllers. The ES used is a variant of *NIPES* [9] which combines novelty search with a dynamically increasing population size and has been shown to be both sample and time efficient. We use the most recent version [5] in which the learning is bootstrapped wherever possible by use of an external archive that stores the best controller found for a given '*type*' of robot. *Type* is defined by a tuple (sensors, wheels, joints)[1].Using two environments and a photo-taxis task, we investigate:

- The effect of the choice of learner on overall task performance
- The influence of the learning method on the diversity of morphologies explored by the evolutionary algorithm

The results raise some interesting issues regarding diversity and convergence that will need to be addressed in the future to make further progress towards the end-goal of autonomous robot evolution.

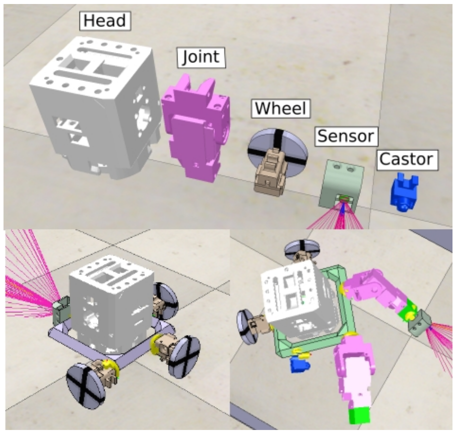

Figure 2: Top picture represents the head, joint, wheel, sensor, and castor organs; bottom left picture the typical successful robot; and bottom right a more complex robot but unsuccessful.

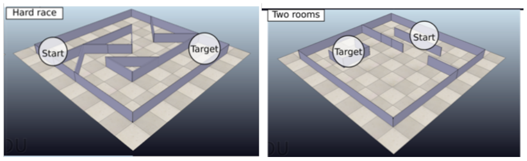

Figure 3: The environments on which the experiments are conducted. A beacon detectable by IR sensors is placed on the target position.

2 RELATED WORK

Cheney *et al* [3] first highlighted the potential issues arising through simultaneous evolution of morphology and control. Several authors have attempted to address these using variations on the hierarchical scheme given in figure 1, using a diverse range of learning mechanisms in diverse morphological spaces.

For example, in [6], an evolutionary algorithm is used to evolve morphologies in a design space of consisting of spheres and limbs, using RL as the learner. Although the robots do not have "sensors", the RL algorithm makes use of a large amount of proprioceptive and exteroceptive state information. They find that the system selects for morphologies that are capable of learning faster as the algorithm runs.

A potential drawback of the hierarchical approach is that decoupling the morphology and controller optimisers can prevent information sharing between them. In [10] Bayesian Optimisation approach is used for both morphology and control loops to design hexapod micro-robots, selected for its efficiency as all optimisation takes place in reality. Here, the controller optimisation process exploits knowledge collected from optimising previous morphologies, providing an information-sharing method that improves data-efficiency by removing the need to start from scratch.

The knowledge-transfer issue was also addressed in our previous work [5] in which we proposed the use of an external archive to store previously discovered controllers. The archive contains discrete cells, each corresponding a specific *type* of robot as described in section 1. One controller is stored corresponding to the highest-performing controller found for the type, providing a starting point for controller learning where a type has been previously discovered. Morphology optimisation is performed using NEAT [13] (optimising a CPPN) and the learner with NIPES [9].

Other variants also exist. In [12], simultaneous evolution is augmented with learning: an evolutionary loop evolves a genome that specifies both morphology and controller in a modular system, but each offspring is further improved by separately applying a learning algorithm (CMA-ES) to refine the controller. [14] introduce a Graph Heuristic Search algorithm with the goal of designing fabricable robots: this interleaves a *design* phase (sampling morphologies), an *evaluation* phase (optimising the controller of a single selected robot), and a so-called *learning* phase which learns a heuristic function used to guide the morphology search phase.

3 METHOD AND EXPERIMENTS

We conduct experiments using the following setups of the hierarchical optimiser:

- The **design-space** consists of two types of components: skeleton and organs (head, joint, wheel, sensor and castors) shown in figure 2.
- **Morphology** is optimised using an evolutionary algorithm described in [1] that uses a generative encoding (NEAT-CPPN) to produce the robot's body-plan (first creating the skeleton then positioning organs).
- The **controller** for each body is a modified version of an Elman network(ElNet)[4] (a recurrent neural network). Network weights are optimised via one of two learning algorithms: (1) greedy selection from a set of random controllers generated by Latin Hyper-cube Sampling (LHS)[11] and (2) the ES algorithm NIPES used in conjunction with an external archive and described in [5].

[1] note however that two robots of the same type may in fact have very different skeletons and layouts of sensors/actuators

We compare two distinct schemes: morpho-evolution (ME) plus learning using Latin Hyper-cube Sampling (MEL-LHS) and morpho-evolution plus learning using NIPES (MEL-NIPES). Both algorithms utilise a population of 20 body-plans.

MEL-LHS. For each body-plan generated via the ME algorithm, 100 sets of ElNet's parameters are sampled using LHS. The fitness attributed to the body-plan within the ME is the best task-performance score obtained among the 100 ElNets. The algorithm runs for 40 generations. Thus, one run uses 80000 evaluations (20 body-plans x 40 generations x 100 controllers).

MEL-NIPES. For each body-plan the network parameters are optimised using NIPES. NIPES starts either from scratch or from a controller stored in an *archive* [5]. This is a three dimensional grid in which each cell correspond to a type of robot as defined in section 1. It is filled over the course of a run with the best controller found per type and enables learning for a previously discovered type to be bootstrapped. NIPES has a maximum budget of 200 evaluations with two early stopping criteria: (1) task-performance reaches a threshold of 0.95 and (2) the robot does not move during the 50 first evaluations. MEL-NIPES runs for 20 generations[2]. The maximum number of evaluations allowed is 80000 (like MEL-LHS) but it typically uses fewer. MEL-LHS is given more generations as the only optimisation process is the ME.

Experiments are conducted on two environments described previously in [1, 5] dubbed Hard Race and Two Rooms on a photo-taxis task to locate a beacon (figure 3)[3]. The fitness assigned to each body-plan is calculated as the normalised distance between the final position of the robot and the position of the beacon (subtracted from 1 to create a maximisation problem).

3.1 Results

Figure 4 compares the effect of the the two learning methods on performance. It is clear that in both environments, MEL-NIPES delivers higher task-performance and lower variance over repeated runs. Moreover, in both environments MEL-LHS fails to find successful solutions ($\geq$ 0.95), despite locating a high-fitness initial solution in both environments. However, NIPES learns rapidly, typically overtaking MEL-LHS within 10000 evaluations.

We next consider whether changing the learner has an impact on the extent of the morphological space explored. Table 1 summarises the number of different *types* discovered by each method and the *overlap* in types: overlap measures similarity between the distributions of type of all robots generated over a run. For both learners, the number of types continues to increase throughout the optimisation process MEL-LHS delivers more types than MEL-NIPES in both environments, and at both 20 and 40 generations. This suggests that the stronger learner decreases the diversity of body-plans discovered. However, there is considerable *overlap* between the two set of robots discovered (82.5% and 93% in the hard-race and two-rooms respectively). This suggests that MEL-LHS explores more types but that those types do not result in high-performing robots. Nevertheless, the stronger learner seems to decrease the exploration power of the algorithm. Figure 6 summarises the type of the best robot found in each of the 10 runs for each experiment. Note that all experiments converge to one of 13 distinct types. NIPES tends to converge to fewer types overall. For a given environment, there is considerable 'agreement' on types between the two algorithms. It is interesting that in the hard-race, half of the MEL-NIPES runs converge to the same single type, but none of the MEL-LHS runs find this, suggesting the 2-wheeled robot has a large basin of attraction for MEL-NIPES. Examples of some of the evolved robots are shown in figure 2.

Figure 5 characterises the best robots found per generation in terms of the three type attributes (wheels, joints and sensors). As previously demonstrated by Cheney *et al.* in the voxel-based soft-robotics domain [3], the morphology also converges very quickly to an approximate type. With MEL-NIPES, in the hard-race, evolution quickly converges on a type with 3 wheels in the first couple of generations; joints are eliminated after around 7 generations; in many runs, the best morphology found per generation contains a single sensor. In contrast, in the two-rooms setup, sensors are

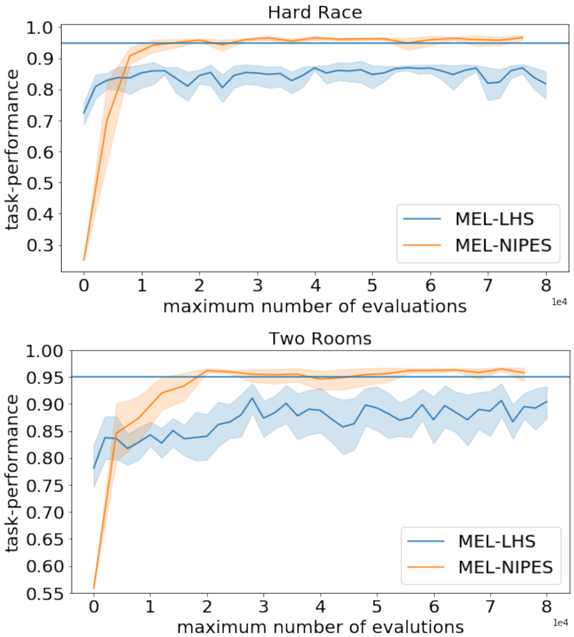

Figure 4: Best fitness over the number of evaluations. For MEL-NIPES, the number of evaluations is the theoretical maximum.

Table 1: The table shows the mean(std) of (1) number of *types* discovered per learning and (2) Overlap between the types of robots generated by MEL-LHS and MEL-NIPES

	Types			**Overlap**
	MEL-LHS		MEL-NIPES	
Gens	20	40	20	
HardRace	68.7 (11.5)	82.5 (17)	50.4 (9.7)	93 (3.4) %
2Rooms	63 (5.2)	73.3 (8.4)	50.2 (13.2)	82.5 (14.4) %

[2] previously shown to be optimal for MEL-NIPES [5]
[3] The source code of these experiments is available at this address: https://bitbucket.org/autonomousroboticsevolution/evorl_gecco_2021

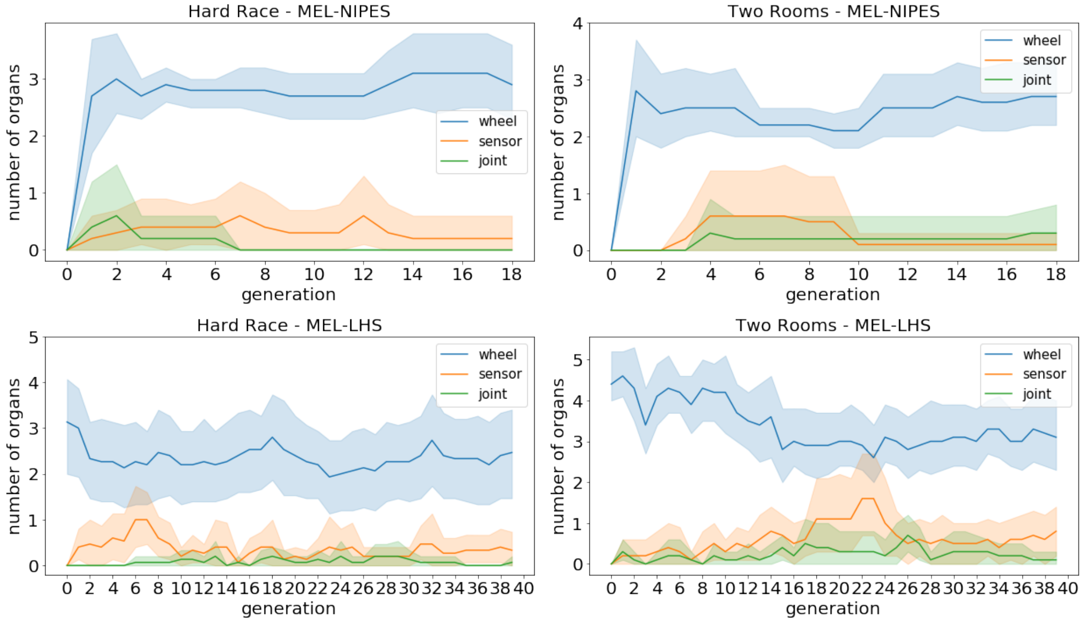

Figure 5: The number of wheels, joints, and sensors of the *best* body-plan over the generations.

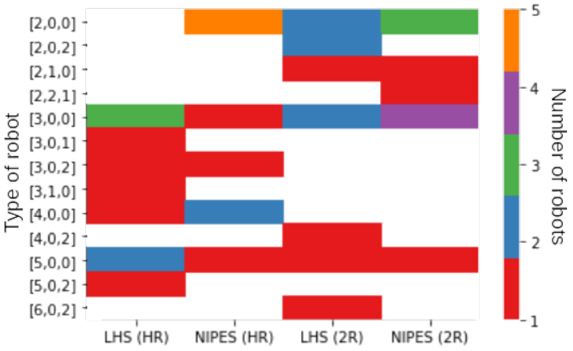

Figure 6: The types of the best robot found in each of the 10 runs in each environment (hard-race HR and two-rooms 2R). y-axis shows [wheels, joints, sensors]

eliminated around generation 10, while some runs maintain robots with one joint throughout a run.

With respect to MEL-LHS, in the hard-race, the overall pattern is similar to MEL-NIPES although with greater instability in the evolutionary dynamics. Note that the best in generation rarely includes a joint, likely explained by the fact that this is harder to find a good controller. In the two-rooms setup, in contrast to MEL-NIPES, MEL-LHS appears capable of maintaining morphologies that include a sensor, while tending to eliminate joints. However, given that this is a photo-taxis task, the low number of sensors suggests that both algorithms converge to sub-optimal body-plans. It also appears that task fitness converges around the same time as the morphologies converge, suggesting that the learners are unable to improve the converged morphologies further.

4 DISCUSSION AND CONCLUSION

We investigated two version of a hierarchical scheme for jointly optimising body-plans and controllers of robots. As in the work of Cheney et al. [3], the two algorithms studied in this article become trapped in a local optima w.r.t morphology. The hierarchical scheme used does not overcome this issue: this is in contrast to the findings of [10] although here this is almost certainly due to the vastly more complex morphological space used. MEL-LHS generates more diversity, but still results in sub-optimal body plans. We hypothesise that although the choice of learner clearly plays a role here, it is not the main issue: improvements to the morpho-evolution component of the algorithm are required as suggested in [3, 6].

The experiments described require evaluation of up to 200 body-plans and 20000 evaluations to converge. While these are reasonable amounts in simulation, if robots need to be fabricated and evaluation time cannot be compressed, these numbers are impractical. Methods to address the premature convergence of body-plans will only increase the number of body-plans requiring testing. Moreover, this study only tackled relatively easy tasks and environments in which wheeled solutions could be found quickly. However, we believe the hierarchical method is worthy of further investigation. Flexibility in allocating budget to either of the two components could allow it to be customised to a particular context, e.g accounting for the relative costs of body-plan fabrication vs learning trials. Equally, the choice of optimiser for each component can be adapted to the context: for example, Bayesian Optimisation is data-efficient[2]; RL is a good choice handling very high dimensional spaces; ES provides a reasonable trade-off between exploration and exploitation.

ACKNOWLEDGMENTS

This work is funded by EPSRC EP/R035733/1

REFERENCES

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

[2] Konstantinos Chatzilygeroudis, Vassilis Vassiliades, Freek Stulp, Sylvain Calinon, and Jean-Baptiste Mouret. 2019. A survey on policy search algorithms for learning robot controllers in a handful of trials. *IEEE Transactions on Robotics* 36, 2 (2019), 328–347.

[3] Nick Cheney, Josh Bongard, Vytas SunSpiral, and Hod Lipson. 2016. On the Difficulty of Co-Optimizing Morphology and Control in Evolved Virtual Creatures. In *Artificial Life Conference 2016*. 226.

[4] Jeffrey L Elman. 1990. Finding structure in time. *Cognitive science* 14, 2 (1990), 179–211.

[5] Léni K. Le Goff, Edgar Buchanan, Emma Hart, Agoston E. Eiben, Wei Li, Matteo De Carlo, Alan F. Winfield, Matthew F. Hale, Robert Woolley, Mike Angus, Jon Timmis, and Andy M. Tyrrell. 2021. Morpho-evolution with learning using a controller archive as an inheritance mechanism. arXiv:2104.04269 [cs.RO]

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

[7] Gregory S Hornby, Hod Lipson, and Jordan B Pollack. 2003. Generative representations for the automated design of modular physical robots. *IEEE transactions on Robotics and Automation* 19, 4 (2003), 703–719.

[8] Sam Kriegman, Douglas Blackiston, Michael Levin, and Josh Bongard. 2020. A scalable pipeline for designing reconfigurable organisms. *Proceedings of the National Academy of Sciences* 117, 4 (2020), 1853–1859. https://doi.org/10.1073/pnas.1910837117 arXiv:https://www.pnas.org/content/117/4/1853.full.pdf

[9] Léni K Le Goff, Edgar Buchanan, Emma Hart, Agoston E Eiben, Wei Li, Matteo de Carlo, Matthew F Hale, Mike Angus, Robert Woolley, Jon Timmis, et al. 2020. Sample and time efficient policy learning with cma-es and bayesian optimisation. In *Artificial Life Conference Proceedings*. MIT Press, 432–440.

[10] Thomas Liao, Grant Wang, Brian Yang, Rene Lee, Kristofer Pister, Sergey Levine, and Roberto Calandra. 2019. Data-efficient learning of morphology and controller for a microrobot. In *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2488–2494.

[11] MD McKay, RJ Beckman, and WJ Conover. 1979. Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code. *Technometrics* 21, 2 (1979), 239–245.

[12] Karine Miras, Matteo De Carlo, Sayfeddine Akhatou, and AE Eiben. 2020. Evolving-Controllers Versus Learning-Controllers for Morphologically Evolvable Robots. In *International Conference on the Applications of Evolutionary Computation (Part of EvoStar)*. Springer, 86–99.

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

[14] Allan Zhao, Jie Xu, Mina Konaković-Luković, Josephine Hughes, Andrew Spielberg, Daniela Rus, and Wojciech Matusik. 2020. RoboGrammar: Graph Grammar for Terrain-Optimized Robot Design. *ACM Trans. Graph.* 39, 6, Article 188 (Nov. 2020), 16 pages. https://doi.org/10.1145/3414685.3417831