

Where Does (Co)evolution Lead to?

Jan Paredis

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

Abstract

This paper investigates the dynamics of a simple coevolutionary system. It consists of a predator-prey system in which one population maximizes its distance to the members of the other population, while the second population tries to minimize the distance to the first population. This results in a coevolutionary pursuer-evader (PE) system whose dynamics can easily be studied.

Next, a simple genotype-phenotype mapping is added to the system. This mapping - as well as other sources of increased selection - push the system towards regions of maximum adaptability (ROMAs). These ROMAs are a generalization of the concept "evolution to the edge of chaos".

Introduction

Three concepts are central in this paper: evolution to the edge of chaos, Genotype Phenotype Mappings (GPMs), and coevolution. Each of these concepts is briefly introduced here.

The knowledge that evolution leads to the *edge of chaos* dates back to the late 80ies, early 90ies, see, for example Packard (1988), Kauffman and Johnsen (1991), and Langton (1990). A more critical account of evolution to the edge of chaos can be found in Mitchell et al. (1994). Paredis (1997) suggested why evolution leads to the edge of chaos:

At each transition between order and chaos there is a high density of different dynamical behaviors, in a volatile environment, it is better to be prepared for change!

However, this hypothesis has not yet been proven empirically or theoretically. Or, as Wang et al. (2011) discuss:

Much has been speculated on the possibility that gene regulatory and other biological networks function in (or evolve to) the critical regime (Gershenson, 2004).

The current paper investigates the evolutionary dynamics of *Genotype to Phenotype Mappings (GPMs)* in a coevolutionary model. Due to the rather unexpected small number of genes in the human genome, the interaction of these

genes during morphogenesis has become an important research topic. It is the GPM which implements morphogenesis. Hence, its importance. In nature this GPM is complex. Or, as Benfey and Mitchell-Olds (2008) describe it:

The difficulty in mapping genotype to phenotype can be traced to several causes, including inadequate description of phenotypes, too little data on genotypes, and the underlying complexity of the networks that regulate cellular functions.

More specific, the concept of evolvability has been introduced. It states that the GPM is under evolutionary pressure as well. It is assumed that these mappings are evolved in order to make the GPM robust and successful in various different (changing) environments. Or, as Benfey and Mitchell-Olds Pigliucci (2010) discuss:

This re-thinking of the genotype-phenotype relationship and its consequences in terms of the related concepts of robustness, modularity and evolvability (Wagner, 2005) are part of an emerging Extended Synthesis in evolutionary biology

Wagner and Altenberg (1996) 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.

Besides GPMs, this paper studies *coevolution*. After the seminal work of Hillis (1990) on predator-prey coevolution for optimization, quite a considerable amount of research has been done on the use of predator-prey coevolution. An overview of this research can be found in (Paredis, 2000). Basically, this research shows how the arms race resulting from predator-prey interactions can be used in order to ob-

tain efficient optimization. Far less research has been devoted to the study of the dynamics of these coevolutionary algorithms. De Jong (1999) proposed this research area for traditional (single population) genetic algorithms. What makes coevolutionary algorithms different is that they have to deal with dynamic environments: the populations change all the time as a response to the changes in the other population. An overview on dynamic worlds can be found in Dempsey et al. (2009). Such a dynamic environment provides a rich basis for the dynamics.

The definition of evolvability is rather varied, or as Pigliucci (2008) expresses it mildly:

The concept of evolvability has many different definitions, generating some conceptual confusion, or at least pluralism.

Dempsey et al. (2009) define evolvability differently (than above) and link it to dynamic worlds as follows:

Evolvability is the potential to incrementally increase the population fitness without first experiencing a decrease as it escapes a local optimum. Where the environment is dynamic a high level of evolvability allows the population to more easily track a moving optimum.

In the current paper, the Coevolutionary Genetic Algorithm (CGA), introduced in Paredis (1994), is used. In the past the CGA has mainly been used as a tool for optimisation, see e.g. Paredis (1995). 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 structure of this article is as follows. First, the standard CGA is described. Next, the simple application which results in Pursuer Evader (PE) dynamics is described. Next, the genotype-phenotype mapping is introduced. The fifth section describes the empirical results associated with the GPM. Next, related research is discussed. The seventh section discusses the model used and its relation to the real world, followed by a road map for future research. Finally, conclusions are given. Note that a short version of the current paper is published in Paredis (2014). This short paper does not include any empirical results.

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 pseudocode 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 (Whitley et al., 1989) 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.

```
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)
```

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 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). In the current paper, all pa-

parameter settings and genetic operators are identical to those described in Paredis (1995) unless mentioned otherwise.

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). 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.

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. 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).

Introducing a Genotype-Phenotype Mapping

In this section, the PE-model is extended with a simple GPM in order to study its contribution to the dynamics. To add a GPM two real numbers from the interval $[0,1]$ are added to the gene string of each evader and pursuer. This string then takes the following form: $(x \ y \ r_1 \ r_2)$. The two first parameters are as before. The last two define the GPM for each of the two parameters (x and y) independently. Hence, the GPM is under evolutionary control.

Each time a new individual is born - this happens once per population and per cycle) the GPM functions as follows. The two iterative mappings, equations (1), and (2), are applied to x and y , respectively, a fixed number of times. This number is called "pregtime". Here, in all experiments, α is set to 0.001. After this, the individual's fitness (through encounters, just as before) is calculated and if fit enough it is inserted in the population at the appropriate rank. Note that the GPM changes the original x and y genes. Or, more precisely, the genotype remains the same, the phenotype - which is originally a copy of the genotype - changes. Also the GPM considers the genes to live on a torus: if it becomes larger than 1, 1 is subtracted from it, x and y cannot become negative because α as well as r_1 and r_2 are all positive.

$$x_{n+1} = x_n + \alpha r_1 x_n \quad (1)$$

$$y_{n+1} = y_n + \alpha r_2 y_n \quad (2)$$

After birth, these iterative functions are applied once during each cycle of the CGA and that during a fixed number of cycles. This constant is called "growtime". Both processes are simple models of a PGM operating in two phases. The first phase represents development during pregnancy, during which no interaction with the world occurs. The second one represents further development after birth, i.e. growth, during which the phenotype interacts with the world. Here, pregnancy as well as growth are represented by small positive increments of the phenotype.

Empirical Results

Given the PE behavior of the standard CGA described before, the question now is: How do r_1 and r_2 evolve in various settings? In order to study this, 100 runs (of 20000 cycles) of the PE CGA with PGM were run. Then all r 's belonging to one population are printed on a $[0,1] \times [0,1]$ plane. Hence, these are the values of the r 's at the end of each of the 100 runs. In this paper, an experiment will be described by a quadruple: the first element is the size of the evader population, the second the size of the pursuer population, the third the pregtime, and finally, the growtime.

In the first case pregtime and growtime are both zero, the r 's play no (evolutionary) role and they will be distributed randomly over the plane. At the end of each run, the r_1 and r_2 of each evader or pursuer are plotted as a dot on a

$[0,1] \times [0,1]$ plane. Figure 1 shows the distribution of the r 's of the evaders in the experiment represented by the quadruple 50-50-0-0: both population sizes are 50 and no GPM is used. Any pattern in this figure is due to random drift and the relatively small number of runs (100). Because of the symmetry between the evaders and pursuers (same population size, and both populations do not use a GPM), the r 's of the pursuers show a similar distribution as the evaders.

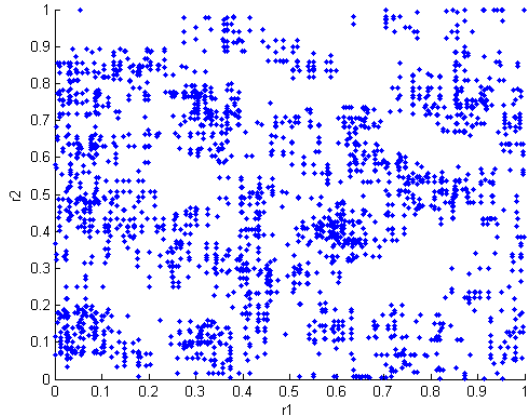


Figure 1: Distribution of r 's of evaders for 50-50-0-0

In the next experiment - 50-50-80-20 - pregame is set to 80 and growtime to 20. Or, in other words, directly after birth of an evader or pursuer the PGM is applied 80 times, during the following 20 cycles the PGM is applied one time per cycle on that individual. Figure 2 shows the distribution of the r 's of the evaders. Clearly, there is structure now: the r 's concentrate around the "edges". They even seem to concentrate most at the "corners". The explanation why this happens is extremely simple: at these places mutation results in the largest variation of change (i.e. change of movement of the x, y genes on the $[0,1] \times [0,1]$ torus). At these edges, it switches from maximal movement to zero movement and vice versa. This allows the individuals of each population to out-manoeuvre the individuals of the other population. The term *regions of maximum adaptability* (ROMAs) is used here to define these regions where the r 's (and hence the GPM) evolve to. Actually, the term ROMA was coined in Paredis (2014). In terms of dynamic systems, these ROMAs are attractors and the density of the r 's give an indication of their strength. Again, due to symmetry (same population size, and same amount of GPM), the distribution of the r 's of the pursuers is similar to the distribution of the r 's of the evaders.

Now, the influence of differences in population size is investigated, again pregame is set to 80, and growtime is 20. In this experiment, the population size of the evaders is set to 20, the pursuers are still with 50. Figure 3 depicts the distribution of the r 's of the evaders. Obviously, there are

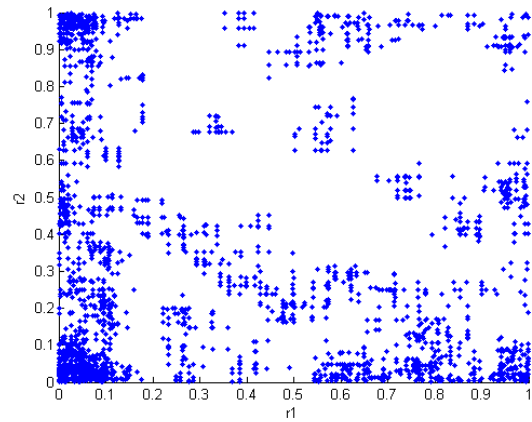


Figure 2: Distribution of r 's of evaders for 50-50-80-20

less dots now because the population size is smaller. Furthermore, there is no strong pressure towards the edges (or corners). This in comparison with the distribution of r 's of the pursuers (figure 4). It can be seen that for the largest population (that is the pursuers) the pressure towards ROMAs is the strongest. The larger populations move slower because at each cycle one offspring is generated. Hence, it is more important, for their survival, that they have access (genetically and / or through the GPM) to a wide variety of behavior. Such that they can out-manoeuvre the smaller, faster population.

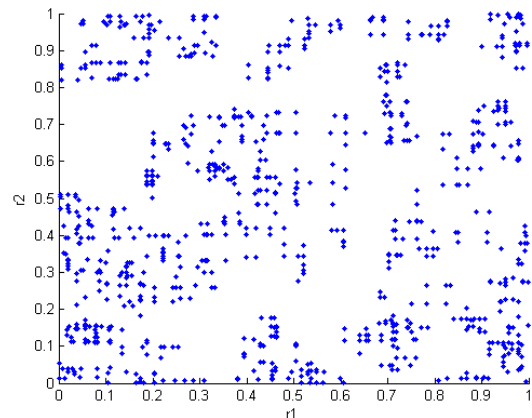


Figure 3: Distribution of r 's of evaders for 20-50-80-20

The same phenomenon is observed when the population of pursuers is smaller (20) than the evader population (50). In this case, the r 's of the evaders are pushed more towards the ROMAs. Clearly, the effect of the GPM, pushing the r 's to the ROMAs, is enforced by the increased pressure on the largest population.

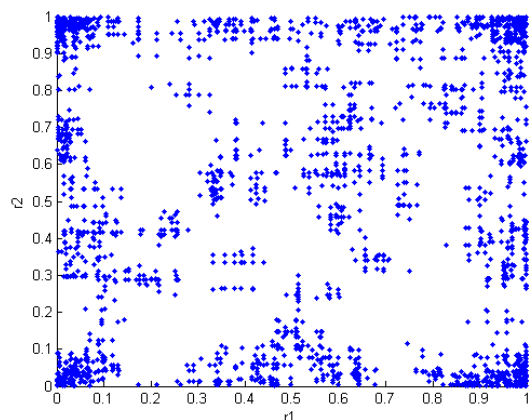


Figure 4: Distribution of r 's of pursuers for 20-50-80-20

A final set of experiments studies the effect of changing the influence of the PGM and the effects of the interplay of its two phases: pregnancy and growth. When both, preptime and growtime, are set to zero then the r 's don't play any role. Hence, both r 's are randomly distributed (as was the case in figure 1). When preptime or growtime are set to hundred (and the other one remains zero) then the r 's belonging to the largest population most clearly moves to the ROMAs. If both population sizes are equal both r 's move to the ROMA in an equal way as was seen in figure 2. This phenomenon is most outspoken when preptime is hundred. In this case, the influence of the PGM is largest: on each individual the PGM is applied 100 times. In the other case, it depends on the age, i.e. the number of cycles executed since its birth, the individual attains. In a dynamic world, the individuals are typically short lived. Figure 5 depicts the evolution of the average age of both populations during a typical run of 20-50-80-20 over 10000 cycles. As a matter of fact, the individuals belonging to the smaller population have the shortest life span. Hence, the effect of the GPM is most outspoken / guaranteed during the pregnancy. As a result, the pressure towards the ROMAs will be the strongest when preptime is set to hundred (and growtime equals zero). Summarizing, pressure towards ROMAs is stronger for (comparatively) larger populations and with increasing role of the PGM. Or, yet in other words, as selection pressure increases the push towards ROMAs becomes stronger.

In fact, one can evolve more parameters of the GPM than only the r 's, such as the α 's. This way, the balance between evolution and the GPM can be evolved and studied. One step further is to evolve the constituent parts of the GPM (the preptime and growtime). This provides insight into the interplay between these components under different circumstances.

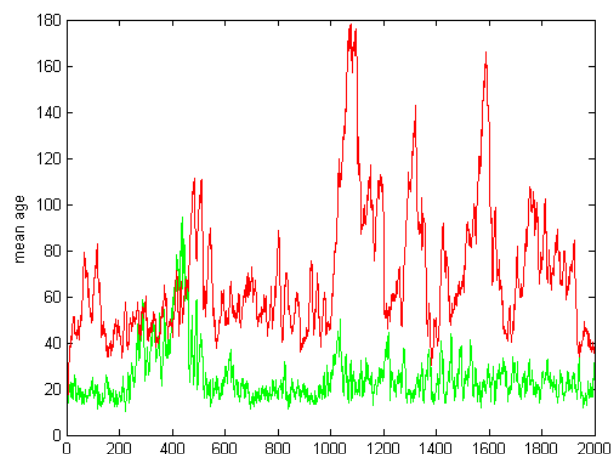


Figure 5: Evolution of the average age of both populations during a typical run of 20-50-80-20 over 10000 cycles. The x-axis represents the number of cycles (at each fifth cycle). The lower graph represents the average age of the evaders.

Related Research

Ebner et al. (2010) investigate the dynamics of coevolutionary interacting species as well. One important difference is the level at which the interactions are modelled. Ebner et al. model at the level of species, whereas the current paper models the interactions at the level of individuals.

Suzuki and Kaneko (1994) describe research closely related to the PE described here. They use a single population of birds. The fitness of a bird is determined by: 1) How hard it is for other birds to imitate its song, and 2) How good it is at imitating the song of the other birds. In fact, in that model, evolution leads towards the edge between chaos and a periodic window.

The research of Castro de Oliveira de Oliveira (2001) is particular relevant here. This work states that eternal transients are needed such that the system never reaches its final destiny. These can be found at the edge of chaos. Literally:

Evolutionary systems stick to critical situations (the edge of chaos), because within all other possibilities (regular or chaotic regimes) they rapidly become trapped into low-dimensional attractors, losing diversity. This trapping feature forbids the system to explore the whole set of available possibilities inside its higher-dimensional space of states.

The current paper provides a far simpler and more general explanation for evolution to the edge of chaos: in a dynamic world evolution leads towards ROMAs. In these regions a maximum repertoire of behaviors is easily accessible through the application of genetic operators and / or the GPM. This is exactly what happens near the edge of chaos: it

is there that the most diverse behaviors are situated. Hence, ROMAs generalize this concept of evolution to the edge of chaos. Or, from a slightly different perspective, ROMAs are the regions where evolvability is high. In retrospect, the results presented here seem rather intuitive. However, extensive experimentation was needed to obtain these results and to understand them.

Discussion

The fact that a very simple model is used here, can give raise to questions like: How realistic is this model? How do the results carry-over to the real world? Are the results not mere artefacts of the model chosen?

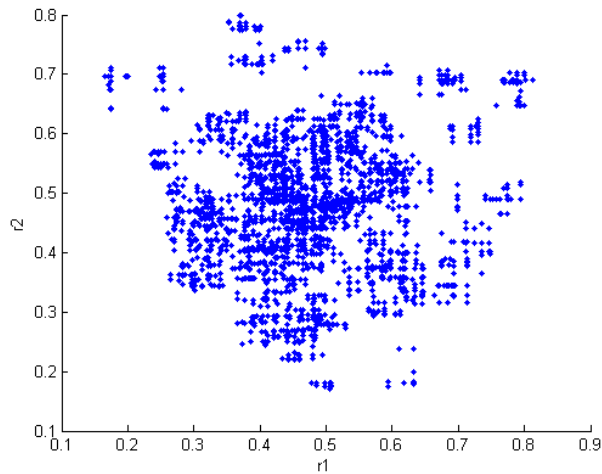


Figure 6: Distribution of r 's of evaders for 50-50-80-20 using non-toroidal mutation of the r 's

As described before, the fact that a toroid model is used here is to allow for infinite, ongoing evolution. Besides this, the toroidal model allows to have different behaviors near each other in the gene space. In the PE model used here, there is no real interaction (epistasis) between the genes of one individual. The interaction is with the individuals of the other population. In nature, epistasis (within and between populations) allows for small changes in genes to result in large effects (i.e. different behavior). Here, the toroidal model plays the role of epistasis: both create regions where small changes lead to large effects. Hence, in other words, epistasis gives rise to ROMAs.

Actually, experiments were done where mutation of the r 's is no longer toroidal (but mutation of x and y still is). In these cases, different patterns of r 's are formed: a cloud is formed with r_1 and r_2 roughly concentrated at the centre of the plane (see figure 6 for the distribution of the r 's of the evaders of 50-50-80-20). As before when selection pressure is increased - through a larger population size (relative to the

other population) or "more" GPM - the r 's concentrate more at the centre of the plane (see figure 7 for 20-50-80-20).

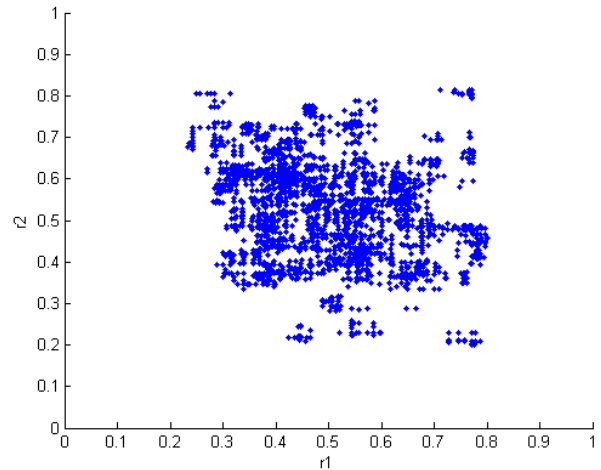


Figure 7: Distribution of r 's of pursuers for 20-50-80-20 using non-toroidal mutation of the r 's

These figures clearly indicate the impact of non-toroidal mutation versus the toroidal mutation used earlier. Whereas increased selection pressure pushes the r 's to the edges and corners in case of toroidal mutation, non-toroidal mutation results in a pressure towards the centre.

Summarizing, the toroidal nature of mutation, might seem artificial, but it compensates for the lack of epistasis in this simple PE model.

Future Research

As stated before, the PE model used here is extremely simple. This is because the goal was to take the simplest model to study interesting coevolutionary dynamics (including PGMs). These simplifications provide a road map for future research which allows to investigate the respective implications / contributions of these restrictions, and their relaxations, to the dynamics.

First of all, the PGM can be made more realistic. Now, it is a one to one mapping. It is known that typically PGMs are one to many mappings. This should further increase evolvability. Furthermore, the fact that the genotype space is similar to the phenotype space is unrealistic.

Another road is to use a different application instead of the pursuer evader system. The PE was used here because it would provide interesting and everlasting dynamics.

Also the fixed (and small) length of the individuals could be relaxed in order to see whether this has an effect on the dynamics.

Obviously, parameters could be set different for each population. Now all parameters, except the experiments with

different population sizes, were kept identical. This could include parameters like reproduction rate, mutation rate, size of mutation Here again, the effect of these changes on the dynamics can be investigated. As already mentioned before, the parameters of the GPM, such as pregame and growtime, could be evolved as well.

A final remark is that although the application discussed here (PE) is spatial. The evolution, however, is not organized in a spatial way, i.e. a mixed model is used here. Clearly, a spatial setup in which selection and reproduction is local has been shown to be more natural and more efficient at optimizing Pagie and Hogeweg (1997). It is expected that the use of a CGA with a spatial component would have an important impact on (the complexity of) the dynamics.

Finally, it is good to note, that the original aim of the research reported here was to investigate the dynamics of the CGA described in Paredis (1995). That is why a non-spatial organisation is used. For the same reason, each population reproduces once per cycle. Clearly, this is not the case in nature. Nonetheless, the general principle discussed here - increased selection pressure providing a push towards ROMAs - is likely to carry over to nature. This because the basic processes of natural evolution are used here: variation, selection, and reproduction with inheritance.

Conclusion

This paper studies the dynamics of a coevolutionary (predator-prey) algorithm by means of a simple pursuer evader application on a torus. This application allows for easy visual inspection of the dynamics. In case both populations have the same size then each population rapidly clusters and the cluster of pursuers chases the cluster of evaders. Occasionally, the pursuers catch up with the evaders. These then split up in sub clusters symmetrically around the pursuers, immobilizing the latter for a while, until re-clustering appears.

Next, a simple genotype-phenotype mapping (GPM) was introduced. This mapping is under evolutionary control as well. As selection pressure increases the GPM evolves towards regions of maximal adaptability (ROMAs). From these regions the individual can easily change its behavior, and has access to a large repertoire of different behaviors. These ROMAs are a generalization of the concept "evolution to the edge of chaos". Or, in other words, the ROMAs are regions with high evolvability.

Acknowledgements

I am indebted to Ivo Bleylevens for getting me started with Matlab. Roos-Marie Bal proofread a draft of this paper. The comments of the ECAL-15 reviewers allowed to improve an earlier draft of this paper.

Jan Paredis (2015) Where Does (Co)evolution Lead to?. Proceedings of the European Conference on Artificial Life 2015, pp. 138-145

References

- Benfey, P. N. and Mitchell-Olds, T. (2008). From genotype to phenotype: systems biology meets natural variation. *Science*, 320(5875):495–497.
- De Jong, K. (1999). Evolving in a changing world. *Foundations of Intelligent Systems*, pages 512–519.
- de Oliveira, P. M. C. (2001). Why do evolutionary systems stick to the edge of chaos. *Theory in Biosciences*, 120(1):1–19.
- Dempsey, I., O’Neill, M., and Brabazon, A. (2009). *Foundations in Grammatical Evolution for Dynamic Environments*, volume 194. Springer.
- Ebner, M., Watson, R. A., and Alexander, J. (2010). Coevolutionary dynamics of interacting species. In *Applications of Evolutionary Computation*, pages 1–10. Springer.
- Gershenson, C. (2004). Updating schemes in random boolean networks: Do they really matter. In *Artificial Life IX Proceedings of the Ninth International Conference on the Simulation and Synthesis of Living Systems*, pages 238–243. MIT Press.
- Hillis, W. D. (1990). Co-evolving parasites improve simulated evolution as an optimization procedure. *Physica D: Nonlinear Phenomena*, 42(1):228–234.
- Kauffman, S. A. and Johnsen, S. (1991). Coevolution to the edge of chaos: Coupled fitness landscapes, poised states, and coevolutionary avalanches. *Journal of Theoretical Biology*, 149(4):467–505.
- Langton, C. G. (1990). Computation at the edge of chaos: Phase transitions and emergent computation. *Physica D: Nonlinear Phenomena*, 42(1):12–37.
- Mitchell, M., Crutchfield, J. P., and Hraber, P. T. (1994). Dynamics, computation, and the "edge of chaos": A re-examination. In *Santa Fe Institute Studies in the Sciences of Complexity-Proceedings*, volume 19, pages 497–497. Addison-Wesley Publishing Co.
- Packard, N. H. (1988). *Adaptation toward the edge of chaos*. University of Illinois at Urbana-Champaign, Center for Complex Systems Research.
- Pagie, L. and Hogeweg, P. (1997). Evolutionary consequences of coevolving targets. *Evolutionary Computation*, 5(4):401–418.
- Paredis, J. (1994). Steps towards co-evolutionary classification neural networks. In *Proceedings of the Fourth International Workshop on the Synthesis and Simulation of Living Systems*, pages 102–108.
- Paredis, J. (1995). Coevolutionary computation. *Artificial life*, 2(4):355–375.
- Paredis, J. (1997). Coevolving cellular automata: Be aware of the red queen. In *Proceedings of the Seventh International Conference on Genetic Algorithms*, pages 393–400.
- Paredis, J. (2000). Evolutionary computation 2: Advanced algorithms and operators, chapter coevolutionary algorithms.

- Paredis, J. (2014). (co)evolution leads towards romas. In *Proceedings of the European Conference on Artificial Intelligence 2014 (ECAI-14)*, doi: 10.3233/978-1-61499-419-0-1079, pages 1079–1080. IOS Press.
- Pigliucci, M. (2008). Is evolvability evolvable? *Nature Reviews Genetics*, 9(1):75–82.
- Pigliucci, M. (2010). Genotype–phenotype mapping and the end of the genes as blueprintmetaphor. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 365(1540):557–566.
- Suzuki, J. and Kaneko, K. (1994). Imitation games. *Physica D: Nonlinear Phenomena*, 75(1):328–342.
- Wagner, A. (2005). *Robustness and evolvability in living systems*. Princeton University Press Princeton.
- Wagner, G. P. and Altenberg, L. (1996). Perspective: Complex adaptations and the evolution of evolvability. *Evolution*, pages 967–976.
- Wang, X. R., Lizier, J. T., and Prokopenko, M. (2011). Fisher information at the edge of chaos in random boolean networks. *Artificial Life*, 17(4):315–329.
- Whitley, D. et al. (1989). The genitor algorithm and selection pressure: Why rank-based allocation of reproductive trials is best. In *Proceedings of the third international conference on Genetic algorithms*, volume 1, pages 116–121.