Evolving a Predator–Prey Ecosystem of Mathematical Expressions with Grammatical Evolution

MANUEL ALFONSECA[1] AND FRANCISCO JOSÉ SOLER GIL[2]

[1]*Escuela Politécnica Superior, Universidad Autónoma de Madrid, Francisco Tomás y Valiente, 11, Campus de Cantoblanco, Madrid 28049, Spain; and [2]Dept. Filosofía y Lógica, Universidad de Sevilla, Spain*

This article describes the use of grammatical evolution to obtain a predator–prey ecosystem of artificial beings associated with mathematical functions, whose fitness is also defined mathematically. The system supports the simultaneous evolution of several ecological niches and through the use of standard measurements, makes it possible to explore the influence of the number of niches and the values of several parameters on "biological" diversity and similar functions. Sensitivity analysis tests have been made to find the effect of assigning different constant values to the genetic parameters that rule the evolution of the system and the predator–prey interaction or of replacing them by functions of time. One of the parameters (predator efficiency) was found to have a critical range, outside which the ecologies are unstable; two others (genetic shortening rate and predator–prey fitness comparison logistic amplitude) are critical just at one side of the range), the others are not critical. The system seems quite robust, even when one or more parameters are made variable during a single experiment, without leaving their critical ranges. Our results also suggest that some of the features of biological evolution depend more on the genetic substrate and natural selection than on the actual phenotypic expression of that substrate. © 2014 Wiley Periodicals, Inc. Complexity 20: 66–83, 2015

1. INTRODUCTION

Ecological simulation is useful because it makes possible to test under controlled conditions, situations very difficult to analyze in real-life systems. Real systems,

Correspondence to: Manuel Alfonseca, E-mail: manuel.alfonseca@uam.es

except where the ecologies are made up of microorganisms, usually take thousands of years to evolve, what makes experimentation unpractical. Conversely, it is difficult to build experimental ecological systems simple enough to perform controlled experiments. This is straightforward in simulated systems.

Ecological simulation has a long history. Ever since Vito Volterra developed his famous predator–prey equations

[1], continuous simulation has been used to represent artificial ecological systems [2–4]. Discrete simulation has also been used frequently, using tools such as cellular automata [5,6] and Lindenmayer systems [7]. Agent-based artificial life ecosystems are relatively old [8] and have fused with artificial life research since the end of the 1980s (see [9] for a relatively recent survey of the field). Typical recent simulations in this field tend to define complicated predator–prey systems, which embody the agents with fuzzy cognitive maps and similar constructs [10,11]. Some of these systems do not address biological ecosystems but are directed to the simulation of social systems, with special application on economy [4,12].

In biological evolution, a genetic substrate, embodied in nucleic acids, is subject to a certain number of random actions (mutation, recombination, etc.). The different genetic compositions are not selected directly. They are translated into phenotypes whose mutual interaction gives rise to natural selection. Our hypothesis is that many of the features of biological evolution depend more on the genetic substrate and the mechanism of natural selection than on the actual phenotypic expression of that substrate. The fact that phenotypes as different as mathematical functions and biological beings give rise to similar features seems to support this hypothesis.

This article describes our experiments to build an evolving predator–prey ecosystem of artificial beings that compete for a limited resource nonspatial environment. The underlying genetic structure is not too dissimilar to that of living beings (a series of genes, represented as integers), subject to genetic algorithms similar to those in biology, but its phenotypic expression is quite different: grammatical evolution (GE) is used to generate, from the genetic substrate, phenotypic counterparts made of simple mathematical expressions. Natural selection is then applied to these phenotypes, after mathematically computing the fitness of the different individuals.

Our goal was to reduce the complexity of the ecosystem to the minimal expression, and test whether some of the typical features of biological evolution can be reproduced successfully in this simplified environment, such as a Volterra-like relationship between predators and prey, and others mentioned in the conclusions. In this way, other features we detect could provide new ideas about biological evolution. We have also studied which values of the genetic parameters generate more stable ecologies, and whether these parameters should actually be constants, or a certain amount of time-dependence is compatible with the stability of the ecologies. Exploring this question can lead to discovering the extent to which the structural changes affect the robustness of ecosystems.

Grammatical evolution, a standard technique in genetic programming (see [13–15]), suggested itself as the proper method, as it separates genomes from phenotypes and improves the closure problem (the need to eliminate individuals with invalid phenotypes), by protecting phenotypes against syntactic errors. Extensions to GE, such as attribute GE or Chistiansen GE [16,17] can also protect from semantic errors. We did not need to use those extensions because our individuals are protected from semantic errors in a different way (see Appendix A).

Our agents are very simple, as they only embody a mathematical expression, which is executed to compare their respective fitness. The environment is also very simple. In some ecological simulations, spatial distribution is important [6,18]. Our agents, however, do not have a space location. Conversely, they can belong to one of the several ecological niches, which evolve simultaneously but independently. We represent niches by applying different fitness functions to those individuals belonging to each niche. We also regulate niche population by making it possible for two niches to share the same fitness function, thus duplicating the population associated to that function.

This is the second set of experiments we have implemented following this technique. The first one [19] focused on the simulation of a parasite–host system, rather than a predator–prey system, like the one described here. Also, in the former study, we analyzed interniche interbreeding, while in this new study, we have performed a rather complete sensitivity analysis of the influence of different system parameters on the result. A detailed comparison between the results of both sets of experiments is left as future work.

This article is divided in the following way: Section 2 describes our procedure (GE and the generation of mathematical expression phenotypes from a genome; the evolutionary algorithm we use; and the predator–prey interaction). Section 3 describes the external parameters in our experiments. Section 4 shows the detailed results of two experiments that we thought particularly interesting among a total of 426 successful experiments we have performed; Section 5 describes a sensitivity analysis that shows the effect of changing several parameters, some of which were found to be critical or semicritical for the stability of the ecologies. Both fixed and variable parameters have been tested. Finally, Section 6 discusses and summarizes our conclusions and lists our future work objectives. Two appendices add some programming considerations and an example of the genotype to phenotype translation using the procedure described in Section 2.

2. GRAMMATICAL EVOLUTION

Grammatical evolution is an evolutionary automatic programming (EAP) algorithm based on strings, independent of the language used. Genotypes are represented by strings of integers (each of which is named gene) and the

context-free grammar of the target programming language is used to map each genotype into a syntactically correct phenotype (a mathematical expression or a program). In this way, GE avoids one of the main difficulties in EAP: the results of genetic operators are guaranteed to be syntactically correct.

Our agents are very simple entities reduced to the minimum information, which "live" and "evolve" together in a controlled non-spatial environment. Each individual consists of a "genome," a vector of n integers in the [0-255] interval. The value of n is random for each initial genome in the [50-199] interval.

We have introduced the concept of "niche," which makes it possible to split the population in several subpopulations, each using a different fitness function. The first element of the genome defines the ecological niche to which the individual belongs. The remaining elements provide the genomic information used to translate the genotype into an equivalent phenotype, which will be subject to evolutionary selection. The role of each element in the genome depends on its position and is redundant (several different integers in the same position give rise to the same phenotype). This is done to emulate the fact that the genetic code in living beings is degenerate [20], although the amount of redundancy used in GE is usually larger.

One of the niches in each experiment is assumed to be the "predators." The remaining niches are made of "prey." Therefore, the minimum number of niches is 2. Predators have an additional externally controlled parameter, their average "efficiency" (Ef). This parameter can be changed for each different experimental run but is constant during the execution of the experiment.

Different strategies were tested for predator–prey interaction.

- First, predators and prey are paired-up. We tested a random pairing versus pairing the best predators with the worst prey (this happens in biological systems, where predators usually capture aged, sick, or tired prey). We found the second strategy to be best for our purposes (in the sense that it gives rise to a higher number of stable experiments, those which endure for more than 200 generations).
- Once paired up, each predator may "eat" its prey, or the prey may "escape," depending on their fitness. Several strategies were also tested here: forcing the predator fitness to be better than the prey versus using a functional comparison of the form

$$\text{Ef.fun}\left(\frac{\text{FP}}{\text{Fp}}\right) \geq ?100$$

where Ef is the predator efficiency mentioned before, FP is the inverse fitness of the prey, Fp the inverse fit-

ness of the predator, ?100 is a uniformly distributed random number in the [0,100) interval and "fun" is a function that can be changed in different experiments. Two functions were tested: a simple logarithm and the logistic curve:

$$k + \frac{2(1-k)}{1+e^{-0.5\text{LN}\frac{\text{FP}}{\text{Fp}}}} \tag{1}$$

where $0 \leq k \leq 1$ is the logistic amplitude coefficient. The logistic curve takes values between k and $2 - k$, which multiplied by Ef (the predator efficiency) gives the probability that the interaction ends in the predator eating the prey. If the condition holds, the predator eats and the prey dies and disappears from the population. Otherwise, the prey escapes. This process is repeated a predefined number of times (external parameter $N1$). At the end of this loop, all the predators in the population that were unable to reach a given number of "meals" (the external parameter $N2$) also die.

The following scheme shows the way in which GE combines traditional genetic algorithms with genotype-to-phenotype mapping.

1. An initial population of N genomes is generated at random. In our experiments, the value of N is a parameter, which can be set for each experiment run.
2. The phenotypes associated to all the members in the initial population are generated using a grammar. In our experiments, each genome is assigned an arbitrary id: a unique function number in the interval [000–N].
3. The genotype population is sorted according to fitness (computed from the phenotypes). In fact, as indicated before, what we usually call fitness is actually an "inverse fitness," as we consider best those functions that get the minimum results (our optimal fitness value is 0). In our experiments, the fitness of a function is defined by a mathematical expression, which can be different for different ecological niches. For instance, one of the (inverse) fitness functions we have used computes the following mathematical expression:

$$\left[\left(\sum|\Delta^4 Z|\right) + \left(\frac{1}{|\sum \Delta^3 Z|}\right)\right] \times c \tag{2}$$

where Z represents the result of applying the function associated to one individual to the input values (in our experiments, all the integers from 1 to 10). This fitness function is smaller (and, therefore, selects) for those mathematical expressions whose fourth difference is minimal and their third difference is maximal, (i.e., polynomials of degree 3). To prevent genome length

shooting up, long genomes are penalized (this is the meaning of the correction multiplier, c).

4. The individuals in the population are ordered by their fitness. In our experiments, this is done independently for all ecological niches, so that evolution takes place independently in each niche. All those individuals whose (inverse) fitness values are greater than 1000 are eliminated (with this limit, over half of purely random genomes are eliminated), together with their associated phenotype functions. This is done to prevent the population to be invaded by individuals with very bad fitness, leaving room for new offspring. In any case, once the population stabilizes with a reasonable fitness, only a small proportion of individuals are eliminated in this way (less than 5%), so the effect of this pruning is effective mainly during the first generations, before the population reaches a steady state.

5. Predator–prey interaction. Each predator (an individual belonging to the predator niche) is paired to a prey (one individual belonging to one of the different prey niches). The predator may "eat" the prey, or the prey may "escape" according to the procedure indicated above. The predator may also "starve," if it fails too much.

6. Create the next generation from a mating pool. In our experiments, the mating-pool is chosen from the 100 best-fitted individuals in the population (or those that remain, if they are less than 100), taken in equal numbers from the different niches, and the future parents in each niche are paired randomly. Four different genetic operations are applied to the offspring:

   • Single-point recombination of parent genomes. This operation is always performed.
   • Mutation (random change of a component of the genome). This operation is performed after recombination has taken place, with a high (80%) probability when the two parents are identical and a lower (p1) percent probability otherwise, to compensate the fact that recombination has no effect in that case. This is not the standard mutation procedure but it has been used before in genetic programming [15,21]. The first element of a genome can also mutate, which means that the offspring may belong to a different niche than their parents. This makes niche colonization after extinction possible.
   • Extension: with a certain percent probability (p2), a randomly selected part of the genome (from 0 to 100%) of one parent is added at the end of the offspring genome. This can happen in living beings in unequal crossing over [22], or when a genome suffers polyploidy and its genetic contents increases.
   • Shortening: with a certain percent probability (p3), one component of the offspring genome is deleted randomly. This can happen in living beings in unequal crossing over, or in a different way, when a genome loses one or more chromosomes and its genetic contents decreases.

7. The offspring genomes are added to the population. In our experiments, if the total number of individuals exceeds N_{max} (the maximum population size), the worst genomes in every niche of the previous population are eliminated (together with their phenotypes) until the number is N_{max} or less. The offspring genomes are associated with phenotype numbers that are or have become free after this operation.

8. The phenotypes associated to all the new members of the population are generated using the same grammar.

9. Go to step 3.

A phenotype is generated from a genotype in the following way:

1. Variable V is initialized with the axiom of the grammar, "E."

2. If V does not contain a nonterminal symbol, the process has finished and the value of variable V is the phenotype expression. If step 2 has been executed 500 times, the process finishes and returns an empty expression. Otherwise:

   a. Let Y be the first nonterminal symbol in V.
   b. Let K be the number of rules in the grammar, whose left part is Y. If $K = 1$, the only available right part replaces the first appearance of Y in V and step 2 is repeated. Otherwise:
   c. Let G be the next element of the genome under translation. If all the elements of the genome have been used, the first one is used again (genomes are circular).
   d. The first appearance of Y in V is replaced by the mth right part of the rule, whose left part is Y (numbered in zero origin), where $m = mod(G,K)$.
   e. Repeat step 2.

Appendix A provides some considerations about the way in which the system has been programmed. Appendix B shows an example of the translation of a genotype into a phenotype, as described by the previous algorithm.

3. ECOSYSTEM AND POPULATION PARAMETERS

In our experiments, we give values to the following external parameters:

- The initial and maximum sizes of the population, N and N_{max}. In all the experiments discussed in this article, these parameters have been set at 1000 and 2000, respectively.
- The number of ecological niches. Two cases were considered: four niches (one predator, three preys sharing the same fitness function), and four niches (one predator, two preys sharing the same fitness function, one prey with a different fitness function). In this way, the initial population of prey is initially three times larger than the initial population of predators, although the population of both stabilizes spontaneously at a different relationship.
- Genetic algorithm parameters controlling mutation, extension, and deletion rates are represented as p1, p2, and p3, respectively. All of them are percent probabilities with their values in the [0,100] interval.
- The predator efficiency Ef, which may vary in the [0,100] interval (it can also be considered as a percent probability).
- The two external predator–prey interaction parameters: $N1$ (number of trials each predator can try to eat a prey in each generation) and $N2$ (minimum number of successful trials).
- The set of values used as arguments for the phenotype functions. In all the experiments discussed in this article, this was a vector of integers from 1 to 10.
- The fitness functions used for each niche.
- The random seed defines the initial conditions of the experiment and affects all the random operations during its execution.
- Interbreeding between different niches was not allowed in these experiments.

4. STUDY OF TWO DETAILED EXPERIMENTS

In this section, we analyze in detail a couple of concrete experiments. The two experiments have been chosen because they illustrate well what happens during the evolution of our ecologies. They cannot be considered typical, however, as every experiment is different and provides interesting suggestions. However, we have performed so many experiments that it is impossible to discuss them all, so this section should be taken as just a sample.

In our first detailed experiment, we used the following parameters: initial population, 1000 individuals; maximum population, 2000 individuals; the random seed was 16,807. There were four niches (one predator and three preys sharing the same fitness function). The fitness functions used were:

- prey: fourth-degree polynomials
- predators: third-degree polynomials

We chose these functions arbitrarily but in such a way that their genetic distance is not large (for our definition of genetic distance, see [19]). In both niches, the polynomials with the largest absolute value were positively selected. Genomes of less than 50 elements are positively selected to prevent runaway genome length.

In each cycle, predators were allowed $N1 = 4$ tries to make a prey. Just one prey captured per cycle was sufficient to keep it alive. Predator efficiency was set at Ef = 24%.

Table 1 shows the dominant functions (with the best fitness) during the evolution of the ecosystem simulated in our first detailed experiment.

This ecosystem endured for 2565 generations and then was halted when the size of its predator population became zero. It could have proceeded with just the prey but this was not done in this case. Figure 1 shows the total size of the prey population as a function of time (generation number), as well as the number of predators. Figure 2 shows a typical Lotka–Volterra plot for a section of the experiment.

Looking at Table 1 and Figures 1 and 2, the following facts can be observed[1]:

- At the beginning of the experiment, when it is generated, the initial population of 1000 individuals is divided equally between the four niches. As all those with a fitness worse (greater) than 1000 are automatically eliminated, the total initial population (136) is smaller. The initial prey/predator relation (114/22 = 5.2) is not significantly greater than 3 (the relation between prey/predator niches).
- In a few generations, a stable equilibrium is reached with a much larger prey/predator relation (the average for the complete experiment was 58). This is exclusively due to the predator/prey interaction. Without it, the relation would stabilize at 3.
- Figure 2 shows the evolution of the prey versus the predator populations in this experiment during generations 121–165. It will be observed that temporary increases in the number of predators coincide with temporary decreases in the number of prey, giving rise to curves somewhat similar to the results of the Lotka–Volterra equations, where a circular shape would have been obtained. This, however, only happens during a certain number of generations, for those equations are applicable to two-species

---

[1]These observations must not be considered as generalized conclusions, since no statistical analysis has been performed on them. They are just interesting remarks which suggest that our experiments do not differ too much in their behavior from biological ecosystems, at least at first sight.
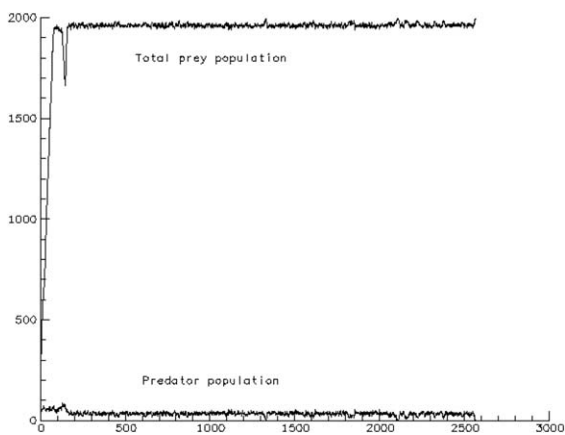
Evolution of the Dominant Functions in the First Experiment

Generation	Prey Best Function	Fitness	Population	Predator Best Function	Fitness	Population
0	$1/x$	0.84	114	$\log_8\left(x+\frac{4!}{x}\right)$	1.25	22
50	$-8-8x\log\ x$	0.3	1368	$\frac{3}{x}$	0.35	47
100			1943	$-x^2\ln x$	0.19	52
150	$-64x^4$	8e−11	1719	$x(x-8\pi x^2)$	2e−4	61
250	$(-8x.8!)^4$	1.3e−25	1978			20
800			1859	$x^3.8!$	1.2e−7	36
850			1966	$-8\pi x^3$	2e−4	29
950			1965	$x(x-9\pi x^2)$	1.7e−4	28
1250	$8.(8x.8!)^4$	1.6e−26	1957			34
1300	$9.(8x.8!)^4$	1.4e−26	1962			35
2000			1943	$9\pi x^2(9-x)$		55
2100			1982	$9\pi x^2.(9-\frac{3}{2}x)$	1.1e−4	14
2350			1967	$9\pi x^2.(9-\frac{8}{3}x)$	6e−5	28
2500			1965	$9\pi x^2.(9-8x)$	2e−5	26
2565			1994			0

ecologies in evolutionary equilibrium, while in our system evolution changes the mixture of species and the situation is different [23].

- The data shown by Table 1 can be interpreted as a predator–prey arms race, where sometimes the predators, sometime the prey, experience significant fitness improvements that give them a visible advantage. A little after generation 120, predator fitness went down from 0.19 to 2e−4, which gave them a great advantage against their prey. The effect on the respective populations is clear in Figure 1:

the prey descended to below 1700, the predators increased to about 80. A little before generation 150, however, the prey discovered a new mathematical function that gave them a tremendous fitness improvement (from 0.3 down to 8e−11), which allowed them to recover their previous population and even reduce somewhat the predator population. The arms race continued during most of the life of the ecosystem,

- Around generation 800, an interesting event is visible in Table 1: predators improve their fitness

FIGURE 1

Results of the first experiment: prey/predator populations as a function of time.

FIGURE 2
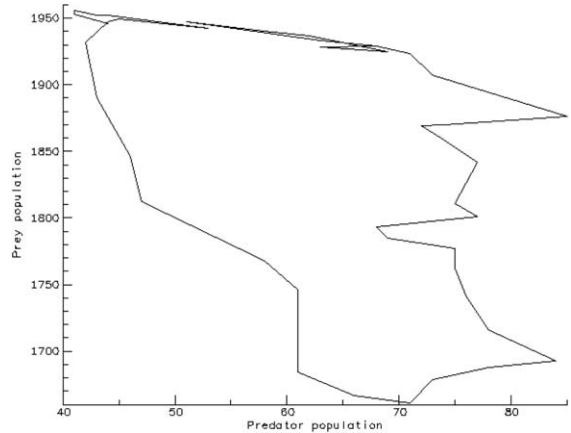
A Lotka-Volterra like plot of prey population versus predator population.

significantly (in fact, they reach their best fitness in the whole experiment, 1.2e −7) but in a few generations they go back to their previous best fitness (2e −4). What happened was: before that improved genome could spread to the whole population of predators, chance made them fail to get a prey and all of them died without leaving descendants. The next best genome then became again dominant for predators.

- The evolution of this ecosystem seems to favor Stephen Jay Gould's theory of punctuated equilibrium evolution [24,25]. It can be seen that during long stretches of time (as between generations 250 and 800, or between 1300 and 2100) there were no improvements in fitness. At other times, however (as between 100 and 250, or between 800 and 950), several consecutive improvements in the genome happen one after another in a short stretch of time.

In a similar, but different experiment, with the same parameters, except for $N1 = 5$ and Ef = 14%, the situation where predators disappeared but prey remained was tested. After 46 generations where the predator niche was empty, this niche was invaded by one individual, descendant from parents of one of the prey niches, which underwent a mutation in the gene defining the niche. It so happened that, in that experiment, the best prey individuals had a mathematical function that maintained a comparable fitness when transplanted to the predator niche. As a consequence, both the predator and the prey niches were occupied during some time (349 generations) by the same species. In other words, when predators disappeared, some of the prey developed cannibalism. Finally, however, the predators again became extinct.

In our second detailed experiment, we used the following parameters: initial population, 1000 individuals; maximum population, 2000 individuals; random seed, 16,807. There were four niches (one predator, three preys, two of them sharing the same fitness function, the other with a different fitness function). The fitness functions selected for were:

- prey 1 (one niche): exponential functions
- prey 2 (two niches): fourth-degree polynomials
- predators: third-degree polynomials

In this case, the fitness functions were chosen in such a way that the genetic distance [19] of the new prey (prey 1) to both the old prey and the predator would be large, keeping the other two niches identical to the preceding experiment. In all cases, the functions with the largest absolute value were positively selected. Genomes of less than 50 elements were positively selected to prevent runaway genome length.

In each cycle, predators were allowed $N1 = 4$ tries to make a prey. Just one prey per cycle was sufficient to keep it alive. Predator efficiency was set at Ef = 25%.

Table 2 shows the dominant functions, which reached the maximum fitness during the evolution of the ecosystem simulated in our second detailed experiment. In this ecosystem, prey failed first after an interesting three-sided arms race. Of course, once the prey disappeared, the predators failed also in the next generation (their fall had begun before, when the prey started to be scarce). Figure 3 shows the size of the three populations (prey 1, prey 2, and predators) as a function of time (generation number).

Looking at Table 2 and Figure 3, the following facts can be observed:

- At the beginning of the experiment, the initial population of 1000 individuals is divided equally between the four niches. As all those with fitness greater than 1000 are automatically eliminated, the total initial population (222) is much smaller. However, due to the difference between their fitness functions, the number of prey 1 individuals is about double than the initial number of prey 2, in spite of the fact that the latter occupy two niches. The initial prey/predator relation (200/22 = 9) is greater than 3 (the relation between prey/predator niches) but not too much, especially if we compare the predator and prey 2 populations, which have similar fitness functions (polynomials).

- In a few generations, a stable equilibrium is reached with prey 2 systematically maintaining a double population to prey 1 (they occupy two niches) and a larger prey/predator relation (the average for the complete experiment was 31). Observe, however, that this relation is significantly smaller than in the previous experiment. Apparently, the diversification of the prey in two different niches allows predators to reach a higher population. In fact, the set of all the experiments performed[2] with three different species appears to be more stable than the previous set, with just two species, which corresponds to the well-known biological equivalent, that an ecosystem is more stable when it contains a greater number of species.

- The final failure of the prey was obviously due to the persistent and overwhelming improvement of predator fitness in the last 200 generations. It should be noticed that predator fitness had to get better than the fitness of both prey niches before forcing them to disappear. In some way, each of the prey species seems to stabilize the population of the other.

- The evolution of this ecosystem also seems to favor Stephen Jay Gould's theory of punctuated evolution. During two long stretches of time (between genera-

---

[2]Not this particular experiment, which has a shorter duration than the previous one.

Evolution of the Dominant Functions in the Second Experiment

| Gener.	Prey 1 Best Function	Fitness	Pop.	Prey 2 Best Function	Fitness	Pop.	Predator Best Function	Fitness	Pop.
0	9^x	5e −6	132	$5/x$	9.0	68	$\log_8\left(x+\frac{!4}{x}\right)$	1.25	22
50	9^x	2e −7 + 6e −11	541	$5/x$	0.36	1053	$\binom{x}{3}$	0.029	61
100			648	$\frac{-4x!}{x}$	0.358	1298			53
200			648	$\frac{-3x!}{x}$	0.34	1283			67
450			648	$x+(-3)^{\frac{x}{6}}$	0.33	1288			63
500			647	$x+(3\pi)^{\frac{x}{6}}$	0.24	1285			67
800	9^{2x+6}	2e −7 + 3e −26	649			1293			57
850			650	$x+e^{-x}+(3\pi)^{\frac{x}{6}}$	0.22	1293	$\binom{\pi x}{3}$	0.0009	54
900	9^{2x+7}	2e −7 + 3e −27	647			1287	$\binom{6\pi x}{3}$	4.4e −6	61
1000			649			1290	$\binom{8\pi x}{3}$	2.2e −6	57
1050			649			1290	$\binom{13\pi x}{3}$	2e −6	59
1100	9^{2x+8}	2e −7 + 4e −28	651			1296			52
1400			649	$x^3.8x$	1.7e −4	1293			55
1450			644	$x^3.64x$	2e −5	1277			70
1500			648			1291	$\binom{72x}{3}$	8e −8	54
1550			559	$x^3.90x$	1.5e −5	1203	$\binom{90x}{3}$	4e −8	69
1600			250			496	$\binom{162x}{3}$	7e −9	80
1623			0			0			9

tions 500 and 800, and again between 1100 and 1400), no improvements in fitness took place. At other times, however (as between 1500 and 1600), several consecutive improvements in the genome happen one after another in a short stretch of time.

In a different but similar experiment, we analyzed during 20 generations, which individual prey were eaten by the predators and how many predators starved. The average per generation was:

Prey 1 eaten: 13.4
Prey 2 eaten: 32.1
Starved predators: 16.8

Predators, therefore, seemed to have a certain preference for prey 2. In this population, the prey were distributed at 33.3 and 66.7%, while predators ate them at 29.5 and 70.5%, respectively. While this difference is not very significant, it may be due to the fact that the fitness of
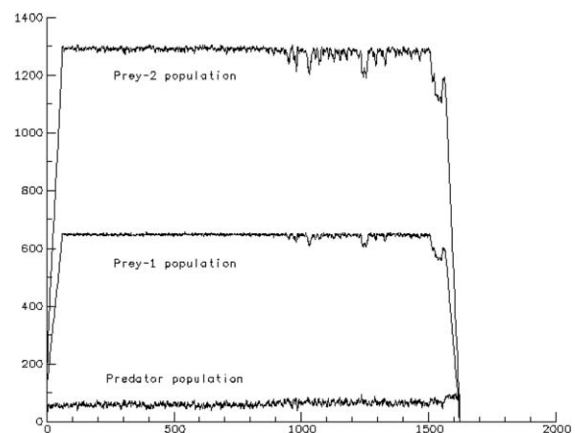
Results of the second experiment: prey/predator populations as a function of time.

prey 1 was usually better than the fitness of prey 2, at least in these experiments (there are exceptions).

5. SENSITIVITY ANALYSIS

In all the experiments described next, performed to study the influence of the different genetic and ecologic parameters on the stability and diversity of the results, the following ranges and basic values were used:

- Mutation rate when the two parents are different. Range: [0%, 80%]. Basic value: 10%. The mutation rate when both parents are identical was kept always at 80%, which forces the maximum value indicated above.
- Extension. Range: [0%, 100%]. Basic value: 5%.
- Shortening. Range: [0%, 100%]. Basic value: 5%.
- Logistic amplitude coefficient k, see Eq. (1). Range: [0,1]. Basic value: 0.75.

From all the experiments performed, those where predators or prey withstood for 200 generations or less were discarded, as they gave rise to very unstable ecologies. We call "successful" those experiments where both predators and prey went beyond 200 generations.

Experiments were performed in batches, sharing the values of all external parameters except predator efficiency, which happens to be a relatively critical parameter. It was discovered that predator efficiency only gives rise to successful experiments in a small part of its range of variation, which depends on the values of the remaining parameters. Thus, the [0,100] interval of possible values gets divided into three sections:

- The [0,a) interval, where predator efficiency is too small and the predator population disappears in no more than 200 generations.
- The [a,b] interval, where predator efficiency is sufficient, but not too large, and one or more experiments endure for more than 200 generations.
- The (b,100] interval, where predator efficiency is too large, and the prey population disappears in no more than 200 generations (immediately followed by the predators, of course).

For every set of external parameter values, the [a,b] interval of predator efficiency was discovered, and all the experiments in that interval for integer values of the parameter were performed. This is what we call a "batch of experiments," where the [a,b] interval rarely contains more than 20 different integer values. In all the results presented below, averages and standard deviation were computed for batches.

In each of the next subsections, we made two types of experiments: first, we tried to find what happens when different constant values are assigned to the parameter under study during a complete batch of experiments. Then we tested the effect of making the parameter variable during the execution of particular experiments. The first type let us deduce how changing the value of some parameter affects the evolution of the ecosystem. The second type tells us whether that parameter should actually be constant or if a certain time dependency can be allowed. While each of the parameters was modified, the basic values were used for all the other parameters.

We performed a total of 426 successful experiments. In all the statistical measurements performed, the first 15 generations were excluded, to allow the ecosystem to go into a permanent regime. In this analysis, we measured the following results:

1. **Successful experiments**: We measured the number of successful experiments, the number of those that exceeded 2000 generations, and the average number of generations in each batch. If a successful experiment exceeded 2000 generations, it was interrupted and its total duration was computed as 2000.
2. **Diversity:** Biological populations are almost never genetically identical, they embody a certain degree of variation, even when they belong to a single species. Among different ways to measure biodiversity, the Shannon diversity index [26–28] is frequently used. This index is defined by the following formula:

$$-\sum_{i=1}^{n} p_i \log p_i \qquad (3)$$

where n is the number of different species and p_i is the frequency of species number i (the number of individuals belonging to that species divided by the total number of individuals).

To study the evolution of diversity in our simulation experiments by means of Shannon's diversity index, we group the individuals in "species." Two individuals belong to the same "species" when their phenotypes are identical, even though their genotypes may not be, due to the redundancy of the genetic code. This accords with the fact that the current definition of biological species is mainly based on a common genome but takes into account that the genetic code is redundant. Phenotypes are considered identical when the mathematical expressions in their phenotype functions are identical. Expressions that always give rise to the same values, but are not identical, are considered to belong to different species. For instance, $(2X)^3$ and $8X^3$, are considered different species, even though their results (and, therefore, their fitness value) are always the same.

We computed three different measures of diversity: maximum diversity, average diversity, and maximum

number of species during the experiment (all the results in the tables are averaged for batches of experiments).

3. **Additional population results:** Average prey/predator population quotient and average number of predators.

5.1. Effect of Changing the Mutation Rate

Table 3 shows the global results of 180 experiments performed varying the mutation rate and the predator efficiency. For each fixed mutation rate between 0 and 80%, 20 experiments were performed for 20 different values of the predator efficiency. In all the successful experiments in this set, predator efficiency belonged to the interval [18,31]. For all values of the efficiency outside this interval, there were no successful experiments at all.

The following behavior can be observed in the table:

- The number of successful experiments (those that endured over 200 generations) is practically independent of the mutation rate. Conversely, the number of stable experiments (over 2000 generations) decreases slightly as the mutation rate increases (specially at the beginning). The same effect can be seen in the average duration of the experiments, where experiments reaching 2000 generations have been assigned that duration, although in fact they would have endured longer, if allowed to proceed. Therefore, this parameter is not critical, as its viability range coincides with its possible range, although stability diminishes progressively.
- The diversity of the experiments (measured in the three ways indicated in the previous subsection) increases clearly with the mutation rate, although it remains stable between 20 and 50% (see also Figure 4, where the error bars show the standard error in the samples). This effect is easy to see in the three diversity measurements we are considering: maxi-

mum diversity, average diversity, and maximum number of different species.

- The average prey/predator population quotient, and the average number of predators, seem to be independent from the mutation rate.

Table 4 shows the global results of 80 experiments performed as the preceding ones, with a time variable mutation rate, a sinusoid between two extremes differing by 20%, with a period of 314 generations. For each variable mutation rate, 20 experiments were performed for 20 different values of the predator efficiency. The result of each batch of experiments is compared with the average of the three experiments with fixed mutation rate corresponding to each variable case, obtained from Table 3. In this case, predator efficiency for all successful experiments belonged to the interval [18,33].
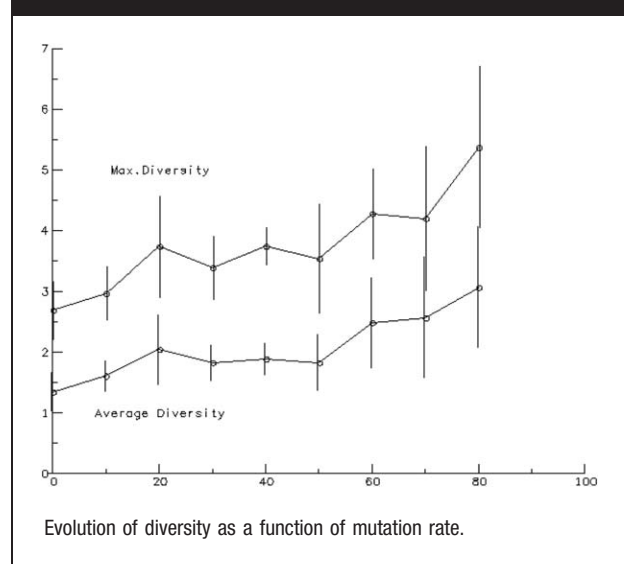
FIGURE 4

Evolution of diversity as a function of mutation rate.

Mut. Rate	>200 Gen.	>2000 Gen.	Ave. Duration	Max. Divers.	Ave. Divers.	Max. Species	Prey / Preds.	Ave. Nr. Preds.
0−20	12	9	1793	3.07	1.58	45.0	44.5	45.3
Fix. ave.	11.3	7.7	1577	3.13	1.67	44.7	46.7	46.4
20−40	10	6	1452	3.27	1.91	42.9	42.7	47.0
Fix. ave.	10.3	6.0	1519	3.62	1.92	61.7	49.9	41.8
40−60	12	6	1468	3.39	1.74	60.3	47.8	43.3
Fix. ave.	11.0	6.3	1546	3.85	2.07	79.8	45.8	44.3
60−80	13	6	1199	4.14	2.32	119.8	48.2	43.8
Fix. ave.	10.0	6.3	1537	4.62	2.71	157.2	47.2	42.6
Var. ave.	11.8	6.8	1477.8	3.47	1.89	67.0	45.8	44.8

The following behavior can be observed in the table:

- The performance of the experiments with a variable mutation rate was quite similar to those experiments performed with fixed rates (the correlation coefficient for the numbers of experiments that reached 2000 generations is 0.98).
- Although they show the same general increase with mutation rate, the three diversity measurements gave slightly smaller values than the corresponding fixed rates.

5.2. Effect of Changing the Shortening Rate

Table 5 shows the global results of 160 experiments performed varying the genomic shortening rate and the predator efficiency. For each fixed shortening rate between 0 and 100%, 20 experiments were performed for 20 different values of the predator efficiency. In this set, predator efficiency for all successful experiments belonged to the interval [19,33].

The following behavior can be observed in the table:

- The number of successful experiments (those that endured over 200 generations) and the number of stable experiments (over 2000 generations) diminish abruptly when the shortening rate increases, until they become zero for a 100% rate. Therefore, this parameter is semicritical, as its viability range is limited at one side of its possible range. The average duration of the experiments, however, does not seem to depend on this parameter or shows a slight decrease.
- The diversity of the experiments decreases clearly when the shortening rate increases, although it remains stable between 5 and 20% (see also Figure 5, where the error bars show the standard error in the samples). This effect is easy to see in our three diversity measurements: maximum diversity, average diversity, and maximum number of different species.
- The average prey/predator population quotient seems to decrease somewhat, while the average number of predators increases slightly, when the shortening rate increases.

Table 6 shows the global results of 80 experiments performed as the preceding ones, with a time varying shortening rate, a sinusoid between two extremes differing by

Short. Rate	>200 Gen.	>2000 Gen.	Ave. Duration	Max. Divers.	Ave. Divers.	Max. Species	Prey / Preds.	Ave. Nr. Preds.
0	12	9	1612	3.43	1.89	52.2	48.8	42.5
5	11	7	1609	2.97	1,61	39.4	47.7	47.0
10	10	9	1937	2.97	1.52	33.5	43.1	47.2
20	11	8	1686	2.98	1.56	34.5	48.2	42.5
40	7	3	1655	2.46	1.41	19.6	38.6	54.7
60	5	1	968	2.67	1.71	24.8	54.8	42.4
80	2	2	2000	2.39	1.20	17.5	31.5	60.0
100	0	0						
Ave.	7.3	4.9	1638	2.84	1.56	31.6	44.7	48.0
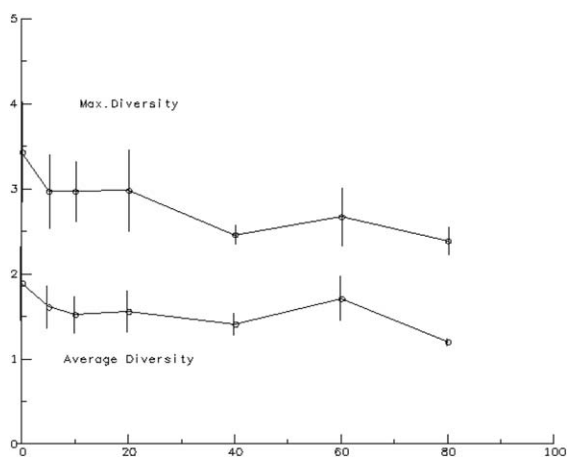
Evolution of diversity as a function of shortening rate.

20%, with a period of 314 generations. For each variable shortening rate, 20 experiments were performed for 20 different values of the predator efficiency. The result of each batch of experiments is compared with the average of those experiments with fixed shortening rate corresponding to each variable case, obtained from Table 5. In this case, predator efficiency for all successful experiments belonged to the interval [19,34].

The following behavior can be observed in the table:

- The performance of the experiments with variable shortening rate was quite similar to those experiments performed with fixed rates (the correlation coefficient for the numbers of experiments that reached 2000 generations is 0.99).
- Although they show the same general decrease with shortening rate, the three diversity measurements gave slightly smaller values than the corresponding fixed rates.

5.3. Effect of Changing the Lengthening Rate

Table 7 shows the global results of 160 experiments performed varying the genomic lengthening rate and the predator efficiency. For each fixed lengthening rate between 0 and 100%, 20 experiments were performed for 20 different values of the predator efficiency. In this set, predator efficiency for all successful experiments belonged to the interval [19,30].

The following behavior can be observed in the table:

- The number of successful experiments (those that endured over 200 generations) remains practically constant for all lengthening rates. The number of stable experiments (over 2000 generations) diminish abruptly toward the end of the lengthening rate range. Therefore, this parameter is not critical, as its viability range coincides with its possible range, although stability diminishes at the end. The average duration of the experiments decreases slightly with the increase of this parameter.
- The diversity of the experiments increases clearly when the lengthening rate increases (see also Figure 6, where the error bars show the standard error in the samples). This effect is easy to see in our three diversity measurements: maximum diversity, average diversity, and maximum number of different species.
- The average prey/predator population quotient seems to increase somewhat, while the average number of predators decreases slightly, when the lengthening rate increases.

Table 8 shows the global results of 100 experiments performed as the preceding ones, with a time varying lengthening rate, a sinusoid between two extremes differing by 20%, with a period of 314 generations. For each variable lengthening rate, 20 experiments were performed for 20 different values of the predator efficiency. The result of each batch of experiments is compared with the average of those experiments with fixed lengthening rate

Effect of Different Variable Shortening Rates on the Results of Experiments

Short. Rate	>200 Gen.	>2000 Gen.	Ave. Duration	Max. Divers.	Ave. Divers.	Max. Species	Prey / Preds.	Ave. Nr. Preds.
0−20	13	8	1684	3.02	1.53	35.9	45.7	45.1
Fix. ave.	11	8.3	1711	3.09	1.64	39.9	46.9	44.8
20−40	9	6	1634	2.66	1.52	27.6	41.1	49.2
Fix. ave.	9	5.5	1670	2.72	1.48	27.0	43.4	48.6
40−60	3	3	2000	2.35	1.37	19.3	33.3	57.3
Fix. ave.	6	2.0	1312	2.56	1.56	22.2	46.7	48.6
60−80	4	2	1409	2.22	1.39	15.3	37.0	54.0
Fix. ave.	3.5	1.5	1484	2.53	1.45	21.2	43.2	51.2
Var. ave.	7.3	4.8	1682	2.56	1.45	24.5	39.3	51.4

Length. Rate	>200 Gen.	>2000 Gen.	Ave. Duration	Max. Divers.	Ave. Divers.	Max. Species	Prey / Preds.	Ave. Nr. Preds.
0	11	8	1621	2.66	1.51	24.6	40.5	48.4
5	11	7	1609	2.97	1.61	39.4	47.7	47.0
10	10	6	1545	2.99	1.58	40.0	46.3	44.6
20	10	7	1736	3.11	1.62	50.8	45.5	44.2
40	11	7	1559	3.67	1.96	82.0	50.9	40.5
60	11	3	1329	3.65	1.85	75.2	52.2	38.5
80	10	6	1477	4.12	2.28	128.7	52.0	38.1
100	8	1	1048	4.61	2.78	165.8	55.1	36.5
Ave.	10.3	5.6	1490	3.47	1.90	75.8	48.8	42.2

corresponding to each variable case, obtained from Table 7. In this case, predator efficiency for all successful experiments belonged to the interval [19,30].

The following behavior can be observed in the table:

- The performance of the experiments with variable lengthening rate was somewhat similar to those experiments performed with fixed rates, with a slightly smaller average duration: the correlation coefficient for the numbers of experiments that reached 2000 generations is 0.51 in this case.
- The three diversity measurements gave values very similar to the corresponding fixed rates.

5.4. Effect of Changing the Amplitude of the Logistic Predator–Prey Curve

Table 9 shows the global results of 140 experiments performed varying the amplitude of the logistic curve defining the predator–prey interaction and the predator efficiency. For each fixed amplitude of the logistic curve between 0 and 1, 20 or more experiments were performed for different values of the predator efficiency. The efficiency interval where successful experiments happen is strongly affected by this parameter. Thus, for the amplitude between 0.75 and 1, the interval where successful experiments appear is the same as in the previous analysis, around [17,30]. For 0.5, however, the interval of stability moves to [20, 38]; for 0.25, to [33,60]; for 0.1, as indicated in the table, just two experiments were successful, with predator efficiencies equal to 71 and 96. So it appears that a smaller value of the amplitude moves the interval up, but a very small value destabilizes the system.

The following behavior can be observed in the table:

- The number of successful experiments (those that endured over 200 generations) is zero at one extreme of the range of variation, grows to a maximum between 0.25 and 0.5, and decreases again slightly between 0.75 and 1. The number of stable experiments (over 2000 generations) remains constant in the viable range. Therefore, this parameter is semicritical, as its viability range is limited at one side of its possible range. The average duration of the experiments, however, increases regularly with the logistic-curve amplitude.
- The diversity of the experiments remains practically constant for all the viable values of the logistic-curve amplitude. This effect can be seen in our three diversity measurements: maximum diversity, average diversity, and maximum number of different species.
- The average prey/predator population quotient and the average number of predators seem to be independent on the logistic-curve amplitude.
- The last row in the table shows the result of one batch of experiments performed with a predator–prey logistic-curve with time-dependent amplitude,
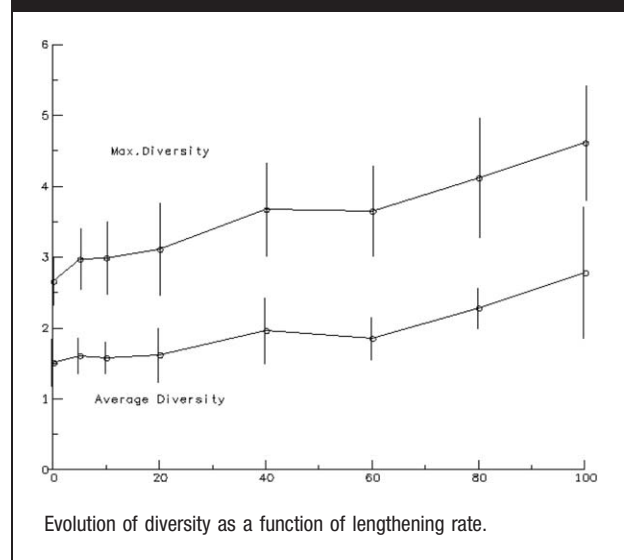
FIGURE 6

Evolution of diversity as a function of lengthening rate.

varying with a period of 314 generations between 0.25 and 0.75. It can be seen that this variability reduced the stability of the viable experiments, while the other measurements (diversity and populations) remained comparable with the fixed cases.

5.5. Effect of Changing Three Parameters at the Same Time

A final batch of experiments was performed to find whether the ecosystem remained viable when three of the parameters vary at the same time. Those parameters were chosen among those that display a critical range:

- The shortening rate was varied with a period of 314 generations between the values of 0 and 20%.
- The logistic-curve amplitude was varied with the same period, in phase with the preceding parameter, between 0.25 and 0.75.
- The predator efficiency was varied by 0.001 with the same period, but in phase opposition with both preceding parameters. This parameter is much more critical than the previous ones. Several batches of

experiments had to be performed before we found a time-dependent predator efficiency that would give rise to viable experiments. This was not possible if the efficiency varied by 0.01 or more, but a maximum variation of 0.001 gave rise to results comparable to the experiments performed with a fixed efficiency.

Table 10 shows the results of this batch of 20 experiments, compared with the results of the fixed case and those where only one parameter was variable.

Apparently, the variability of the parameters does not affect much the different measurements, although the stability of the triple variable case is somewhat smaller and the diversity slightly higher.

6. CONCLUSIONS

In this article, we have designed a procedure that generates artificial predator–prey ecologies that exhibit many of the features of natural evolution, among them the following:

TABLE 10

Effect of Varying Three Critical Parameters at the Same Time

Params.	>200 Gen.	>2000 Gen.	Ave. Duration	Max. Divers.	Ave. Divers.	Max. Species	Prey / Preds.	Ave. Nr. Preds.
Fixed	11	7	1609	2.97	1.61	39.4	47.7	47.0
Var. short.	13	8	1684	3.02	1.53	35.9	45.7	45.1
Var. log.amp.	13	3	826	2.81	1.70	29.7	50.8	42.7
Var. pred.eff.	12	10	1763	3.09	1.53	35.5	43.1	47.3
3 var. parms.	11	5	1229	3.28	1.92	48.9	46.3	44.4

Rows represent: all fixed parameters; variable shortening of genomes; variable amplitude of logistic curve for the predator−prey interaction; variable predator efficiency; and all three parameters variable at the same time

- A Volterra-like relationship between predators and prey, as shown by the cycle displayed in Figure 2. Although this figure corresponds to a part of a single experiment, cycles of this kind also appear in other experiments, although the fact that our ecologies are not in equilibrium makes this situation unusual.
- The prey/predator population relationship stabilizes automatically between one and two orders of magnitude, as in biological populations.
- Differentiation of the prey in more than one species increases the stability of the ecosystem.
- When the predator niche becomes empty, it can be colonized by the offspring of a prey niche, which sometimes gives rise to something similar to cannibalism. With the Lotka–Volterra equations this cannot happen: when the predator disappears, the prey becomes extinct too, because it proliferates too quickly and exhausts all its sources of food. In our system, the fact that we set a maximum population eliminates this effect, therefore, a single-niche prey ecosystem can endure indefinitely. As the Lotka–Volterra equations are simplifications that apply to ecological systems in evolutionary equilibrium, they should not be taken as the absolute standard of comparison. In real biological systems, niche colonization by nearby niches undoubtedly happens [27].
- Although our results are still too few, and no systematic analysis has been made, they seem to favor S. J. Gould theory of punctuated evolution [24] rather than phyletic gradualism [29].

We believe that these results provide some support for the hypothesis that some of the features displayed by biological evolution may depend to some extent on chance modifications of the genome plus natural selection, rather than on the particular form adopted by the phenotypes. Of course, in living beings things are much more complicated, and the genotype–phenotype relationship is not one-sided, as in our simplified experiments, therefore, this hypothesis may well be a too far-fetched extrapolation. However, the fact that phenotypes as different as mathematical functions and biological beings give rise to a few similar features seems to support the idea that some at least of these features may be a consequence of the mechanism, rather than of the actual form taken by the phenotype.

An interesting question that may be raised in this respect is the following: what is the significance of these simulation experiments? Are they a mere metaphor, or do they provide us with ideas that we can use in the study of real ecological systems? We believe they are something more than a metaphor: they can become a working analogy, with the potential to teach useful concepts applicable to real life.

To perform our experiments, we have used the following ideas:

- GE, which separates genomes from phenotypes (this is a standard technique in genetic programming).
- Individual genotypes are represented by means of mathematical expressions. Fitness functions become simple mathematical tests on those expressions. Mathematical expressions (using lambda-calculus, rather than APL2, without GE) have been used before in artificial life experiments [30,31]. In principle, lambda-calculus and APL2 should be equivalent for the representation of mathematical functions.
- Simultaneous evolution of several "niches" is attained by means of changes in the genome interpretation (the first element selects the niche), and using several fitness functions (one per niche).
- Predators are represented as individuals belonging to one of the available niches which prey on those in the other niches. At the end of every generation, predators try to eat prey in several bouts. If they do not reach a minimum number of captures (usually one in our experiments) they die. Prey eaten also dies, obviously. The result of each predator–prey

encounter depends on their respective fitness and also on chance. In each bout, predators are offered available prey chosen from those with less fitness. Predators with highest fitness are offered the prey with least fitness. This procedure does not require the division of the ecological space in discrete areas with space coordinates.

We have performed a sensitivity analysis by modifying the different parameters of the genetic algorithm and got the following results:

- Increasing the mutation rate does not affect the number of successful experiments, the average prey/predator population quotient and the average number of predators, but it decreases their stability and increases diversity. A variable mutation rate produces results comparable to a fixed rate equal to its average.
- Increasing the shortening rate affects negatively the number of successful experiments and their stability. This parameter is semicritical, as its viability range is limited at one side of its possible range. Diversity also decreases. A variable shortening rate produces results comparable to a fixed rate equal to its average.
- Increasing the lengthening rate does not affect the number of successful experiments, but decreases their stability. Diversity increases. A variable shortening rate produces results comparable to a fixed rate equal to its average.
- The amplitude of the logistic curve used to compare fitness in the predator–prey interaction is a semicritical parameter, as the number of successful experiments drops to zero at the lower end of the scale. Their stability, however, increases slightly or remains constant. Diversity also remains constant. A variable logistic curve amplitude reduces stability but keeps the other measurements the same.
- Predator efficiency is a very critical parameter. Making it variable even very slightly (by 0.01 or more) reduced to zero the number of successful experiments.
- Making the three critical parameters (shortening, logistic curve amplitude, and predator efficiency) variable at the same time, without leaving their viability region, slightly decreases the stability of the experiments and increases their diversity.

This analysis will help us focus on the best performing values of the parameters, which will speed-up our future experiments. It can also help to detect, which are the critical parameters in real ecological systems.

In the future, we intend to explore the following issues:

- To compare our approach using GE with a similar implementation using more traditional genetic algorithms

- To measure the relative ease with which the offspring generated during our experiments can migrate from one niche to another, depending on the genetic distance of their fitness functions
- To study the effect of predator species being specialized to a certain prey niche
- To study the effect of having two predator species that compete with one another
- To study the effect of having three or more different prey species
- To analyze the effect of using different fitness functions for the predator/prey ecological niches
- Our fitness function is currently absolute: the same function is used during the whole program runs. In the future, this function can be made relative, so as to make more realistic the competition between individuals and niches, and the arms race between predator and prey
- To perform a more complete analysis of the apparent emergence of punctuated equilibrium in this set of experiments
- To perform a similar sensitivity analysis with our parasite–host experiments described in [19], and to compare the results to our predator–prey experiments described in this article

APPENDIX A: PROGRAMMING CONSIDERATIONS

Both the expressions and the GE environment are written in the APL2 language [32], which has been selected as the language of choice for the following reasons:

- APL2 is a very powerful language, especially for the generation of expressions, with a large number of primitive functions and operators available.
- The APL2 expression grammar is very simple and can be implemented with just four nonterminal symbols, which makes the GE process simpler.
- APL2 instructions can be protected to prevent semantic and execution errors giving rise to program failures. In this way, we can rest assured that all the expressions associated to the different individuals will execute, although their results may not correspond to a good fitness. The GE technique also becomes simpler thanks to this feature, because it is not necessary to include any semantic information.
- Being an interpretive language, APL2 makes it possible to create programming functions at execution time, thus providing the feasibility of computing fitness during the execution of the genetic algorithm. With a compiling language such as C or C++, this would be very difficult.

A phenotype is an APL2 function of the following form:

[0] $Z \leftarrow Fnnn\, X$

[1] $Z \leftarrow (\rho X)\rho 0$

[2] $\rightarrow (5 < \rho \square LC)/0$

[3] $''\square EA\, 'Z \leftarrow APL2\_expression'$

Only the APL2 expression in line 3 is generated from the genome. The remainder of the functions is the same for all.

- Line [0] defines a monadic function with explicit result, called *Fnnn*.
- Line [1] assigns to the function result a vector of zeros.
- Line [2] stops the execution of the function if function call depth is greater than five (this eliminates infinite recursion).
- Line [3] executes the expression generated from the genome and, if no error is detected, returns its value as the result of the function. Otherwise, a result of all zeros is returned (this is what line [1] is for).

The following fitness expression selects for third degree polynomials in APL2 notation:

$$\left((+/|2-/2-/2-/2-/Z)+\div|+/2-/2-/2-/Z\right)$$
$$\times(.2\,5\,1000)[+/50\,1000 < \rho X]$$

where *X* is the genome of the individual. This is the same expression represented in common mathematical notation by Eq. (2).

The grammar describing APL2 expressions is used to generate a phenotype from a genotype using GE:

$E ::= 0 \mid o0 \mid 0o0$ (mathematical expression)

$0 ::= N \mid X \mid (E)$ (operands)

$o ::= + \mid - \mid \times \mid * \mid \div \mid \bigcirc \mid \Gamma \mid L \mid \otimes \mid ! \mid |$ (operators)

$N ::= 0|1|2|3|4|5|6|7|8|9$ (digits)

where *X*, the digits and operators $\{+, -, \times, *, \div, o, \Gamma, L, \otimes, !, |\}$ are the terminal symbols of the grammar, while $\{E, O, o, N\}$ are the nonterminal symbols, or variables, that is, intermediate symbols that will transform into other symbols using one of the indicated rules. Table 11 shows the functions that the APL2 operators compute.

APPENDIX B: EXAMPLE OF GENOTYPE TO PHENOTYPE CONVERSION

Let the genome be [89,40,58,130]. In step 1, we start with *V* = "E."

1. The first nonterminal symbol in *V* is *E*. The number of right parts of the rule with left part *E* is *K* = 3.

The next element in the genome is *G* = 89. Therefore, $n = \text{mod}(89,3) = 2$. The second right part (in zero origin) for the rule with left part *E* is OoO. We replace *E* by OoO in *V*. After this step, *V* = "OoO."

2. The first nonterminal symbol in *V* is *O*. The number of right parts of the rule with left part *O* is *K* = 3. The next element in the genome is *G* = 40. Therefore, $n = \text{mod}(40,3) = 1$. The first right part (in zero origin) for the rule with left part *O* is *X*. We replace *O* by *X* in *V*. After this step, *V* = "XoO."

3. The first nonterminal symbol in *V* is *o*. The number of right parts of the rule with left part o is *K* = 11. The next element in the genome is *G* = 58. Therefore, $n = \text{mod}(58,11) = 3$. The third right part (in zero origin) for the rule with left part *o* is *. We replace *o* by * in *V*. After this step, *V* = "X*O."

4. The first nonterminal symbol in *V* is *O*. The number of right parts of the rule with left part *O* is *K* = 3. The next element in the genome is *G* = 130. Therefore, $n = \text{mod}(130,3) = 1$. The first right part (in zero origin) for the rule with left part *O* is *X*. We replace *O* by *X* in *V*. After this step, *V* = "X*X."

5. Now *V* does not contain any nonterminal symbol, therefore, the generation is complete and the result is expression "X*X," that is, *X* to the *X* power in APL2. The APL2 function generated is

[0] $Z \leftarrow Fnnn\, X$

[1] $Z \leftarrow (\rho X)\rho 0$

[2] $\rightarrow (5 < \rho \square LC)/0$

[3]$''\square EA\, 'Z \leftarrow X * X$

REFERENCES

1. Volterra, V. Leçons sur la Théorie Mathématique de la Lutte pour la Vie, Gauthier-Villars: Paris, 1931.
2. Alfonseca, M.; Carro, R.; de Lara, J.; Pulido, E. Education in ecology at the Internet with an object-oriented simulation language, Proceedings of Eurosim 98, Helsinki, 1998; pp 118–123.
3. de Lara, J. Modelling an ecosystem, 2000. Available at: http://www.ii.uam.es/~jlara/investigacion/ecomm/ecologia/simul.html. Last accessed on February 4th 2014.
4. Zhang, Y.; Zhang, Q. Dynamical analysis of a delayed singular prey–predator economic model with stochastic fluctuations. Complexity, in press; doi:10.1002/cplx.21486. Published online 5 December 2013.
5. Hawick, K.A.; Scogings, C.J. A minimal spatial cellular automata for hierarchical predator–prey simulation of food chains, Institute of Information and Mathematical Sciences, Technical Report CSTN-040, 2010..
6. Hol, F.J.H.; Wang, X.; Keymer, J.E. Population structure increases the evolvability of genetic algorithms. Complexity 2012, 17, 58–64.
7. Alfonseca, M.; Ortega, A.; Suárez, A. Cellular automata and probabilistic L systems: An example in Ecology. In: Grammars and Automata for String Processing: From Mathematics and Computer Science to Biology, and Back; Martin-Vide, C.; Mitrana, V., Ed.; Taylor & Francis, London & New York, 2003; pp. 111–120.
8. Conrad, M.; Pattee, H.H. Evolution experiments with an artificial ecosystem. J Theor Biol 1970, 28, 393–409.
9. Dorin, A.; Korb, K.B.; Grimm, V. Artificial-life ecosystems: What are they and what could they become? Artificial LIFE XI: Proceedings of the Eleventh International Conference on the Simulation and Synthesis of Living Systems, MIT Press, Winchester UK, p. 173–180, 2008.
10. Gras, R.; Devaurs, D.; Wozniak, A.; Aspinall, A. An individual-based evolving predator–prey ecosystem simulation using a fuzzy cognitive map as the behavior model, Artif Life 2009, 15, 423–463.
11. The Wolfram Demonstrations Project, 2012. Predator–Prey Ecosystem: A Real-Time Agent-Based Simulation. Available at: http://demonstrations.wolfram.com/PredatorPreyEcosystemARealTimeAgentBasedSimulation/. Last accessed on February 4th 2014.
12. Axelrod, R. Advancing the art of simulation in the social sciences. Complexity 1997, 3, 16–22.
13. O'Neill, M.; Ryan, C. Grammatical Evolution: Evolutionary Automatic Programming in an Arbitrary Language. Kluwer: Norwell, MA, 2003.
14. Dempsey, I.; O'Neill, M.; Brabazon, A. Foundations in Grammatical Evolution for Dynamic Environments. Studies in Computational Intelligence Springer, Berlin Heidelberg, 2009, p. 194.
15. Byrne, J.; McDermott, J.; O'Neill, M.; Brabazon, A. An analysis of the behaviour of mutation in Grammatical Evolution. Proceedings of the 13th European Conference on Genetic Programming, EuroGP, Istambul, 2010, Vol. 6021, LNCS, Springer, pp. 14–25.
16. de la Cruz, M.; Ortega, A.; Alfonseca, M. Attribute grammar evolution, In *Artificial Intelligence and Knowledge Engineering Applications: A Bioinspired Approach*, ed. J. Mira, J. R. Álvarez, Lecture Notes in Computer Science 3562, p. 182–191, Springer, Berlin Heidelberg.
17. Ortega, A.; de la Cruz, M.; Alfonseca, M. Christiansen grammar evolution: Grammatical evolution with semantics, IEEE Trans Evol Comput 2007, 7, 77–90.
18. Goodnight, C.; Rauch, E.; Sayama, H.; De Aguiar, M.A.M.; Baranger, M.; Bar-yam, Y. Evolution in spatial predator–prey models and the "prudent predator": The inadequacy of steady-state organism fitness and the concept of individual and group selection. Complexity 2008, 13, 23–4.
19. Alfonseca, M.; Soler Gil, F.J. Evolving an ecology of mathematical expressions with grammatical evolution. Biosystems 2013, 111, 111–119; doi:10.1016/j.biosystems.2012.12.004.
20. Watson, J.D.; Baker, T.A.; Bell, S.P.; Gann, A.; Levine, M.; Oosick, R. Molecular Biology of the Gene; Pearson/Benjamin Cummings: San Francisco, 2008.
21. Ortega, A.; Dalhoum, A.A.; Alfonseca, M. Grammatical evolution to design fractal curves with a given dimension, IBM J Res Dev 2003, 47, 483–493.
22. Graur, D.; Li, W.-H. Fundamentals of Molecular Evolution: 2nd Ed.; Sinauer Associates: Sunderland, MA, 2000.
23. Brauer, F.; Castillo-Chavez, C. Mathematical Models in Population Biology and Epidemiology, Springer-Verlag, Berlin Heidelberg, 2000.
24. Gould, S.J.; Eldredge, N. Punctuated equilibria: the tempo and mode of evolution reconsidered. Paleontology 1977, 3, 115–151.
25. Geary, D. The legacy of punctuated equilibrium. In: Stephen Jay Gould: Reflections on His View of Life; Allmon, W.D. et al, Eds.; Oxford University Press: Oxford, 2008; pp. 127–145.
26. Shannon, C.E. 1948 A mathematical theory of communication. Bell Syst Tech J 1948, 27, 379–423; 623–656.
27. Magurrran, A.E. Measuring Biological Diversity; Blackwell Science: Oxford, 2004.
28. Tuomisto, H. A consistent terminology for quantifying species diversity? Yes, it does exist. Oecologia 2010, 4: 853–860; doi: 10.1007/s00442-010-1812-0.
29. Dawkins, R. The Blind Watchmaker; W.W.Norton & Company: New York, 1996.
30. Fontana, W. Algorithmic chemistry. In: Artificial Life II, SFI Studies on the Sciences of Complexity, Vol. X; Langton, C.G.; Taylor, C.; Farmer, J.D.; Rasmussen, S., Eds.; Addison-Wesley, Redwood City CA, 1991; pp. 159–209.
31. Fontana, W.; Buss, L.W. The barrier of objects: From dynamical systems to bounded organizations. In: Boundaries and Barriers; Casti, J.; Karlqvist, A., Eds.; Addison-Wesley, Redwood City CA, 1996; pp. 56–116.
32. Alfonseca, M.; Selby, D. APL2 and PS/2: The language, the systems, the peripherals, APL Quote Quad (ACM SIGAPL) 1989, 19, 1–5.