INTRINSIC AND EXTRINSIC ADAPTATION IN A SIMULATED COMBAT ENVIRONMENT

THESIS

Presented to the Graduate Council of the University of North Texas in Partial Fulfillment of the Requirements

For the Degree of

MASTER OF SCIENCE

By

Steven P. Dombrowsky, B.S.
Denton, Texas
May, 1995
INTRINSIC AND EXTRINSIC ADAPTATION IN A
SIMULATED COMBAT ENVIRONMENT

THESIS

Presented to the Graduate Council of the
University of North Texas in Partial
Fulfillment of the Requirements

For the Degree of

MASTER OF SCIENCE

By

Steven P. Dombrowsky, B.S.

Denton, Texas

May, 1995

Genetic algorithm and artificial life techniques are applied to the development of challenging and interesting opponents in a combat-based computer game. Computer simulations are carried out against an idealized human player to gather data on the effectiveness of the computer-generated opponents. It is found that genetic algorithms and artificial life generate opponents that are more effective than those generated by a random process (which is the approach used in many games). It is found that neither genetic algorithms nor artificial life has a distinct advantage. Genetic algorithms have several advantages over artificial life, including the generation of more effective opponents. Artificial life has several advantages over genetic algorithms, including the generation of distinct species, a feature that adds to the potential interest of the game. The effect of varying certain key parameters such as the form of the fitness function, interval between breeding, and speciation threshold are also investigated.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>Life, Natural Selection, and Evolution</td>
<td></td>
</tr>
<tr>
<td>Adaptation Through Artificial Life and Genetic Algorithms</td>
<td></td>
</tr>
<tr>
<td>Survey of Applications of Artificial Life Games</td>
<td></td>
</tr>
<tr>
<td>Artificial Life Applied to a Dynamic Feedback Game</td>
<td></td>
</tr>
<tr>
<td>Mechanics of DFALG</td>
<td></td>
</tr>
<tr>
<td>Fundamental Areas of Interest</td>
<td></td>
</tr>
<tr>
<td>2. ADAPTATION AND ARTIFICIAL LIFE</td>
<td>20</td>
</tr>
<tr>
<td>Increasing Adaptation Using Artificial Life Simulation Using a Non-propagating Population</td>
<td></td>
</tr>
<tr>
<td>Variations in Artificial Life Control Parameter Thresholds</td>
<td></td>
</tr>
<tr>
<td>3. ADAPTATION AND GENETIC ALGORITHMS</td>
<td>38</td>
</tr>
<tr>
<td>Genetic Algorithms Compared to Artificial Life</td>
<td></td>
</tr>
<tr>
<td>Variations in Breeding Periods of Genetic Algorithms</td>
<td></td>
</tr>
<tr>
<td>Variations in Fitness Functions of Genetic Algorithms</td>
<td></td>
</tr>
<tr>
<td>4. SPECIATION AND FITNESS</td>
<td>54</td>
</tr>
<tr>
<td>Extended Runs Using Artificial Life</td>
<td></td>
</tr>
<tr>
<td>Emergent Species From a Cloned Species</td>
<td></td>
</tr>
<tr>
<td>Affects of Environment Size on Speciation</td>
<td></td>
</tr>
<tr>
<td>5. CONCLUSION AND OPEN PROBLEMS</td>
<td>64</td>
</tr>
<tr>
<td>APPENDIX A</td>
<td>68</td>
</tr>
<tr>
<td>APPENDIX B</td>
<td>70</td>
</tr>
<tr>
<td>BIBLIOGRAPHY</td>
<td>73</td>
</tr>
</tbody>
</table>
CHAPTER 1

INTRODUCTION

Recently there has been a surge of activity among computer scientists to make use of paradigms and analogies from nature to solve various problems arising in computer science. Some examples of this trend are genetic algorithms (GAs) with inspiration from genetics and evolution, artificial life (AL) with inspiration from biology and nature, and artificial intelligence (AI) with inspiration from neurobiology and cognitive science.

On a different dimension, with advances in graphics and hardware technology the popularity of games is on an increase, which is supported by an expanding market. Of all the different varieties of games on the market such as sport, adventure, strategic, combat, and vehicular, the most traditional and more common is the combat game [2,11]. This style of game basically consists of shooting at opponents which requires nothing more than good hand-eye coordination for winning. Unfortunately, opponents typically are very simple and exhibit the same basic programmed strategy, offering the player little challenge after his skills have been honed. The aim of this study is to revolutionize combat game entities using the paradigms of AL and GAs.
The work reported here is an attempt to formulate a combat game simulation using AL and GA paradigms to evolve entities, which are the opponents in the combat game simulation, that will adapt to the player's strategies and skills. The specific objectives of this study are:

1. To apply the framework of AL and GAs to a combat game simulation.

2. To demonstrate how AL and GAs affect the natural evolution of opponents in a combat game simulation.

3. To empirically demonstrate the inherent advantages of varying control parameters of AL and GAs when applied to a combat game simulation.

The thesis is organized as follows. In Chapter 2, adaptation of entities using AL is examined. It will be shown that AL will allow entities to adapt to a player's strategy and skills and variation in AL techniques will result in different entity evolution. In Chapter 3, adaptation using GAs is examined. It will be shown that GAs are similar to AL and variation in GA techniques will result in different entity evolution. Chapter 4 is devoted to the speciation, the formation of new classes of species, and fitness, the overall evaluation of the success of adaptation of an entity. Conclusions and open problems are in Chapter 5.

Life, Natural Selection, and Evolution

What is life? Common definitions range from "the period of time from birth to death" to "an organismic state characterized by capacity for metabolism, growth, reaction
to stimuli and reproduction"[16]. Given these definitions how can life be recognized? It has been suggested [13] that assembling a list of properties of life and then testing candidates on the basis of whether or not they exhibit the properties on the list. The main problem with this approach is that there is disagreement as to what should be on the list, although there are some rather constant features that have been agreed upon, including the following [16]:

1: Living things are organized.
2: Living things take matter and energy from the environment.
3: Living things reproduce.
4: Living things respond to their environment
5: Living things adapt to the environment.
6: Living things tend to maintain a balanced steady state.

We will use the above characteristics as our list of criteria for defining life.

In 1837 when Charles Darwin started to keep a journal entitled Transmutation of Species, he began to toy with the idea that the environment could select the individuals that were allowed to breed and these would be the hardiest [16]. Briefly, natural selection includes:

1: Overproduction of offspring.
2: Natural variation within a population.
3: Limited resources and the struggle for survival.
4: Selection by the environment for those with traits that enable the individual to survive and reproduce.
From this list it is concluded that the environment (both the living and nonliving) is the key element that determines which individuals are to survive and reproduce. The traits that permit survival in the environment will increase in the population as other less advantageous traits are weeded out.

Evolution is the continuous genetic adaptation of organisms or populations through mutation, random drift and natural selection. It is important to remember that individuals do not evolve, populations evolve. This gives rise to speciation, the process whereby new classes of species are formed. As mentioned earlier, natural selection provides for a species to adapt to its environment and sometimes gives rise to new species. Driven by the forces of natural selection, species can become more similar, a process called convergent evolution.

Adaptation Through Artificial Life and Genetic Algorithms

Life was previously defined by emphasizing several key characteristics of living organisms. Also discussed were issues such as natural selection and evolution which provide for adaptation of species to their environment. These areas are a subsection of the study of life known as biology. Sometimes biology can be slow in giving its secrets. Biologists often ask what will happen to a species in an environment over the next thousand years or environmentalist may wonder what will happen to this species if a certain aspect of the environment is changed. These questions lead
to the study of adaptation. Adaptation is the modification of an entity or its parts that makes it more fit for existence under the conditions of its environment [16]. This statement is the basis of many fundamental theorems of biology.

Through physiological regulation and genetic modification, entities strive not only to maintain their own steady state, but also to exert homeostatic control over their environment. Since entities adapt using physiological regulation and genetic modification, adaptation can be divided into two categories, intrinsic and extrinsic. Intrinsic adaptation is evolution of the physiological state, which occurs automatically as a result of the dynamics of a system caused by evolution of many interacting subsystems. Extrinsic adaptation on the other hand, is genetic modification, which is the evolution governed by a specific fitness function [9]. The fitness function is the determining factor that decides which entities are fit and unfit.

AL is considered to be intrinsic, since it allows for physiological regulation. Physiological regulation is evident as entities achieve homeostasis which implies that they have found the equilibrium of energy consumption, metabolism, mobility, and procreation of viable offspring. At a more basic level, AL is defined by [7] as a new science dedicated to the invention and study of artificial life,
i.e., entities that simulate, emulate, or instantiate structures and processes characteristic of life forms. AL can provide insight into real living systems and provide new methods of chemical synthesis, self-improving techniques for controlling complex systems, and ways to automatically generate optimally tweaked computer programs. In the future, AL will lend itself to robotics, virtual reality, and the retrieval of information from unmanageably huge databases [14].

In general terms, GAs attempt to find a maximum on the solution curve of a problem without exhaustively solving for every possible combination of variables [17]. Our application of GAs will use a direct genetic modification to bring about adaptation and therefore are considered to be extrinsic. GAs begin by generating an initial population of random entities. In order to introduce variation into the population, the highest fit entities are selected from the population to breed and produce the next generation of entities. The selection of highly fit entities is accomplished by the fitness function, which evaluates who is fit and who is not. In order to avoid local optima or premature convergence, the concept of mutation is applied to the offspring of the selected entities, which perturbs the alleles of the genes. Using generational replacement, the offspring replace parents in population for the next round of evaluation [1].
Survey of Applications of Artificial Life Games

Although the field of AL is rather new, many application exist including techniques for controlling complex systems, virtual reality, information retrieval, and entertainment. This thesis will examine the use of AL in a combat game. Several well known AL games already exist including Conway’s Game Of Life, Rucker’s Artificial Life Lab, Karakotsios’s SimLife, and Maxis’s line of simulation games.

Probably the most famous game application using AL is Conway’s classical Game Of Life. The Game Of Life is described as a game, but unlike common computer games, nobody ever wins or loses. By using a set of rules and cellular automata, Conway was able to produce an artificial world defined by its own physical laws. The Game Of Life is like a spreadsheet program where each cell’s actions is defined by its neighbor. This almost makes the player’s role nonexistent [12].

Another similar game which uses concepts from Conway’s Game Of Life is Rucker’s Artificial Life Lab. Artificial Life Lab allows the player to create and mutate artificial creatures. Rucker, like Conway, uses cellular automata to give his creatures life like status. Rucker also folds in the concepts of GAs which enhances progression of his cellular automata universe, giving the player the ability to interactively explore the world of artificial life [14].
A third game using AI techniques is Karakotsios's SimLife. SimLife is not so much a game as an exercise in desktop biology, an opportunity to assume the role of a digital deity in a biosphere that plays by digital rules. The science behind this game is so rich that SimLife could actually be used as a tool to advance our understanding of biology. Even a casual player will find hours of diversion creating mutant forms of life, putting them down on Earth and letting nature take its course [5].

Maxis, the creator of SimCity, has produced a line of entertainment simulation games based on AI concepts. These include El-Fish which allows you to cross-breed or mutate exotic fish in an electronic aquarium, SimFarm which allows players to try to build their own large or small farm, and SimAnt which puts you in charge of an ant colony [6].

Artificial Life Applied to a Dynamic Feedback Game

The common thread that binds all of the above examples together, is the fact that the player is really just a student or an inquisitive scientist. There is no purpose or goal, only to start the simulation and see where it goes. In this study we consider a combat game in which the player aims to destroy the entities without getting destroyed. AI and GA techniques will be used to create entities that (1) become better as the game progresses by adapting to the player's strategy and skill, and (2) are more interesting and unpredictable than randomly generated entities.
Another drawback to the entertainment type AL simulations is that they are based on cellular automata. Cellular automata are data structures composed of homogeneous finite-state machines. The data structure is typically a one, two, or three dimensional array. The state transitions are without variation, no randomness or control structures are allowed. Each state transition is defined by a set of predetermined rules involving a cell's current state and the states of neighboring cells. This design does lend itself to chaotic pattern generation characteristic of biological patterns, but does not provide for a combat game simulation [8].

For these reason we have chosen to apply AL to a combat game simulation which provides a dynamic virtual environment where entities can adapt and evolve in response to a simulated player. An initial population of entities is created and interacts with a simulated player who provides the stimuli that cause the population to evolve.

This dynamic feedback artificial life game (DFALG) also has an underlying scenario which provides for a more interesting environment. In a 2 dimensional virtual environment a simulated player has a food source which is used to attract entities. The entities require food for energy since they use energy for mobility, attacking, mating, firing shots, and feeding offspring entities. The objective is for the player to destroy the entities by
firing shots at them as they enter the field of view, which is a perimeter around the player. The entities retaliate to the predation of the player, by returning firing.

Mechanics of DFALG

In order to understand the subtleties and underlining features of DFALG, which produces adaptation of entities in a virtual environment, we must look at the mechanics of DFALG. DFALG entities use a simple finite state machine to determine a current condition or action. The states are normal living, feeding of young, mating, and attacking. Normal living is the most common state and all other states must return to normal living before entering another state. Each state has rules that determine the entities actions and responses to the environment. All characteristics of the entities are determined by the alleles of the genes in their genotype. A detailed description of these genes can be found in APPENDIX A and are referenced here as G1 through G20.

DFALG is a prototype combat game simulation that uses a technique known as a game tuning shell, which allows the developer to manipulate certain characteristics of the virtual environment to obtain the desired results. The set of DFALG control parameters are listed in APPENDIX B and are referenced here as CP1 through CP36.

Normal living in DFALG is the most common of all states and the hardest to model. The number of rules that could be
used to model normal living are numerous and often hard to define. DFALG uses the normal living state as the main transition between all other states and therefore delegates some of its rules to other states. The following rules define the normal living state of DFALG:

1. Excludes all other states.
2. Includes loss of energy by a base amount determined by control parameter CP17.
3. Energy loss is enhanced by the allele of gene G19.
4. Leads to all other states, including death.
5. Includes non-attacking motion determined by the phenotype.
6. Child entities grow to adults; period of childhood is determined by gene G1.
7. Entities age and can only live normally to an age determined by the allele of gene G10.

Earlier it was pointed out that one of the characteristics of life is that it must be organized. DFALG defines this characteristic to mean the ability of adult entities to feed child entities. Feeding has been modeled using the following rules:

1. Only adult entities can feed child entities.
2. Adult entities lose the energy they feed to child entities.
3. Adult entities will not give up the last of their food to a child entity, therefore, death cannot occur from feeding.
4. Willingness to feed is determined by the allele of gene G11.
5. Adult entities can only feed offspring within their own species, where the species threshold is determined by control parameter CP25.

6. Adult entities can only feed offspring if within a distance determined by control parameter CP29.

7. Adult entities get extra life for feeding offspring.

The entities require food for energy since they use it for all functions of life. The quest for food is what brings them closer to the player. When an entity enters the field of view of the player, the entity's state becomes one of attack. The following are the underlying rules that govern entity attacks on the player:

1. Entities get energy by maneuvering close to the player.

2. Energy is absorbed at rate of increase equal to the inverse square of the range from the entity to the player.

3. Absorbed energy is enhanced by the metabolism gene G14.

4. Maneuvering is controlled by the alleles of genes G2 through G9, which control velocity, acceleration, rates of change in velocity and acceleration, and determination of the rate of change to be random or constant throughout the attack.

5. Error in the x and y vectors of the shots fired are determined by the allele of genes G12 and G12.

The driving force behind adaptation in DFALG is the ability of entities to produce offspring. AL, which is intrinsic adaptation, uses mating to propagate its species. GAs, which are extrinsic, use selective breeding.

AL uses natural selection by allowing only those entities which are within the same species and have an overall similar fitness to mate. Once offspring are born,
they are placed into the population with the parents. The underlying rules of mating are as follows:

1. Can only mate if within a certain distance which is determined by control parameter CP9.

2. If two entities are within mating distance, they must pass speciation and fitness thresholds determined by control parameters CP25 and CP27.

3. Cannot mate while attacking.

4. Interval between mating determined by the allele of gene G20.

5. Mating duration determined by the allele of gene G15.

6. When mating occurs probability of gene crossover is determined by control parameter CP12.

7. When mating occurs probability of mutation within the gene is determined by control parameter CP13.

8. Mutation is also affected by ages of the parent entity using an exponential function.

9. The probability that two entities will mate regardless of speciation and fitness thresholds is determined by the allele of gene G16.

GAs uses selective breeding by selecting only those entities with the high fitness. This is accomplished through the use of a fitness function, which determines who is fit and who is not. Once offspring are born, they replace their parents in the population, a technique called generational replacement. The underlying rules of breeding are as follows:

1. Interval between breeding is determined by control parameter CP4.

2. The population does not increase.

3. Distance between entities and speciation does not matter.
4. Use only the fitness to determine which entities to breed.

5. When breeding occurs, probability of gene crossover is determined by control parameter CP1.

6. When breeding occurs, probability of mutation is determined by control parameter CP2.

Several references to death were made during the discussion on the different states of DFALG. Death is an important feature and must play a role in any system that exhibits the process of evolution. Evolution involves a continuing iteration of selection, which implies differential death. In natural life death occurs as a result of accident, predation, starvation, disease, or if these fail to kill the entity, it will eventually die from senescence resulting from an accumulation of wear and tear at every level of the entity [13]. DFALG on the other hand, models death as a subset of the above. It uses predation, starvation, and senescence.

Death by predation is one of the key characteristics that makes DFALG a game and not an entertainment simulation. Predation allows for a winner. Entities are the natural prey of the player. They attack the player for food or face starvation. Another way to view this is the player is a hunter and his food source is the bait.

Starvation is another means of death in DFALG. Entities must move within range of the player to obtain food. Food energy is absorbed by the entity at a rate equal to the inverse of the square of the range to the player.
This concept is carried deep within DFALG since the ability to outmaneuver the player is directly controlled by the genotype driven by the alleles of the genes. Only after several generations of evolution will entities with the optimal maneuvering characteristics emerge.

Finally, death can occur by senescence. Each entity has a gene that encodes maximum life span. This time period can be cut short by the other two means of death or it can be extended based on the amount of feeding that was carried out in its lifetime.

Fundamental Areas of Interest

In keeping with the objectives of this study, certain areas will be the main points of interest. These areas are fitness, death rate, life span, hit ratio, energy level, and speciation. Fitness will be used as an overall judgement of adaptation, whereas death rate and life span will give key insight to population dynamics. Energy level and hit ratio will be used in relation to game dynamics. Finally speciation will be used to show evolution and adaptation.

There is a special relationship between adaptation and fitness. Adaptation refers to the condition or state of fitness of an entity at a certain time under a given set of conditions, whereas fitness is the overall ranking of how an entity is adapting. This leads to four characteristics of fitness:

1. The fittest survive, therefore those who survive are the fittest.
2. Fitness cannot be based solely on the grounds of superiority in size and strength.

3. Survival implies being able not merely to live but also to produce offspring. Less fit entities do not necessarily die from predation, starvation, or senescence. Entities may live just as long as their fitter co-existing entities, but produce fewer or less fit offspring.

4. Fitness is determined largely by the particular set of circumstances under which the entity lives. Should the environment alter, there will be a concomitant change in the requirement for survival.

Fitness therefore, requires a combination of characteristics that make possible effective propagation under the given environmental conditions.

The same function is used to determine fitness in both intrinsic or extrinsic adaptation. The difference is how the fitness ranking is used. Intrinsic adaptation uses fitness as a threshold to determine whether or not two entities share a common fitness ranking. Extrinsic adaptation uses fitness as a determination of selecting the best to breed to produce the next generation. Unless otherwise specified, DFALG uses the following fitness function:

\[
\text{FITNESS} = (20 \times \text{HIT RATIO}) + (\text{LIFE SPAN}) + (2 \times \text{BATTLE TIME}) + (10 \times \text{SHIELD COUNT}) + (2 \times \text{ATTACK COUNT}) + (20 \times \text{ENERGY LEVEL})
\]

where shield count is the number of hits an entity can withstand before death, and attack count is the number of attacks the entity has made against the player. The purpose of the fitness function is to assign a high fitness ranking to those entities that have scored hits against the player,
have high longevity, have not been hit by the player, have aggressively attacked the player, and have a high energy level. This fitness function rewards more for hit ratio and energy level, two of the characteristics crucial to DFALG entity evolution. The reason for using battle time, shield count, and attack counts in the fitness function is to punish those entities that do not attack the player or who unsuccessfully attack the player. It is of interest to examine the average fitness of all dead entities at a given point in the simulation.

As discussed above, death can occur from predation, senescence, or starvation. Useful insight into the simulation can be gained by examining the mortality count. Mortality count can show periods of survival of the fittest when the rate of mortality count appears to increase quadratically. It can also show the beginning stages of homeostasis as the rate slows.

Life span is another good indicator of adaptation. It is defined to be the number of simulation updates between the creation of the entity and its death. It is of interest to examine the average life span of all dead entities at that given point in the simulation. In normal adaptation, this value will increase as a function of time and show periods of survival of the fittest and homeostasis.

Energy level describes the amount of energy that is contained by an entity. It is of interest to examine the
average energy level of all dead entities at a given point in the simulation. The average value will give indication of successful attacks, which will supply a large amount of food, and feeding, which will help the offspring live to adulthood.

Hit ratio is defined to be the number of successful hits on the player divided by the number of shots fired by an entity. It is of interest to examine the average hit ratio of all dead entities at a given point in the simulation. As mentioned during the explanation of fitness, hit ratio contributes to determining which entities are fit and unfit. This added dimension will provide for evolutionary changes in hit ratio.

Speciation in DFALG is defined as the amount of variation allowed between two entities before they are considered to be in separated classes of species. Speciation is calculated as the difference between the genotypes of two entities in relation to all the genotypes of the population. This is accomplished by taking the dot product between the chromosome vectors of two entities, which determines the amount of variation between them and comparing this value to the percent difference of the largest genotype allowed.

Speciation is important to the game designer, and ultimately the player, because a variety of entities will make the game more exciting. The classes of species
produced will be **polymorphic**. That is, they will look similar, but may be slightly different. Speciation will provide for several different classes with different characteristics. Each class will adapt to the player’s combat strategy and skill in different ways [16].
ADAPTATION AND ARTIFICIAL LIFE

In the simulation the criterion for successful adaptation, whether be extrinsic or intrinsic, is simple and unequivocal: survival. Any population that fails to cope with changes in its environment becomes extinct. On the other hand, in successfully meeting the challenges of habitat alterations, a population can evolve into something new and descendants may differ markedly from their ancestors. We will first look at intrinsic adaptation in our virtual environment using AL. Next, the stability and optimization of propagating populations will be compared to that of non-propagating populations. Finally, we will consider the effects of varying key AL control thresholds.

Increasing Adaptation Using Artificial Life

It will be shown later in the chapter that increasing adaptation can be achieved by using AL techniques. The focus will be on the key characteristics of fitness, life span, energy level, and hit ratio. Recall that fitness of an entity is defined to be the score derived by a weighted function which rewards an entity for exceptional behavior and punishes for poor behavior, life span of an entity is defined to be the number of simulation updates between the creation of the entity and its death, energy level is
defined to be the amount of energy an entity has at a given point in time, and hit ratio is the ratio of the number of hits scored to the number of shots fired by that entity. Experimentation shows that as these characteristics stabilize the entity population achieves homeostasis. **Homeostasis** is a relatively stable state of dynamic equilibrium or tendency toward such a state.

Fitness will be considered first. Since fitness is the score derived by the weighted fitness function, which

![Figure 1. AL - Fitness and Mortality Count](image-url)
reflects effectiveness of the entity’s mapped strategy
defined by its phenotype, it initially starts out with a
stepwise characteristic as shown in figure 1. This stepwise
characteristic in period A, a period of survival of the
fittest, is explained by the following: (1) the instability
of the populations as they adapt to the environment, in
otherwords, the overall fitness of each entity dying during
this period is more likely to have a larger dispersion in
frequency and (2) the contribution of the other key
characteristics. It has already been shown that life span,
energy level, and hit ratio contribute to the fitness of an
entity. The contribution of each of these characteristics
will be examined later. The fitness function then begins to
stabilize. This correlates directly to the population of
entities as they adapt and try to achieve a homeostatic
state. After period A has ended, signified by the
mortality count changing from a near quadratically
increasing trend to an almost linearly increasing trend
[10], fitness also increases linearly and becomes stable.

Life span will be considered next. Life span is the
period of time between the creation and death of an entity.
Increasing life span provides strong evidence for the
increasing adaptation of entities. Initially, entities
exhibit a very short life span, which is explained by
natural selection, as shown in figure 2. As one would
expect, weak entities are dying off very quickly, which also
Figure 2. AL - Life Span and Mortality Count

contributes to the almost quadratically increasing mortality count. During period A the entities strive for adaptation and life span exhibits the stepwise characteristic described earlier. As period A ends, life span enters the beginning stages of homeostasis. In order to obtain the same amount of adaptation during homeostasis, the simulation must run twice as long as it did during period A.

Next, energy level is considered. Energy level has direct relationship to the phenotype. As mentioned earlier, the amounts of energy lost for normal movement, mating, feeding, and attacking are driven by alleles of genes in the
genotype, but other characteristics such as the ability to maintain a lengthy attack as well as belonging to a strong and large species which will enable a young entity to obtain more energy are driven by the phenotype and will all contribute to the energy level. For the purpose of this simulation the initial energy level randomly starts from 0.5 to 1.0. This explains the initial large value for energy level as seen in figure 3. As expected, energy level also exhibits the stepwise characteristic of period A. Recall, energy level contributes to the fitness function, where the stepwise characteristic was originally observed. Also
observe that after period A, the trend of the energy level is downward and stabilizing. This is explained by the entities achieving their homeostatic state as the entities adapt to the strategy and skill of the player. In order to maintain high levels of energy the entities would have to aggressively attack the player, but this aggressiveness will result in a high mortality count. Therefore, the entities must maintain a balance between attacking and energy consumption.

Figure 4. AL - Hit Ratio

Finally, hit ratio is considered. Hit ratio is a feature specific to DFALG. In order to maintain some
consistency in the virtual environment, hit ratio was
developed using genotype as well as phenotype information.
Therefore, the dramatic fluctuations in hit ratio, as seen
in figure 4, during period A, are contributed to the way hit
ratio is formed. Since it is a combination of several
attributes from the phenotype and genotype, it magnifies the
instability found during this period. But not even the
instability of hit ratio can withstand the forces of
adaptation. As the entities begin to reach homeostasis and
the period of survival of the fittest is ending, hit ratio
becomes stable.

All of these characteristics, fitness, life span,
energy level, and hit ratio are the basic attributes of
DFALG, which is based on genetic and biological paradigms.
In order to compare and contrast other features of DFALG,
the effects of a non-propagating population applied to DFALG
must be examined. This was done by removing all
possibilities of mating or breeding. Without mating or
breeding there is no intrinsic or extrinsic adaptation,
which essentially means no evolution. By examining the key
characteristics as compared to the same runs using AL in
DFALG a clear understanding of the advantages of AL will be
seen.

Simulations Using a Non-propagating Population

Not using intrinsic or extrinsic adaptation will result
in less than optimum performance from the entities due to
the lack of evolution. For the purposes of this simulation model this was achieved by not allowing entities to mate or breed. In other words, no propagation of the species, directed by either method, intrinsic or extrinsic. To enhance the non-propagating condition, when the entity population decrease there is a 50% probability that the simulation will create entities at random. This is a common technique used in game designs and provides a more competitive virtual environment. Using the key characteristics the non-propagating condition will be compared to AL using intrinsic propagation. In contrast to AL using propagation, non-propagation will show a definite lack of ability to reach homeostasis.

The lack of ability to reach homeostasis is shown in the fitness characteristic in figure 5. The fitness climbs slowly as those fit enough to survive the adaptation period get scored. Although it looks as if the fitness of the non-propagating population is roughly equivalent to fitness of the propagating population recall that this is the period of survival of the fittest and both populations are losing weak entities. This is supported by each population starting with the same random initial population. As what would normally be the end of the period of survival of the fittest approaches, fitness for non-propagation entities continues to grow at its current rate, but the propagating population begins to diverge and emerge at a higher fitness. This is
due to the fact that the propagating population has started to evolve and the non-propagating population is only being repopulated through random creations. As the period of survival of the fittest ends and the propagating population is continue to evolve as a function of the environment and the non-propagating entities tend toward a stable fitness supported by the stronger more fit entities from the random initial population. Fitness then begins to stabilizes but eventually starts to decrease as weak entities are created, which ultimately lower the average fitness. Remember, in an evolving and adapting population fitness should continue to
improve since the population of entities never stops evolving, it just slows its rate.

The population eventually becomes stable as the

Figure 6. Non-propagating vs Propagating (Mortality Count) simulated player is able to maintain a balance between the random creations and the rate at which the entities are destroyed as shown in figure 6. After period A when the propagating entities' mortality count appears to slow and become linear, the non-propagating entities mortality count also becomes linear but at a higher rate of increase. When the simulation ends the ratio of non-propagating deaths compared to propagating deaths is almost 2 to 1.
Another way to view this is through life span. During period A the non-propagating population exhibits the same period of survival of the fittest as the weaker entities are dying. The difference is that during period A the propagating entities are starting to produce more adapted offspring whereas the non-propagating entities are being repopulated only with random entities. In one aspect, the initial population of non-propagating entities are evolving as the less fit entities are weeded out, but they fail to

---

**Figure 7. Non-propagating vs Propagating (Life Span)**
adapt by producing more fit offspring. As shown in figure 7, when homeostasis begins for the propagating entities, they continue to evolve and adapt throughout the simulation. The non-propagating entities only stabilize and actually start to decline, which indicates the simulated player is winning.

The energy level of the entities also adds insight into the stabilization of the non-propagating entities. As shown in figure 8, the energy level of the entities, although decaying at the same rate, begins to decay sooner
than AL propagating entities. This is attributed not only to the inability of the non-propagating entities to adapt, but it is also explained by the random creation of the non-propagating entities. Since the non-propagating entities are created at random, their initial energy level can begin much lower than the energy level of offspring from the propagating entities who give their energy to the offspring to insure their survival. Without these features, the non-propagating entities begin to decay sooner than AL propagating entities. After decay sets in, both populations of entities exhibit the same basic trend. As the non-propagating entities are unable to achieve homeostasis within the virtual environment, the rate of decay increases. As expected, the energy level also stabilizes.

Finally, the comparison between both populations types shows significant improvement of the hit ratio of AL propagating entities over non-propagating entities. As before, this is attributed to the random creation of non-propagating entities which prevents them from achieving homeostasis. From the beginning of the simulation, non-propagating entities fail to produce significant hits. Although, trends in improvement are seen in figure 9 they are quickly dampened by the random creation.

**Variation in Artificial Life Control Parameter Thresholds**

Variation of AL control parameter thresholds will result in different results of various DFALG attributes and
characteristics. The key AL control parameters which direct the adaptation of entities are speciation and fitness. Speciation is defined as the amount of variation in the genotype of an entity that is allowed before it will be considered part of a species. Fitness, on the other hand, is the score derived by the fitness function which is controlled by the phenotype. Experimentation shows that variation in these two key control parameters will correlate to different final values of the key and other simulation characteristics.
Twelve simulations were run varying only the speciation threshold and holding the fitness threshold constant. The speciation threshold is the amount of variation allowed between species. A smaller threshold indicates tighter restrictions. Varying the speciation threshold has several affects on several characteristics of the entities.

Overall fitness tends to stay the same during the simulation, indicating that throughout the simulation, several of the characteristics counter one another. An example of this is the energy level and hit ratio. As the threshold is decreased, allowing for more probability of cross species breeding, these two characteristics decrease. This is explained by the loss of direction of the evolution of the population. Without a good speciation threshold, entities will interbreed with other species who have traits that will counter good traits of the first entity. This is also supported by the number of attacks necessary to achieve the same amount of food. The more chaotic breed population needs to attack more frequently to obtain the necessary level of energy to prevent starvation.

The chaotic tendency of the population caused by a low speciation threshold can be correlated to the number of mates and offspring. The lower the speciation threshold, the higher probability that entities will mate. As shown in table 1, the number of mates and offspring increase as the speciation threshold is lowered.
Another important factor that is produced from speciation threshold variation is polymorphism. Polymorphism is many diverse entities that exist within the same species, as shown by the decrease in the number of beginning and ending species. With high speciation thresholds more distinct species are categorized. With lower speciation threshold the same random initial population is considered more similar.

Table 1. Speciation Threshold Variation

<table>
<thead>
<tr>
<th>% VAR</th>
<th>FT</th>
<th>LS</th>
<th>EN</th>
<th>HR</th>
<th>MT</th>
<th>OF</th>
<th>SP</th>
<th>AT</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>241.3</td>
<td>83.71</td>
<td>.365</td>
<td>.227</td>
<td>159</td>
<td>267</td>
<td>10/6</td>
<td>754</td>
</tr>
<tr>
<td>20</td>
<td>246.0</td>
<td>88.3</td>
<td>.358</td>
<td>.248</td>
<td>166</td>
<td>271</td>
<td>7/6</td>
<td>1069</td>
</tr>
<tr>
<td>25</td>
<td>235.3</td>
<td>91.7</td>
<td>.320</td>
<td>.228</td>
<td>201</td>
<td>277</td>
<td>4/5</td>
<td>1013</td>
</tr>
<tr>
<td>30</td>
<td>229.1</td>
<td>74.6</td>
<td>.314</td>
<td>.229</td>
<td>281</td>
<td>377</td>
<td>3/3</td>
<td>1187</td>
</tr>
<tr>
<td>35</td>
<td>253.5</td>
<td>91.0</td>
<td>.288</td>
<td>.214</td>
<td>190</td>
<td>267</td>
<td>3/2</td>
<td>894</td>
</tr>
<tr>
<td>40</td>
<td>245.1</td>
<td>93.2</td>
<td>.276</td>
<td>.177</td>
<td>176</td>
<td>259</td>
<td>2/2</td>
<td>1079</td>
</tr>
<tr>
<td>45</td>
<td>246.1</td>
<td>95.9</td>
<td>.277</td>
<td>.201</td>
<td>234</td>
<td>269</td>
<td>2/2</td>
<td>813</td>
</tr>
<tr>
<td>50</td>
<td>246.4</td>
<td>87.8</td>
<td>.282</td>
<td>.193</td>
<td>251</td>
<td>298</td>
<td>2/2</td>
<td>1125</td>
</tr>
<tr>
<td>55</td>
<td>244.5</td>
<td>93.6</td>
<td>.227</td>
<td>.174</td>
<td>192</td>
<td>263</td>
<td>2/2</td>
<td>1244</td>
</tr>
<tr>
<td>60</td>
<td>288.9</td>
<td>70.2</td>
<td>.304</td>
<td>.213</td>
<td>296</td>
<td>406</td>
<td>2/2</td>
<td>1272</td>
</tr>
<tr>
<td>65</td>
<td>226.3</td>
<td>79.6</td>
<td>.316</td>
<td>.178</td>
<td>257</td>
<td>319</td>
<td>1/1</td>
<td>778</td>
</tr>
<tr>
<td>70</td>
<td>233.7</td>
<td>89.8</td>
<td>.271</td>
<td>.199</td>
<td>235</td>
<td>290</td>
<td>2/2</td>
<td>928</td>
</tr>
</tbody>
</table>

FT = fitness  LS = life span  EN = energy level
HR = hit ratio  MT = number of mates  OF = offspring
SP = beginning/ending species  AT = entity attacks

Next considered are the effects of variation on the fitness threshold. The same twelve simulation were run varying only the fitness threshold and holding the speciation threshold constant. Since fitness is a derived
score from a fitness function, a higher threshold indicates similarly fit entities.

As shown in Table 2, fitness threshold variations have little affect on the overall fitness of the entities for the same reasons as the speciation threshold variation. The subtle differences come between beginning and ending speciation and the impact of the fitness threshold variation on a constant speciation threshold.

At first, the impact that fitness threshold variation has on beginning and ending speciation is slight. As the fitness threshold is lowered, allowing less fit entities to mate with entities of their same fitness and higher, ultimately produces an average population of similar species. Normally, as shown in AL speciation, as natural evolution takes effect the less fit species die off. As the fitness threshold is lowered and a more average population is produced, the ending speciation climbs back to the initial level.

This concept of an average species also explains the slight variations in the key characteristics. Unlike speciation threshold variation which produced chaotic variations in the key characteristics, the generality of the population tends to hold the characteristics constant with only slight differences.

At first, this information may appear small and insignificant. But to a game designer, this information can
hold a multitude of opportunities to design more complex and subtle games. The interaction of the different variations produce different effects. Variations of the speciation threshold produce a variety of conditions from stable evolution to a chaotic existence. Fitness threshold variation on the other hand, can produce conditions ranging from a stable evolution to a more general population with less diversity. These options give the game designer the ability to design games with various attributes. For example if designer wants a very aggressive entity and is willing to sacrifice energy levels and the formation of new species, then a 40% variation in speciation could be used.

Table 2. Fitness Threshold Variation

<table>
<thead>
<tr>
<th>% VAR</th>
<th>FT</th>
<th>LS</th>
<th>EN</th>
<th>HR</th>
<th>MT</th>
<th>OF</th>
<th>SP</th>
<th>AT</th>
</tr>
</thead>
<tbody>
<tr>
<td>70</td>
<td>241.3</td>
<td>.365</td>
<td>.227</td>
<td>159</td>
<td>267</td>
<td>10/6</td>
<td>754</td>
<td></td>
</tr>
<tr>
<td>65</td>
<td>242.8</td>
<td>.356</td>
<td>.209</td>
<td>147</td>
<td>258</td>
<td>10/7</td>
<td>926</td>
<td></td>
</tr>
<tr>
<td>60</td>
<td>239.1</td>
<td>.369</td>
<td>.203</td>
<td>187</td>
<td>248</td>
<td>10/7</td>
<td>858</td>
<td></td>
</tr>
<tr>
<td>55</td>
<td>243.7</td>
<td>.375</td>
<td>.205</td>
<td>188</td>
<td>230</td>
<td>10/6</td>
<td>838</td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>239.5</td>
<td>.378</td>
<td>.212</td>
<td>185</td>
<td>229</td>
<td>10/6</td>
<td>763</td>
<td></td>
</tr>
<tr>
<td>45</td>
<td>234.0</td>
<td>.383</td>
<td>.212</td>
<td>178</td>
<td>240</td>
<td>10/8</td>
<td>825</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>244.5</td>
<td>.383</td>
<td>.191</td>
<td>207</td>
<td>256</td>
<td>10/8</td>
<td>890</td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>239.2</td>
<td>.359</td>
<td>.208</td>
<td>181</td>
<td>252</td>
<td>10/7</td>
<td>826</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>237.9</td>
<td>.375</td>
<td>.218</td>
<td>225</td>
<td>281</td>
<td>10/6</td>
<td>734</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>234.9</td>
<td>.377</td>
<td>.222</td>
<td>210</td>
<td>257</td>
<td>10/9</td>
<td>938</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>240.3</td>
<td>.375</td>
<td>.215</td>
<td>198</td>
<td>259</td>
<td>10/10</td>
<td>1033</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>240.3</td>
<td>.375</td>
<td>.215</td>
<td>198</td>
<td>259</td>
<td>10/10</td>
<td>1033</td>
<td></td>
</tr>
</tbody>
</table>

FT = fitness  LS = life span  EN = energy level
HR = hit ratio  MT = number of mates  OF = offspring
SP = beginning/ending species  AT = entity attacks
CHAPTER 3

ADAPTATION AND GENETIC ALGORITHMS

In contrast to intrinsic adaptation (adaptation by natural selection and evolution), extrinsic adaptation describes adaptation using selective evolution or breeding. It has been discussed that entities often adjust to changing conditions through appropriate physiological adjustments over the course of time through natural selection and evolution, which is the basis for intrinsic adaptation. But adaptation can also be achieved through extrinsic methods. These methods of selective evolution are simulated using GAs.

Recall that GAs are based on the mechanics of natural selection and through the use of a predefined fitness function are used to find optimal solutions of functions and processes [3]. The term GA can be misleading, since GAs do not use all of the dynamics of AL. They have although proven to be very powerful optimization tools which incorporate features inspired by the mechanisms of reproduction, allele variation, and selection found in natural evolution [18]. GAs will be compared to AL using the key characteristics. Then variation in GAs breeding intervals will be discussed. Finally, variations in the fitness function will be examined.
Genetic Algorithms Compared to Artificial Life

GAs optimize entities similar to that of AL through phenotype selection and genotype variation. At another level, the comparison of GAs to AL provides for verification of the optimization of AL [15]. Using the key characteristics and a 100 simulation update breeding interval (applying GAs every 100th simulation update), it was shown that GAs provide similar result to that of AL.

Figure 10. GAs vs AL (Fitness)

GAs and AL are characterized by similar trends in their overall fitness. Figure 10 shows that this is correct. Although the slight variation towards the end of period A
and in the later half of the simulation indicates that the GAs provided a slightly different solution (where a solution is a population of entities), the relative difference is only 3%.

Next, the mortality count was compared between GAs and

![Graph showing mortality count for GAs vs AL](image)

**Figure 11. GAs vs AL (Mortality Count)**

AL as shown in figure 11. At first it looks as if the GAs have produced an optimal population early on in period A, characterized by a very low mortality count, but this can be misleading. Recall that GAs use generational-replacement, which replaces all parents with all offspring. Therefore,
at every 100th simulation update the current generation is replaced by the offspring of that breeding population. If all parents replaced by offspring were considered part of the mortality count, the expected stepwise increase at every 100th update would mask the mortality count by predation, starvation, and senescence.

When examining the life span characteristic it appears for the most part that the same general trend is again followed between GAs and AL as show in figure 12. One thing is apparent, as the simulation progresses, the life span of

![Graph showing life span comparison between GAs and AL](image)

Figure 12. GAs vs AL (Life Span)
the GAs becomes progressively higher. This exemplifies one of the fundamental differences of GAs and AL. Remember that GAs are extrinsic in nature, they force evolution and steer the solutions towards an optimal point, whereas AL follows natural selection, which will strive for a steady state or homeostasis. AL will optimize during its period of survival of the fittest and then continue to evolve slowly as a function of its environment. GAs on the other hand, will react based on the fitness of the current population of entities. Although the life span of the optimized GA entities is increasing, recall that the overall fitness varied only slightly. It will be shown next that other characteristics were optimized differently to obtain the increase in life span.

When examining the energy level, the strength of the fitness function in GAs is clear. As mentioned earlier, the default fitness function rewards for high energy levels. This is observed in figure 13 during period A, as GAs quickly optimize on this characteristic and closely maintain this value during the simulation. AL on the other hand, maintains a higher level during period A, fluctuates as entities strive for homeostasis, and eventually stabilizes (at a much lower level than does GAs) as it is achieved.

Finally, the review of hit ratio in figure 14 shows that GAs do not optimize to the same entity population as does AL. This is in part due to the way that hit ratio is
Figure 13. GAs vs AL (Energy Level)

determined using information from the phenotype as well as the genotype. This gives AL an advantage, since phenotype and genotype information is rooted strongly in AL dynamics. GAs can only react to the population since they are extrinsic. They make alterations to the genotypes and hope that improved behavior will emerge from the phenotype. This characteristic can be improved for GAs through modifications to the fitness function, which will be discussed next.

**Variations in Breeding Periods of Genetic Algorithms**

GAs find optimal or new optimal solutions base on how fit those solutions are as defined by a predetermined
fitness function. Another factor that affects the final solution produced by GAs is the number of breeding cycles. GAs are run sequentially, generation after generation. In their application to DFALG, GAs are used as an optimize. Therefore, how often the optimizing GAs are run becomes an issue. In the simulation the breeding intervals of 100, 300, and 500 simulation updates were selected.

The fitness characteristic shows the affects of varying the breeding period. Figure 15 shows the effects of the variation of the breeding period on fitness. The 300 update interval, which is called GA(30), produces the optimal
effect. This is solely because of the information of the population of entities at the time of breeding. Remember that GAs react to the fitness of the entity, which is driven by its genotype. The 100 and 500 update intervals, which are called GA(10) and GA(50) respectively, also optimize to a fitness close to GA(30). This closeness in overall fitness is again explained by the different solutions obtained from variations of the same approach. None of the solutions are right or wrong, but offer different final optimized entity populations.
Next, observe the effects of the variations on mortality count in figure 16. Notice that the mortality count for GA(10) is much lower than that of GA(30) and GA(50). This is because GA(10) has optimized an entity population that is not as aggressive as GA(30) and GA(50). This is supported by the number of attacks by entities optimized with each variation. The total number of attacks for each breeding period variation of GA(10), GA(30), and GA(50) are 973, 1075, 1328 respectively. Notice that GA(30) and GA(50) have optimized an entity population that is much more aggressive than GA(10). The disadvantage to this
aggression is the exposure to death by predation, as shown in the different counts of mortality.

The first noticeable feature about the life span characteristic in figure 17 is the closeness in trend for each of the variations. If this is compared to the mortality count, it seems at first to contradict the large difference between GA(10) and the other two variations. This converse trend is explained by the solution produced by GA(10). As previously mentioned, GA(10) produces entities that are less aggressive which reduce their exposure to
death by predation resulting in a longer life span. On the other hand, since GA(30) and GA(50) optimize solutions that are more aggressive than GA(10), more of the population dies by predation. Yet at the same time, the life span of the entities is still equal to that of GA(10). This is supported by GA(10) having the largest ending population as compared to GA(30) and GA(50).

The energy level characteristic in figure 18 does not provide much, if any, significant information on its own.

Figure 18. GAs with Different Breeding Periods (Energy Level)
The true insight comes when using the hit ratio in conjunction with the energy level. The hit ratio in figure 19 explains the tightness in trend of the fitness characteristic. Since GA(10) has optimized a solution that resulted in a low hit ratio, its contribution will counteract its stronger features in the trend of fitness.

![Figure 19. GAs with Different Breeding Periods (Hit Ratio)](image)

The explanation of why GA(10) is the lowest of all variations is simple. Using the energy level characteristic shown in figure 18, GA(10) maintains the lowest energy level. This again is explained by the less aggressive solution optimized by GA(10). GA(10) entities attack only
long enough to obtain energy to sustain life and have less chance of dying by predation, as well as not being given enough time to attack the player. GA(30) and GA(50) have optimized to a more aggressive entity, although GA(50) entities are not as optimized as GA(30) entities. The species of more aggressive entities provide for higher hit ratios and higher energy levels, but ultimately result in higher mortality counts due to prolonged attacks with the player.

Variations in Fitness Functions of Genetic Algorithms

The fitness function is the essence of the GAs. GAs do not care about the problem or the working environment. It is up to the game designer to determine what is fit and unfit, and to make the decision of what is to be achieved. In order to show that the fitness function determines the characteristics of the simulation using GAs we have modified the normal fitness function to obtain a desired effect. Since it has already been shown that GA(10) produces a low hit ratio it will be used for variations in the fitness function that will ultimately produce a higher hit ratio.

The fitness function used for GAs(10), as described earlier during the discussion of the mechanics of DFALG, is repeated here for clarity:

\[
F_0 = (20 \times \text{HIT RATIO}) + (\text{LIFE SPAN}) + (2 \times \text{BATTLE TIME}) + (10 \times \text{SHIELD COUNT}) + (2 \times \text{ATTACK COUNT}) + (20 \times \text{ENERGY LEVEL})
\]
Next, the amount that hit ratio affects the fitness is increased as shown in the following variations:

\[
F_1 = (100 \times \text{HIT RATIO}) + (\text{LIFE SPAN}) + (2 \times \text{BATTLE TIME}) + (10 \times \text{SHIELD COUNT}) + (2 \times \text{ATTACK COUNT}) + (20 \times \text{ENERGY LEVEL})^2
\]

\[
F_2 = (\text{HIT RATIO})^2 + (\text{LIFE SPAN}) + (2 \times \text{BATTLE TIME}) + (10 \times \text{SHIELD COUNT}) + (2 \times \text{ATTACK COUNT}) + (20 \times \text{ENERGY LEVEL})
\]

\[
F_3 = (\text{HIT RATIO})^3 + (\text{LIFE SPAN}) + (2 \times \text{BATTLE TIME}) + (10 \times \text{SHIELD COUNT}) + (2 \times \text{ATTACK COUNT}) + (20 \times \text{ENERGY LEVEL})
\]

The effects from fitness function variations are shown in figure 20. During period A, the fitness of each variation initially starts out the same. This is to be
expected since each variation is using GA(10). Since GA(10) breeds every 100th simulation update, the first signs of affects are seen at this point. Once the first initial breeding takes place, the stronger (more emphasis on hit ratio, F2 and F3) fitness functions begin to dominate the simulation. As period A ends, F2 and F3 emerge as the dominate fitness functions and continues this dominance to the end of the simulation.

The hit ratios for the varied fitness functions can be seen in figure 21. The results are once again exactly as
expected. Fitness functions F2 and F3 provide the higher hit ratios. Recall from the variations of the breeding period that GA(10) using fitness function F0 produced a final hit ratio of .19, but now using F2 and F3 produce hit ratios of .26 and .24 respectively, which are increases of 37% and 26%.
CHAPTER 4

SPECIATION AND FITNESS

Previous chapters concentrated on the semantics of AL and GAs and how the population (made up of several classes of species) responded to the different approaches of adaptation. Chapter 2 varied the speciation threshold which is used to determine possible mating, but did not explain the outcome of speciation at the end of the simulation. Affects that extended time has on speciation and fitness will be examined. It will also be shown that the simulation can form new species from a single species. Finally, the affects of the virtual environment size on speciation and fitness will be examined.

Extended Runs with Emphasis on Speciation and Fitness

Using the same initial simulation parameters used in the basic AL simulations of chapter 2 as a basis and modifying only the maximum simulation run time resulted in the fitness trend as shown in figure 22. As discussed in chapter 2, the fitness trend increases during period A and then begins to slow after the entity population achieves homeostasis. The question lies in what happens beyond the basic simulation end time of 2500 simulation updates. As shown in figure 22, overall fitness continues to increase but at a slower rate. This is explained by the slow in
evolution and the continual success of the currently adapted population of entities. This also explains why overall fitness is not static or only slightly increasing. Recall from chapter 2 during the dissuasion of non-propagating entities, that a static or slightly increasing overall fitness is characteristic of a population of entities that has failed to adapt to the players skills and strategies.

Speciation has proven to be a very valuable concept in AL. It has also been shown to be equally valuable when applied to DFALG. Like fitness, it is of interest to examine the effects of an extended simulation on speciation.
As shown in figure 23, the initial random population starts with ten classes of different species. During the period A when survival of the fittest is taking place, the species count does not decrease. This is explained by one of the characteristics of fitness described earlier. Recall that survival implies not merely being able to live, but also to produce offspring. Less fit entities do not necessarily die from predation starvation or senescence. They may live just as long as their fitter co-existing entities, but produce fewer or less fit offspring. This implies that during period A all weak entities, regardless of what class of
species they belong, are dying. This does not imply that the entire species is weak, but does show that sufficient variation within a species is enough to weaken a particular entity.

As period A ends and homeostasis begins, the classes of species that contain less fit entities begin to weaken in the sense that they produce less offspring and are unable to maintain a viable species. Although homeostasis is achieved by the population as a whole, the regulation of individual classes of species is continually changing. The shift in census of each individual class of species is constant and frequent in the initial stage of homeostasis. As the simulation is allowed to continue, the frequency of species class shifting slows as fewer and stronger classes of species emerge.

**Emergent Species from a Single Species**

It has been shown in the previous section that through natural selection and evolution weaker, less viable species will lose the survival of the fittest and eventually die off. This act in the dynamics of the virtual environment allows for the expansion of stronger more viable species. If a few dominating species can evolve from a population of several species is the converse true; can species emerge from a single dominating species? New classes of species can be formed from a single species.
First, an initial population of entities was produced without variation. In other words, each entity is exactly alike. In order to obtain speciation from a cloned population, the probability of gene mutation during a mate was increased by a factor of 10. This allowed for a greater probability of gene variation within each mate.

It could be assumed that this would be the solution and a population of several small species would be produced. But like life itself, DFALG is more robust. It is true that several small species were produced as shown in figure 24. But these single member species were unable or unfit to mate
and eventually died out, which is characteristic of the noise in the species count. Eventually, over time, as mutated offspring are produced that are fit enough to survive and find a common mate, new stable species emerged.

![Comparison of Overall Fitness](image)

**Figure 25. Comparison of Overall Fitness**

In addition to the emergent species, the overall fitness also shares some interesting characteristics during evolution from a single species to multispecies population. As shown in figure 25, the fitness of the cloned initial species (signified by FITNESS 1) population varies from the fitness of the initial multi-species AL population (signified by FITNESS 2). During period A, while the
population contains only one or two species, the fitness increases rapidly. This is explained by the small amount of speciation (the amount of variation between species). Since the two existing species are so similar there are no weak entities pulling the fitness of the species down. As explained previously in chapter one, during the period of survival of the fittest and the initial stages of homeostasis the weaker entities are succumbing to natural selection. This give the fitness a slower increasing trend as compared to the fitness of the cloned population.

As entities mate and the similarity of entities within a species becomes more unstable, the fitness begins to slow its increasing trend. Since the initial population was cloned, the entities that have survived into period B share similar programmed maximum life span. This explains the jump in fitness after period B begins. The fact that these entities have survived this long indicates that they are fit, recall this was one of the initial claims about fitness discussed in the introduction. In addition to simultaneous death of the older entities, several new classes of species begin to emerge and then become extinct, as seen in period B of figure 25. This explains the stagnate fitness, as the unfit entities (created by the high mutation probability) struggle to survive but again succumb to natural selection. Notice that there are now three stable classes of species and as period B ends the fitness decreases slightly in
response to the death of the weaker entities of the stable species.

As period C begins the existing stable classes of species begin to exhibit homeostasis. As compared to FITNESS 2, the fitness trends are almost exact. Recall from the first section of this chapter that at the end of the extended simulation run the number of classes of species is exactly the same as the ending number of species for the initial cloned population run.

**Affects of Environment Size on Speciation**

Biologists tend to describe the number of species in a particular environment in terms of richness and diversity. The richness and diversity of species increases with environment complexity and closeness to a food source [16]. The fact that the virtual environment of the prototype combat game is relatively simple, leads to the conclusions that the number of species will not be as numerous as in a complex environment. This is supported by the preceding sections where the final number of species was small at the end of the simulation. The affect of making the virtual environment larger also has an adverse effect on the fitness of the entities in the species, since the food source is now farther away and harder to reach.

The virtual environment was increased using the ratio sizes as listed in table 3. These changes in virtual environment size allowed the entities to be affected by both
distance from food source and complexity in the environment. Although the affects of the simple virtual environment have already been shown, increasing the size of the virtual environment will only make it simpler, since entities will be less likely to mate, feed offspring, and attack the food source.

Table 3. Environment to Entity Ratios

<table>
<thead>
<tr>
<th>SIZE</th>
<th>UNITS OF SQUARE SPACE</th>
<th>UNITS OF SQUARE SPACE PER ENTITY</th>
<th>UNITS OF SQUARE SPACE PER ENTITY POPULATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>277200.0</td>
<td>5544.0</td>
<td>272.2</td>
</tr>
<tr>
<td>2</td>
<td>1000000.0</td>
<td>20000.0</td>
<td>1000.0</td>
</tr>
<tr>
<td>3</td>
<td>2250000.0</td>
<td>45000.0</td>
<td>2250.0</td>
</tr>
<tr>
<td>4</td>
<td>4000000.0</td>
<td>80000.0</td>
<td>4000.0</td>
</tr>
</tbody>
</table>

The results of the AL simulations using the four different environment sizes are summarized in table 4. As stated earlier, the distance from the food source will have an adverse effect on the fitness of the entities. As the size of the environment is increased, the overall fitness of the entities decreases. This is to be expected since the food source is harder to reach, the probability of feeding offspring is less likely, and the number of attacks (which are necessary to obtain the food) are less frequent.

The results also show a slight impact on the number of species when the size of the virtual environment is increased. Compared to life, DFALG is extremely simple. The expansion of the virtual universe only simplifies it
more. Given this, the number of final species decreases with the increase in size.

Table 4. Results of Increasing Environment Size

<table>
<thead>
<tr>
<th>SIZE</th>
<th>FITNESS</th>
<th>NUMBER OF BEGINNING AND ENDING SPECIES</th>
<th>CENSUS OF BEGINNING AND ENDING SPECIES</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>493.9</td>
<td>7/3</td>
<td>3,2,7,2,1,2,3/1,4,5</td>
</tr>
<tr>
<td>2</td>
<td>209.9</td>
<td>8/3</td>
<td>5,4,4,1,1,2,2,1/8,1,1</td>
</tr>
<tr>
<td>3</td>
<td>195.5</td>
<td>8/1</td>
<td>5,4,4,1,1,2,2,1/4</td>
</tr>
<tr>
<td>4</td>
<td>174.0</td>
<td>8/2</td>
<td>5,4,4,1,1,2,2,1/4,3</td>
</tr>
</tbody>
</table>
Along with advancements in software, hardware, and computer graphics, computer games have also advanced since the days of pong and pacman. As the complexity of computer games has increased, so has the need for a more advanced opponent. Typically, computer game opponents are very simple. They challenge the player by being programmed to move a little quicker, shoot a little faster, or withstand more damage. The emphasis is that they are programmed and ultimately are lifeless and static in nature. Using AL and GAs, the way that opponents in computer games are generated has been revolutionized.

By applying the framework of AL and GAs to a combat game simulation, affects on the natural evolution of opponents in response to a simulated player’s skill and strategy was demonstrated. AL provides intrinsic adaptation by using the automatic evolution provided by natural selection. GAs provide extrinsic adaptation by using selective breeding and optimizing the population of opponents.

Intrinsic adaptation brought about by AL has several advantages and disadvantages. The first advantage is that it follows the natural direction of evolution. It also
allows for expansion into almost any area imaginable and can be developed into the finest detail. Some of the disadvantages are that small changes can dramatically affect the biosphere of the virtual environment. A simple change to add a feature can cascade into a pyramid of supporting changes. For example adding sex to the opponents opens several avenues of thought that must be pursued, such as how to determine sex of offspring and preventing mating of similar lineage.

Extrinsic adaptation brought about by GAs also has several advantages and disadvantages. The first advantage is that GAs are easier to integrate into the virtual environment. Since GAs are oblivious to the application to which they are being applied, they are not bogged down by the intricate and complex details of AL. GAs also provide for slightly faster optimization of opponents in response to a player’s skill and strategy. The optimization is so easily adjusted by modifying the fitness function or other GA control parameters. Unfortunately GAs are not truly connected to artificial life. Their names, constructs and techniques are borrowed from AL, but the bottom line is they are solely optimization algorithms.

It has been shown that AL and GAs applied to a dynamic feedback game simulation will produce optimized opponents, yet several open problems to this approach still exist. Such as the development of a more complex environment,
varying the skill and strategy of a simulated player in middle of the game, creating a species-generating genetic algorithm, and adding a real player.

The development of a more complex environment needs to be examined. Recall that the virtual environment of DFALG is relatively simple as compared to the dynamics of life. The more lifelike that the virtual environment is, the more it will have the predictable features of life. These features will give even more flexibility and diversity to the game designer in developing new game scenarios.

Another important aspect which should follow or accompany the development of a more complex environment is the affect of changing the simulated player’s skill and strategy after homeostasis has been achieved. The homeostatic state achieved by evolved opponents will continue until a stimulus changes the environment. Changing the simulated player’s skill and strategy will provide the necessary stimulus to disrupt the steady state equilibrium and force the opponents to again achieve homeostasis.

A more effective way to boost GAs’ performance is to create a species-generating genetic algorithm. Using this type of GA, new species would be created whenever the GA detects two significantly different areas being explored within its population. To decide when the population is diverse enough to merit a split, a predetermined threshold is used to compare the highest ranking entities of the
population. Once the threshold is crossed the entity can be assigned to the species to which it is most closely related. These distinct species generally evolve independently of one another, although they can occasionally swap a few of their better entities [4].

Finally, the integration of the ability to support a real player needs to be accomplished. Recall that this is a prototype game using a simulated player. Using a simulated player provided the advantage of repeatable results during the development. The next logical step in DFALG’s evolution is to transform it into a real combat game.
APPENDIX A

DESCRIPTION OF GENES
<table>
<thead>
<tr>
<th>GENE</th>
<th>DESCRIPTION</th>
<th>RANGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>G1</td>
<td>Duration of childhood</td>
<td>0..30 units</td>
</tr>
<tr>
<td>G2</td>
<td>Velocity angle of attack</td>
<td>0..90 deg</td>
</tr>
<tr>
<td>G3</td>
<td>Velocity angle of attack modifier</td>
<td>0/1</td>
</tr>
<tr>
<td>G4</td>
<td>Velocity vector</td>
<td>50..150 units</td>
</tr>
<tr>
<td>G5</td>
<td>Acceleration angle of attack</td>
<td>0..180 deg</td>
</tr>
<tr>
<td>G6</td>
<td>Acceleration angle of attack modifier</td>
<td>0/1</td>
</tr>
<tr>
<td>G7</td>
<td>Acceleration vector</td>
<td>500..900 units</td>
</tr>
<tr>
<td>G8</td>
<td>Acceleration vector modifier</td>
<td>0/1</td>
</tr>
<tr>
<td>G9</td>
<td>Type of change (random or constant)</td>
<td>0/1</td>
</tr>
<tr>
<td>G10</td>
<td>Life span</td>
<td>0..MAXLIFE updates</td>
</tr>
<tr>
<td>G11</td>
<td>Probability an adult entity feeds newborn or child entity</td>
<td>0.0..1.0</td>
</tr>
<tr>
<td>G12</td>
<td>X component error for entity shot</td>
<td>0..20 deg</td>
</tr>
<tr>
<td>G13</td>
<td>Y component error for entity shot</td>
<td>0..20 deg</td>
</tr>
<tr>
<td>G14</td>
<td>Metabolism rate</td>
<td>0..1.0 per update</td>
</tr>
<tr>
<td>G15</td>
<td>Mating duration</td>
<td>0..1.0 updates</td>
</tr>
<tr>
<td>G16</td>
<td>Probability of maverick mating</td>
<td>0..1.0</td>
</tr>
<tr>
<td>G17</td>
<td>Energy loss during mating</td>
<td>0..1.0 total</td>
</tr>
<tr>
<td>G18</td>
<td>Energy loss for firing</td>
<td>0..1.0 total</td>
</tr>
<tr>
<td>G19</td>
<td>Energy loss for normal living</td>
<td>0..1.0 per update</td>
</tr>
<tr>
<td>G20</td>
<td>Period of inter breeding</td>
<td>0..5 updates</td>
</tr>
</tbody>
</table>
APPENDIX B

DESCRIPTION OF SIMULATION CONTROL PARAMETERS
CP1: Probability of choosing a locus on the chromosome for crossing during breeding.

CP2: Probability of mutation during breeding.

CP3: Maximum number of generation per breeding period.

CP4: Time period between each breeding period.

CP5: The maximum population size.

CP6: Number of shots allowed by an entity (always 1)

CP7: Number of shots allowed by the player (Max of 5)

CP8: Radius of the players field of view.

CP9: Maximum mating distance.

CP10: Maximum offspring per mate.

CP11: Random number initializer.

CP12: Probability of choosing a locus on the chromosome for crossing during a mate.

CP13: Probability of mutation during a mate.

CP14: Minimum energy level required to mate.

CP15: Energy lost during a mate by each parent entity.

CP16: Energy lost by entity for firing a shot.

CP17: Energy lost by entity for normal living.

CP18: Number of hits the player can withstand.

CP19: Number of consecutive shots fired by player (Maximum 5).

CP20: Radius of priority as entities converge.

CP21: Radius of explosion of entity shots.

CP22: Turns breeding on (1) and off (0).

CP23: Turns mating on (1) and off (0).

CP24: Amount of energy the player has at start.
CP25: Percent of speciation similarity necessary before mating is allowed.
CP26: Allow entities to absorb energy from the environment
CP27: Percent of fitness similarity necessary before mating is allowed.
CP28: Maximum entity life span.
CP29: Maximum distance allowed before feeding can occur.
CP30: Selection of the size of the virtual universe.
CP31: Selection of the fitness function.
CP32: Initial entity population size.
CP33: Use maverick mating on (1) and off (0)
CP34: Start with a random species yes (1) and no (0)
CP35: Simulation end time
CP36: Number of Monte Carlo runs
BIBLIOGRAPHY


