Simulation of Continuously Evolving Ecosystem

Anti Kivi, Sander Sats, Tetiana Rabiichuk[1]

[1]Institute of Computer Science, University of Tartu

Inspiration

We wished to explore the area of artificial life, by creating our own world populated with creatures. We were inspired by Bibits world [1] developed by Léo Caussan. It started as a small hobby project in 2016, and now it has a dedicated Youtube channel with 12 thousands subscribers, and it was recently presented at 2020 Conference on Artificial Life (ALIFE 2020). Another valuable source of inspiration was a Primer [2] series of videos on evolution by Justin Helps. We have gained useful insights about genetic algorithm from the book "The Nature of Code" [4] by Daniel Shiffman.

Fig. 1: The Bibits. [1].

A brief description of our world

Each creature is alive as long as it has non-zero health. During lifespan, the creature can eat food, which increases its health, and hence life span. The creatures can multiply asexually and sexually (each creature has a preference towards certain type of reproduction determined by its genes). We wanted to create a continuous evolution, therefore, sexual reproduction happens in real time with certain probability when two creatures bump into each other. We have also explored the possibility of turning bigger creatures into predators: if the creature is big enough, it can consume smaller creatures. Each creature has a vision field (when an object is inside the vision field, it can be detected by the creature). The size of the vision field is also determined by the DNA of the creature. The size and the speed of the creature are also determined by DNA, since they have an impact on the ability of the creature to successfully survive (get food) and reproduce. However, having some biological privilege (bigger size, higher speed, bigger vision field) comes with the cost. During lifetime, the health of the creature decreases (and can be restored through food consumption) by the following law:

$$health\_loss \propto size^3 \cdot speed^2 + vision\_field\_size \qquad (1)$$

Genetic algorithm

Application of genetic algorithm usually consists of 3 main steps: *selection*, *crossover*, *mutation*. In traditional approach, the genetic algorithm is given an objective function to minimize/maximize. Each member of the population is represented with a vector of parameters (genes) that determine the value of the objective function for this member of the population. For each generation, for each member of the population the value of the objective function is computed. Then, on the basis of the computed values, members of the population with better values of objective function (but usually not exclusively the best, to avoid converging into local minimum/maximum) are chosen to create a new generation (*selection* step). For each pair of parents from the subset of population created on the selection step, the reproduction consists of two steps: *crossover* and *selection*. *Crossover* step determines which components of the gene the child gets from which of the parents. The secret of the success of the genetic algorithm is also due to variability in the population. One way to achieve the variability is to create a diverse initial population. However, the selection step can potentially remove "unfitted outliers" who, even though do not possess the best value of the objective function, can possess a valuable gene component. Therefore, in the selection step members of the population with not the best scores are usually included as well, but in the smaller amount. Another way to induce variability is through random mutation of the genes of the child obtained in the crossover step.

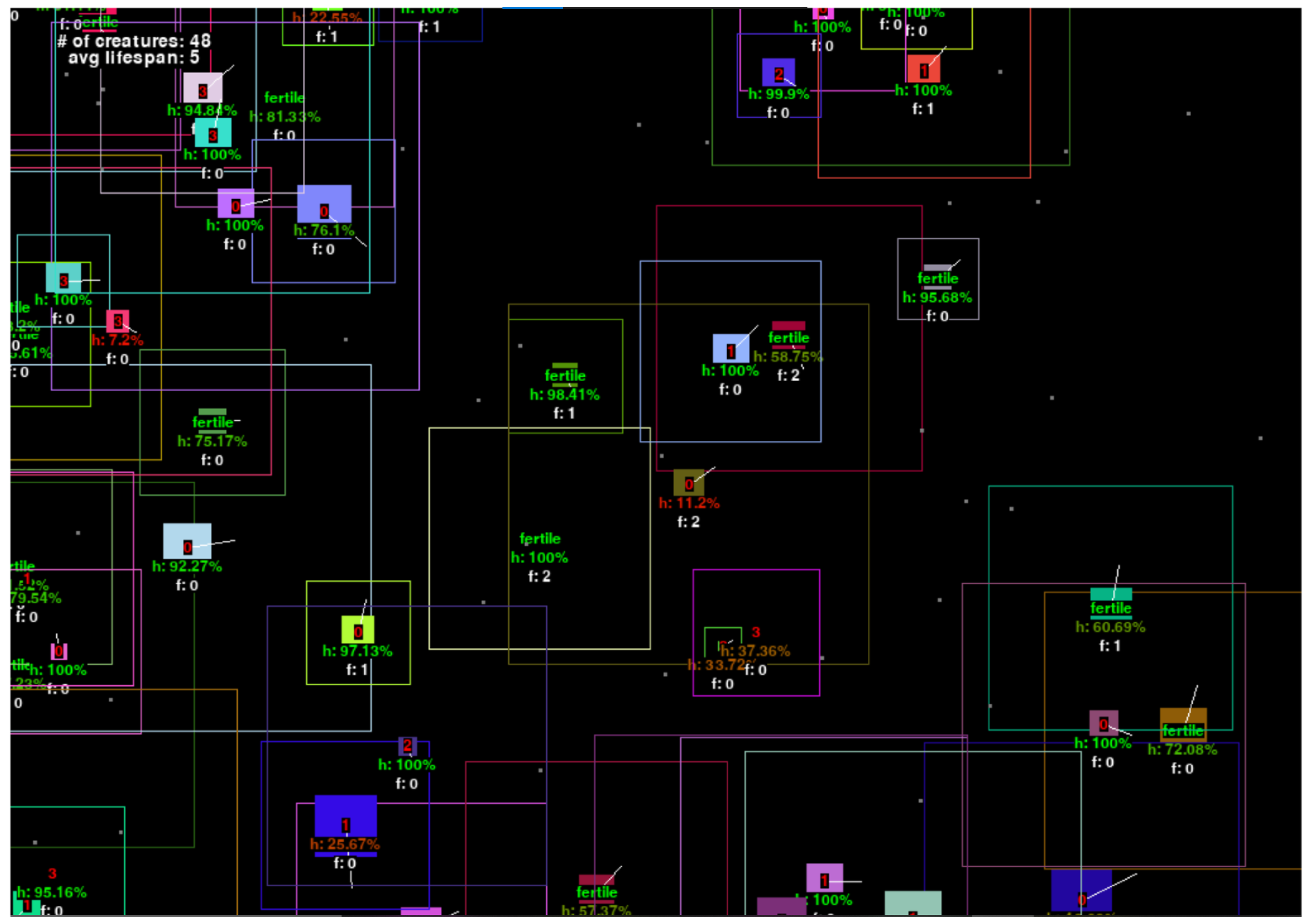

Fig. 2: A view of the world. "h" stands for health, "f" stands for total number of food pellets consumed. The outer rectangle indicates the boundaries of the vision field of the creature.

In our model, since we model a continuous evolution process, the creatures do not possess a unique, well-defined objective function that can be optimized. The end goal of each creature is to survive and reproduce, which in itself contains a range of sub-tasks. Therefore, in our model there are no *generations*. Creatures live and die in the continuously evolving world. The selection step for the reproduction is therefore different from the traditional approach. We introduced sexual and *asexual* reproduction. In the first case, two creatures "living their life" in the environment reproduce with a certain probability when they bump into a potential partner. In the second case (which also is happening with a fixed probability), the child gets a replicated copy of parent's genes, with a certain probability of mutations. One can say that in our context "the fittest" are the creatures who survive longer, since the longer the creature survives, the more chances it gets to reproduce.

Results and conclusions

We started off with creatures randomly wandering around the world, occasionally bumping into food and living off-springs. Next, we upgraded "the intelligence" of the creature. If food was detected in the vision field, the creature moved towards the food. Each creature has a reproduction cool down: it cannot mate immediately after it had off-springs. In case there was no food in the vision range, however, potential partner was detected, and both partners were ready to mate, our creature moved towards potential partner. We also experimented with different impacts of having off-springs on parent's health. In order to make food the only source of energy in the world, child's initial health depends on the current health of parent/parents.

Next, we created a brain of the creature: a feed-forward neural network. We wanted to observe what behaviour would emerge in a specified world conditions if we start from completely randomly initialized brains. Now, our creatures possess two types of genes: one of them encodes physical parameters (*size*, *speed*, *color*, *visual field size*, *multiplication type preference*), another type of genes codes "brain parameters": weights of the neural network that models creature's brain and determines its behaviour (movement direction) based on the inputs provided. However, we have not observed any tendency towards food consumption in order to survive. Creatures usually just gather at the outskirts of the world and reproduce.

Hence, we decided to reinforce food consumption trait. Since our initial plan was to create a continuously evolving world, we did not want to implement a traditional genetic algorithm approach. We decided to combine both approaches. Creatures still could reproduce through direct creature interaction via sexual reproduction. But we also introduced an objective function of total number of food pellets consumed. In the previous versions of the world, every once in a while (regulated by creature spawn time parameter), a completely random creature was introduced to increase diversity. We decided to substitute random creature spawning with traditional genetic algorithm approach, and instead of a random creature add descendants of the current best eaters in the population. However, again there were no obvious improvements in the tendency of the creatures to stir towards food. One of the possible reasons could be that the network tries to learn a continuous output in range $[0, 1)$ that later gets remapped to $[0, 2\pi)$ range, while in the examples we have seen (for instance, Super Mario [3]) the output layer is discrete, hence the mapping that needs to be learned is much simpler.

Our project is available at
`https://github.com/liivur/algorithmics-project`.

References

[1] Léo Caussan. *The Bibits: an attempt at creating digital life*. `https://leocaussan.itch.io/the-bibites`.

[2] Justin Helps. *Primer*. `https://www.youtube.com/primerlearning`.

[3] SethBling. *MarI/O - Machine Learning for Video Games*. `https://www.youtube.com/watch?v=qv6UVOQ0F44&t`.

[4] D. Shiffman, S. Fry, and Z. Marsh. *The Nature of Code*. D. Shiffman, 2012. ISBN: 9780985930806. URL: `https://books.google.at/books?id=hoK6lgEACAAJ`.