Comparing Encodings for Performance and Phenotypic Exploration in Evolving Modular Robots

Frank Veenstra, Emma Hart
Edinburgh Napier University
f.veenstra@napier.ac.uk

Edgar Buchanan, Wei Li
University of York

Matteo De Carlo, Agoston E. Eiben
Vrije Universiteit Amsterdam

ABSTRACT

To investigate how encodings influence evolving the morphology and control of modular robots, we compared three encodings: a direct encoding and two generative encodings—a compositional pattern producing network (CPPN) and a Lindenmayer System (L-System). The evolutionary progression and final performance of the direct encoding and the L-System was significantly better than the CPPN. The generative encodings converge quicker than the direct encoding in terms of morphological and controller diversity.

CCS CONCEPTS

• **Computing methodologies** → **Search methodologies**; **Evolutionary robotics**;

KEYWORDS

Evolutionary Robotics, Encodings

ACM Reference Format:
Frank Veenstra, Emma Hart, Edgar Buchanan, Wei Li, and Matteo De Carlo, Agoston E. Eiben. 2019. Comparing Encodings for Performance and Phenotypic Exploration in Evolving Modular Robots. In *Genetic and Evolutionary Computation Conference Companion (GECCO '19 Companion), July 13–17, 2019, Prague, Czech Republic.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3319619.3322054

1 INTRODUCTION

In evolutionary robotics, the encoding used to map the genotype to phenotype greatly influences how the search space is traversed. Being able to implement an efficient encoding can thereby speed up the evolutionary search process. Direct encodings have been implemented in modular robotics and are representations that directly map the genotype to the phenotype [1, 6]. Generative encodings indirectly map the genotype to the phenotye as has been done through using Lindenmayer-Systems (L-Systems) [4, 7, 10]. Another generative encoding utilized Compositional Pattern Producing Networks (CPPNs) to construct virtual robots [2]. To investigate which encoding strategy performs best for evolving locomotion in simulated modular robots, we implemented a direct encoding, an L-System and a CPPN to construct both the morphology and controllers of our modular robots.

GECCO '19 Companion, July 13–17, 2019, Prague, Czech Republic

ACM ISBN 978-1-4503-6748-6/19/07…$15.00
https://doi.org/10.1145/3319619.3322054

2 METHODS

Virtual robot experimentation platform (V-REP) [3] was used for our experiments [1]. The modular robots were composed of two types of modules: a cube module and a servo module (described in [10]). A cube module was initially created to which additional modules could be attached. The encoding strategies determined how the male connection sites of the modules would be connected to the available female connection sites of the robot. A simple sinusoidal wave function was attached to every module to control the robot in a decentralized manner. Due to the great number of possible configurations of a robot, we limited our modular robots to a tree depth of 5 modules, and a maximum number of 20 modules.

The direct encoding contained all the information about every single module and could attach additional modules to available female connection sites. The mutation operators of the direct encoding were: (1) add a module, (2) remove a module (and connected modules), (3) change the orientation of an attached module, and (4) mutate control parameters of a module. The L-System's axiom symbol [5] represented the cube module. Four other symbols defined servo modules. The mutable parameters of the L-System were the production rules for each symbol (orientation and connections site of modules; see [10]). The mutation operators were: (1) resize the list of production rules of a symbol, (2) change the production rule

[1]The source code, preliminary results, and some videos can be found here: https://github.com/FrankVeenstra/EvolvingModularRobots_GECCO_2019

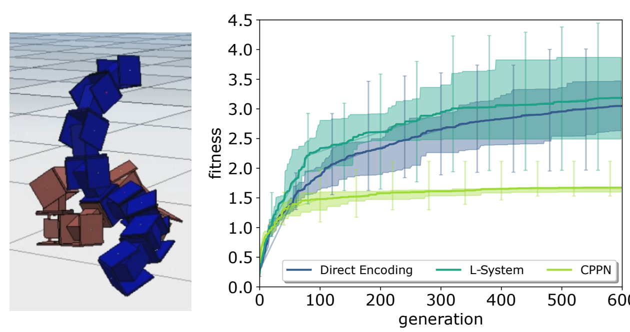

Figure 1: (left) Example of an evolved robot (L-System). (right) Performance difference between the direct encoding, L-System and CPPN using optimal mutation rates. The solid line represents the average maximum fitness achieved between each of the ten runs across generations. The area for each run represents the 25^{th} and 75^{th} percentiles. The error bars represent the 0^{th} and 100^{th} percentiles.

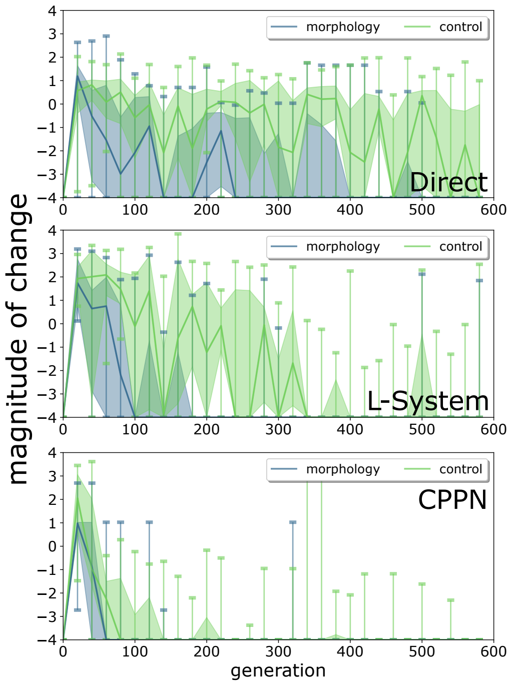

Figure 2: Magnitude of phenotypic change. The graphs show the average change in the morphology (blue) and control (green) lines. The area represents the 25^{th} to 75^{th} percentiles and the error bars the 0^{th} to 100^{th} percentiles.

of a specific symbol, (3) swap two production rules, and (4) mutate the controller corresponding to a symbol.

The CPPN was recursively queried for each available connection site, and had an operator that expands the robot morphology similar to the L-System. The network of the CPPN is modeled as a recurrent neural network with the activation functions of the neurons used in [8]. The mutation operators used by the CPPN were: (1) mutate connections (all layers), (2) add neurons (recurrent layer), (3) remove neurons (recurrent layer), and (4) mutate activation parameters and activation functions of the recurrent layers.

A steady state evolutionary algorithm [9] was implemented for which we only optimized the mutation rate for each different encoding. The offspring were selected with a random selection operator. The parent and offspring populations were combined and 50% of the worst individuals were removed to form the population of the next generation. The fitness metric used for the robots was distance moved within 20 seconds. For the direct encoding and the CPPN, a mutation rate of 12% was chosen, while a mutation rate of 32% was ideal for the L-System. We repeated 10 evolutionary runs for 54,000 evaluations and a population size of 90 (600 generations).

3 RESULTS

Figure 1 depicts an example of an evolved robot and the difference between performance of each encoding. The direct encoding and the L-System performed similarly while the CPPN converged prematurely, underperforming with respect to the other encodings. Statistical testing showed no significant difference between the L-System and direct encoding (Mann-Whitney U test, p-value of 0.37) in the final generation; the difference between the direct/L-System encoding and the CPPN was significant in both cases (p-value of 0.00012 and 0.00016 respectively). In addition, the magnitude of change from one generation to the next in both the controllers and morphologies of evolved robots is plotted in Figure 2. The morphological change was measured by comparing the morphological tree structures of elites between generations. The controller change was based on the difference of the control parameters set for each module. For all encodings, morphological change of the elites decreased quicker than change of the controller. Qualitatively, most phenotypic change occurred in the direct encoding, least change in the CPPN, and a change in between for the L-System.

4 CONCLUSION

Different encodings alter the traversal of the search space when evolving modular robots. In our experiments the CPPN performed significantly lower compared to the other encodings. Moreover, in all three encodings, morphologies tend to fixate faster than the controllers. The direct encoding promoted both types of adaptation for longer; this is followed by the L-System, with the CPPN faring worst. Ultimately, more experiments are required to say more about the difference in performance and phenotypic diversity between encodings. This will be useful when evolving robot morphologies and controllers where a vast search space may be highly convoluted and difficult to map.

ACKNOWLEDGEMENTS

This project has received funding from "Autonomous Robot Evolution: Cradle To Grave", EPSRC reference: EP/R035733/1.

REFERENCES

[1] 2013. EDHMoR: Evolutionary designer of heterogeneous modular robots. *Engineering Applications of Artificial Intelligence* 26, 10 (2013), 2408 – 2423.
[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 15th Annual Conference on Genetic and Evolutionary Computation (GECCO '13)*. 167–174.
[3] M. Freese E. Rohmer, S. P. N. Singh. 2013. V-REP: a Versatile and Scalable Robot Simulation Framework. In *Proc. of The International Conference on Intelligent Robots and Systems (IROS)*.
[4] Gregory S. Hornby, Hod Lipson, and Jordan B. Pollack. 2003. Generative representations for the automated design of modular physical robots. *IEEE Trans. Robotics and Automation* 19, 4 (2003), 703–719. https://doi.org/10.1109/TRA.2003.814502
[5] Aristid Lindenmayer. 1968. Mathematical models for cellular interactions in development I. Filaments with one-sided inputs. *Journal of Theoretical Biology* 18, 3 (1968), 280 – 299. https://doi.org/10.1016/0022-5193(68)90079-9
[6] Daniel Marbach and Auke Jan Ijspeert. 2003. Co-evolution of Configuration and Control for Homogenous Modular Robots.
[7] Karine Miras, Evert Haasdijk, Kyrre Glette, and A. E. Eiben. 2018. Search Space Analysis of Evolvable Robot Morphologies. In *Applications of Evolutionary Computation - 21st International Conference, EvoApplications 2018*. Vol. 10784. Springer, 703–718.
[8] Kenneth O. Stanley. 2007. Compositional Pattern Producing Networks: A Novel Abstraction of Development. *Genetic Programming and Evolvable Machines* 8, 2 (June 2007), 131–162. https://doi.org/10.1007/s10710-007-9028-8
[9] G. Syswerda. 1991. A Study of Reproduction in Generational and Steady-State Genetic Algorithms. *Foundations of genetic algorithms* (1991).
[10] Frank Veenstra, Andres Faina, Sebastian Risi, and Kasper Stoy. 2017. Evolution and Morphogenesis of Simulated Modular Robots: A Comparison Between a Direct and Generative Encoding. In *Applications of Evolutionary Computation*, Giovanni Squillero and Kevin Sim (Eds.). Springer International Publishing, Cham, 870–885.