Exploring the Evolution of Genotype Phenotype Mappings

Document status and date:
Published: 01/01/2017

DOI:
10.1109/CEC.2017.7969324

Exploring the Evolution of Genotype Phenotype Mappings

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

***Abstract*—This paper investigates the evolution of two types of simple Genotype Phenotype Mappings (GPMs): a many-to-one mapping and a one-to-many mapping. Both GPMs are under genetic control. For both types of mappings different Regions Of Maximum Adaptability (ROMAs) are found. These ROMAs are the regions - in a paramterized space of GPMs - evolution leads to. The attraction towards these ROMAs increases as selection pressure increases. Finally, this paper discusses the evolution of pleiotropy and the ROMAs it leads to.**

I. Introduction

The Human Genome Project (International Human Genome Sequencing Consortium) [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). Or, as [2] describes the complexity of GPMs in nature:

> 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.

Wagner and Altenberg [3] 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.

In the current paper, the Coevolutionary Genetic Algorithm (CGA), introduced in [4], is used. This algorithm was inspired by the seminal work of Hillis [5]. In the past, the CGA has mainly been used as a tool for optimalisation, see e.g. [6]. 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 GPMs which - despite their simplicity - still exhibit realistic, complex dynamics.

Many techniques and models have been used to implement GPMs, e.g.: Lindenmayer sytems [7], Cellular automata [8], Random Boolean Networks [9], or reaction diffusion sytems [10], amongst others. These are rather complex models, here simple (mostly) linear equations are used to implement the GPMs.

Another distinction between approaches that evolve GPMs is whether the GPM is part of the genome, as in [11], or whether the genotype and GPM coevolve as seperate populations, as in [12]. Here, the former approach is followed. The symbiotic CGA, introduced in [13], could be used to implement the latter approach. The advantages and disadvantages of each approach are described in [14].

At an abstract level, a parameterized space of GPMs is used here. Each individual has an instantiation of the paramaters, i.e. a GPM, on its genotype. The current paper investigates to which parts of the space of GPMs evolution leads.

The current paper is a sequel to [15], [16] which introduced the concept *Region Of Maximum Adaptability* (ROMA). These ROMAs are exactly the region(s) in the space of GPMs evolution leads to. This previous work used a one-to-one mapping as a GPM. In the current paper, more complex GPMs are investigated: one-to-many and many-to-one mappings. In addition to this, two models of pleiotropy are discussed to allow for interaction of genes in the GPM. Actually, sections 2 and 3 of the current paper originate from [16]. Just like in the earlier reseach a Pursuer-Evader (PE) model is used. Here, two populations live on a torus. The fitness of the individuals of both populations is calculated through continuous sampling of the distance between itself and the individuals belonging to the other population. One population - the evaders - maximize this distance, the other population - the pursuers - minimize this distance. The PE model was chosen because it leads to ongoing evolution.

The structure of this paper is as follows. After this introduction the CGA is described. Section three describes the pursuer-evader (PE) sytem. Section four describes two types of GPMs and gives experimental results on their evolution. Section five describes the degree of pleiotropy a GPM evolves to. After which, a discussion and a section on future research are given. Finally, conclusions are drawn.

II. 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. The fitness of these individuals is then

978-1-5090-4601-0/17/$31.00 ©2017 IEEE

```
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)
```

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.

After the creation of the two initial populations, the main *cycle* of a CGA is executed. The pseudo-code of this cycle is given above. First, 20 encounters are executed between SELECTed individuals. This selection is linearly biased towards highly ranked individuals: similar to GENITOR (Whitley [17]) the top individual is 1.5 times more likely to be selected than the median individual. The pay-off of this encounter - here this is the distance bewteen both individuals - is then 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 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 [6] unless mentioned otherwise.

III. 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] x [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 also occurs: the pursuers breaking up to immobilize the evaders for a while, as the snapshot in figure 2 depicts.

When the two populations have different population sizes then their respective speed changes. This is because at each cycle both populations reproduce exactly 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

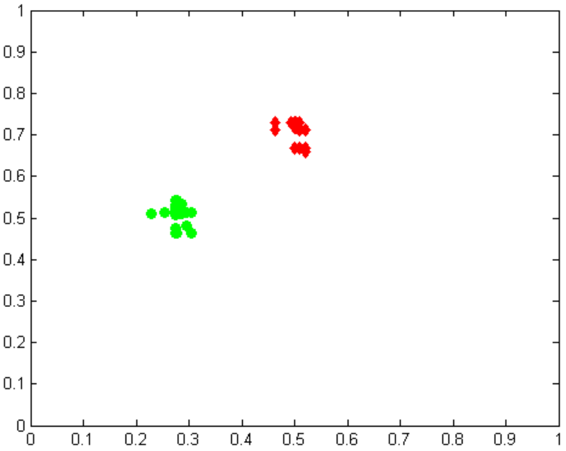

Fig. 1. A cluster of pursuers (black diamonds) pursuing a cluster of evaders (grey cirles)

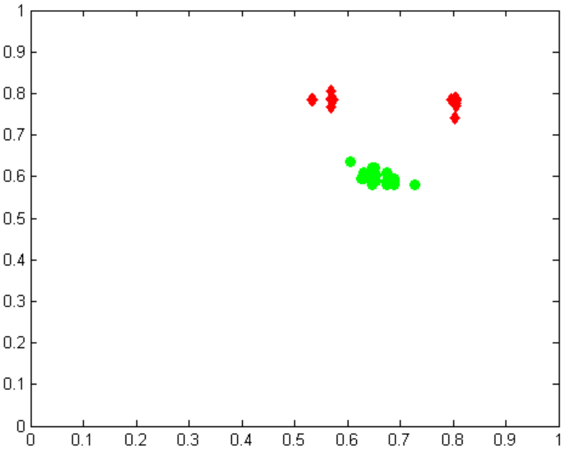

Fig. 2. A cluster of pursuers (black diamonds) splits up in two parts immobilizing a cluster of evaders (grey cirles)

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).

IV. Two types of GPMs

GPMs are known to be complex many-to-many mappings. Here, these mappings are decomposed into many-to-one mappings and one-to-many mappings, both are decscribed in the following two subsections. One example of each type of mapping will be discussed.

A. A many-to-one mapping: Discretisation

In order to have a simple GPM under evolutionary control, two real numbers in the interval [0,1] are added to the individual. Hence, an individual consists of a quadruple (x y r_1 r_2). As before, x and y are the position of the individual on the torus. The last two parameters, r_1 and r_2, determine the degree of discretisation applied to x and y, respectively. This discretisation process maps the genotype (x and y) into the phenotype. In the experiments described here, the number of bins varies between 5 and 200. Hence, the code below describes the discretisation process for x and y.

```
nrofbins= round(5 + 195 * r1)
x = round(x * nrofbins) / nrofbins

nrofbins= round(5 + 195 * r2)
y = round(y * nrofbins) / nrofbins
```

It is good to note that a small r (r_1 or r_2) results in few equally sized, large bins, e.g if r is zero that results in 5 bins of size 0.2. If, on the other hand, r is one then there are 200 bins of size 0.005.

Given the PE behavior of the CGA described before, the question now is: *To what values do r_1 and r_2 evolve in various settings?* This was studied by executing 100 runs (of 20000 cycles each). At the end of each run, all r's belonging to one population are printed on a [0,1] x [0,1] plane. Hence, these are the values of the r's at the end of each of the 100 runs.

In the first experiment, both populations have the same size, i.e. 50. Each experiment is described by two numbers: the population size of the evaders and of the pursuers. Hence, this first experiment is described as 50-50. Figures 3 and 4 show the distribution of the r's of the evaders and pursuers in this experiment. Remember that small r's correspond to small numbers of large bins.

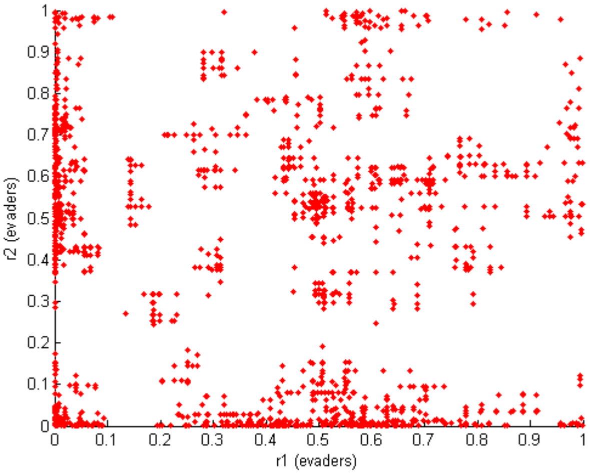

Fig. 3. Distribution of r's of evaders for 50-50

In both figures, there is a clear concentration of the r's near the axes. Or, in other words, the axes (including the origin) are the ROMAs for this type of GPM. Because of the symmetry between the evaders and pursuers, the r's of the pursuers show a similar distribution as the r's of the evaders, and hence, their ROMAs are similar. These ROMAs indicate that the evaders,

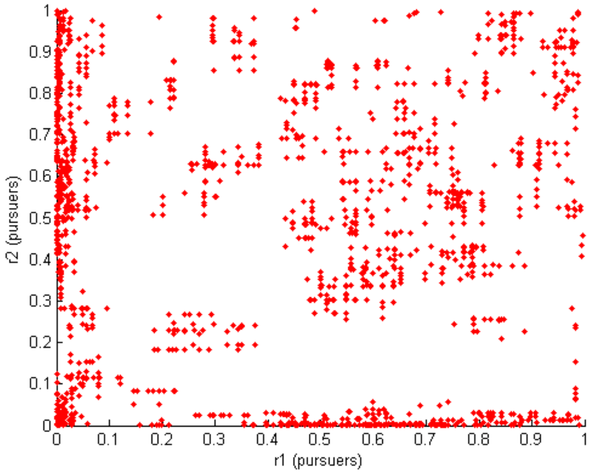

Fig. 4. Distribution of r's of pursuers for 50-50

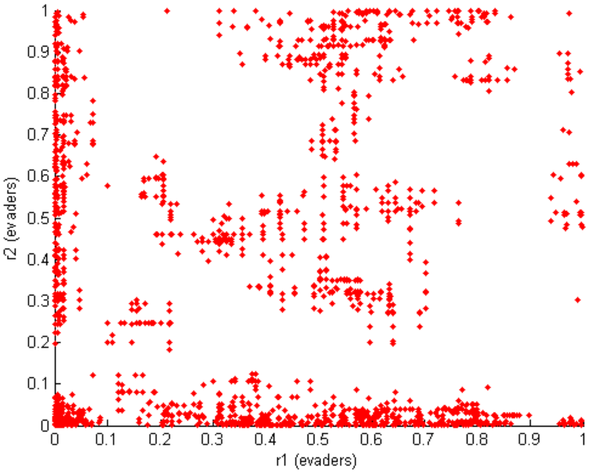

Fig. 5. Distribution of r's of evaders for 50-20

as well as the pursuers, prefer to make relatively large steps, during the GPM / discretisation, in at least one of the (x or y) dimensions. It is not clear why the r's evolve towards (at least one of) the axes. Also studying the evolution of the r's during single runs did not reveal any pattern: both populations of r's cluster and then stay roughly at the same location or slowly move, often along the axes.

A second experiment - 50-20 - uses different population sizes for the two populations. Now, the population size of pursuers (20) is smaller than the evader population size (50). Figures 5 and 6 depict the distribution of the r's of the evaders and pursuers, respectively. Obviously, there are less dots in the figure of the pursuers, because of the smaller population size. Furthermore, a stronger pressure towards the axes can be seen for the larger (evader) population. The density of the dots is an indication of the strength of the pressure towards the ROMAs. The larger population moves slower (on the x,y torus) because at each cycle one offspring is generated for each population. Or, in other words, the selection pressure is higer on the larger (evader) population. Again, there is pressure towards larger bins (almost zero r) in at least one of the two dimensions. But now, increased selection pressure results in stronger pressure of the r's towards the ROMAs.

B. A one-to-many mapping: Expansion

Here, a simple one-to-many mapping is introduced. Just as in the previous section, the map is described by the same parameters, r_1 and r_2. But now, these describe the radius along each dimension (x and y) in which a random number is drawn to replace x and y, respectively. The code below shows how the expansion process works. A small random value is added or subtracted (with equal probability) to or from x and y. The code below shows how the genotype is transformed into phenotype by the expansion operator. The inbound function enforces that the phenotype remains on the torus [0,1] x [0,1].

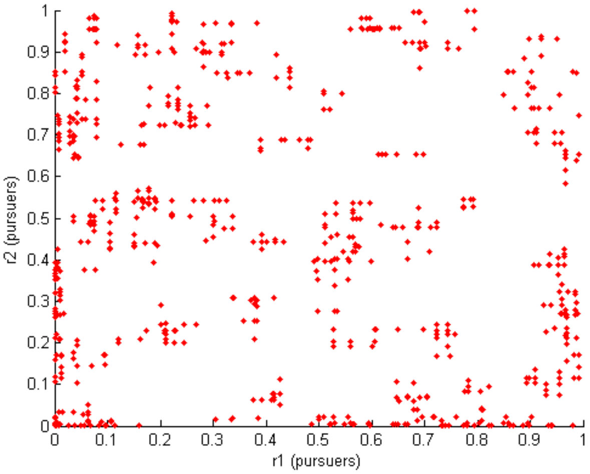

Fig. 6. Distribution of r's of pursuers for 50-20

The function rand generates a real valued number uniformly drawn from the interval [0,1].

```
x = inbound(x +/- 0.1 * rand * r1)


y = inbound(y +/- 0.1 * rand * r2)
```

Again, the same experiments as in the previous section are performed. First, both populations have an equal population size of 50 (figures 7 and 8). Now, both r's are pushed towards the upper right corner. This potentially maximizes the amount of movement from the genotype to the phenotype. Hence, in this case, the upper right corner constitutes the ROMA and to a lesser extend the edges r_1=1 or r_2= 1.

Here, as well, this attraction to the ROMAs is stronger as the selection pressure increases, i.e. the selection on the larger population is fiercer than the selection on the smaller

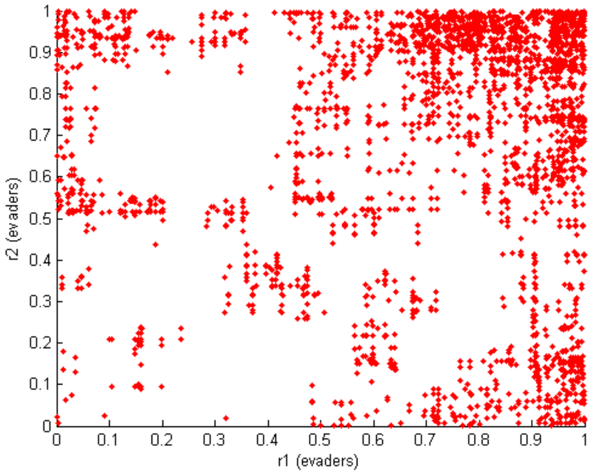

Fig. 7. Distribution of r's of evaders for 50-50

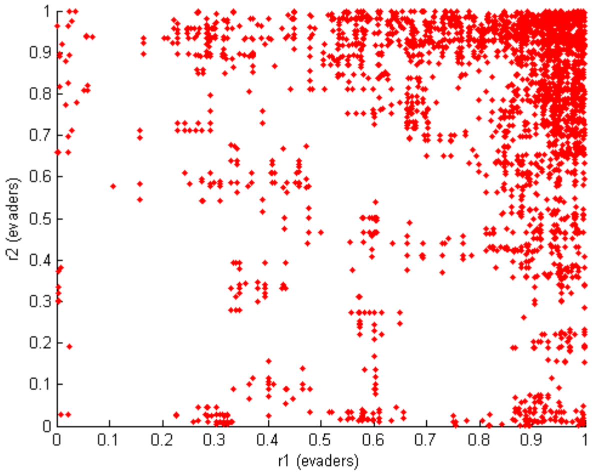

Fig. 9. Distribution of r's of evaders for 50-20

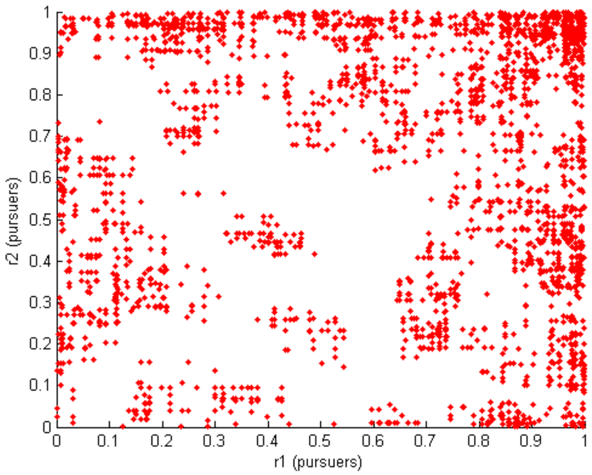

Fig. 8. Distribution of r's of pursuers for 50-50

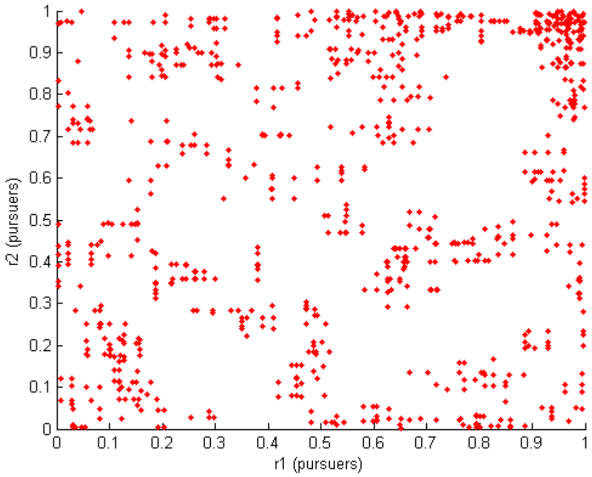

Fig. 10. Distribution of r's of pursuers for 50-20

population (see figures 9 and 10). Actually, the region around the origin of the r's of the evaders is almost empty.

The examples above illustrate that it is often hard to understand what the ROMAs will be, even for the simple models used here. Let alone to predict them. This will be the case for the ROMAs in the next section as well.

V. Evolving Pleiotropy

In the section above, the GPMs did not allow for interaction between the genes. This section describes the degree of pleiotropy, i.e. interaction between the genes, a GPM evolves to in the PE model. Two models are distinguished: a linear model of interaction (as in [18]), and a non-linear one (based on [19]).

A. Linear Pleiotropy

In this model the r's determine how much the gene values get mixed in producing the phenotype. Now 4 r's - r_1, r_2, r_3, and r_4 - are evolved per individual (besides the x and y). This leads to the following GPM:

```
x= (r1 * x + r2 * y) / 2
```

```
y= (r3 * y + r4 * x) / 2
```

In this section, only the ROMAs of the evaders of 50-20 are given because they are the most outspoken. Furthermore, only r_1 and r_2 are depicted. This because r_3 and r_4 lead to similar ROMAs as r_1 and r_2. Figure 11 shows that the r's evolve towards the 4 corners of the graph. These are the extreme cases of mixing the genes: no mixing or complete mixing. It is not clear why the r's group around the origin. This allows the coordinates to be reset to zero or to a small value.

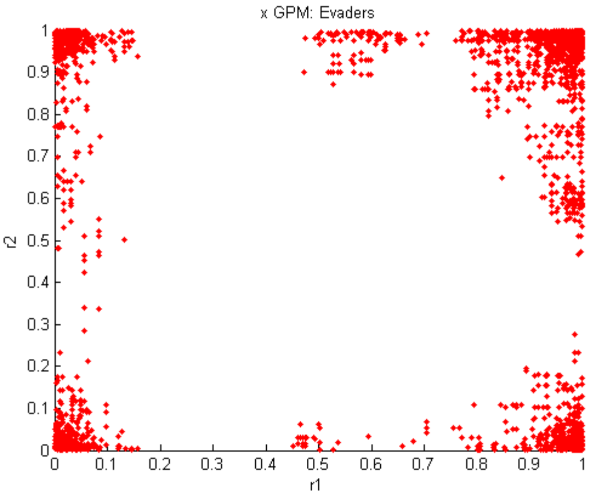

Fig. 11. Distribution of r's of evaders for 50-20 (linear pleiotropy)

B. Non-linear Pleiotropy

A non-linear GPM is obtained by adding a continuous sigmoid transfer function to the linear model above. Or, in other words, the sigmoid is applied to both the linear combinations.

Figure 12 shows that the r's evolve towards the corners as well, but much stronger than in the linear model Also, the pressure towards these ROMAs is less for the pusuers, as is shown in Figure 13. Here, only the upper right corner attractor is present. Actually, the r's are mainly spread out over the entire upper right half of the graph. Again, it is hard to find an explanation for this behavior. As was the case before, the r's associated with the y gene of both populations are similar to the r's associated with the x gene. Or, in other words, The ROMAs associated with the x and y genes have the same pattern.

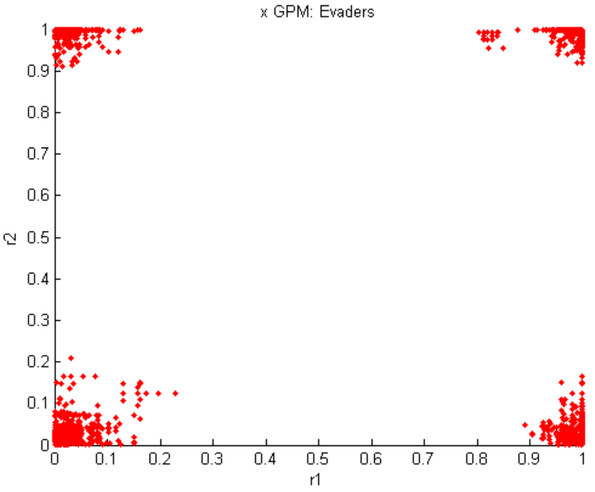

Fig. 12. Distribution of r's of evaders for 50-20 (non-linear pleiotropy)

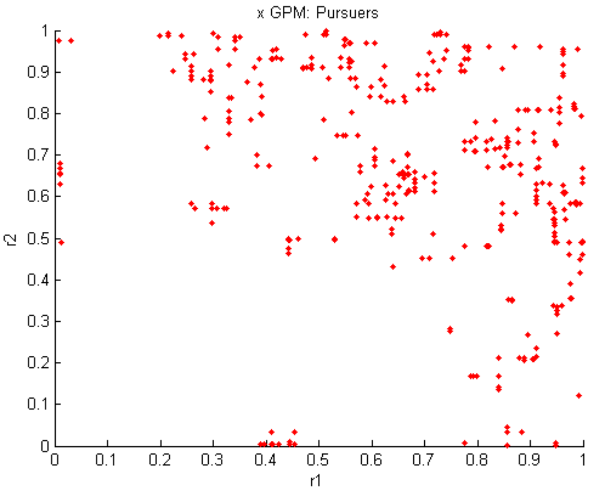

Fig. 13. Distribution of r's of pursuers for 50-20 (non-linear pleiotropy)

VI. Discussion

Paredis [16] argued that the toroidal model, on which x and y evolve, compensates for the lack of epistasis in the simple PE model: it allows for small (mutational) changes to have large (numerical) consequences. Actually, the expansion GPM takes into account the toroidal nature of the model. Discretisation does not have to take this into account because it "shrinks" the space. Hence, it will never get out of the bounds.

There is a second difference between the two GPMs: discretisation and expansion. The first one is deterministic whereas the second one (expansion) is stochastic. Actually, the expansion is completely random in terms of size as well as direction. Athough the former has an upperbound. Despite this random nature, the r's clearly evolve towards stable ROMAs . Furthermore, in terms of dynamical systems, the discretisation operator creates attractors, with the size of the bassin of attraction determined by the r's. The expansion operator, on the other hand, is a repellor.

Given the extremely simple models used here, the following question is certainly justified: Do the results presented carry over to nature? The fact that the basic processes (variation, selection, and reproduction with inheritance) are used, the general principle discussed here - increased selection pressure provides a push towards ROMAs - is likely to carry over to nature. Furthermore, the results presented here correspond with [16].

Finally, it should be stressed that the ROMAs are really stable: the experiments above all have been repeated numerous times. Each time, the same (ROMA) pattern appeared.

VII. Future Research

A strongly related issue to the research presented here is neutrality. Many-to-one GPMs are one source of neutrality. It would be interesting to investigate the relation between neutrality and the partial nature of the fitness evaluation. Finally, the spatial structure of the neutral regions is very simple here. In nature, the shape of these neutral pathways is much more complex. This is a second road for future research: the introduction of more complex (fractal?) neutral pathways.

An interesting, promissing, approach is to consider the GPM as a dynamical system [20]. Similar to the work described above, evolution can then move in a parameterized space describing a class of dynamical systems. It would then be interesting to see to which dynamical system(s) evolution leads.

VIII. Conclusion

This paper studied the evolution of two types of GPMs: many-to-one and one-to-many mappings. For both cases, the ROMAs were determined. These ROMAs are the region(s) in the space of mappings where evolution tends to lead to. Furthermore, it was shown that the attraction towards these regions increases as the selection pressure increases.

Finally, two pleiotropic models were used. From the experiments, it can be seen that the system evolves to the extreme cases of mixing: complete mixing or no mixing at all.

Even for the simple models used here, it is difficult to explain, or predict, to which GPMs (i.e. ROMAs) evolution leads. Despite this, the ROMAs are very stable: the same (ROMA) pattern arises each time a series of runs is done.

References

[1] International Human Genome Sequencing Consortium, "Finishing the euchromatic sequence of the human genome," *Nature*, vol. 431, no. 7011, pp. 931–945, 2004.

[2] P. N. Benfey and T. Mitchell-Olds, "From genotype to phenotype: systems biology meets natural variation," *Science*, vol. 320, no. 5875, pp. 495–497, 2008.

[3] G. P. Wagner and L. Altenberg, "Perspective: Complex adaptations and the evolution of evolvability," *Evolution*, pp. 967–976, 1996.

[4] J. Paredis, "Steps towards co-evolutionary classification neural networks," in *Proceedings of the Fourth International Workshop on the Synthesis and Simulation of Living Systems*, 1994, pp. 102–108.

[5] W. D. Hillis, "Co-evolving parasites improve simulated evolution as an optimization procedure," *Physica D: Nonlinear Phenomena*, vol. 42, no. 1, pp. 228–234, 1990.

[6] J. Paredis, "Coevolutionary computation," *Artificial life*, vol. 2, no. 4, pp. 355–375, 1995.

[7] A. Lindenmayer, "Mathematical models for cellular interactions in development." *Journal of theoretical biology*, vol. 18, no. 3, pp. 300–315, 1968.

[8] E. F. Codd, *Cellular automata*. Academic Press, 1968.

[9] S. Kauffman and S. Levin, "Towards a general theory of adaptive walks on rugged landscapes," *Journal of theoretical Biology*, vol. 128, no. 1, pp. 11–45, 1987.

[10] A. Gierer and H. Meinhardt, "A theory of biological pattern formation," *Kybernetik*, vol. 12, no. 1, pp. 30–39, 1972.

[11] R. E. Keller and W. Banzhaf, "The evolution of genetic code in genetic programming," in *Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation-Volume 2*. Morgan Kaufmann Publishers Inc., 1999, pp. 1077–1082.

[12] S. Margetts, "Adaptive genotype to phenotype mappings for evolutionary algorithms," Ph.D. dissertation, Citeseer, 2001.

[13] J. Paredis, "The symbiotic evolution of solutions and their representations," in *Proceedings of the 6th International Conference on Genetic Algorithms*. Morgan Kaufmann Publishers Inc., 1995, pp. 359–365.

[14] G. Wilson and M. Heywood, "Introducing probabilistic adaptive mapping developmental genetic programming with redundant mappings," *Genetic Programming and Evolvable Machines*, vol. 8, no. 2, pp. 187–220, 2007.

[15] J. Paredis, "(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*. IOS Press, 2014, pp. 1079–1080.

[16] ——, "Where does (co)evolution lead to?" in *Proceedings of the European Conference on Artificial*. MIT Press, 2015, pp. 138–145.

[17] D. Whitley *et al.*, "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*, vol. 1, 1989, pp. 116–121.

[18] G. Wagner, "Multivariate mutation-selection balance with constrained pleiotropic effects." *Genetics*, vol. 122, no. 1, pp. 223–234, 1989.

[19] C. Espinosa-Soto, O. C. Martin, and A. Wagner, "Phenotypic plasticity can facilitate adaptive evolution in gene regulatory circuits," *BMC evolutionary biology*, vol. 11, no. 1, p. 5, 2011.

[20] J. Jaeger and N. Monk, "Bioattractors: dynamical systems theory and the evolution of regulatory processes," *The Journal of physiology*, vol. 592, no. 11, pp. 2267–2281, 2014.