Creating an Evolution Simulator Using Agent-Based Modeling and Simulating

TJHSST Computer Systems Lab 2007-2008

by Natasha Wallage

Abstract

The main purpose of this program is to simulate accurately the genetic evolution of a species. It will attempt to do so using the most common evolutionary device known as genetic drift, a means of microevolution. This project is actually one of Agent-Based Modeling and Simulating (ABMS) in which the organisms are agents that react with one another and their environment. It has successfully linked trait value changes with environmental disasters that create stress on a species.

Fig 1. A snapshot of the final version of the evolution simulator. The bunnies represent the Flubber class while the plants represent the Blubber class.

Introduction

Evolution is often thought of as the changes that occur in an organism to better adapt it to its environment. However, this is not completely true. Evolution occurs in both positive and negative directions. It is completely random and the result could be in favor of the organism or it might not. It just so happens that the organisms resulting from an inefficient “evolution” do not survive and thus the species are left with the more fit species that will reproduce and populate. I am trying to simulate evolution and track the change in a species' traits to better understand how evolution really works and, more importantly, to experiment with genetic algorithms as represented in Computer Science.

The computer will simulate an environment and the user can modify that environment. Modification of an organism environment can force it to adapt and in essence, better survive. Because of this, those that are not adaptable to the change will die off and those that are will live on and reproduce, thus creating a genetic drift in the species. This is the theory, and according to this theory, one should be able to predict the changes in a species genetic makeup due to a change in its environment. If the environment becomes hotter, those creatures with higher temperature tolerance should be less affected than those without a high temperature tolerance, and so one would expect to see the species evolve to have a greater tolerance to higher temperatures. My project tracks speed and lung capacity as the adaptable traits.

Agent-Based Modeling

The actual evolution simulator is an ABMS with the Flubber class forming the ‘agents.’ An agent is “autonomous and self-directed.” It can “function independently in its environment and in its dealings with other agents.” [8] Mostly, an ABMS focuses on the interactions between the agents. In this project, I observed both the interactions between the agents and the interaction between the agents and their environment.

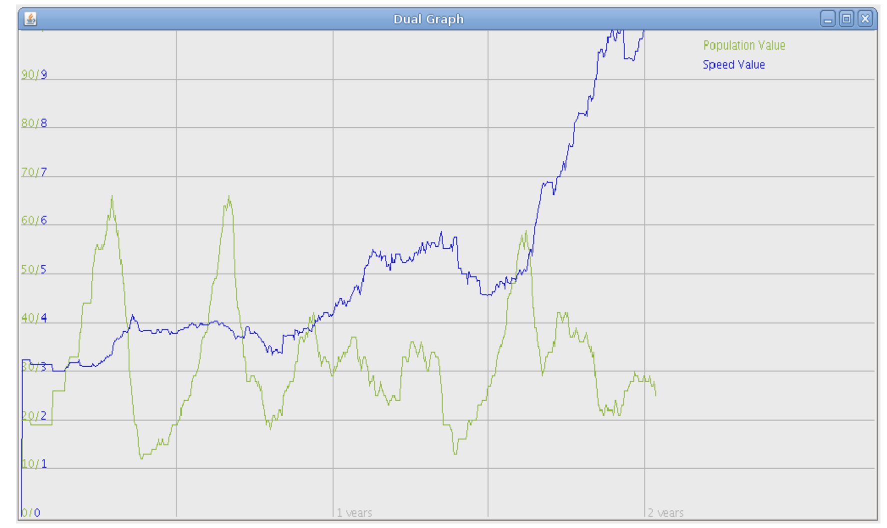

Fig 2. The DualGraph class graphing population and average population speed value over time.

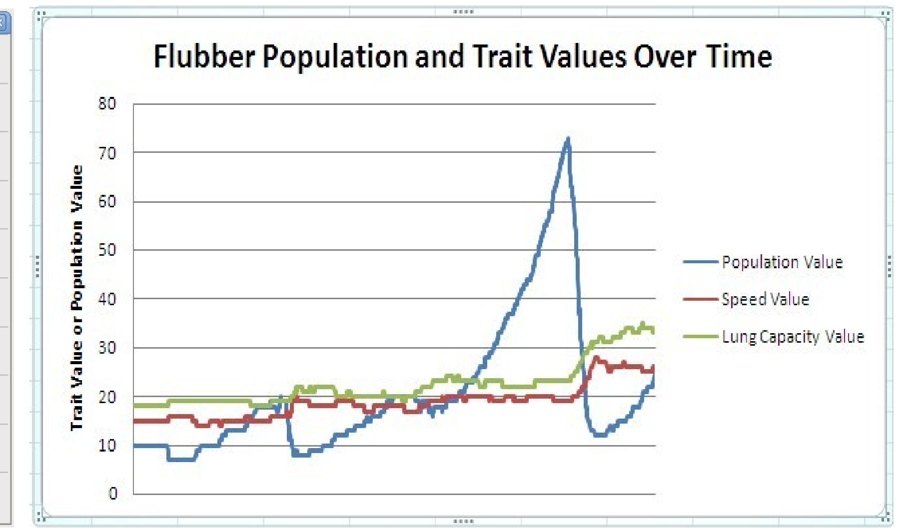

Fig 3. Output from the MultiGraph class reconstructed in Microsoft Excel.

Main classes

1. Blubber class

The sole purpose of the Blubber class is to provide food for the Flubber class. It stored the amount of energy it contains. As it gets older, it will grow bigger and thus the amount of energy it contains will increase. (If a Flubber eats an older Blubber it will gain more energy.)

2. Flubber class

The Flubber population is the species that is being studied. It contains changeable traits in its ‘DNA’ which determine the characteristics of its children (unless they be claimed by mutation).

• States: Each Flubber has four states: hungry, sexy, drowning, and curious. A Flubber bases its next action on its current condition. If it is *hungry*, it will search for food to eat or return to a nearby location from memory. If it is *sexy*, it will search for a mate of the opposite sex that is also sexy. If it is *drowning* it will retrace its steps until it is no longer lacking in oxygen. Finally, if the Flubber is none of the afore mentioned conditions, it is *curious*. When a Flubber is curious, it will freely roam about its environment, noting the location of food that it ‘sees.’

• Memory: Flubbers have two basic memories. One is devoted to known food locations while the other is devoted to recent steps taken.

• Genes: When two Flubbers mate, their genes are collected to determine the genetic make-up of the offspring, though randomly mutation will occur and the child will receive a gene that is not of its mother or father.

3. Environment class

This class contains all the Flubber and Blubber objects in the environment and allows them to react with one another. It has two HashMaps; one for the Flubber population and one for the Blubber population. This class also provides the GUI for the simulation and is the portal for user input.

• States: The Environment class also has four different states; normal state, drought, flood, and disease. The *normal state* is the default state of the Environment in which the plants grow at a somewhat speedy rate and there is little or no stress on the species whatsoever. When a *drought* is present, about eighty percent of the food in the Environment will die and the re-growth and regeneration of the plant life will be sluggish. Any water that was previously in the Environment will disappear (dry up). A *flood* fills the entire Environment with water, leaving on the plant life above the water. This means that the Flubber class can only survive by remaining on top of the Blubber plant life. However, the re-growth and regeneration of the plant life is increased while the Environment remains in this state. A *diseased* Environment will remove any individual Flubbers that do not meet the immunity required to survive. Depending on the severity of the disease and the diversity of the Flubber population, the disease may kill anywhere from no Flubbers to all Flubbers in the population.

Results and Conclusions

In my first model (Fig 4), I included both predators and prey. This made the system very complex and unstable. If there were too many predators, they would end up consuming all the prey and then starve. If there were only a few predators, they would have a hard time catching the prey and would usually die before breeding. Then the predators died out, the prey population would surge and the food from the environment could not grow fast enough to feed the gigantic herbivorous prey population. Eventually, the prey population would die out due to starvation. This also happened when the initial prey population was too large, thus when the prey died from starvation, the predators also died out. My current model (Fig 1) has proved to be much more successful than its predecessor. Because there are no predators, the system is less susceptible to balancing problems. I have run this simulation consistently for 10 hours and the population of Flubber agents does not die out.

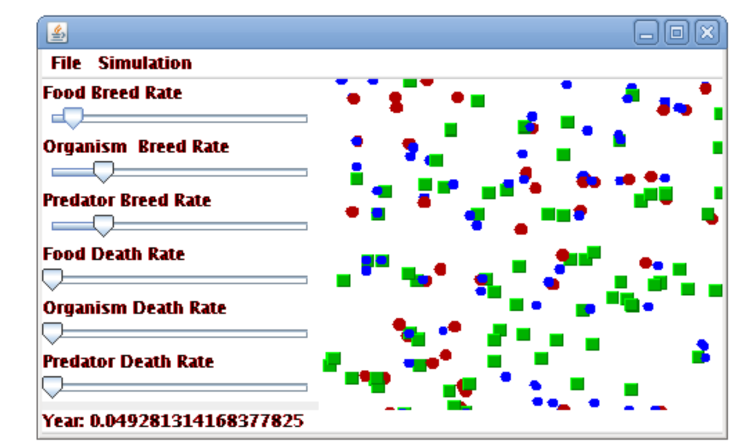

Fig 4. The first simulator that included plant life, herbivores, and carnivores. This model was extremely fragile and unsuccessful. The predators are represented by the red dot, the herbivores with the blue, and the plant life with the green squares. The slider bars on the left allowed for the user to control the death rates and breed rates for each population.

With the new model in place, the program runs quite smoothly and the graphs provide a good graphical display of what is happening. It behaves almost as one would predict with an oscillating population value and a trait value that shows an upward trend (evolution for the better). From the graphs, you can see that the trait values usually show a rapid increase when the population is very high. Because the traits I am tracking are speed and lung capacity, this makes a lot of sense. As the population gets high, the availability of food decreases and the ability to get to food quicker becomes an important survival skill; therefore, the agents with greater speed will live on to reproduce while those with lesser speed will die out, increasing the trait average for the population. On the other hand, if there is a flood, the agents with the greatest lung capacity will live on while those with low lung capacity will die out. This means the average trait value of lung capacity will increase over time. Overall, the project was a success and the evolution simulator shows strong similarities to real life.

References

[1] Chris Colby, “Introduction to Evolutionary Biology”, http://www.talkorigins.org/faqs/faq-intro-to-biology.html
[2] Charles Darwin, “The Origin of Species”, http://www.talkorigins.org/faqs/origin.html
[3] John Wilkins, “Evolution and Chance”, http://www/talkorgins.org/faqs/chance/chance.html
[4] Laurence Moran, “Random Genetic Drift”, http://www.talkorigins.org/faqs/genetic-drift.html
[5] Jaroslaw Puszcynski, “Artificial life Portal”, http://www.alife.pl/portal/main/e/index.html
[6] Jeff Smith, “Genetic Algorithms: Simulating Evolution on the Computer”, http://www.developer.com/tech/article.php/964131
[7] Eric Turner, “Evolution Simulator”, http://www.tjhsst.edu/ rlatimer/techlab07/TurnerProposal07.pdf
[8] Macal and North, “Tutorial on Agent-Based Modeling and Simulation”, http://www.cas.uiuc.edu/networkreadings/north1.pdf