Simulating Morphological Evolution in Large Robot Populations

Florian Golemo Marjeta Markovič Jasper Schoormans
Peter de Jonge Maik Couwenberg Evert Haasdijk
{f.golemo | m.markovic | j.schoormans | p.a.de.jonge | m.couwenberg}@student.vu.nl,
e.haasdijk@vu.nl

Dept. of Computer Science, VU University Amsterdam
The Netherlands

ABSTRACT

Computational capacity and memory are limiting factors when simulating large numbers of robots with complex bodies: available physics engines struggle to handle more than a couple of dozens of complex robot bodies. This limits the possibilities of investigating the evolution of robot morphology to small populations with few generations. We present a method to simulate large evolving populations of robots with complex and varying morphologies. By simulating individual robots in parallel, we sacrifice the possibility of interaction between robots (other than to exchange genomes), but gain the opportunity to simulate substantial populations, not so much limited by the capabilities of the simulator itself as by the number of processors at our disposal.

Categories and Subject Descriptors: I.2.8 [Artificial Intelligence]: Problem Solving, Control Methods, and Search — *Heuristic methods*

Keywords: Morphological Evolution; Artificial Life; Evolutionary Robotics

1. INTRODUCTION

Evolutionary robotics that considers the co-evolution of morphology and control often focusses on evolutionary algorithms that optimise body shapes for tasks such as locomotion and pushing objects [6, 4, 1]. In such research, the evolutionary algorithm is employed as an optimiser and proceeds along the lines of the generate-test-select loop that was established in the 1970s.

Casting the co-evolution of morphology and control (or body and mind) as a force for adaptation rather than for optimisation, another, less common, area of research considers populations of robots that live and breed in an environment without specific objectives. The success with which robots tackle the environment determines the rate at which their genes spread: reproductive success is not determined by measuring competence in some task but depends solely on robots behaving so that they survive and spread their genes as the environment allows [8, 9]. Such an ecosystem enables research into the evolution of mind and body in a new way and offers profound opportunities for novel research in artificial life and embodied artificial intelligence with implications for evolutionary robotics and evolutionary biology [5, 3].

Evaluations of individuals in evolutionary systems with a classic control loop can be trivially parallelised to achieve impressive speed gains. Objective free systems such as those we consider here, however, implement evolution through local interactions between individuals: evolution is not centrally orchestrated and individuals are born asynchronously. The decentralised and asynchronous nature of the evolutionary cycle makes parallel evaluation a much less straightforward proposition in these cases.

However, physically realistic simulations of robot populations have to contend with severe limitations of present-day simulators that struggle to simulate more than a couple of dozen individuals of moderate complexity [9]. We expect that the study of the dynamics of evolution, in particular the interplay between the development of controller and morphology benefits greatly from simulations with larger populations, numbering hundreds or even thousands of individuals. Such large numbers, we expect, are required for the evolution of populations of morphologically diverse and complex robots.

This paper describes a method that allows researchers to simulate populations of robots of that magnitude by parallelising the simulation of individual robots. Simply put, we propose to simulate each individual robot in isolation from the rest of the population. A separate mating process analyses the traces of each robot after its simulation has completed to identify *mating events* and initiates the creation and simulation of any offspring. Thus, simulating the exchange of genetic material is separated from simulating the robot behaviour.

Our method circumvents the need to simulate many robots in a single process and so allows the simulation of very large populations, in particular in computer clusters, while maintaining acceptable runtime and memory requirements for the experiments. Of course, there is a price to pay: the robots cannot interact with or even perceive each other in any way other than to exchange genomes. We argue that

GECCO '15, July 11 - 15, 2015, Madrid, Spain

ISBN 978-1-4503-3488-4/15/07...$15.00

DOI: http://dx.doi.org/10.1145/2739482.2768494

even with this limitation, the resulting system enables significant research into evolutionary processes that involve mind and body.

2. METHOD

The method we propose relies on three processes: one to simulate the individuals, one to analyse the log of each individual's actions to identify mating events and one that generates new individuals through recombination and mutation. These processes link up in a cycle.This split into three processes results in each individual being simulated and analysed asynchronously. There may be substantial time between the simulation of individuals born at the same time step depending on when each parents' simulation is analysed. This is particularly the case in computer clusters with job scheduling. This also makes it straightforward to suspend experiments at any time. Restarting the three processes with the database of a suspended run causes the loop simply to pick up where it left off.

Initialisation and termination. The initialisation phase starts the three processes, generates individuals for the initial population and inserts these into the experiment's database so that the simulation process registers attends to them. If no individuals are being simulated, created or analysed, the population has died off and the processes terminates. The system can also implement a time check to stop the simulation after a pre-defined amount of simulated time by ignoring mating events that occur past this time.

Simulation. The simulation process checks the database for any new individuals; it launches the simulator of choice for each individual separately and stores the log of its behaviour for analysis. Typically, individuals are simulated for a set time, but it is possible to set up the simulation so that other factors such as energy consumption influence the length of a robot's life. The simulation logs a robot's behaviour (for instance, a trace of the robot movement through the environment) and stores that for analysis by the mating process.

Mating. The mating process continuously checks if any simulations have been completed and analyses the resulting logs. It implements the rules of the environment that determine when two robots reproduce. It identifies mating events by analysing the robot behaviour in relation to that of logs of other robots. It may enforce constraints, for instance limiting the number of offspring a robot may have, or a time limit between a robot producing two offspring. A mating event that passes these tests is entered into the database for the creation process to pick up.

This process may seem similar to the central control loop of traditional evolutionary algorithms. It is crucial to note, however, that the mating process actually only models purely local interactions between robots. One could see it as an additional part of the simulator that simulates the exchange of genetic material through local communication. Including this aspect in the simulation process itself would necessitate costly synchronisation between instances of the simulator. This would substantially reduce the speed gains and the simplicity of the method. A limitation of separately simulating mating in this way is that mating events cannot influence other aspects of robot behaviour at all as this would introduce synchronisation issues.

Creation. The creation process reads any new mating events from the database. The description of a mating event contains the genetic information of the parent(s). This is then used to generate the genome for a new individual with applicable recombination and mutation operators. The new individual is then stored for the simulation process to pick up, closing the loop.

We base our implementation of morphological evolution on CPPNs that prescribe the arrangement of voxels to form a robot's body as described in [2].. There are four types of voxel: rigid, compliant and two versions of oscillating voxel. The latter ones provide the basis of the robot's movement. The creation process implements NEAT [7] to generate CPPNs that define the make-up of the individuals. The simulation process relies on the *voxelyze* simulator kindly provided by Nick Cheney.The robots move through a toroidal arena without any obstacles. The initial population of robots is placed randomly throughout the arena. Each robot is simulated for a specified, fixed lifetime.

Our implementation of the mating process implements rules for procreation that are based on those in mEDEA, and can be simply phrased as "meeting is mating." When two robots come within some minimum distance, the *mating range*, of each other, this precipitates a mating event. The mating event describes the time and location where the parents met along with the CPPNs of both parents (their genomes). Note that there is no selection on the basis of any performance metric, there is no concept of a fitness function at all: mating occurs as the result of the environmental rules only. This means that, just as in biology, fitness can only be an a posteriori metric that is determined by an individual's reproductive success over its lifetime.

The creation process handles a mating event by applying NEAT's recombination and mutation operators. The resulting individual is then instantiated and simulated in the simulation process. The new individual starts life at the next time step, on the coordinate point midway between the parents.

3. REFERENCES

[1] J. C. Bongard and R. Pfeifer. Evolving complete agents using artificial ontogeny. In *Morpho-functional Machines: The New Species (Designing Embodied Intelligence)*, pages 237–258, 2003.

[2] N. Cheney, R. MacCurdy, J. Clune, and H. 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, pages 167–174, 2013.

[3] M. Hoffmann and R. Pfeifer. The implications of embodiment for behavior and cognition: animal and robotic case studies. *CoRR*, abs/1202.0440, 2012.

[4] H. Lipson and J. Pollack. Automatic design and manufacture of robotic lifeforms. *Nature*, 406(August):974–978, 2000.

[5] J. Long. *Darwin's Devices: What Evolving Robots Can Teach Us About the History of Life and the Future of Technology.* Basic Books, 2012.

[6] K. Sims. Evolving 3{d} morphology and behavior by competition. *Artificial Life*, 1(4):353–372, 1994.

[7] K. O. Stanley. Compositional pattern producing networks: {A} novel abstraction of development. *Genetic Programming and Evolvable Machines*, 8(2):131–162, 6 2007.

[8] J. Ventrella. Genepool: Exploring the interaction between natural selection and sexual selection. In *Artificial Life Models in Software, 1st ed.*, pages 81–96. Springer, 2005.

[9] B. Weel, E. Crosato, J. Heinerman, E. Haasdijk, and A. E. Eiben. A robotic ecosystem with evolvable minds and bodies. In *2014 IEEE International Conference on Evolvable Systems*, pages 165 – 172, 2014.