

Lamarckism Boosts Sample Efficiency in Robot Evolution

(Master Thesis Project Proposal)

Jacopo Di Matteo (13465635)

`jacopo.di.matteo@student.uva.nl`

December 30, 2022

This document is the “Master Thesis Project Proposal” of Jacopo Di Matteo (ID: 13465635) for the approval of a master thesis not part of the marketplace.

1 Thesis Committee

A master thesis at the UvA lasts **eight** months (**42** credits), and the proposed research question or direction should be novel and of high scientific interest. The conditions to do a thesis external to the marketplace are:

A Project Proposal: *This document.*

A Supervisor: Prof. Dr Agoston E. Eiben, `a.e.eiben@vu.nl`.

A Second Assessor: Prof. Dr Mike H. Lees, `m.h.lees@uva.nl`.

A Projected Timeline: from 31-10-22 to 30-06-23.

Alongside *Prof. Dr Agoston E. Eiben*, a **Co-Supervisor** will act as additional help and reference point: Jie Luo, `j2.luo@vu.nl`

2 Introduction

In the following document, a project plan is laid out, with three different stages, and the resulting research questions and hypotheses.

Before the project outline, an introduction to the material is given, this justifies the proposed plan, outlines pivotal design choices and explains key terminology.

3 Background Information

Evolutionary robotics (ER) is a cross-disciplinary field at the intersection between Evolutionary Computing and Robotics. The overarching system methodology in ER is the design of Evolutionary Algorithms (EAs) to automatically generate morphologies (bodies) and controllers (brains) of Robots, which can then be assembled in the real world and made to operate on a given task.

Before further explaining ER, a short introduction to Evolutionary Computing and Evolutionary Algorithms is in order.

3.1 Evolutionary Computing & Evolutionary Algorithms

Evolutionary Computing (EC) is a sub-field of Artificial Intelligence (AI). EC utilises the conceptual framework of modern Evolutionary Theory to design algorithms to solve a wide selection of problems, such as optimisation and constraint satisfaction. EC can be considered the digital counterpart of Evolutionary Biology.

When applied to a given problem, for example, constraint satisfaction, the term “*individual*” is used as a placeholder for “a candidate solution to the problem at hand”. A “*population*” is a group of two or more different individuals.

Typically an Evolutionary Algorithm (EA) consists of the repeated application of four consecutive steps:

1. The parent selection process: where one or more individuals are picked from the population for mating.
2. A recombination operator: when two or more parents are selected, this operator combines them into a new individual.
3. A mutation operator: where small stochastic changes are made to the new individual.
4. A survivor selection process: where, through the use of a fitness function, the next generation of individuals is chosen from the pool of parents and children.

Parent selection and survivor selection are inspired by the Darwinian notion of “survival of the fittest”, these provided the “direction” of evolution. While the recombination and mutation draw inspiration from Modern Synthesis and the concepts of “mutation and recombination of DNA”, these provide “diversity” for evolution.

Selecting the right fitness function, which measures the “quality” of potential solutions, is of primary interest as this often dictates whether or not the evolutionary process will converge to a good solution.

Extending the biological analogy, problem solutions are encoded in “*genotypes*”, while their physical interpretation (the manifestation of the actual solution) is called “*phenotypes*”. Special mappings have to be designed to generate a phenotype given a specific genotype (the analogous process in biology is Ontogenesis, the biological process of transforming DNA into a fully formed organism). Care must be taken in the design of the genotype, and the genotype-to-phenotype mapping, to ensure that the solution(s) to the original problem exists within the phenotype space (the space of all possible phenotypes).

Parent selection and survivor selection are performed on the phenotypes, while recombination and mutation are stochastic operations used to shuffle the parental genotypes. Starting with a population of randomly selected (possible) solutions, a well-designed EA expects that the population’s average fitness improves with the cyclic application of the aforementioned steps, each cycle is called a “*generation*”.

Usually, no guarantee of convergence, or of finding the optimal solution, can be made. An EA concludes when a sufficiently good solution is found, or a set amount of generations have passed.

3.2 Evolutionary Robotics

As stated previously, ER is concerned with the automatic generation of robots to solve a specific task.

In practice this means that an ER system needs to produce the following outputs:

1. A fully functional and constructible robot morphology, i.e., the physical body of the robot.
2. A controller which enables the robot to perform a given task, i.e., the brain that lets the robot behave.

This implies that the designer of an ER system needs to make the following design choices:

1. The evolutionary algorithm for the body.
2. The evolutionary algorithm for the brain.

Both of these require independent choices for their respective genotype, phenotype, fitness functions, choice of recombination and mutation operator.

At this point we can touch on a crucial aspect of ER, **where does evolution take place?**

In a future where the whole evolutionary process can be performed in the real-world, robot evolution could then be carried out *in-vivo*. However, in the present day, such evolutionary processes can only be performed, end-to-end, in simulated environments; the assembling, testing and disassembly of robots is a very expensive procedure, both in terms of time and human effort, making a fully-physical implementation of robot evolution untenable.

A driving force in ER research is bringing the evolution of robots to the physical world, either as simulation/real-world hybrids or in fully-embodied systems. Until this is achieved, all systems that evolve robots must have a simulated environment, this requirement adds a third dimension to the design problem: the selection and design of an appropriate simulated world.

Nevertheless, because robots should be tangible and solve problems in the real world, the final product of this digital evolution is often manufactured physically.

3.2.1 The Ramification of the Joint Evolution Morphologies & Controllers

The system outlined above implies the concurrent evolution of morphologies and controllers. However, the joint evolution of morphologies and controllers gives rise to the well-documented “*mismatch problem*”[?]: if we start with two paternal robots, each with a functioning brain-body pair, their mating can result in an **unviable** child.

This problem arises as a consequence of mutation and recombination which, when applied to a controller, can change the parental genotypes to the point where the resulting controller can no longer adequately utilise the morphology it is born with. This means, at best, the robot will behave strangely, and at worst it will not function at all.

To overcome this issue, one proposed solution is the addition of an “*Infancy Stage*”, where a robot learns to control the body it’s born with by undergoing a learning period. To give a practical example, in the case where the controller of the robot is an Artificial Neural Network (ANN), evolution represents the changing topology of the network, while the learning phase would represent its training.

The inclusion of a learning phase for the robot means that beyond the “evolution of the brain” the designer of the ER system needs to consider the design of the “learning algorithm” to improve the brain. Thus we need to extend once again the set of design choices needed for the design of an ER system, which now includes:

1. Design of an EA for the evolution of the morphologies.
2. Design of an EA for the evolution of the controller architecture.
3. Design of Learning Algorithm for the learning phase of the controller.
4. Design of the simulated environment.

This, of course, implies that the controllers of the robots need to both evolve and learn. These four design elements seamlessly integrate into the “The *Triangle of Life*”, a systematic methodology for the development of ER systems.

3.2.2 Triangle of Life & Embodied Cognition

Embodied cognition is the theory that (human) intelligence is a product of the interaction between the body, the brain and the environment. Its Evolutionary Robotics equivalent is the Triangle of Life, where the creation of intelligent robots is encapsulated in:

1. Morphogenesis stage: where the genotype is used to generate the robot.
2. Infancy stage: where the robot learns to control his body, with the end goal of performing well in a designated task.
3. Operational stage: where the robot performs its designated task and reproduces.

The Triangle of Life is both a theoretical framework and a systematic methodology for the design of ER systems. Beyond being a guiding principle for the design of systems which evolve intelligent robots, the division into separate phases allows for the researchers to focus on either of the three stages, decreasing the degrees of freedom and lightening the conceptual load.

3.2.3 Morphogenesis Stage: Physical & Digital Creation

It’s important to elucidate the manufacturing process of the robots, as well as their digital representation. The rise of rapid prototype manufacturing, spearheaded by the revolution in affordable consumer-level 3D printers, means that robots can be easily and cheaply built using these technologies. This leap in technology lead to unprecedented flexibility in the design of the Morphospace of evolvable robots; the space of all morphologies which can be obtained through evolution (another term borrowed from Biology).

A detailed discussion of this topic falls outside the scope of this proposal, but a very good one can be found here [?]. What’s important for the thesis here proposed, and by extension, this project proposal, is that the robots studied will be modular, and based on the open-source **RoboGen** platform [?]. This can be thought of as a collection of different “robotic organs”, which the EA learns to arrange to create a well-performing robot.

The use of modular robots allows for fast assembly, reusability of parts, simple extensibility to new organs, and a rich variety of possible morphologies. Digital counterparts of these organs can be designed and ported into the simulated worlds, which results in a simple one-to-one mapping to the real world.

A comparison of different robotic systems, including modular robots, and their benefits and drawbacks can be found here [?].

3.2.4 Infancy Stage: The Problem of Sample Efficiency

The learning stage of the triangle of life, while improving the performance of the robots at a given task, and aiding in the prevention of body-controller mismatch, increases the computational time in a multiplicative manner. This means an additional N learning steps increase the overall length of the evolutionary process by N times (each generation takes N times longer). In practice, this translates to simulations that last days rather than hours.

For this reason, improving the efficiency of the learning stage, in other words, reducing the number of N steps needed to achieve acceptable fitness (also called “sample efficiency”), is a fundamental problem in ER. Reducing evolutionary trials can save days from the total simulation time.

An important notion involving the infancy stage is the learning delta: the difference between the ability of a robot to perform a task right after being born and after the learning period is over. This is a very useful tracker of the plasticity of robots, and recent experiments showed [?] that the learning delta increases with passing generations. This indicates that the robots become “more learnable”.

This stage will be the focal point of the proposed thesis, in particular the extension of learning algorithms with Lamarckian-inheritance.

Lamarckian-inheritance Lamarckian inheritance states that a parent organism can pass on to its offspring physical characteristics that it acquired through use or disuse during its lifetime.

When applied to the controllers of robots in ER, this implies that whole or parts of the paternal controllers are spared the effects of recombination and mutation and the controllers are directly passed to the offspring, allowing it to “inherit the experiences” of its parents.

Establishing the technical specifics of Lamarckian inheritance when applied to controllers of robots in ER systems is the objective of this thesis.

3.2.5 Operational Stage: Fitness Function & Simulated World

The design of a good fitness function, a mathematical metric for how well a robot performs the desired task, is crucial and almost entirely dictates the direction of the evolution of the robots.

The creation of fitness functions for complex tasks, or robots that need to perform multiple tasks, is a difficult open problem. However, in the case of this thesis, only a task with a clear metric is considered: “**targeted locomotion**”, the ability of a robot to go from a starting location x to a target location y in the least amount of time.

While, as mentioned previously, simulated environment design is also a critical aspect of ER systems, it falls outside the scope of this thesis. Additionally, the hope of hybrid systems (which integrated real-life feedback) makes this research point less urgent.

The current expectation for this thesis is that a **flat plane** will be used for the majority of performed experiments.

3.3 Relevant Literature

Most of the background theory of EC and EA was drawn from the book [?].

Information on the simulation of robot evolution and progress towards full-embodied can be found in [?].

The Triangle of Life was first outlined in [?], and the need for a learning phase to avoid the mismatch problem was recently reviewed in [?].

An analysis of choosing the underlying Morphospace, and experimental results using various robotic platforms, can be found in [?].

The effects of adding the “Infancy stage” to an ER system have only been recently empirically quantified in [?], whereas an analysis of the performance of various learning algorithms on a test suite of robots can be found in [?].

Previous work on Lamarckian evolutionary computing can be found in [?], while a similar theory applied to ER which uses “controller archives” is explored in [?].

3.4 Closing Remarks on Background Information

As explained above, the addition of a learning phase to the ER system means that the time it takes to produce functioning robots is vastly increased.

This opens the door to plentiful research on the “Infancy stage”, particularly the optimisation, in terms of both speed and fitness, of the learning phase; this is the subject of the proposed master thesis, which will be outlined in the remainder of this project proposal.

4 Project Outline

This master thesis is indented to extend the findings from [?] and [?], by researching the changes in “sample efficiency” when extending the learning stage of an ER system to include Lamarckism.

In general, it can be stated that the primary goal of this thesis is: *“Exploring Lamarckian-inheritance for the learning phase of robot controllers in robot evolution”*. While a secondary goal is: *“Exploring the effects of different learning algorithms on automatically generated “robot-test suites”*.

To achieve this, the master thesis will be composed of three different stages.

4.1 Stage 1

In the first stage of the project, a set of learning algorithms must be chosen to form the basis for later stages.

Some of the different learning algorithms which could be tried are:

- Novelty-driven, increasing population, evolution-strategy (NIPES)
- CMA-ES
- Reinforcement Learning (RL)
- Bayesian Optimisation (BO)
- RevDe (KNN)
- Back-propagation

- CGPDN
- HyperNEAT
- Simulated Annealing

Further research into these must be undertaken before a definitive list can be established.

As outlined above, minimising the number of learning steps is a crucial component for the acceleration of robotic evolution. For this reason, the primary metric by which the algorithms will be evaluated is “best fitness reached” relative to the number of learning steps performed.

Another important metric for consideration is “genetic diversity”, which is a measure of premature convergence. Other metrics could also be implemented and require further research.

At this stage, the learning algorithms will remain decoupled from evolution, and will instead be evaluated on a test suite of pre-made robots. One such test suite of hand-crafted robots already exists [?], and the auxiliary target of this thesis is the creation of an algorithmic method for the generation of robots (without evolution) to test learning algorithms.

4.1.1 First Open Question

1. *How do different learning algorithms compare when applied to robot controllers, particularly on the task of directed locomotion?*

4.2 Stage 2

Following the experimental results from Stage 1, a subset of the tested learning algorithms is chosen for integration into an ER system, mimicking [?].

In this stage the robot evolution will be Darwinian, conducting an experimental study on the working of this system will form the benchmark for Stage 3.

4.3 Stage 3

The chosen learning algorithms are integrated into the ER system in a Lamarckian way. The results of these are then compared with the results from Stage 2.

Below is a non-exhaustive list of possible quantitative metrics for the comparison of Darwinian learning versus Lamarckian learning:

- Population-based:
 - Best fitness
 - Average fitness
 - Genetic diversity
- Individual-based:
 - Speed
 - Accuracy
 - Morphological complexity
 - Network size/complexity

Morphological complexity is measured with a series of different markers such as:

- Number of parts
- Displacement velocity
- Extremity count
- Joint count
- Symmetry

Morphometrics is the field of Evolutionary Biology which deals with the quantitative measure of the physical traits of an organism with the end goal of organising them into clear categories; the above is a form of robotic morphometry.

Further research into related works, and experimentation, are required to establish definitely which of these are best suited to answer the research questions posed in this project proposal.

4.3.1 Second & Third Open Question

2. *Will the morphologies be different between Lamarckian and Darwinian?*
3. *How will the learning delta be affected by the use of Lamarckian controllers?*

4.4 Hypotheses

1. Lamarckian evolution speeds up the system in terms of evolutionary trials.
2. Lamarckian evolution reduces the learning delta over time.

4.5 Tech Stack

The evolutionary experiments will be performed in Revolve 2 with the MuJuCo backend.

Revolve 2 is a platform developed by the Vrije Universiteit Amsterdam (VU) Computational Intelligence (CI) group. Revolve 2 enables the automation of large-scale (simulated) evolutionary robotics experiments. The platform is reliant on a “backend” physics engine¹.

MuJuCo is a free and open-source physics engine for fields that require quick and accurate simulation, such as robotics².

The research I will conduct will primarily involve MuJuCo, to exploit the M1 iMacs available at the Robot Lab.

4.6 The Vrije Universiteit Amsterdam, Computational Intelligence Laboratory (VU CI Lab)

One of the research groups at the Vrije Universiteit Amsterdam (VU) is the Computational Intelligence (CI) Lab. The lab primarily focuses on “evolutionary computing, machine learning and complex systems for optimisation, modelling, robotics and sensory data processing”.

¹<https://github.com/ci-group/revolve2>

²<https://mujoco.org>

Part of their portfolio of research is Evolutionary Robotics (ER), which is the application of Evolutionary Algorithms (EA) to tackle the problem of automated robot generation for task completion; this involves the generation of both “robot bodies” (morphologies) and “robot controllers” (brains), their simultaneous evolution, and finally task performance.

4.6.1 Previous Personal Involvement With the VU CI Lab

In the year 2021, I took part in the Evolutionary Computing course taught by professors from the VU CI Lab. This course offered 2 options for coursework, the first option was the “standard assignment”, which consist of several programming challenges involving evolutionary algorithms. The second option was the “advanced research assignment” which allows (around 15) students to work on small research projects, to tackle research questions in 3 months.

Having chosen the “advanced research assignment” I got to work with the CI lab, in particular, I was supervised by Dr Eiben. Thanks to this project I became very passionate about EAs, ER and the work being done at the CI lab, to the extent of wanting to pursue a PhD in this field. Later in the same academic year, I was a Teaching Assistant for the Computational Intelligence course, a 2nd-year bachelor course taught by the VU, also taught by professors from the CI lab, which further strengthened my connection with the group.

References