Using a Genetic Algorithm to Replicate Allopatric Speciation

Gary B. Parker and Thomas B. Edwards
Department of Computer Science
Connecticut College
New London, CT, USA
parker@conncoll.edu and tedwards@conncoll.edu

Abstract—In this paper we describe a method using genetic algorithms to replicate natural allopatric speciation. There are many versions of evolutionary computation (EC) that have some characteristics of speciation, but none that match natural processes. In an effort to develop such a form of EC, we created a simple model that we used to experiment in developing an EC that mimics natural speciation. Our long term goal for speciation is to have a single population eventually become two populations that are reproductively isolated even though they reside in the same environment. In previous work we developed an environment where we replicated adaptation, survival of the fittest, and migration of a population. In this paper, we report on research where we used this environment to explore the possibility of speciation. We use a genetic algorithm that alters the agents in the environment as we allow a population of intermixing individuals to develop and become established. We then add a physical barrier that separates the individuals in the population and then remove the barrier after several generations to see if the initially single population becomes two reproductively isolated populations despite no longer being physically isolated. In this way we were able to replicate the initial stages of allopatric speciation.

Keywords—Allopatric, Sympatric, Speciation, Bioinspired, Genetic Algorithm, Artificial Life, Agent Modeling, Rule Base, Evolutionary Computation

I. INTRODUCTION

Speciation is the process of creating a new distinct species from an ancestral species. There are a variety of definitions for what makes a distinct species. The long term goal for this research is to have the individuals of one species unable to produce viable offspring with individuals from another species. The short term goal is to have individuals of differing species choose not to mate and if they do produce offspring, the offspring do not mate. This would result in separate gene pools for each species. There are two types of speciation, sympatric speciation and allopatric speciation. Sympatric speciation is the process where completely new species evolve from a common ancestral species despite the new species occupying the same geographic area [1]. Allopatric speciation is the process where a new species evolves from a common ancestral species due to some form of geographic isolation.

The overarching theme of our speciation research is to see if it is possible to replicate or simulate both allopatric and sympatric speciation using an artificial computer environment and a genetic algorithm (GA) [2]. GAs consist of a population of individuals that are intended to be the solutions to a problem. The traits of these individuals are encoded in chromosomes, which are often constructed from a sequence of 0’s and 1’s so that they are amenable to genetic operators such as crossover and mutation. New individuals are evolved by finding a fitness value for each current individual, which is used for selection. Crossover between chromosomes is used to create a new chromosome, which is then subjected to random mutations. This new chromosome is used to create a new individual that is introduced into the population.

In this paper we present our results for replicating allopatric speciation. The motivation for the work is to both test biological theories and to find ways to improve genetic algorithms. The ability to replicate and model evolution using artificial species could directly translate to actual processes occurring naturally in the world. This might provide insight into the true evolutionary processes. Additionally the work could improve the use of genetic algorithms to a point where learning can progress to form more complicated solutions to problems using improvements through speciation.

Other research has been done that tends toward speciation. Most of this has been focused on the use of niching to improve the use of GAs for multimodal function optimization [3]. The idea is to compute a similarity measure between chromosomes and use this to form niches with subpopulations in the solution landscape. This is an effective way to ensure distribution in GA chromosome solutions and researchers have expanded the idea to create new ways to ensure niches are maintained [4]. Although a good solution for multimodal function optimization, niching with its distinct subpopulations is not what we consider speciation. The individuals within the niches are determined by a similarity function as opposed to mating preferences as in natural systems. The similarity function is artificially creating the subpopulations. In our research we want the subpopulations to develop naturally.

A system that makes use of the concept of niching for the evolution of agents is NeuroEvolution of Augmenting Topologies or NEAT [5]. NEAT is similar to our research because it uses evolutionary processes to learn agent controls, the learning is in real time, and there is a concept of speciation. NEAT uses a compatibility function to determine if two agents are part of the same species or not, similar to how in nature things that are more structurally different are usually different species. NEAT has access to all the historical data for an agent and this information is used to tell which gene corresponds to another gene and which agents belong to which species.
Additionally, NEAT uses explicit fitness sharing which makes similar agents share their fitness pay off. This means that there is competition within species, but not as much between two different species. In our system we want species to develop through natural reproductive isolation -- they choose not to mate as opposed to us restricting them from mating. This is more in line with the natural processes needed for biological speciation. Our system does not make agents share their fitness payoff, although limited resources could tend to have this result. It allows competition within a species and between species because agents are constantly vying for the food resources in the environment regardless of the species type.

The Speciation Island Mode (SIM) has some similar characteristics to our work as well as it tries to evolve new subpopulations. The goal of this research was to investigate “a novel model for parallel evolutionary algorithms (EAs) based on the biological concept of species” [6]. Here species represent potential solutions to problems and islands are different processors which can have solutions allocated to them. The processors can then perform varying processes such as fitness calculations in their own sub-population. Population outliers of islands are detected. The outliers are removed from the population and then placed on another island. The outliers and the other population can then evolve independently of each other. Allopatric speciation is occurring due to the physical separation of the individuals, however outliers are detected and actively removed from the population without natural migration. So although allopatric speciation for EA’s is used it does not truly occur naturally in a biological sense. Additionally, there is no notion of true speciation since the populations are never recombined to test for reproductive isolation. Our work uses the idea of allopatric speciation for GAs in a strict biological sense.

This idea of islands and speciation has also been explored on in other instances such as the Evolving Neural Networks with Collaborative Species research done by Potter and De Jong [7]. In this case, cooperative speciation is applied to get solutions to a problem. The research uses the same premise of having islands (processors) which each independently evolve their own populations. However the islands communicate with each other because the islands select representatives to be combined with the other representatives into a single structure. The structure is then evaluated to see how well the problem was solved. Credit to the solution heads back via the representatives to their respective islands so the EAs get data on how to evolve the next wave of island agents. This system is known as Cooperative Coevolutionary Algorithms and is intended to solve large problems as opposed to replicate speciation [8]. The species start out separate and remain that way, whereas our species start as one species and split into two. In addition, their species are intended to work together whereas our organisms are in competition for food resources.

In addition to speciation we use a unique method of variable population sizes for GA’s. The Genetic Algorithm with Varying Population size (GAVaPS) implemented a way to have the population size vary by introducing aging in the agents [9]. In GAVaPS, an individual will die when its age exceeds its lifetime value. The paper discusses different processes for which a lifetime value can be decided for an individual, meaning that the lifetime value, which remains constant as the agent ages, is determined for each individual. Our research is similar to the idea of GAVaPS, but has key differences. We allow for variable population size by implementing age as well, however we do not pre-determine a lifetime value for our agents. Instead we use age to impact how much life (energy) an agent loses each round. Once the energy drops to zero or less, the agent will die. Thus we do not determine a lifetime value for the maximum age an agent can get. We use an energy score which is impacted by age to decide if an agent should die.

Other GA’s have been created to allow variable population sizes as well, such as APGA, ProFIGA, and the PSO-GA hybrid algorithm (PGHA) [10]. The PGHA has similarities to our research because “parents are neither dead after their reproduction right away, nor living forever”. Agents can survive after reproducing but the longer they live, the higher the probability that they die. The PGHA uses an actual probability to determine if an agent is alive after n rounds. The algorithm selects agents based on fitness and allows for a variable population size. In our work, selection is based on location (the proximity of two agents) and the selectivity of the two agents. In addition, we use variable population size, however we use age as a factor that directly impacts an agent’s energy as the simulation progresses, as opposed to the PGHA method of using a probability to determine life and death.

II. Environment

The model developed was designed to be as simple as possible, yet complex enough to create situations where speciation would occur. We use a grid environment where agents are capable of moving, eating, and interacting with each other. This grid can be of variable length and width depending on the test and application needs. In the research reported in this paper we used a 100 x 100 grid environment -- Figure 1 shows what this looks like before adding agents. The grid is made up of discrete blocks (spaces), which can be empty or have a seed and/or an agent within them. A seed is represented as a square while an agent is represented as a circle. During each time cycle, the agents all move one block as specified by their controller and food pops up in the grid.

The food in the environment is made up of seeds with three different sizes: large, medium, and small. The density of each seed type can be specified, which dictates the numbers of each seed type added to the grid during each time cycle. The grid is divided into 3 sections: the left third, middle third, and right third; where differing distribution probabilities for each seed type can be specified. For example, with a large seed distribution of 80/10/10 in the grid, a large seed would have an 80% chance of occurring in the left third of the grid, a 10% chance of occurring in the middle third of the grid, and a 10% chance of occurring in the right third of the grid. Having the grid environment split up into different sections simulates different distinct overlapping environments, which could be used to replicate different climates or soil types. New seeds will populate the environment after every time cycle in the simulation (when all the agents have had a chance to perform some action). The number of new seeds added is determined by the designated density for each seed type. Their placement is random, but determined by the distribution for each seed type. The food does not move – once seeds are placed, they stay in that position until they are eaten.
Fig. 1. 100 x 100 grid environment populated with seeds but with no agents. The large seeds are mainly in the left third, medium seeds are mainly in the middle third, and small seeds are mainly in the right third. This probability distribution can be changed to fit the testing needs.

Agents move around within the environment, eat seeds, and have the potential to interact with each other. The number of agents in the environment will increase and decrease depending on the actual environment. The physical characteristics of the agents are color and size. The initial population of agents receive random colors and sizes. An agent’s color is stored as an RGB value, but as new agents are created, their color is inherited from the parents’ colors with some chance for mutation (adding/subtracting from a parents RGB value).

Fig. 2. A 10 x 10 grid with agents (circles) and seeds (squares) of varying sizes. The agents have varying sizes of large, medium, and small. The size of the agent determines which seed that agent is most capable of eating. If a large agent was surrounded by 2 small seeds, a medium seed, and a large seed, then the best option for the agent would be to move to and eat the large seed. An example of the varying sizes of agents and seeds as well as the general distribution of seeds can be seen in Figure 2.

Each agent has two chromosomes. One of the chromosomes (details discussed in the next section), dictates what activity it will do given its surroundings. One of the possible actions of an agent is to mate with another agent in its adjacent vicinity. If this successfully occurs, another agent is produced and placed into the environment within three blocks from the parents. The second chromosome of each agent specifies what it is looking for in a partner in order to reproduce. It dictates the desired partner size, maximum color difference between partners, and desired age range in the partner (Figure 3, Tables 1, 2, and 3).

Agent 1 - Size: Medium, R color: 100, G color: 100, B color: 100, Age: 55
Chromosome: 1 1 0 0 1 1 0 1 0 0 1
Size Desires: 1 1 meaning any size
Max Color Dif: 0 0 1 meaning 189
Ideal Age Range: 1 0 1 meaning 110 - 124
Actual Age Range: 0 0 1 meaning up and down 1 index spot, 95 - 139

Agent 2 - Size: Large, R color: 50, G color: 50, B color: 50, Age: 135
Chromosome: 0 0 1 1 0 0 0 0 1 0 0
Size Desires: 0 0 meaning medium
Max Color Dif: 1 1 0 meaning 664
Ideal Age Range: 0 0 0 meaning 35 - 49
Actual Age Range: 1 0 0 meaning up and down 4 index spots, 35 – 109

Fig. 3. Example of two agents that would successfully reproduce with each other given their chromosomes and specific information.

There are four possibilities for desired partner size: medium, small, large, and any. The first two bits of the chromosome represent what an agent is looking for in a partner (Table 1).

Following the size preference bits, the next three bits dictate the maximum desired color difference in a partner agent (Table 2). To find the difference in color we find the difference between the various RGB values of both agents, take the absolute value, and then add them up. This means that the smallest color difference between agents can be 0 and largest can be 765 (3 * 255). The range of color difference between two agents goes from 0 to 765. The range is broken up into chunks with the following ranges: 0-94, 95-189, 190-284, 285-379, 380-474, 475-569, 570-664, and 665-765.

<table>
<thead>
<tr>
<th>Size Preference</th>
<th>Bit Value</th>
<th>Decimal Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Medium</td>
<td>0 0</td>
<td>0</td>
</tr>
<tr>
<td>Small</td>
<td>0 1</td>
<td>1</td>
</tr>
<tr>
<td>Large</td>
<td>1 0</td>
<td>2</td>
</tr>
<tr>
<td>Any</td>
<td>1 1</td>
<td>3</td>
</tr>
</tbody>
</table>

TABLE I.  THE BIT COMBINATION FOR THE DESIRED MATE SIZE OF A PARTNER AGENT.
Agents grow older every time cycle in the simulation. The age of reproductive fertility was defined to be between the ages of 35 and 154. For age desirability of a mate, this was broken up into eight categories: 35-49, 50-64, 65-79, 80-94, 95-109, 110-124, 125-139, 140-154. The next three bits of the mating selectivity chromosome dictate which of these categories are considered the ideal age of the partner (Table 3). The next three bits of the chromosome represent an added range (plus and minus index value) from the ideal age category for mating.

All of these three mating characteristics can best be seen in an example in Figure 3. We can see in the figure that the two agents would reproduce. Agent 1 desires an agent of any size and Agent 2 desires an agent of medium size. In both cases the agents are compatible. The color difference between the two agents is 150 and both agents allow for a maximum color difference that is more than 150, so the color traits show the agents are compatible. Finally the desired age range for Agent 1 is 95 - 139 and Agent 2 has an age of 135. The desired age range for Agent 2 is 35 - 109 and Agent 1 has an age of 55. The age of both agents fall into their partners desired range. Thus the age traits show that they are compatible. From this, the agents choose to mate and attempt to create a child.

### III. AGENT CONTROLLER

The actions of each agent are controlled by a rule based system, which is made up of a set of different rules (antecedent / consequent) that control the actions of the agent. If the antecedent of the rule is true, then the rule fires and the action in the consequent is taken. If more than one rule has the potential to fire, some means of conflict resolution is needed. In our research, we use priority ordering (this is learned by the GA) so the highest priority rule with a true statement fires. If there is a tie in priority, a random tied rule is chosen to fire. The rule based system was used to determine what specific action the agent would take at each time cycle in the environment.

The agents are capable of performing various actions: move to a free space, move to a free space and attempt to reproduce, move to a large seed space and do not eat it, move to a medium seed space and do not eat it, move to a small seed space and do not eat it, move to a large seed space and eat it, move to a medium seed space and eat it, move to a small seed space and eat it. At each time cycle the immediate surroundings of an agent (one square up, down, left, and right) are analyzed in conjunction with the antecedent of the rules to see which rules are set to fire. This will be used to determine, depending on the rule priorities, the action from the rule base the agent performs.

### TABLE IV. PERCENT CHANCE AGENTS SUCCESSFULLY PRODUCE AN OFFSPRING

<table>
<thead>
<tr>
<th>Seed Size</th>
<th>Small</th>
<th>Medium</th>
<th>Large</th>
</tr>
</thead>
<tbody>
<tr>
<td>Small</td>
<td>100</td>
<td>30</td>
<td>15</td>
</tr>
<tr>
<td>Medium</td>
<td>30</td>
<td>100</td>
<td>30</td>
</tr>
<tr>
<td>Large</td>
<td>15</td>
<td>30</td>
<td>100</td>
</tr>
</tbody>
</table>

An agent starts with 250 energy. All actions have a cost, and will decrease an agent’s energy. When agents attempt to reproduce, it will cost both agents a standard 80 amount of energy. Reproduction between agents of varying sizes will not always be successful. The probability that agents of differing sizes successfully produce an offspring can be seen in Table 4. Any action other than reproduction reduces the agent’s energy by the square root of agent’s age, so older agents use more energy to survive each time cycle. If an agent eats a seed it will get a certain amount of energy depending on the agent’s size and the size of the seed being eaten as depicted in Table 5. For example, if we had a large agent that had 200 energy and 100 age, and it moved to and ate a large seed, the resultant energy would be: 200 + 30 − √100 = 220.

An example of a chromosome with its resultant priorities can be seen in Table 6. This table depicts a single agents’ chromosome broken into its separate parts and what each part of the chromosome dictates. The chromosome is broken into eight four bit sections. Whichever section has the highest decimal value relative to the agent’s surrounding is the action performed.
According to this example the agent would most want to move to a large seed space and not eat the food. However if one of the adjacent spaces does not contain a large seed this action cannot be performed; the next highest priority rule will be checked to see if the antecedent is satisfied. In this case, the agent would try the rule that would move it to a free space. This process of finding the highest rule to fire is continued until a rule is fired. In the case of a tie, one of the tied rules is randomly picked to fire. Pseudocode for the algorithm can be seen in Figure 4.

For each agent in the environment:
- Get the adjacent points (at most 4)
- For each adjacent point:
  - Determine the appropriate rule/rules for that point
  - Add point/rule pairs to a RulesToFire list
- Determine the rule priority for the agent from its chromosome
- Fire highest possible rule and move agent to that space
- If there is a tie in rule priority:
  - Pick random rule to fire
- If multiple possible positions for firing rule:
  - Pick random space to move to
- If rule selected is mate:
  - Determine if both agents are willing to mate
  - If agents mate:
    - Create new agent
    - Place it at random space within 3 spaces of parents
- Reduce energy of agent
  - If attempted to mate
    - Reduce energy of both partners by 80
  - Else reduce energy by the square root of agent age
- Add energy to agent if food consumed
- Add one to the age of the agent
- Update agent information in environment

![Figure 4](image-url)

**Table VI. An example action chromosome that determines priorities for the rule based system controller.**

<table>
<thead>
<tr>
<th>Rule Base</th>
<th>Free Space</th>
<th>Reproduce</th>
<th>Large Seed</th>
<th>Med Seed</th>
<th>Small Seed</th>
<th>Large Seed</th>
<th>Med Seed</th>
<th>Small Seed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chromosome</td>
<td>1 1 0 1</td>
<td>0 1 1 1</td>
<td>1 1 1 0</td>
<td>1 0 1 0</td>
<td>0 0 0 0</td>
<td>1 0 1 0</td>
<td>0 1 1 1</td>
<td>1 0 0 0</td>
</tr>
<tr>
<td>Decimal Value</td>
<td>13</td>
<td>7</td>
<td>14</td>
<td>10</td>
<td>0</td>
<td>10</td>
<td>7</td>
<td>8</td>
</tr>
</tbody>
</table>

Offspring. The GA operates on two chromosomes (mating preferences and rule priorities), each made up of 0’s and 1’s. Selection in our system is determined by physical proximity (adjacent spaces) of the two individuals and they need to both agree to mate depending on the other’s size, color, and age. The idea of selection weighted by agent fitness does not fully apply here. Instead, any agent can attempt to reproduce, however the fitter agents should have a greater chance to reproduce because they will have more energy for reproduction and survive longer. Crossover is single point with 100% chance of crossover and mutation is bitwise independent with a very low probability of mutation (1/300 per bit).

The GA used in our research is similar to a steady-state GA, except that we do not use standard chromosome replacement with our GA – old individuals die when out of energy and new individuals are added to the population with no regard to the overall size of the population.

**V. Results**

Initial tests were done to check the model [11]. These tests showed that populations could evolve to optimize for the available food sources, adapt to changing environments, migrate to changing locations of food sources, and adjust when both locations and types of foods changed.

Tests were then created in an attempt to replicate allopatric speciation. A population of medium agents who inhabited the middle third of the grid was established. The environment was set up such that only medium seeds existed in the grid in a 10/80/10 distribution. The initial random population of individuals was created, and as expected all the big and small agents died out while a population of medium agents established themselves in the middle third of the grid. Once the population was fully established, a physical barrier (wall) was put up in the middle of the grid, and thus in the middle of the population. At the same time the wall was put up, medium seeds were no longer being produced and large and small seeds began to be produced predominantly in the left and right third of the grid respectively. As expected from previous tests, half of the medium population migrated to the large seed section and evolved to a population of large agents. The other half of the medium population migrated to the small seed section and evolved to be a population of small agents. The two separate populations were allowed to evolve independently while the wall remained in existence. After thousands of time cycles the wall was removed.

Using the established populations from this simulation run as a starting point, two tests were performed. The first test
were examples where first and second degree hybrids would appear. In some cases the hybrids would die out, but there were others where the hybrids would persist and reproduce with agents from the main populations. Our goal was to show that any hybrids is not ideal, we feel the results are positive. Even though first degree hybrids existed, they did not reproduce, meaning they would not affect the gene pool of either of the two main species populations.

In future work, we’d like to expand the agent characteristics that were used for selectivity. Alterations that forced the agents to choose a mate based on color and age, as opposed to mainly size, could be the next step in a more complete replication of speciation.

ACKNOWLEDGMENT
We thank Associate Professor Randall DeJong from the Department of Biology from Calvin College for his instruction/guidance on speciation and help in developing a simulated environment where speciation could occur.

REFERENCES