

Evolving Genotype Phenotype Mappings as dynamical Systems

Citation for published version (APA):

Paredis, J. (2017). Evolving Genotype Phenotype Mappings as dynamical Systems. In *WIVACE 2017 Book of Abstracts* (pp. 31-40). WIVACE.

Document status and date:

Published: 01/01/2017

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.umlib.nl/taverne-license

Take down policy

If you believe that this document breaches copyright please contact us at:

repository@maastrichtuniversity.nl

providing details and we will investigate your claim.

The Evolution of Genotype Phenotype Mappings as Dynamical Systems

Jan Paredis

Department of Data Science and Knowledge Engineering, Maastricht University,
P.O. Box 616, 6200 MD Maastricht, The Netherlands
j.paredis@maastrichtuniversity.nl

Abstract. This paper investigates the evolution of Genotype Phenotype Mappings (GPMs). Here, the GPMs are represented as dynamical systems. It is investigated to which regions of a parameterized space of GPMs evolution leads. These regions are called Regions Of Maximum Adaptability (ROMAs). These ROMAs are stable but hard to predict.

Keywords: Coevolution, Dynamics, Dynamical Systems, Genotype Phenotype Mappings

1 Introduction

The Human Genome Project [1] successfully determined the human DNA sequence. This project gave rise to two unexpected findings relevant for the current paper. First, the human genome was shorter than expected. Moreover, a large part was neutral: it did not encode anything. This led to the conclusion that the complexity resides in the expression of the DNA, i.e. the genotype phenotype Mapping (GPM).

Wagner et al. [2] put forward the hypothesis that GPMs are under genetic control and that evolutionary algorithms (EAs) can be used to investigate this. One of the advantages of such an approach is that experiments can be done that are impossible in nature (e.g. because of the lack of control over system parameters). The research proposed here is an instantiation of their proposal in a simple artificial coevolutionary context. The aim of the current research is not to develop biologically plausible models. Nature is far too complex for that. Here, a simple model is used to study the dynamics of GPMs. The idea to model the GPMs as dynamical systems was, amongst others, suggested in [3] and [4].

In the current paper, the Coevolutionary Genetic Algorithm (CGA), introduced in [5], is used. This algorithm was inspired by the seminal work of Hillis [6]. In the past, the CGA has mainly been used as a tool for optimisation, see e.g. [7]. Now, the dynamics of the CGA is studied. The coevolutionary interactions in nature are often complex. The goal of this paper consists of the design of a SIMPLE coevolutionary application and GPM which - despite their simplicity - still exhibits realistic, complex dynamics.

The current paper is a sequel to [8], [9], and [10] in which gradually more complex GPMs are evolved. Moreover, [8] introduced the concept *region of maximum adaptability* (ROMA). It is the region in a parameterized space of GPMs evolution leads to. Actually, sections 2 and 3 of the current paper originate from [9]. Just like in the earlier research a pursuer evader (PE) model is used. This model was chosen because it leads to ongoing evolution.

The structure of the paper is as follows. After this introduction the CGA is described. Next, pursuer-evader (PE) systems are discussed. Section four describes GPMs as dynamical systems followed by a section with empirical results of the evolution of GPMs. Section 6 discusses the relevance of the results. Finally, future research is described and conclusions are drawn.

2 A Coevolutionary Genetic Algorithm

Here, the basic CGA is described, as a first step it creates, two populations (called pop1 and pop2). Typically, the individuals in these initial populations are (uniformly) randomly generated. Next, the fitness of these individuals is calculated. This fitness depends on the particular application, but it is the result of a number - here 10 - of ENCOUNTERS of an individual with individuals of the other population. These encounters result in a pay-off which is stored in the history of the individual. The actual fitness is the average of these (10) history elements. Because these encounters represent predator-prey interactions, success for one individual (in an encounter) is failure for the other one. Hence, the value of an encounter is stored in the history of one individual involved in the encounter. The other individual stores the negative of this value in its history. Once all initial fitnesses are calculated, both populations are sorted on fitness: the individual with the highest fitness on top the least fit one at the bottom.

Next, the main *cycle* of a CGA is executed. The pseudo-code of this cycle is given below. First, 20 encounters are executed between SELECTed individuals. This selection is linearly biased towards highly ranked individuals: similar to GENITOR [11] the top individual is 1.5 times more likely to be selected than the median individual. Next, the pay-off of this encounter is calculated and stored in the history, removing the payoff of the least recent encounter from the history. Hence, the history is implemented as a queue. Finally, the fitness (the average of the history) of both individuals involved in the encounter is re-calculated. Possibly, this changes the ranking of the individual in its population. Note that the predator prey interaction results in a negative pay-off for the individual of the second population.

After these 20 encounters the CGA produces one offspring for each population: it SELECTs two parents. A new individual is generated from these parents through the application of MUTATION (probability of mutating a gene is 0.1) and (uniform) CROSSOVER. The fitness is calculated by executing 10 encounters between the new individual and SELECTed members of the other population (again using the negative payoff for individuals which belong to the second population). In case this fitness is higher than the fitness of the bottom individual

```

DO 20 TIMES
ind1 := SELECT(pop1)
ind2 := SELECT(pop2)
payoff := ENCOUNTER(ind1,ind2)
UPDATE-HISTORY-AND-FITNESS(ind1,payoff)
UPDATE-HISTORY-AND-FITNESS(ind2,-payoff)
ENDDO

p1 := SELECT(pop1)          ; pop1 parent1
p2 := SELECT(pop1)          ; pop1 parent2
child := MUTATE-CROSSOVER(p1,p2)
f := FITNESS(child)
INSERT(child,f,pop1)
p1 := SELECT(pop2)          ; pop2 parent1
p2 := SELECT(pop2)          ; pop2 parent2
child := MUTATE-CROSSOVER(p1,p2)
f := FITNESS(child)
INSERT(child,f,pop2)

```

then the new individual is placed in the population at its appropriate rank. All individuals with a lower fitness go one position down and the bottom individual is deleted. This basic cycle is repeated a large number of times (e.g. 20000 cycles). The sampling process to calculate (and update) the fitness is called lifetime fitness evaluation (LTFE). In the current paper, all parameter settings and genetic operators are identical to those described in (Paredis [7]) unless mentioned otherwise.

3 Pursuer-Evader Dynamics

In this particular application, each individual consists of two genes: real numbers in the interval $[0,1]$. The pay-off of an encounter between two individuals consists of the cartesian distance between the two pairs of genes. The first population maximizes the distance to the individuals of the other population. The negative payoff of the members of the second population results in a minimization of the distance to the individuals of the first population. This because in both populations fitness is maximized.

Each individual can be represented as one point on the plane $[0,1] \times [0,1]$. Furthermore, in order to allow for an unbounded evolution, this plane is considered to be a torus. Hence, the distance is the minimum of the two possible distances (one crossing (an) "edge(s)"). Furthermore, mutation can cross the "edges" as easy as it can move in the plane. Or, in other words, 0.95 is equally likely to be mutated into, for example, 0.085 or 0.05. Finally, a standard uniform crossover is used: new offspring receives each gene from one of its parents randomly and independently.

The dynamics of this application is fairly simple. The initial (random) populations are scattered randomly over the plane. In the first experiment described below equal population sizes consisting of 50 individuals are used. Fairly soon (typically in less than thousand cycles) during evolution two clusters appear (one for each population) where one cluster chases (pursuer) the other (evader). Figure 1 provides a snapshot of such a chase. From time to time different behavior is observed. Sometimes the pursuers catch up on the evaders. At this moment the cluster of evaders breaks up. Most of the time the evader cluster breaks up in two or four sub clusters, which are located symmetrically with respect to the pursuers. These sub clusters virtually immobilize the pursuers while the evader sub clusters move radially and finally become one cluster again. Due to sampling errors and finite population sizes the evaders cluster (i.e. unite) again before the sub clusters have gone all the way. Once the evaders are clustered again, the "standard" pursuing of two clusters continues. Obviously, the symmetrical case may occur as well: the pursuers breaking up to immobilize the evaders for a while, as the snapshot in figure 2 depicts.

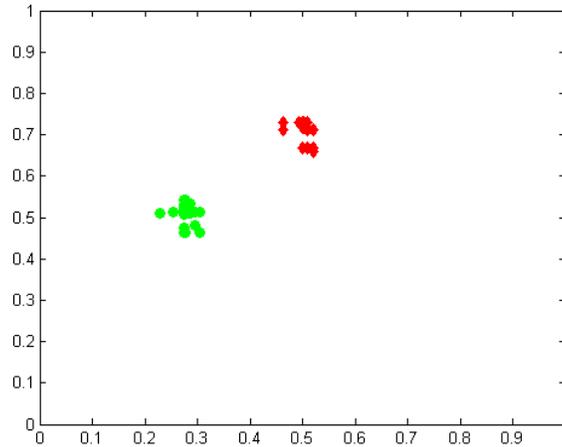


Fig. 1. A cluster of pursuers (black diamonds) pursuing a cluster of evaders (grey circles).

When the two populations have different population sizes then their respective speed changes. This is because at each cycle both populations reproduce once. Hence, the smaller population evolves the fastest, i.e. moves faster on the plane. In case the pursuer population is smaller, the pursuers regularly catch up with the evaders. When this happens the evaders split up, again immobilizing the pursuers until the evaders form one cluster again. Then the chase resumes.

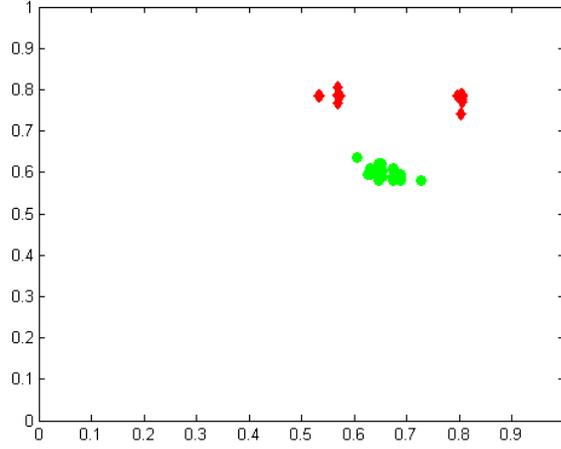


Fig. 2. A cluster of pursuers (black diamonds) splits up in two parts temporarily immobilizing a cluster of evaders (grey circles).

In the other case, the evader population is the smallest population. Here, the evader population successfully keeps ahead of the pursuer population. Occasionally, the evaders even have to slow down in order not to get too close to the pursuers (remember: the world consists of a torus).

4 GPMs as Dynamical Systems

As a starting point, a simple dynamical system with a 2 dimensional phase space is used here to represent a GPM. It contains point attractors. The number of attractors and the size of their basins is under genetic control. The equations below - taken from [12] - describe the dynamics of this system. Figure 3 shows the corresponding phase space.

$$\dot{x} = \sin(2 * \pi * x) * \cos(2 * \pi * y) \quad (1)$$

$$\dot{y} = \sin(2 * \pi * y) * \cos(2 * \pi * x) \quad (2)$$

Now, two additional parameters - r1 and r2 - are added to transform the dynamics. These r's constitute a parameterized space of GPMs. Both parameters are in the interval [0,1] and are under genetic control: i.e. each individual - evader or pursuer - has these parameters in addition to the x and y positions on the torus as their genetic representation. The equations including these rs are given below. The constant 200 provides an upper bound to the number of attractors. The periodicity of the goniometric functions used allows for multiple attractors.

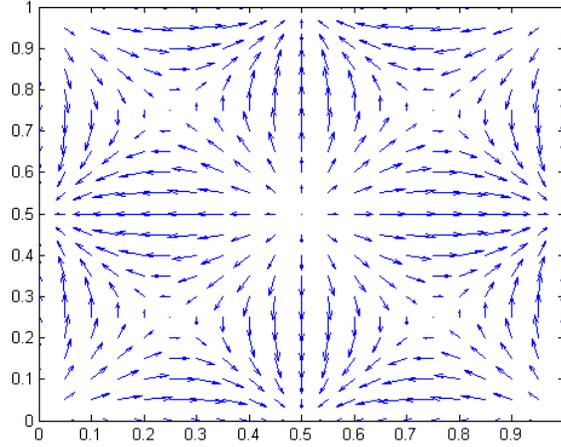


Fig. 3. Phase space diagram for equations (1) and (2).

It is important to note that each time a new individual is created its GPM (using its r s) is applied (100 times with a time step of 0.1) starting from its location on the torus, i.e. its x and y . The resulting phenotype is then used for fitness calculation.

$$\dot{x} = \sin(200 * \pi * r1 * x) * \cos(200 * \pi * r2 * y) \quad (3)$$

$$\dot{y} = \sin(200 * \pi * r2 * y) * \cos(200 * \pi * r1 * x) \quad (4)$$

A second dynamical system, described by the equations 5 and 6, introduces two extra r s - $r3$ and $r4$ - in order to allow for assymetries in the x and y direction of the phase space.

$$\dot{x} = \sin(200 * \pi * r1 * x) * \cos(200 * \pi * r2 * y) \quad (5)$$

$$\dot{y} = \sin(200 * \pi * r4 * y) * \cos(200 * \pi * r3 * x) \quad (6)$$

5 Empirical Results

Now, the algorithm is run 100 times with a population of 50 evaders and 20 pursuers. The notation 50-20 is used for this setup. Earlier research [9] showed that pressure towards the ROMAs is stronger for larger populations. This because the reproduction rate is the same for both populations. Hence the selection pressure on the evaders is largest. For this reason, the distribution of the r 's of the evaders will be depicted here, because they are the most outspoken. Figure 4 shows the distribution of the r 's of the evader population at the end of each of the 100

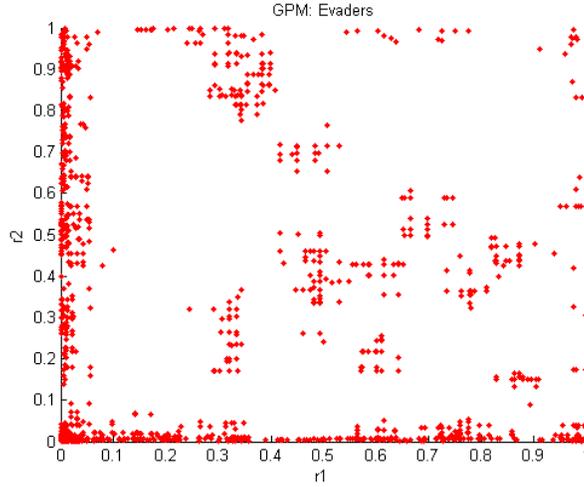


Fig. 4. Distribution of r 's of 100 runs of 50-20 using equations 3 and 4.

runs. Clearly, the r 's are drawn towards both axes. So the region near both axes are the ROMAs. It is important to note that it is difficult to understand - let alone predict - what the ROMAs will be. But, on the other hand, these ROMAs are stable: repeating the experiments results in the same ROMAs.

Each point in figure 4 represents a dynamical system, instantiating equations 3 and 4. Here, two examples of dynamical systems in the ROMA will be shown. Figure 5 represents the system where both r 's are equal to 0.02, i.e. near the origin. It consists of 8 equally spaced point attractors. In the second system, shown in figure 6, r_1 is equal to 0.02 as well, r_2 , on the other hand, is equal to 0.98. Hence, both systems are part of the ROMA. A couple of observations are relevant here. Both systems have different dynamics but their geometrical look is similar: the attractors and saddle points are at the same positions. Their type (attractor or saddle point) might be different. Because of the relatively small number of attractors these GPMs are $n:1$ mappings with a relatively large n . Furthermore, if r_1 and r_2 are unequal and they are switched then a similar phase space is obtained only all vectors are reversed.

When the dynamical system represented by the equations 5 and 6 is used, the ROMA in the r_1 - r_2 plane is the y -axis, i.e. r_1 is small, near zero. In the r_3 - r_4 plane the distribution is most dense near the x -axis. Or, in other words, there is selective pressure towards small r_4 s. One could say that the ROMA of figure 4 is decomposed in two parts.

6 Discussion

The fact that the basic evolutionary processes (variation, selection, and reproduction with inheritance) are used, the general principle discussed here - increased

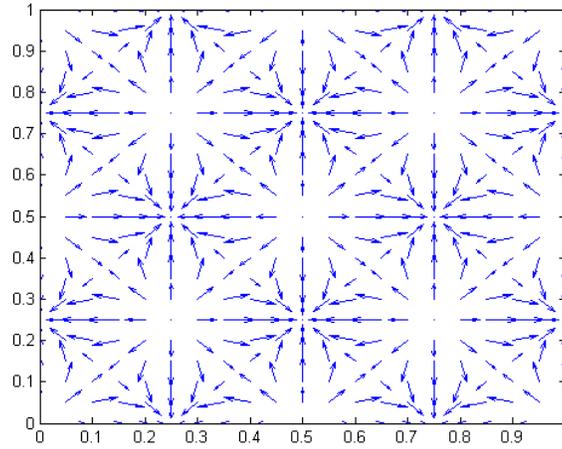


Fig. 5. Phase space of the system described by Eqns. (3) and (4) with $r_1 = r_2 = 0.02$.

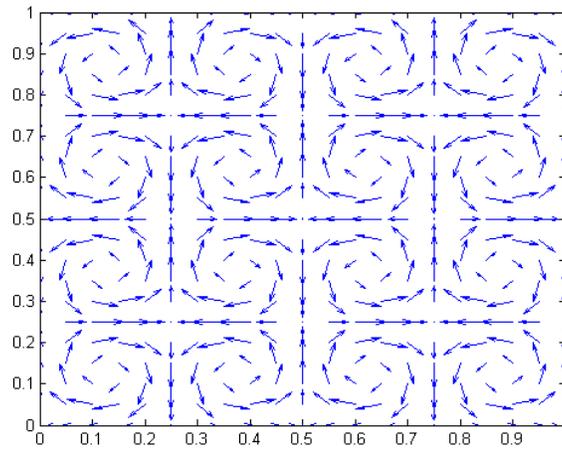


Fig. 6. Phase space of the system described by Eqns. (3) and (4) with $r_1 = 0.02$ and $r_2 = 0.98$.

selection pressure provides a push towards ROMAs - is likely to carry over to nature. Furthermore, the results presented here correspond with [10]. More specific, the many to one mapping, called discretisation, in [10] also provides a push towards the axes.

7 Future Research

A logical follow up is to explore different types of parameterized spaces of dynamical systems.

8 Conclusion

This paper provides a proof-of-principle of the evolution of GPMs as dynamical systems. It confirms the existence of ROMAs - regions in a parameterized space of GPMs - evolution leads to. These ROMAs are stable but hard to predict.

References

1. International Human Genome Sequencing Consortium: Finishing the Euchromatic Sequence of the Human Genome. *Nature* (431) 7011, pp. 931–945, Nature Publishing Group (2014).
2. Wagner, Gunter P and Altenberg, Lee: Perspective: Complex adaptations and the evolution of evolvability. *Evolution*, pp. 967–976, JSTOR (1996).
3. Alberch, P.: From genes to phenotype: dynamical systems and evolvability. *Genetica*, 84(1), pp. 5–11, Springer (1991).
4. Jaeger, J. and Monk, N.: Bioattractors: dynamical systems theory and the evolution of regulatory processes, *The Journal of physiology*, 592(11), pp.2267–2281, (2014).
5. Paredis, J.: Steps towards co-evolutionary classification neural networks. *Proceedings of the Fourth International Workshop on the Synthesis and Simulation of Living Systems*, pp. 102–108, MIT Press (1994).
6. Hillis, W. D.: Co-evolving Parasites improve Simulated Evolution as an Optimization Procedure. *Physica D: Nonlinear Phenomena*, (42)1-3, pp. 228–234, Elsevier (1990).
7. Paredis, J.: *Coevolutionary Computation*, *Artificial life*, 2(4), pp. 355–375, MIT Press (1995).
8. Paredis, J.: (Co)evolution Leads towards ROMAs, *Proceedings of the European Conference on Artificial Intelligence 2014 (ECAI-14)*, pp. 1079-1080, IOS Press (2014), doi: 10.3233/978-1-61499-419-0-1079.
9. Paredis, J.: Where does (Co)evolution Lead to?, *Proceedings of the European Conference on Artificial Life*, pp. 138-145, MIT Press (2015).
10. Paredis, J.: Exploring the Evolution of Genotype Phenotype Mappings, *Proceedings IEEE Congress on Evolutionary Computation* (2017).
11. Whitley, D.: The GENITOR algorithm and selection pressure: Why rank-based allocation of reproductive trials is best, *Proceedings of the third International Conference on Genetic Algorithms*, vol. 1, pp. 116–121, Morgan Kaufmann(1989).
12. Jordan, D. W., and Smith, P.: *Nonlinear ordinary Differential Equations: Problems and Solutions*, Oxford University Press, USA (2007).