

Modeling the Evolution of Mimicry

Mohiul Islam

A Thesis
in
The Department
of
Computer Science
and Software Engineering

Presented in Partial Fulfillment of the Requirements
for the Degree of Master of Computer Science at
Concordia University
Montreal, Quebec, Canada

January 2011

© Mohiul Islam, 2011

CONCORDIA UNIVERSITY
School of Graduate Studies

This is to certify that the thesis prepared

By: Mohiul Islam

Entitled: Modeling the Evolution of Mimicry

and submitted in partial fulfillment of the requirements for the degree of

Master of Computer Science

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

_____ Chair
Dr. E. J. Doedel

_____ Examiner
Dr. T. Radhakrishnan

_____ Examiner
Dr. N. Kharna

_____ Supervisor
Dr. P. Grogono

Approved by

Chair of Department or Graduate Program Director

Dr. Robin A. L. Drew, Dean
Faculty of Engineering and Computer Science

Date

Abstract

Modeling the Evolution of Mimicry

Mohiul Islam

A novel agent based, artificial life model, for the evolution of mimicry is presented. This model is a predator-prey co-evolution scenario where pattern representation phenotype is simulated with Cellular Automata, while behaviors of pattern recognition is configured with Hopfield Network. A visual three dimensional toroidal cube is used to construct a universe in which agents have complete freedom of mobility, genetic representation of behavior and reproduction capability to evolve new behaviors in successive generations. These agents are classified into categories of predator and prey species. Genome of prey species control their mobility and palatability, while 2D Cellular Automata (CA) is used to represent a pattern, where the rule to generate the CA is also genetically represented. Through evolution, successive generations of prey species develop new patterns to represent them both visually and to the predators. Predators are agents with the primary purpose of providing selection pressure for the evolution of mimicry. They are equipped with Hopfield Network memory to recognize new CA pattern and make intelligent decisions to consume the prey based on their level of palatability. Using the above construction of ideas, successful emulation of the natural process of mimicry is achieved. Also complex behavior pattern of Batesian and Mullerian mimicry is simulated and studied.

Acknowledgments

My first appreciation is to my supervisor Dr. Peter Grogono. It is not only a privilege to work with him, but also the inspiration and encouragement that I have received from him during the period of this work, made it possible.

My second appreciation is to my parents Mobinul Islam and Latifa Mobin. Their hard work and sacrifice in life made it possible for me to be able to come all this way.

My third appreciation is to my sister Shagufta Islam, who also made it easier for me to be able to give all the time that I required to complete my thesis.

Contents

List of Figures	viii
List of Tables	x
1 Introduction	1
2 Review	3
2.1 Introduction	3
2.2 Evolutionary Computation	3
2.2.1 History	4
2.2.2 Evolutionary Algorithms	5
2.2.3 Genetic Algorithm	7
2.2.4 Evolutionary Strategies	8
2.2.5 Evolutionary Programming	9
2.2.6 Genetic Programming	9
2.3 Artificial Life	9
2.3.1 Complex Adaptive System	10
2.3.2 Echo	12
2.4 Conclusion	16
3 The Inspiration: Mimicry	17
3.1 History	17
3.2 Batesian Mimicry	17
3.2.1 Definitions of Mimicry	19
3.3 Mullerian Mimicry	20
3.4 Evolutionary Dynamics of Mimicry	22

3.4.1	Mimicry Ring	24
3.5	Conclusion	25
4	The Model: Evolution of Mimicry	26
4.1	Introduction	26
4.2	Past Work	27
4.2.1	Models by Franks and Noble	28
4.3	FormAL Framework	30
4.3.1	Agents	30
4.3.2	Spatial representation of the environment	30
4.3.3	Mobility	32
4.4	The Prey: Mimics and Models	33
4.4.1	Pattern representation by Cellular Automata	33
4.4.2	Species diversity	34
4.4.3	Genome	35
4.4.4	Reflection of punctuated equilibrium	35
4.4.5	Genetic representation of palatability	36
4.4.6	Interaction between other Prey and Predators	36
4.5	Predator	39
4.5.1	Learning	39
4.5.2	Design of Memory with Hopfield Network	41
4.5.3	Genome	44
4.5.4	Mobility and reproduction capability	44
4.5.5	Reproduction process	45
4.6	Conclusion	46
5	The Results	48
5.1	Introduction	48
5.2	Mimicry Ring Reports	48
5.3	Initial configuration with two prey species	49
5.4	Initial configuration with four prey species	55
5.5	Increased initial population with four prey species	58
5.6	Initial population with six prey species	60

5.7	Initial configuration with only unpalatable species	63
5.8	Initial configuration with only palatable species	67
5.9	Initial configuration with single prey species	69
5.9.1	Unpalatable	69
5.9.2	Palatable	71
5.10	Analysis of Batesian Mimicry	72
5.11	Analysis of Mullerian Mimicry	73
5.12	Conclusion	73
6	Future Work: Application	74
7	Conclusion	76
	References	78
	Acronyms	83

List of Figures

3.1	Plate from Bates (1862) illustrating Batesian mimicry	18
3.2	A very well-known example of mimicry, the viceroy butterfly	22
3.3	Punctuated equilibrium vs. phyletic gradualism	23
4.1	Cellular Automata	34
4.2	A population of predators	45
5.1	Population distribution of mimicry rings (2 prey species, 10k iterations) . .	51
5.2	Population distribution of mimicry rings (2 prey species, 5k iterations) . . .	52
5.3	Number of mimicry rings (2 prey species)	53
5.4	Population distribution of generated mimicry rings (2 prey species)	54
5.5	Graphical representation of the model (simulation time: 7600)	55
5.6	Population distribution of mimicry rings (4 prey species)	56
5.7	Number of mimicry rings (4 prey species)	57
5.8	Graphical representation of the model (simulation time: 11200)	57
5.9	Population distribution of mimicry rings (4 prey species, increased population)	59
5.10	Number of mimicry rings (4 prey species, increased population)	60
5.11	Population distribution of mimicry rings (6 prey species)	61
5.12	Number of mimicry rings (6 prey species)	62
5.13	Population distribution of mimicry rings(4 prey species all unpalatable) . .	64
5.14	Number of mimicry rings (4 prey species all unpalatable)	65
5.15	Population distribution of mimicry rings(4 prey, all unpalatable but reduced predator memory)	66
5.16	Population distribution of mimicry rings (4 prey species all palatable) . . .	68
5.17	Number of mimicry rings (4 prey species all palatable)	68

5.18	Population distribution of mimicry rings (1 prey species unpalatable)	70
5.19	Number of mimicry rings (1 prey species unpalatable)	70
5.20	Population distribution of mimicry rings (1 prey species palatable)	71
5.21	Number of mimicry rings (1 palatable prey species)	72

List of Tables

4.1	Parameters to control the environment.	31
4.2	Three dimensional representation of the environment	32
4.3	Parameters to control mobility of agents.	33
4.4	Cellular Automata rule	34
4.5	Distribution and purpose of each gene of the 17 bit prey genome.	35
4.6	Genetic representation of palatability	36
4.7	Parameters to control prey population and visibility.	38
4.8	Distribution and purpose of each gene of the 5 bit predator genome.	44
4.9	Parameters to control predator population and pattern recognition capability.	47
5.1	Parameters to mimicry ring report.	49
5.2	Agent configuration of 2 prey species	49
5.3	Difference in prey pattern genotype and phenotype	53
5.4	Agent configuration of 4 prey species	55
5.5	Agent configuration of 4 prey species with increased population	58
5.6	Agent configuration of 6 prey species.	60
5.7	Agent configuration of 4 prey species all unpalatable.	63
5.8	Agent configuration of 4 prey species all palatable.	67
5.9	Agent configuration of 1 prey species unpalatable.	69
5.10	Agent configuration of 1 prey species palatable.	71

Chapter 1

Introduction

Mimicry is a process of deception. It is an evolutionary process with the help of which organisms survive by deceiving its predator. But this deception happens only if the environment contains similar appearing noxious organisms which the predators find unpalatable. Palatable organisms mimic the unpalatable ones through the process of evolution for survival of its species.

Artificial Life is an interdisciplinary study of life and life-like processes that uses synthetic methodology. This thesis consists of such a work which concentrates specifically on the life-like process of the evolution of mimicry. The synthetic tools used for this simulation is a three dimensional graphical environment, Cellular Automata and Neural Network.

Chapter 2 is intended to provide background knowledge upon which this thesis is based. It reviews different fields of science where the concept of evolution has been applied to solve complex computer science problems, and achieved better result than other conventional methods. It also discusses the field of Artificial Life which is an inter disciplinary field of science where scientist take concepts from biology, economics and physics to build complex and adaptive models to emulate nature and its life like behavior.

Chapter 3 discusses an inspiration from the field of evolutionary biology which is the primary purpose of building the model of this thesis. Mimicry is an evolutionary process, which not only provides us with insight into the process of evolution as it happens in nature but also is fascinating to study and learn how nature adapts itself in complicated situation for the survival of its species. In addition, this chapter discusses the different kinds of mimicry such as Batesian and Mullerian, while including how the concept of mimicry ring

helps us explain each of these cases.

Chapter 4 contains description of an agent based artificial life model, which is the primary purpose of this thesis. Inspired by the evolutionary mimetic process we present a model whose purpose is not only to emulate the evolution of mimicry but also to come up with complicated scenario of Batesian and Mullerian mimicry with the help of mimicry ring. This model is based on a FormAL (Formal Artificial Life) framework which is similar to Holland's Echo model but with a slightly different approach.

Chapter 5 deals with the results extracted from the model and its analysis to evaluate successful emulation of mimicry. The results are extracted by applying different initial conditions to the model. As this work is with complex adaptive system, the data that we extract from such system is also expected to be convoluted. But our effort in this section is to come up with simplified analysis of the data to be able to comprehend complex activities in the model.

Chapter 6 tries to provide direction for possible future work. The goal of this thesis was not only to come up with a model that emulate natures mimetic process but also to apply this method to solve problems in the field of computer science. This approach is similar to the way evolutionary algorithms and genetic programming help us in optimizing solutions of complicated problems.

Chapter 7 is an effort to bring a conclusion to justify the work done on this thesis.

Chapter 2

Review

“The theory of evolution by cumulative natural selection is the only theory we know of that is in principle capable of explaining the existence of organized complexity” - Richard Dawkins [Daw96]

2.1 Introduction

This chapter contains a discussion on the field of Evolutionary Computing and Artificial Life. Both of these field of computer science are inspired by the process of evolution as it happens in nature. The model presented in this thesis is also based on an evolutionary process existing in nature. This thesis is part of the field of Artificial Life, motivated by the work of Echo [Hol96] and other such simulated Complex Adaptive Systems. In addition to the model discussed in chapter 4, this thesis also put some effort in finding application for the evolutionary process of mimicry (chapter 6). Similar to evolutionary computation, we intend to apply the deceptive process to solve a problem in computer science.

2.2 Evolutionary Computation

The field of evolutionary computing is the collective name for a range of problem-solving techniques based on principles of biological evolution, such as natural selection and genetic inheritance.

2.2.1 History

Turing(1948) Ideas for evolutionary computation initiated from Alan Turing from his 1948 report titled “*Intelligent Machinery*” where the expression of “*genetical or evolutionary search*” was used [Tur48]. He suggested a range of ideas for systems which could be said to modify their own programs.

Bremermann(1962) Bremermann was the first to apply the concept of biological evolution to execute computer experiments for solving optimization problems [Bre62]. He was the first to consider the problem of minimizing a real-valued fitness function. He selected very simple functions to optimize in order to provide a more tractable analysis of the evolutionary process.

Rechenberg(1964) Rechenberg’s contribution is notable mainly for providing Evolutionary Strategies. He invented a highly influential set of optimization methods which were successfully applied to challenging problems such as aerodynamic wing design. [Rec73]

L. Fogel, Owens and Walsh (1965) A seminal work that merged the fields of evolutionary computation and computational intelligence was by Fogel, Owens and Walsh in “*Artificial Intelligence through a simulation of evolution*” [FOW66]. In this book, evolutionary process was applied to finite state automata to predict symbol strings generated from Markov processes and non-stationary time series. Evolutionary prediction of such kind was motivated by the concept that prediction is a keystone to intelligent behavior (defined in terms of adaptive behavior, in that the intelligent organism must anticipate events in order to adapt behavior in light of a goal).

Holland(1975) Perhaps the most significant work in terms of applications in the field of evolutionary computation comes from the inventor of genetic algorithms, John H. Holland. His ground breaking book “*Adaptation in natural and artificial systems*” [Hol92] is where he developed *The Schema Theorem* which is the foundation for explanation of the power of Genetic Algorithms.

Koza(1992) Koza’s invention is the idea of Genetic Programming, which is an automated method for creating a working computer program from a high level statement of a problem.

Starting from a high-level statement of “*what needs to be done*” it searches through all possible permutation of steps and finds the optimum solution through evolutionary methods [Koz92].

Contemporary terminology denotes the fields of **Evolutionary Strategies (ES)**, **Evolutionary Programming (EP)**, **Genetic Algorithm (GA)** and **Genetic Programming (GP)** to be under one umbrella termed as **Evolutionary Computation (EC)** while the algorithms are called **Evolutionary Algorithm (EA)**.

2.2.2 Evolutionary Algorithms

Evolutionary algorithms are inspired by the strategy of *survival of the fittest* from evolutionary biology, and uses different methods to optimize mathematical expressions which are used to evaluate a pool of population of solutions over generations. There are many different forms of evolutionary algorithm. All of them can be generalized to follow their pattern from natural selection. The mathematical expression, also termed as **fitness function** is used to evaluate the fitness of each individual from the set of existing solution space. An individual or a solution which provides higher output from the fitness function is considered as more fit to survive in the environment. So given a problem the algorithm initializes the environment with a random set of population or solutions. Then it uses the fitness function to select a better set of solutions from the existing ones. After that different operators of **mutation** and **recombination** are applied to the selected set of solution to come up with a new generation of solution set, born from the existing ones. Again the fitness function is applied to the current set of population to come up with a better set and this process iterates until the evaluation from the fitness function is satisfactory to give the best set of solutions.

According to Eiben [ES03] this process has two fundamental forces that form the basis of evolutionary systems:

- “*Variation operators (recombination and mutation) creates the necessary diversity and thereby facilitate novelty.*”
- “*Selection acts as a force pushing quality.*”

Important components of evolutionary algorithms are

Representation (Individuals or Solution Space) Evolutionary Algorithms(EA) are considered as robust problem solvers as it provides evenly good performance over a wide range of problems. To apply evolutionary algorithm to this wide range the most important part is representation of the solution space. In EA representation is analogous to the complex pathway that exists between **genotype** and **phenotype** of a biological organism.

Evaluation Function (Fitness Function) This is the function with which the solution is evaluated. The outcome of applying evolutionary algorithm to the problem depends on this function. It is defined to evaluate each of the solutions of the problem. In biology evaluation function is analogous to the decision of survival of any species in an environment.

Population The population is the solution set of the problem over which EA will be applied. It is used with the same meaning as in biology. The diversity of the population is very important. If the initial random population of solutions are diverse then there is more possibility of reaching an optimal solution space very quickly, and it also avoids reaching local optimum instead of the global optimum.

Variation operators, recombination and mutation Considering arity (number of object as input) variation operators are divided into two kinds. The mutation operator accepts only one operand. The process of mutation is similar to the way it happens in biology. It is a stochastic process, meaning, its output the child depends on the outcomes of a series of random choices. The operator is responsible for causing a random unbiased change. Mutation has very different roles in different fields of EC, for example in Genetic Programming it is not used at all, while in genetic algorithms it is a very pivotal process as it provides fresh blood to the existing set of solutions. Getting a new mutated child is like stepping into a new solution space outside of the ones which already exist in the parent pool. A recombination operator is responsible for merging two parent genotypes into one or two offspring genotypes. It is also a stochastic process depending on the choice of the parts of the parent individuals. Recombination operators with high arity (using more than two parent) are mathematically possible and easy to implement but does not have any biological equivalent. According to Eiben [ES03] *“the principle behind recombination is simple - by mating two individuals with different but desirable features, we can produce an offspring that combines both of those features.”* It is important that variation operators are defined

based on the representation of the problem under consideration.

2.2.3 Genetic Algorithm

Genetic Algorithms(GA) have the most wide set of applications. It was first conceived by John H. Holland for studying adaptive behavior as in *Adaptation in natural and artificial systems* [Hol92]. They are mainly considered as function optimizers. In general GA have many diverse representation schemes considering the solution set of the problem. Depending on the problem scenario an appropriate representation could be Binary, Grey Code, Integer, Real-Valued, Floating-point or Permutation based.

Using GA for a binary representation, mutation can be single point or multiple point, where a random location is selected from the binary genome and the bit is flipped. For integer representation random resetting is used instead of bit flipping. Creep mutation is another scheme which was designed for ordinal attributes and works by adding a small (positive or negative) value to each gene with a random probability. For a real-valued or floating-point mutation none of the above strategy is applicable. As for this case mutation is applied to a random gene maintaining a certain range of value within an upper and a lower bound and using a standard for randomization, such as Uniform, Gaussian or Cauchy distribution. For permutation representation, swap mutation is a strategy used by simply swapping the location of two randomly selected genes. Insert mutation is another strategy where a random gene is selected and inserted to a random location in the genome, causing the other genes to move a single space. In scramble mutation, location of a collection of randomly selected genes are scrambled, while in inversion mutation serialized location of a set of genes are inverted.

GA also have many different recombination operators for each representation. For binary and integer representation one-point crossover and N -point-crossover are the most frequently used. These two operators work by dividing the parent genome at one or multiple random points and combining them to create an offspring. Uniform-crossover [Syw89] is another method which works by treating each gene independently and making a random choice to which parent it should inherit from. Recombination for floating-point representation can use the same operators like binary and integer, and when it does it is termed as discreet recombination. Another type is intermediate or arithmetic recombination where the offspring do not just receive a part of their parent genome, but instead, their genomes

are derived from their parent using an arithmetic formula. Permutation representation has many interesting recombination operators, out of which partially mapped crossover was first proposed by Goldsberg and Lingle [GL85] for the traveling salesman problem and since then it has become one of the most widely used operators for adjacency type problems. Other useful operators for permutation representation are Edge crossover, Order crossover and Cycle crossover, where their name provides more or less an appropriate description of their operation.

In regards to parent selection the most commonly used process by GA is fitness proportion selection, where the selection probability of a parent depends on the absolute fitness value of the individual compared to the absolute fitness values of the rest of the population. A fitness value is evaluated from the evaluation or the fitness function. Introduced by Holland [Hol92] this procedure has a few problems. Premature convergence being one where individuals that are a lot better than the rest take over the entire population. Also when fitness values are all very close, there is almost no selection pressure in evolution of the fittest. Inspired by the observed drawbacks of fitness proportion selection a method was proposed by Baker [Bak87] termed as Ranking selection. This procedure preserves a constant selection pressure by sorting the population on the basis of fitness, and then allocating selection probabilities to individuals according to their rank. Both of the above mentioned selection procedures requires knowledge of the entire population. But when the population size is very large Tournament selection is an effective process [ES03]. In this case a randomized pre-selection is done over the large population and then they are ranked according to their fitness value and selected as parent.

Genetic Algorithms are one of the most effective strategy used in the field of evolutionary computation to solve optimization problems. Then again there are limitations. When the solution set of a problem do not have a gnomonic representation, genetic algorithms cannot play any role in improving it. So the set of optimization problems to which GA can be applied is not universal.

2.2.4 Evolutionary Strategies

The key important property of the set of algorithms termed as “Evolutionary Strategy” is **self-adaptation** of strategy parameters. The parameters of Evolutionary Algorithm are included in the chromosomes and they co-evolve with the solution set while running

the algorithm. To summarize these set of algorithms, they usually have real-valued vectors, their recombination process is discrete or intermediary, their mutation process uses Gaussian perturbation while parent selection is uniform random. In terms of their attribute feature they are fast and good optimizer for real-valued optimization.

2.2.5 Evolutionary Programming

The primary objective of evolutionary programming was to generate artificial intelligence by simulating evolution as a learning process [FOW66]. Capability of a system to adapt its behavior to meet some specified goal in the environment was considered as intelligence. Fogel used finite state machines as predictors and evolved them. Usually representation for evolutionary programming is with real-valued vectors. Its parent selection is deterministic, where each parent creates one offspring via mutation. There is no recombination operator for evolutionary programming. Mutation is applied with Gaussian perturbation. While survivor selection is probabilistic.

2.2.6 Genetic Programming

Genetic programming is typically applied to machine learning tasks such as prediction and classification. Its attribute feature behaves similar to neural network and usually has huge populations of solution combination. The algorithm itself is a very slow process. These sets of algorithms use non-linear chromosomes such as trees and graphs (data structures). Mutation is mostly avoided for genetic programming. Recombination operators work by exchanging subtrees. Mutation if applied will be a random changing process in the tree. Parent selection process is usually fitness proportional while the survivor selection process is usually replacement of generations. From a technical point of view, genetic programming is simply a variant of genetic algorithm working with a different data structure: the chromosomes are trees.

2.3 Artificial Life

Artificial Intelligence is the field of computer science where scientists take concepts from psychology and implement into computer science. In contrast to that, in Artificial Life scientist take concepts from biology, specially evolutionary biology and genetics and implement

into computer science.

Christopher G. Langton is considered as one of the founders of the field of Artificial Life. He invented the term while organizing the first ‘*International Conference on the Synthesis and Simulation of Living Systems*’ at the Los Alamos National Laboratory in 1987.

In defining Artificial Life Langton states the following,

“ ‘Art’ + ‘Life’ = *Artificial Life: Life made by Man rather than by Nature. Our technological capabilities have brought us to the point where we are on the verge of creating ‘living’ artifacts. The field of Artificial Life is devoted to studying the scientific, technological, artistic, philosophical, and social implications of such an accomplishment.*”

Artificial Life (AL) has been described by Taylor, also as a tool for biological inquiry [TJ93]. While providing a brief survey over different AL models he talks about *Wetware systems* which work at the molecular level, the *Software systems* which work at the cellular level and the *Hardware systems* which works at the organism level. *Wetware* artificial life systems are the most similar to natural life and indeed are actually derived from natural life. Most of the experiments are attempts to direct an artificial evolutionary process toward the production of Ribonucleic Acid (RNA) molecules with specific catalytic properties. The initial contribution of *software systems* in artificial life was from John von Neumann when he designed the first artificial-life model (without referring to it as such), the famous self-reproducing, computation-universal cellular automata [VN66]. He tried to understand the fundamental properties of living systems, especially self-reproduction and the evolution of complex adaptive structures, by constructing simple formal systems that exhibit those properties. *Hardware systems* at the organism level can deal with an organism’s sensory and nervous system, its body and its environment. Even though an organism’s nervous system is usually considered as fantastically complex, the other interesting complex components can be its geometric, mechanical, dynamical or thermal properties.

2.3.1 Complex Adaptive System

Complex Adaptive System (CAS) are special cases of complex systems. They are complex as they are diverse and made up of multiple interconnected elements and adaptive as they have the capacity to change and learn from experience. Examples of complex adaptive systems

include the stock market, social insect and ant colonies, the biosphere and the ecosystem, the brain and the immune system, the cell and the developing embryo, manufacturing businesses and any human social group-based endeavor in a cultural and social system such as political parties or communities. This includes some large-scale online systems, such as collaborative tagging or social bookmarking systems.

The following definition of Complex Adaptive System is comprehensive.

“A Complex Adaptive System (CAS) is a dynamic network of many agents (which may represent cells, species, individuals, firms, nations) acting in parallel, constantly acting and reacting to what the other agents are doing. The control of a CAS tends to be highly dispersed and decentralized. If there is to be any coherent behavior in the system, it has to arise from competition and cooperation among the agents themselves. The overall behavior of the system is the result of a huge number of decisions made every moment by many individual agents.”

- John H. Holland

Holland also provided some common kernel of similarities and dissimilarities between different complex adaptive systems in *Adaptation in Natural and Artificial Systems* [Hol92]:

1. *“All complex adaptive system involve large number of parts undergoing a kaleidoscopic array of simultaneous non linear interactions.”* As the behavior of the whole system is non-linear, the usual mathematical techniques of linear approximation such as, linear regression, normal coordinates, mean field approaches, and the like make little progress in the analysis of complex adaptive systems. As because different interactions between agents are simultaneous, parallel computing is helpful in term of simulating and analyzing these systems.
2. *“The impact of these systems in human affairs center on the aggregate behavior, the behavior of the whole.”*
3. *“The interactions evolve over time, as the parts adapt in an attempt to survive in the environment provided by the other parts.”*
4. *“Complex adaptive systems anticipate.”*

2.3.2 Echo

Echo is a class of simulation model for complex adaptive system, designed primarily for gedanken experiments rather than precise simulations. Echo provides a population of evolving, reproducing agents distributed over a geography, with different inputs of renewable resources at various sites. Each agent has simple capabilities - offense, defense, trading and mate selection - defined by a set of chromosomes. Even though these capabilities are simple and defined simply, they provide a rich set of variations illustrating the four kernel properties of complex adaptive systems described in section 2.3.1.

Criteria

Echo has been constructed based on a set of criterion. The following is a short summary on it extracted from Holland's *Hidden Order: How Adaptation Builds Complexity* [Hol96].

- Simplicity is considered as the first and foremost criterion. Instead of emulating a real system Echo is more of a thought experiment. Simplicity is attained by carefully constraining agent interaction and also giving the agents only a primitive internal model.
- Actions and interaction of each agent in echo has been designed to be part of a wide range of CAS setting. Their mobility is also part of a wide range of geographical setting. Input to the environment which is considered as a stimuli or resource, can also be diversified.
- Fitness of each agent, is an evolving criterion. Instead of considering fitness as an external exogenous factor, it should be dependent on the context provided by the site and other agents. The fitness value, instead of being a constant should evolve along with the system.
- *“The primitive mechanisms in Echo should have ready counter parts in all CAS”*. Interpretation of results from the Echo model should be direct and ready-made. As simulation is nothing but manipulation of numbers and symbols, it is always possible to make fanciful and deceiving interpretation of them, taking the opportunity to prove one's point. So by applying primitive mechanism for interpretation it is possible to avoid misinterpretation. Another advantage is with the help of interpretations,

selected mechanisms can be shown to be sufficient to generate the phenomenon of interest. Even though simulation cannot establish that a given mechanism is actually present, but it can establish its sufficiency or plausibility.

- The architecture of the Echo model should be flexible enough so that other well-studied mathematical models of particular CAS can be incorporated whenever required. Among other models the significant ones include, Dawkins' *biological arms race* [Daw90] and Brower's *survival of mimics* [Bro88] from ecology; Wicksell's *Triangle* [MMS89] and *overlapping generation models* [And89] in economics; the *prisoner's dilemma game* [Axe84] in political science; Holland's *two-Armed bandits* [Hol92] in operational research and Perelson's *antigen-body matching* [Per99] in immunology.
- As mathematical analysis provides the surest route to arrive at a valid generalization point, most aspects of the Echo model should be amenable to mathematical analysis.

In developing Echo using the above set of criterion, a step by step approach has been taken. Among these steps of improvement, the work by Hraber titled "The Ecology of Echo" [HJF97] is more from an ecological perspective and has been discussed in the following.

Organization

As Echo is an agent-based model, it has a collection of agents distributed in a two dimensional site which contains resources. Agents in Echo have freedom and flexibility to move around in this environment. These agents are capable of replicating themselves, while also having the capability of making complex interactions. Interactions include combat, trading and mating.

Resources and Sites Resource in a site of Echo form its very foundation. They are represented by letters (a,b,c,d). Everything in Echo is constructed by combining these letters as strings. These resources are treated much like an 'atom' and by combining them as a molecular string, agents are constructed. Geographically, Echo is a collection of interconnected sites. Each site contains a fountain which is the source of resource for the agents in Echo. The amount of resource available in each site varies in time. A site with no resource can be compared to a 'desert' while another site with increased availability of letters can be a water spring or a pond.

Agents “Agents have a genome that is roughly analogous to a single chromosome in an haploid species”. An agents *tag* is an external phenotype of its respective genotype (such as color). Agents have internal conditions or rules of interaction, which are genetically represented and they encode the agents internal preferences, behavioral rules and so forth. So interactions between multiple agents occur based on their conditions (rules for interactions) and tags (appearance). This results in sophisticated interaction between agents such as deception (mimicry, bluffing) and in-transitivities. “The six (external) tag and (internal) condition genes possessed by every agent are the offense tag, defense tag, mating tag, combat condition, trade condition and mating condition”. These genes are used to determine what sort of interaction between agents will take place and also the outcome of their interaction.

Agents have two ways of reproduction, one being self replication, while the other is by interaction between a pair of agents.

Replication The replication process for agents is asexual reproduction. An agent needs to collect sufficient resource to start the self-replication process. After replication, the newly created agent receives a fraction of its parent’s resource. Spontaneous mutation is applied to the gene for this process, where different mutation rate is applied to different parts of the gene.

Interactions There are diverse interactions between agents consisting of combat, trading and mating. Agents need to be in the same location in site for any interaction to take place. Testing for which selection of interaction to take place, is always conducted in the order of combat, trade and mating.

Combat The interaction of combat is analogous to real world antagonistic behavior, where one agent is killed resulting in the survivor agent to receive all its resource. Depending on each agent’s offense and defense tag and the tag prefixes, the winner is decided. A *combat matrix* is also calculated with the genome of individual agents, which helps in calculating the winner.

Trading When two agents are chosen to interact and they do not engage in combat, they are given the opportunity to trade and mate. Trading and mating are very much a mutual

agreement between agents compared to combat, which is more of a destructive process. “Agents *A* and *B* will trade if *A*’s trading condition is a prefix of *B*’s offense tag and vice versa”. Trading takes place for excess resources between agents, where excess is defined as the amount which the agent possess above the amount required for a replication process, plus some reserves. The value of the reserve is controlled with the help of a simulation parameter. A deceptive process is introduced here while trading takes place, when an agent might not have excess resource but will still be engaged in trade, as it is not possible for the other agent to know the resource of its opposite interactor.

Mating Only if two agents find each other suitable, they will engage in the mating process. Suitability for mating is decided on certain conditions based on the mating tag of individual agents. A form of two point crossover takes place to perform the mating action and produce diverse new agents.

Discussion of Results

The Echo model has good resemblance with natural ecosystem. It has stable patterns of interactions among genetic variants, or trophic networks. It consists of different levels of resources along with relative species abundance. Even different cataclysmic events such as meteor, drought and so fourth are introduced. Also several isolation effects are observed where a number of sites are allowed to evolve independently. In addition to that Echo contains interesting flow of resources.

Two separate experiments are done with the above simulation to observe whether species abundance pattern can be used to find ecologically plausible behavior in Echo. To make a falsifiable comparison with biological systems, two ecological patterns are considered, each of which has similar quantitative properties in many different ecosystems that exist in nature. The objective is to compare these naturally occurring ecosystems with Echo. The ecological patterns are the Preston curve and the species-area scaling relation. This allowed comparison of Echo diversity patterns with empirical patterns of species abundance.

Preston’s canonical log normal distribution is the most widely accepted formalization of the relative commonness and rarity of species. The species-area scaling relation is also an important ecological pattern. The theory of island biogeography is predicted on species-area relations. In conservation biology, this relation has been used to predict the effects

of different reserve size, with species diversity. Although the species-area relation can be derived from Preston's canonical log-normal distribution, a satisfactory explanation of the processes that regulate this relation has not been advanced.

Now the question is, does Echo exhibit the robust ecological phenomenon of Preston curve and species area scaling relation? For Preston's curve plot, the data show, qualitatively the pattern of genome abundances in Echo population resembles the general patterns found in biological systems, although some differences are observed. But for species area curve some discrepancy was observed between the expected result and the observed values. Several factors were raised for this discrepancy. One was the choice of the simulation parameter. Second being the definition of different species in Echo as agents having different genome is considered from different species. Another factor being the balance between rates of speciation and extinction of population.

Finally to answer the above question, it has been concluded that, Echo showed good quantitative agreement with naturally occurring species abundance distribution and species area scaling relation.

2.4 Conclusion

This chapter discusses evolutionary computation, the field that takes concepts from evolutionary biology to solve and optimize problems in compute science. It also discusses the interdisciplinary field Artificial life where scientist discuss diverse knowledge from biology, economics, physics and use computer science to simulate and analyze life as it is in nature. The discussion of this chapter is expected to provide a base upon which the following chapters will begin construction of the core idea of this thesis.

Chapter 3

The Inspiration: Mimicry

“It is not the strongest of the species that survives, nor the most intelligent that survives. It is the one that is the most adaptable to change.” - Charles Darwin

3.1 History

Henry W. Bates first published in 1862 his findings about the similarities and dissimilarities between Heliconiinae and Ithomiinae butterflies, after 10 years of research in the Brazilian rain forest. For the next hundred years, it simulated heated discussion among all groups of people, scientists, philosophers, theologians, teachers and amateur naturalists. Bates collected ninety-four pieces of butterfly. He grouped them according to their similar appearance. He found butterflies having similar appearance, exhibiting morphological features which point to completely different species even families. Out of the ninety four species, sixty seven are now classified as Ithomiinae, while twenty seven of them are Heliconiinae.

3.2 Batesian Mimicry

Even though Heliconiids are conspicuously colored, they are extremely abundant. They are also slow in mobility. Still predators in the surrounding area, mostly insectivorous birds do not prey on them, because of their inedible and unpalatable nature. Also because of this phenomenon other edible and palatable species such as ithomiinae and pieridae, pretend to be heliconiids and thus enjoy protection.

Repulsive animals, such as heliconiids are very conspicuously colored. Having this no-

ticeable property, they are easily recalled by predators. Their wing pattern works as a warning to predators. Once a predator has the knowledge of their inedible and unpalatable property, they would probably never attempt to try it again. As this is true, if any organism within close family and species, but being edible and having a deceptive resemblance to those conspicuously colored species will be avoided by the predators. Wickler [Wic68] expresses,

“Such unpalatable appearing and yet edible animals thus possess a false warning pattern, they ‘act a part’. An actor is a mime, and so the representation of a false warning pattern was called mimicry. Since Bates was the first to point out this phenomenon, it has received the name Batesian Mimicry in his honor.”

In general, the animal which is avoided by predator for unpalatable behavior is called the **model** and the imitating animal is called the **mimic**.

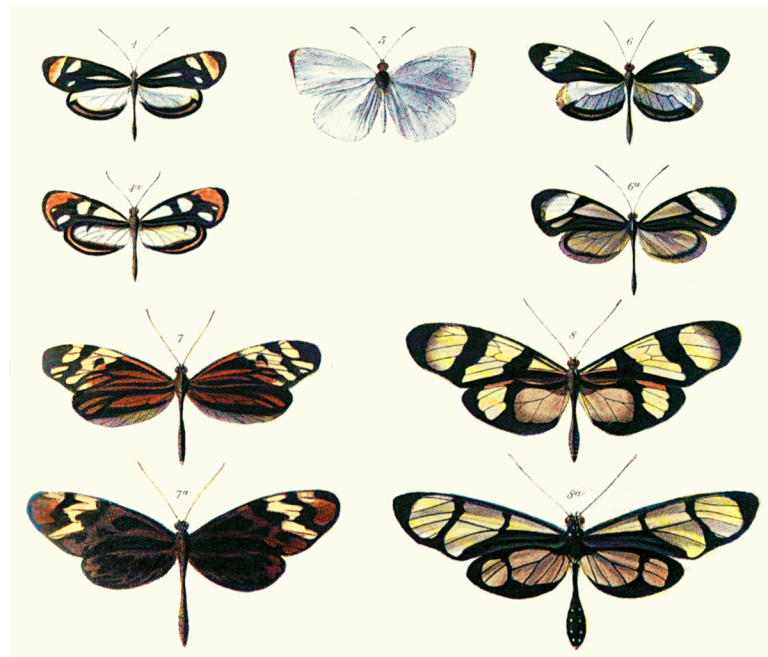


Figure 3.1: Plate from Bates (1862) illustrating Batesian mimicry between *Dismorphia* species (top row, third row) and various *Ithomiini* (Nymphalidae) (second row, bottom row). [Bat62]

3.2.1 Definitions of Mimicry

The following account on the definition of Mimicry is mostly extracted from Wickler's *Mimicry in plants and animals* [Wic68].

According to Wickler the definition of mimicry provided by Bates [Bat62] is quite restricted and is similar to the following

“resemblance in external appearance, shapes and colors between members of widely distinct families.”

The following more sophisticated definition was provided at the international Zoological Congress in Washington, 1963,

“Mimicry is the close resemblance of one organism to another which, because it is unpalatable and conspicuous, is recognized and avoided by some predators at some times.”

The best known definition of mimicry quoted to the present day, are those given by Wallace and listed in Poulton's *The color of animals, their meaning and use* [Pou90]. The clauses are presented in the following:

- *“that the imitative species occur in the same area and occupy the same station as the imitated”*. This increases the probability that a predator will affect both parties concerned. Nevertheless, the model could live in Africa and the mimic in Europe, or vice versa, if the deceptive predator were a migratory bird.
- *“that the imitators are always the more defenseless”*. On the contrary, for the case of Mertensian mimicry among coral snakes the more offensive party may occasionally be the mimic.
- *“that the imitators are always less numerous in individuals”*. What is meant here is that the deceived animal should encounter the mimic less often than the model. This is true when the deceived animal must learn to recognize model and mimic and when positive and negative experiences carry equal weight. But it is known from various learning experiments that when an animal has learned about warning coloration, negative experiences or punishment stimuli (such as inedibility) can have

a stronger and more lasting effect than positive experiences. Accordingly it should be expected that there would be an excess of mimics over models in proportion to the degree of predominance of negative experiences over positive ones. When the deceptive animal possesses an innate reaction to the model, the number of mimics can be virtually unlimited.

- *“that the imitators differ from the bulk of their allies”*. The most remarkable cases of mimicry are those where close relatives form divergent models. It is nevertheless highly possible that groups of closely related species may be similar mimics of the same model. Of course, it must always be remembered that species can look very different from their relatives because of adaptations which have nothing to do with mimicry.
- *“that the imitation, however minute, is external and visible only, never extending to internal characters or to such as do not affect the external appearance.”* This criteria, like the previous one is aimed at defining mimicry as a special adaptation. But if one of two closely related species both of which are poisonous and possess similar warning coloration, should secondarily lose its poisonous qualities, it would become a mimic and yet resemble its model in almost all its non-mimetic characters.

According to Wickler [Wic68], these five requirements, if fulfilled, indeed increase the probability that any given case of mimicry will be correctly diagnosed. But such supplementary clauses cannot be used as a definition of mimicry in general, though this is unfortunately often attempted.

3.3 Mullerian Mimicry

Bates was not able to explain some phenomena of mimicry. Occasionally two inedible unrelated butterfly species are amazingly similar in appearance. An explanation for this was provided by Fritz Muller in 1878. Like Bates, Muller observed and caught butterflies in Brazil. When there are multiple inedible species it is hard for predators to recognize each of them to know which one to consume and which one to avoid. Because of predator’s limited memory, all these species still lose their number even after being inedible. So to save this loss, and to prevent more sacrifice of their own kind, inedible species from different family

also tend to evolve to have similar appearance. This phenomena is referred to as Mullerian Mimicry in the name of Fritz Muller.

According to Huheey [Huh88] “*Mullerian mimicry is normally viewed as a system of mutual warning coloration arising from the convergence of two or more warningly colored species, though two closely related species could also be Mullerian mimics through parallel evolution*” [Mul79]. In Muller’s view, predation load on each species are reduced when their warning signals are shared or converged. Naive young predators create the most impact in learning predation as they start their life with no knowledge or experience over the prey species. For cases like these, all unpalatable species should benefit from mutual advertising. If mutation gives opportunity for a single unpalatable prey to resemble a warningly colored species, it is expected to be more likely to survive predation than if the change was in opposite direction. Predators surviving on warningly colored species need to generalize; meaning if similar patterns signal unpalatability, predators recognition rate becomes much higher. Thus, variation in color pattern is tolerated, and mutations leading to even poor resemblance provides some protection.

It is worth observing the fact that Mullerian mimicry is not deception, as it is to the predator’s advantage to be able to recognize all deterrent and noxious species. Preys resemblance does not have to be close enough to cause misidentification on the part of the predator, but merely similar enough to remind it of a bitter past experience with similar prey. As Huheey continues “*Thus although there is selection for convergence to perhaps the most common, average pattern, or more likely the one associated with the most deterrence (both from the intensity of the traumatic stimuli and the frequency of predators receiving them), the convergence need not be rapid or precise. Indeed, the resemblance among Mullerian mimics is often not close, and the existence of very close mimics raises doubts that the mimicry is Mullerian.*”



Figure 3.2: A very well-known example of mimicry is the viceroy butterfly (top) which has a pattern very similar to the unpalatable monarch butterfly (bottom). It was considered for a long time that this is an ideal example of Batesian mimicry. But recently Brower discovered that the viceroy is actually just as noxious as the monarch, making this a case of Mullerian mimicry. [RB91] Image source: Wikipedia

3.4 Evolutionary Dynamics of Mimicry

The dynamics of mimicry has been investigated by Turner [Tur88], where he states that the evolution of mimicry can be explained best by the process of punctuated equilibrium instead of phyletic gradualism.

Punctuated Equilibrium In evolutionary biology, punctuated equilibrium is the theory which proposes that most sexually reproducing species will remain in an extended state called *stasis* while experiencing little evolutionary change for most of their geological history. When evolution occurs it is localized in rare, rapid events of branching speciation, called cladogenesis. Cladogenesis is the process by which species split into two distinct species, rather than one species gradually transforming into another. Thus, “*punctuated equilibria is a model for discontinuous tempos of change (in) the process of speciation and the deployment of species in geological time*” [GE77].

Phyletic Gradualism In contrast to punctuated equilibrium, phyletic gradualism states that species continue to adapt to new environmental and biological selection pressures over the course of their history, gradually becoming new species. Phyletic gradualism holds that a species population changes gradually; that is there is no clear line of demarcation between an ancestral species and a descendant species unless a splitting (cladogenetic) event occurs or the gradually-changing lineage is divided arbitrarily. During this process, evolution occurs at a smooth, steady and incremental (but not necessarily constant and slow) rate on a geological time scale.

On this issue Turner states,

“According to punctuated equilibrium, therefore, evolution is not more or less both gradual and continual, as has been supposed for most purposes in most post-Darwinian theories. Rather, most morphological change is held to take place within geologically short periods, separated by vastly longer periods of comparative stasis [EG72]. The rapid, large changes all occur during speciation, that is to say during the branching of the evolutionary tree (cladogenesis);” [Tur88]

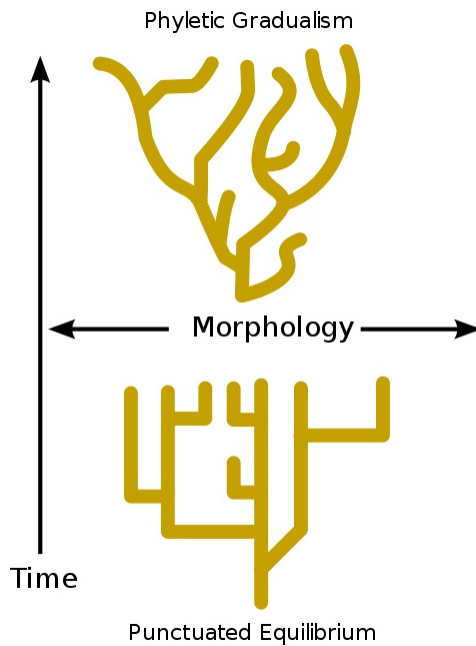


Figure 3.3: Punctuated equilibrium, bottom, consists of morphological stability and rare bursts of evolutionary change, while the top explains phyletic gradualism. Image source: Wikipedia

To explain the dynamic evolutionary process of mimicry, Turner came up with a synthetic theory [Tur88], which was originated by Poulton [Pou12] and Nicholson [Nic27], termed as the **two stage model**. This theory states that mimicry normally arises in two steps. A comparative large mutation achieves a good approximate resemblance to the model; it is followed by gradual evolutionary changes that refine the resemblance, in many cases to a high degree of perfection [She72] [For64]. This two-stage theory has been applied for the explanation of Mullerian mimicry as well.

3.4.1 Mimicry Ring

Any theory of Mullerian mimicry has to take into account the phenomenon of the coexistence of multiple mimicry rings. If we examine the local butterfly fauna in any area of the world, we will find that between all the aposomatic (warningly colored and defended) species present there are normally only a limited number of different patterns, normally far smaller than the number of species. Each cluster of species, all sharing a common pattern, is termed as Mullerian mimicry ring. Thus, in the rain forest of South and Central America, most of the long-winged butterflies (ithomiids, danaiids, and heliconids) belong to one of only five different rings.

Formation of Mimicry Rings

In regards to the formation of mimicry ring Turner [Tur88] has given a theory which takes into account the level of protection of different rings of butterfly species and their difference in phenotypic warning patterns. Different level of variation of these two parameter can give us very different formation of mimicry ring evolution. The cases are considered below.

- If we imagine two different rings of similar warning patterns in one single habitat, similar enough that the envelop of protection afforded to one gives some protection to the more extreme variant of the other (in a simplified language, suppose that predators that have sampled species A sometimes avoid the more extreme members of species B because they mistake them as A), then these two species are subject to natural protection for mutual convergence of their patterns. Eventually they will form a rather accurate Mullerian mimicry ring.
- By contrast, if the two species have patterns so dissimilar that predators encountering

one never imagine that it might be the other (the envelop of protection do not overlap the other species' phenotype distribution), then there is no selection whatever for convergence, and the two patterns will remain distinct indefinitely.

- But mutations of color patterns are occurring all the time. Suppose that species B happens to produce a mutation whose pattern falls within the envelop of protection afforded to species A. The mimicry need not be perfect, but if it is good enough, the mutation will have an advantage and will spread. In this way species B may, by a single mutation, become a Mullerian mimic of species A; in other word species B may switch from one Mullerian mimicry ring to another. Clearly, if the mimicry is not perfect, the two patterns will then gradually converge, making the resemblance ever closer.

Like Batesian mimicry, Mullerian mimicry can evolve in two stages: the mutational, one way convergence stage followed by the gradual, mutual convergence stage. It is worth mentioning that in the first stage only the less protected species can adopt the pattern of the better protected species; mutations in the other direction is not favored.

3.5 Conclusion

Mimicry is one of the most fascinating cases of evolutionary biology. This phenomenon has been studied extensively as infinite number of evidence exists in nature. One hundred years after Bates first clearly defined the concept, a review of literature listed 1,500 papers arguing for or against it. And this figure is more than 40 years old as it has been taken from Wilcker [Wic68], published in 1968. It is almost impossible to survey all the known examples of Mimicry, and this would in any case be pointless. Then again many examples can be found in Cott's *Adaptive Coloration in Animals* [Cot57].

Chapter 4

The Model: Evolution of Mimicry

“Modeling, it should be clear is an art form. It depends on the experience and taste of the modeler. In this it is much like cartooning. The modeler (cartoonist) must decide which features to make salient (exaggerate), and which features to eliminate (avoid), in order to answer the questions (make the political point)” - John H. Holland [Hol96].

4.1 Introduction

The objective of this thesis is to design an agent based artificial life model for simulating the natural process of the evolution of mimicry.

There are mainly two species of agents. These agents have properties and behavior similar to the **model**, the **mimic** and the **predator** in the evolution of mimicry. We represent evolution of pattern for the model and the mimic with the help of Cellular Automata (CA). Cellular Automata can be easily represented by simple rules, which can be expressed as a binary string. The predator will be equipped with a Hopfield network [Hop82], to be able to have pattern recognition capability. The process of evolution will of course be running at the genetic level.

The choice of Hopfield Network memory for a predator can be considered appropriate as the number of patterns which can be recognized by this network is inversely proportional to the accuracy of recall. As more patterns are memorized, Hopfield network tends to make more errors. This behavior will be appropriate for the simulation of Mullerian mimicry. Mullerian mimicry happens because of limited memory of the predators. Because of this

limited memory, multiple inedible butterflies seems to converge to a single ring.

The environment is designed as three dimensional, while the space will be of toroidal nature. This idea has been taken from the *Laws and Life* project by Peter Grogono [GCS⁺03].

4.2 Past Work

Various models of mimicry has been simulated and explored. The model by Turner [TS96] and the mathematical model of Huheey [Huh88] tend to focus on the selective pressure on prey brought about by the particular learning abilities of the predator, and employ simple Monte Carlo or mathematical approaches.

Sherratt [She02] provides an innovative perspective on the evolution of warning signals by considering co-evolving predator and prey populations. The model's predators are deterministic, in that they have a fixed behavioral strategy over their lifetime, and cannot learn from experience. For both cryptic and conspicuous prey, each predator has fixed policy of either attacking or avoiding.

The latest work on modeling evolution of warning signals and mimicry with individual based simulation is done by Franks and Noble. Their initial work [FN02] seems to focus on putting some conditions of mimetic evolution in an individual based model with multiple species preyed upon by a single abstract predator, where the appearance of each prey species can evolve but their palatability is fixed.

On 2003 [FN03] another model for the origin of mimicry ring has been proposed by Franks and Noble. Accordingly theory suggests that all Mullerian mimics in an ecosystem should converge into one large ring, while this convergence will be encouraged by presence of Batesian mimics. So an evolutionary simulation to observe the above mentioned phenomenon has been presented in this piece of work.

Franks and Noble continue to test the influence on mimicry ring evolution by Batesian mimics in their work on [FN04]. Usually mathematical models of mimicry has fixed prey coloration and appearances, which enables a comparison of predation rates to demonstrate the level of protection a mimic might be afforded. In this model prey colorations are free to evolve. This phenomenon is used to examine the effect of Batesian mimicry on Mullerian mimics and mimicry rings.

4.2.1 Models by Franks and Noble

The **first** model by Franks and Noble [FN02] is where “*multiple species are preyed upon by a single abstract predator; the appearance of each prey species can evolve but their palatability is fixed*”. Each individual had a single gene: a value representing their external appearance or phenotype. The phenotypes are constrained to a ring of values from 1-20 (where 20 and 1 are neighbors). The distance of one phenotype from another represents their levels of similarity.

A single abstract predator was modeled with a simple reinforcement learning system. The predator’s experience of each phenotype was represented by a score, which would be maintained by probability to consume the next prey species depending on similarity or difference in phenotype.

The existence of mimetic effect were measured with the initial and final distances between prey species’ phenotypes. Three experiments were noted. Firstly, with one palatable and one unpalatable species. For this Batesian mimicry was evolved. For the second experiment, with two unpalatable species, Mullerian mimicry was evolved if the two prey species have some initial resemblance. Experiment 3 was carried out with two unpalatable and one palatable species, where the phenotype of the palatable species moved towards that of one of the unpalatable species, which is in other words Batesian Mimicry.

The **second** model by Franks and Noble [FN03] is based on two working hypothesis:

1. *All of the Mullerian mimics in a given ecosystem should eventually converge into one large ring in order to gain maximum protection.*
2. *If the Mullerian mimics do not converge into one large ring, then the presence of Batesian mimics could entice them to do so, by influencing the rings to converge.*

Although there are many mathematical and stochastic models of mimicry in the biological literature, this model gives attention to the evolution of mimicry ring phenomenon from an artificial life perspective.

Prey Similar to the first model this also contains a population of prey species each having an appearance and palatability level. Different species of prey were each assigned a fixed palatability level on a scale between zero and one (least to most palatable), where 0.5 is neutrally palatable. Palatable species have values greater than 0.5, and unpalatable species

have values lower than 0.5. Each prey species have used two genes with values compositely representing their external appearance or phenotype. Both of these genes were constrained to values from 1 - 200. The Euclidean distance of one phenotype from another represented their level of similarity.

Predator Similar to Turner’s stochastic model [TKE84], predators were modeled with a Monte Carlo reinforcement learning system. The predator’s experience of each phenotype was represented by an attack probability, which was initialized to ambivalence at 0.5. After eating prey of a particular phenotype, the predator would make a post-attack update of the relevant probability according to the palatability of the prey consumed. The predator would use its experience of different prey appearances to help it decide on whether or not to attack at the next opportunity.

In contrary to the stochastic model [TKE84], predators would generalize on the basis of experience. A set of probability formula was used to come up with the current probability to consume a prey species based on its palatability and the previous probability of consumption based on experience using generalization rate and the Euclidean distance between the experienced phenotype and the consumed prey phenotype.

Results According to the first experiment, which started with 20 unpalatable prey species, hypothesis 1 of a single large mimicry ring was not established. There was existence of multiple mimicry rings of very different frequency (population). Second experiment was carried out with some palatable species along with the unpalatable ones which was able to reach conditions of Batesian mimicry also with less mimicry rings, where hypothesis 2 borne out. It can be observed that,

“The presence of Batesian mimics would provide positive selection pressure on mutants and would, therefore, increase the probability that they would evolve an initial resemblance to another unpalatable species. Also, Batesian pressure on mimicry rings has the potential to push one ring into the range of another, helping to bridge a large phenotypic difference between them.” [FN03]

4.3 FormAL Framework

The “*FormAL framework*” is a collection of ideas and concepts taken from Peter Grogono’s Formal Artificial Life (FormAL) project [GCS⁺03] and are used to build a framework for modeling the evolution of mimicry. The primary goal of the FormAL project was “*To study the emergence of complexity*”. While the principal behind it was “*not to include a variable in an agent unless the variable is genetically controlled (or, at least, genetically influenced)*”. We will see that this principal was not completely established in simulating mimicry as it was not possible to come up with a genetic representation of Hopfield Network which was used as a pattern recognizer for the predator species.

4.3.1 Agents

In FormAL, an **Agent** is a simulated organism. It is designed simply, but with the following qualities:

- It has behaviors to be able to reproduce itself using genetic information.
- Capable of modifying the structure of genome between generations.
- Able to interact with other agents.
- And also to survive and reproduce in a challenging environment.

4.3.2 Spatial representation of the environment

The framework consists of a three dimensional visual environment where agents of the individual based simulation gets complete freedom of movement defined from their genetic representation.

Space The space of the environment is a three-dimensional lattice of discrete points. The coordinate of each point in space is of the form $(x, y, z) \in \Sigma^3$, where Σ be the set $\{0, 1, \dots, S - 1\}$. S is a universal constant being a small positive integer, which in this simulation has been kept at 20 (the value of the *WorldSize* parameter in table 4.1).

Time Time, being an integer ($t \geq 0$), advances in discrete steps in the simulation, where at each step the agents update themselves.

Cell The entire three dimensional toroidal environment is divided in multiple cells. A cell is a three dimensional cubical section of the hyperspace. Accordingly in table 4.1 $ISize$ is the parameter that controls the number of cells in the environment. The size of a cell is evaluated from the entire world size. And the total number of cell is calculated as $ISize^3 = 216$.

Parameter	Value	Description
ISize	6	Number of cell in single dimension of the 3-D toroidal cube
World Size	20	Size of a single dimension of the 3-D toroidal cube
Cell Size	$WorldSize/ISize$	Size of each cell
Total Number of Cells	$ISize^3 = 216$	Total number of cells in the environment

Table 4.1: Parameters to control the environment.

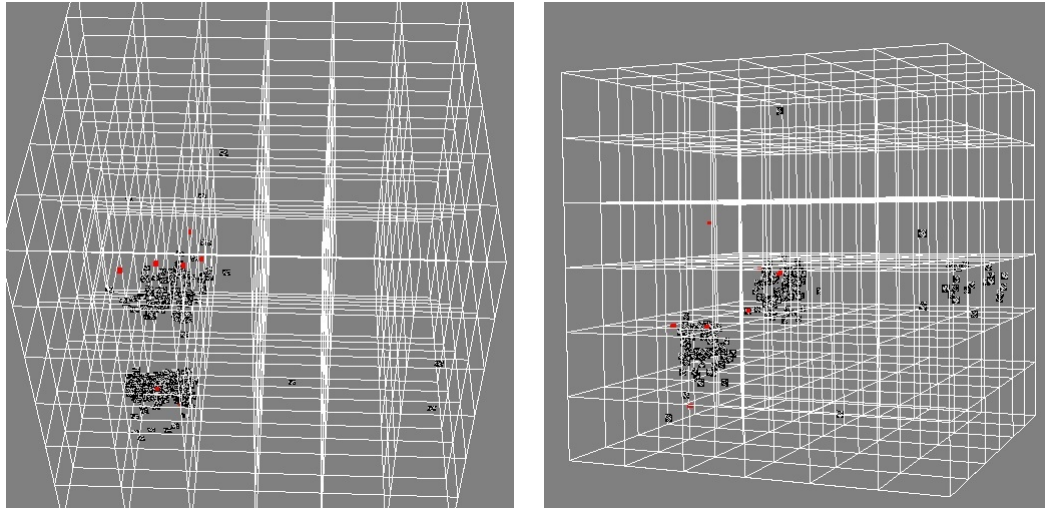
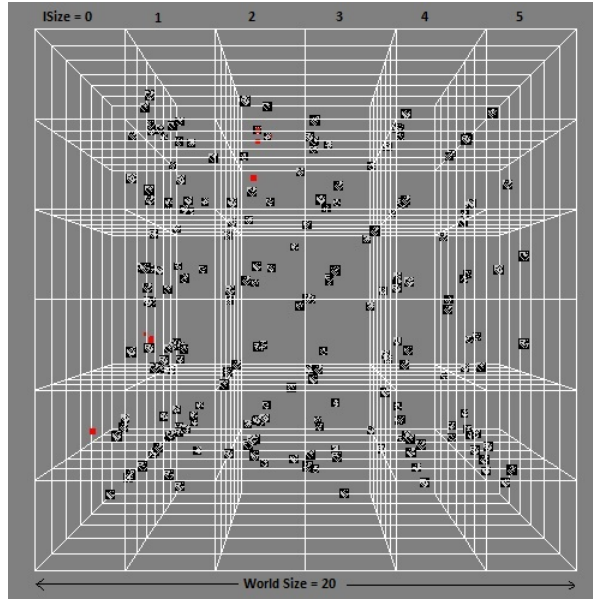


Table 4.2: Three dimensional representation of the environment divided in cells. Presence of different species of agents inside.

4.3.3 Mobility

An agent's position is calculated once during each step of update in time. The agents *position*, *force*, *acceleration* and *velocity* are all vector components. The *force* component is calculated from agent's mobility gene. And it is used to compute agent's *acceleration*. If the *force* and *velocity* are both zero, then the agent has no effect in motion. Otherwise, Newton's law is used to obtain the *acceleration*, which is integrated to obtain the new *velocity* and new *position*.

Algorithm 4.1 Algorithm for updating movement of the Agents

```
for each step in time do  
     $acceleration \leftarrow ForceFactor * force - Friction * velocity$   
     $velocity \leftarrow DT * acceleration$  {DT is considered as Differential Time Set}  
     $deltaPos \leftarrow DT * velocity$   
     $position \leftarrow position + deltaPos$   
end for
```

Parameter	Value	Description
Force Factor	40	A unit vector is multiplied by this amount before being added to the force vector.
Differential Time Step (DT)	0.01	The time step used for first-order integration of the motion equations.
Friction	5	The friction constant used for motion calculations.
Work Factor	1	$Workdone = WF * force * distance$, where WF is this constant.

Table 4.3: Parameters to control mobility of agents.

Algorithm 4.1 explains the calculation of mobility of each agent at each time step of the simulation, while table 4.3 contains the values of the parameters used to apply to this algorithm.

4.4 The Prey: Mimics and Models

4.4.1 Pattern representation by Cellular Automata

Only the significant characteristics which are required for a successful model of mimicry is implemented. As for this simulation we are considering heliconius butterfly, the representation of their wing pattern is with the help of cellular automata. So basically every prey organism will contain a genetic representation of cellular automata with which the predator will identify the prey and store its level of palatability in memory.

Cellular Automata

Cellular automata are computer generated patterns applying very simple rules. First invented by Jon von Neuman and later extended by Stephen Wolfram, this field of study has created a lot of enthusiasm in the evolution of complexity with blocks of simplicity, in this case rules. We choose cellular automata as it can be easily represented with the help of a binary genome and then evolutionary operations on the genomic representation, such as mutation and crossover can easily be applied.

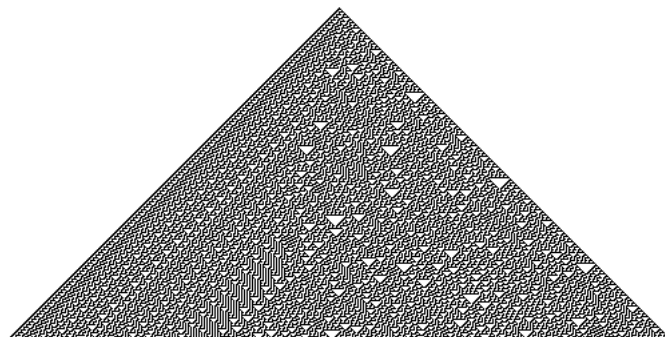


Figure 4.1: Cellular Automata Rule 30. Image source: Wikipedia

Current Pattern	111	110	101	100	011	010	001	000
New state of center cell	0	0	0	1	1	1	1	0

Table 4.4: Cellular Automata rule

In generating the pattern of figure 4.1, the genetic representation would be the ‘New state of center cell’. As figure 4.1 is a rule 30 cellular automata, the eight bit binary representation of 30 would be the genetic representation for this pattern. Generation of automata pattern from this binary string can be considered analogous to the complex pathway between the genotype and the phenotype that exists in all living organism.

4.4.2 Species diversity

Franks and Noble have used different models to diversify species. In [FN02] they have used linear difference in number “constrained to a ‘ring’ of values from 1-20 (where 20 and 1 are neighbors)” to distinguish species diversity. Here the distance of one phenotype from another represents their level of similarity.

To use CA as prey pattern, an agent’s genome is constructed as an 8 bit binary value with a decimal range between 0 to 255. Each of this 256 values have a unique CA pattern associated with it. When storing this pattern in Hopfield memory we take a linear representation of this 2-D pattern, while to consider similarity between two pattern we evaluate the hamming distance between their linear representation.

With CA based pattern representation, population of prey species with a specific CA pattern can be grouped as one single species. And by restricting inter species reproduction we can control the diversity of patterns. But there is mutation applied when similar species mate with each other, so new species do born out of generations of existing species. So that is why we have two separate mutation rate while reproducing prey species. One being the “Pattern Mutation Rate” (default values are mentioned in table 4.7) with which we control mutation of the first 8 bits of the genome while the “Genome Mutation Rate” is used to control mutation of rest of the 9 bit genome (table 4.5). Similar efforts of multiple mutation rate at varying location has been used in developing Echo as mentioned in section 2.3.2 [HJF97].

4.4.3 Genome

The Genome of the prey species consists of 17 bits as presented in table 4.5. The first eight bits represent the rule, which is used to generate Cellular Automata pattern. The next two bits are used to represent palatability of the organism explained detail in section 4.4.5. The next six bits is the magnitude of the force with which mobility of the organism is calculated. The 17th bit is used to evaluate reproduction capability of the organism, depending on which the prey species is either capable or not capable to participate in reproduction.

Pattern(8)	Palatability(2)	Mobility(6)	Reproduction(1)
10101101	01	110001	1

Table 4.5: Distribution and purpose of each gene of the 17 bit prey genome.

4.4.4 Reflection of punctuated equilibrium

Punctuated equilibrium has been defined in section 3.4 as the evolutionary process that is more inclined to cladogenesis instead of gradualism. Also Turner [Tur88] emphasizes on punctuated equilibrium to describe the evolution of mimicry instead of phyletic gradualism.

The design of the model under discussion also follows Turner’s explanation in terms of evolving mimicry. As it can be observed, new CA patterns evolve from existing ones in prey population just by a single mutation in the pattern gene. Mimics do not follow a gradual process of evolution to look close to models but rather the change happens randomly through a single step mutation. The mutations that are favored, helps the mimics to survive while the unfavored ones fail to persist. It can be observed later in table 5.3 how CA patterns of prey species can have vastly different configuration for a unit change in their representative gene.

4.4.5 Genetic representation of palatability

The palatability of each prey species is fixed and has been represented with 2 bits of the genome giving it a range of 0 to 3 with four levels of palatability. The combinations are as follows:

Gene (Index 8 to 9)	Palatable
00	True
01	True
10	False
11	False

Table 4.6: Genetic representation of palatability

This representation in table 4.6 is unlike Franks and Noble [FN03] where palatability level has been used on a scale between zero and one (least to most palatable), where 0.5 is neutrally palatable.

4.4.6 Interaction between other Prey and Predators

The prey have been defined to have many conglomerate behavior in the environment. Prey interaction with other prey species and with predators make the evolution of mimicry possible. Mobility of prey species and their reproduction capability are two important behaviors which result from interaction.

Mobility

The mobility genes of the prey consist of 6 bits. These six bits are used to calculate the force with which each prey try to move towards any neighborhood cell. The algorithm sorts all neighboring cell descending to the number of prey species. Then it selects the cell which contains the highest number of prey with zero predator. If all the neighboring cells contain predators, then the algorithm sorts the neighboring cells descending on the number of predators and chooses the one which contains the least. Algorithm 4.2 explains in detail.

Algorithm 4.2 Algorithm for updating position of the Prey species

for each step in time **do**

$forceMagnitude \leftarrow convertToDecimal(genome[10 - 15])$

Get list of neighborhood cells, sorted by population of prey species.

From the sorted cell list choose the one with the least number of predators and most number of prey.

Move towards the selected cell with $forceMagnitude$ unit of force.

end for

Reproduction

Every prey species starts reproducing when it reaches the “*Reproductive age limit*”. If it is capable of reproducing, which is decided based on its 17th bit gene, the prey will randomly select another prey species with similar pattern and palatability from the same cell and mate with it. Now the other prey also needs to be genetically capable of reproduction and needs to reach its age limit. A new genome is created from the existing genome of the two prey by applying single point crossover operation. Mutation is performed separately on the pattern gene and the rest of the genome, with two different rates to control them using the values in table 4.7. So there is two point mutation for the genome.

Algorithm 4.3 Algorithm for reproduction of the Prey species

for each step in time **do**

capableToReproduce \leftarrow true or false depending on the 17th bit of the genome, and maturity on reproduction age.

if *capableToReproduce* == *true* and *currentCellPopulation* > 1 **then**

anotherPrey \leftarrow Select random prey from same cell.

if *anotherPrey* is alive and *capableToReproduce* and of similar *pattern* and *palatability* **then**

Perform genetic cross over and mutation to create new genome.

Create new prey with new genome, and release into environment.

Record reproduction time for both prey.

end if

end if

end for

Parameter	Value	Description
Prey Size	2 to 5	Size of the prey species in the 3D FormAL environment.
Reproduction age limit	100	Minimum number of iterations or time steps a prey species need to be present in the simulation to get reproduction capability
Reproduction interval	1000	Number of iterations a prey need to wait before reproducing again.
Pattern Mutation Rate	0.05	Rate of Mutation of the pattern genome.
Genome Mutation Rate	0.5	Rate of mutation of the rest of the genome excluding the pattern gene.
Demise Age	2000	Age at which the prey species will be removed from the environment.

Table 4.7: Parameters to control prey population and visibility.

4.5 Predator

Predators in the system are designed to provide selection pressure to *models* and *mimics* for the evolution of mimicry. Similar to prey species, they are agents in the FormAL environment capable of mobility and reproduction. In addition to it, these agents are equipped with Hopfield Network Memory to be able to learn and recognize patterns of the prey species. Their mobility and reproduction capability are controlled at the genetic level, while their memory is not genetically controlled, as it was not possible to come up with a genetic representation for Hopfield Network. So every new predator species gets to born with zero memory and with no inheritance from parents. A set of parameters are defined to control predators' population and learning ability in the environment. Details of these parameters are given in table 4.9.

4.5.1 Learning

The objective of a predator's interaction with prey is always to consume it. But based on the prey's pattern and palatability, the predator will either be able to consume it or throw it back to the environment. At this event the predator needs to learn the pattern with which the prey has been represented. The pattern represents palatability of the prey species, at least to the predator. To store this new pattern into memory, each predator is equipped with a Hopfield Network Memory. Every time a new interaction is made by the predator its memory is initialized with all the existing pattern that has already been encountered and the new one. The learning procedure used for this memory is Hebbian Learning.

Hebbian Learning

Hebb's postulate of learning is the oldest and most famous of all learning rules; it is named in honor of the neuropsychologist Donald Hebb(1949). Hebb's book *The Organization of Behavior* [Heb49] states the following:

“When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic changes take place in one or both cells such that A's efficiency as one of the cells firing B is increased.”

The following *general learning rule* is adopted in neural network studies: *The weight vector* $\mathbf{w}_i = [w_{i1} \ w_{i2} \ \dots \ w_{in}]^t$ *increases in proportion to the product of input* \mathbf{x} *and learning signal* r . The learning signal r is in general a function of \mathbf{w}_i, \mathbf{x} , and sometimes of the teacher's signal d_i . Thus we have,

$$r = r(\mathbf{w}_i, \mathbf{x}, d_i) \quad (4.5.1)$$

The increment of the weight vector \mathbf{w}_i produced by the learning step at time t according to the general learning rule is,

$$\Delta \mathbf{w}_i(t) = cr[\mathbf{w}_i(t), \mathbf{x}(t), d_i(t)]\mathbf{x}(t) \quad (4.5.2)$$

when c is a positive number called the *learning constant* that determines the rate of learning.

For Hebbian learning rule the learning signal is equal simply to the neuron's output. We have

$$r \triangleq f(\mathbf{w}_i^t \mathbf{x}) \quad (4.5.3)$$

The increment $\Delta \mathbf{w}_i$ of the weight vector becomes

$$\Delta \mathbf{w}_i = cf(\mathbf{w}_i^t \mathbf{x})\mathbf{x} \quad (4.5.4)$$

The single weight w_{ij} is adapted using the following increment:

$$\Delta w_{ij} = cf(\mathbf{w}_i^t \mathbf{x})x_j \quad (4.5.5)$$

This can be written briefly as

$$\Delta w_{ij} = o_i x_j, \text{ for } j = 1, 2, \dots, n \quad (4.5.6)$$

Here, \mathbf{o} is the output vector. The learning rule requires the weight initialization at small random values around $\mathbf{w}_i = \mathbf{0}$ prior to learning. The Hebbian learning rule represents a purely feed-forward, unsupervised learning. The rule implements the interpretation of the classic *Hebb's postulate of learning* stated above.

The rule states that if the cross product of output and input, or correlation term $o_i x_j$ is positive, this results in an increase of weight w_{ij} ; otherwise the weight decreases. It can be seen that the output is strengthened in turn for each input presented. Therefore, frequent input patterns will have most influence at the neuron's weight vector and will eventually produce the largest output.

Training for Hopfield Network used for predator's memory is done by Hebbian Learning. Initially the weights are all set to zero. Using Hebbian rule, the outer product of the input - output vector pairs are calculated for each pattern. As Hopfield is a feedback network, the output of the network is also the input. The outer vector matrix of all the patterns are summed to come up with the final weight matrix. Each component of the weight matrix $\mathbf{W} = \{w_{ij}\}$ is given by:

$$w_{ij} = \sum_{p=1}^P s_i(p)t_j(p), i \neq j \quad (4.5.7)$$

$$w_{ij} = 0, i = j$$

where P is the number of patterns. Vectors \mathbf{S} and \mathbf{T} are respectively, the input and the desired output of the network.

4.5.2 Design of Memory with Hopfield Network

Hopfield Network

Algorithm for the Hopfield model is described in the following:

1. *Learning.* Let $\xi_1, \xi_2, \dots, \xi_\mu$ denote a known set of N-dimensional fundamental memories. Use the outer-product rule (i.e. Hebb's postulate of learning) to compute the synaptic weights of the network as

$$w_{ji} = \begin{cases} \frac{1}{N} \sum_{\mu=1}^M \xi_{\mu,j} \xi_{\mu,i} & j \neq i \\ 0 & j = i \end{cases} \quad (4.5.8)$$

where w_{ji} is the synaptic weight from neuron i to neuron j . The elements of the vector ξ_μ equal ± 1 (bipolar). Once they are computed, the synaptic weights are kept fixed.

2. *Initialization.* Let ξ_{probe} denote an unknown N-dimensional input vector (probe)

presented to the network. The algorithm is initialized by setting

$$x_j(0) = \xi_{j,probe}, j = 1, \dots, N \quad (4.5.9)$$

where $x_j(0)$ is the state of neuron j at time $n = 0$ and $\xi_{j,probe}$ is the j th element of the probe $\boldsymbol{\xi}_{probe}$.

3. *Iterate Until Convergence.* Update the elements of state vector $\boldsymbol{x}(n)$ asynchronously (i.e., randomly and once at a time) according to the rule

$$x_j(n+1) = \text{sgn} \left(\sum_{i=1}^N w_{ji} x_i(n) \right), j = 1, 2, \dots, N \quad (4.5.10)$$

Repeat the iteration until the state vector \boldsymbol{x} remains unchanged.

4. *Outputting.* Let \boldsymbol{x}_{fixed} denote the fixed point (stable state) computed at the end of the step 3, The resulting output vector \boldsymbol{y} of the network is

$$\boldsymbol{y} = \boldsymbol{x}_{fixed} \quad (4.5.11)$$

Step 1 is the storage phase, and steps 2 through 4 constitute the retrieval phase.

Input to memory

Each prey contains an evolving cellular automata which is represented by a binary genome. This two dimensional pattern is serialized to be available as a one dimensional binary array, which is taken as input for any predator organism trying to interact with the prey. This binary representation of the pattern gets converted to a bipolar representation. Each input pattern consists of $m \times n = mn$ components, each component representing one pixel of the pattern (m and n representing each dimension). The m by n pattern configuration will be serialized by putting all row vectors in one single row sequentially.

Predator attack algorithm

As soon as a predator reaches its attack age it selects random prey species around its vicinity and starts attacking them. This attack process also involves recognition of prey pattern, which is at the heart of mimicry simulation for this research. Two parameters have been

defined to limit predator memorization and recognition process as both of these processes are computationally expensive. The “Hopfield Minimum Memory Size” (default value in table 4.9) is the number of memory a predator needs to store before making intelligent decisions about attacking a new prey species. After a predator is born, it start attacking prey without any caution. But after every attack the predator will store its pattern and palatability level inside its memory. As soon as it reaches the minimum memory size, it will start making intelligent decision about attacking the next species. It will try to recognize the pattern and if found palatable, prey will be consumed. Otherwise prey will be thrown back into the environment. If the pattern is not recognized predator will try to consume it and in the process will store its palatability and pattern into memory.

In this way the predator memory is limited to “Hopfield Maximum Memory Size”. After reaching this memory predator will not store any more new pattern but will try to associate with the existing ones it has already stored. Algorithm 4.4 and 4.5 explains the process in every detail.

Algorithm 4.4 Algorithm for attacking Prey species

```

for each step in time do
  if Predator has reached Minimum attack age then
     $preyToAttack \leftarrow$  select random prey species from the same cell.
    if  $currentMemorySize < MinimumMemorySize$  then
       $attemptToKill(preyToAttack)$ ; {Function defined in algorithm 4.5}
    else
      recognize  $preyToAttack$  pattern from memory.
      if prey is not recognized or prey is palatable then
         $attemptToKill(preyToAttack)$ 
      else
        Prey is recognized as unpalatable, so release it back into environment.
      end if
    end if
  end if
end for

```

Algorithm 4.5 $attemptToKill(preyToAttack)$

```

Kill  $preyToAttack$ .
if Memory size  $< MaxMemorySize$  then
  Associate  $preyToAttack$  pattern and palatability and store into memory.
  Recalculate Hopfield Memory Weights using Hebbian Learning outer product rule.
end if

```

4.5.3 Genome

Each predator has a 5 bits genome as explained in table 4.8. The first 4 bits represent its mobility and magnitude of the force component with which it moves around the environment. The last bit controls reproduction capability of each species.

Mobility(4)	Reproduction(1)
1101	1

Table 4.8: Distribution and purpose of each gene of the 5 bit predator genome.

4.5.4 Mobility and reproduction capability

Movement behavior of a predator is calculated from its genome. The first 4 bits of the genome are converted from binary to decimal to determine the magnitude of force at which it will move towards the maximum crowd of prey present within its neighborhood. So magnitude of movement of a predator varies within a range of 0-15 units. If no prey is present in the neighborhood then this force is active in trying to keep predators distributed all over the cells. A predator chooses the neighborhood cell which contains the least number of predators. When the neighborhood contains zero predators, it would select any one of them randomly and move towards that cell.

Algorithm 4.6 Algorithm for updating position of the Predator species

for each step in time **do**

$forceMagnitude \leftarrow convertToDecimal(genome[0 - 3])$

 Get list of sorted neighborhood cells. Sorted by population of prey species.

 From the sorted cell list choose the one with the least number of predators and most number of prey.

 Move towards the selected cell with $forceMagnitude$ unit of force.

end for

The screen shot in figure 4.2 is from the simulation with only 40 predator species. It is presented to show the behavior of predator species in the simulation, in absence of any prey species. It can be observed the predators are distributed all over the cells with a constant mobile behavior to switch to another neighboring cell depending on the one containing the least number of predator and also most number of prey species. This behavior of predator has been designed to enforce the predatory behavior of these agents and also to have increased predator prey interaction in the simulation in terms of one agent chasing the

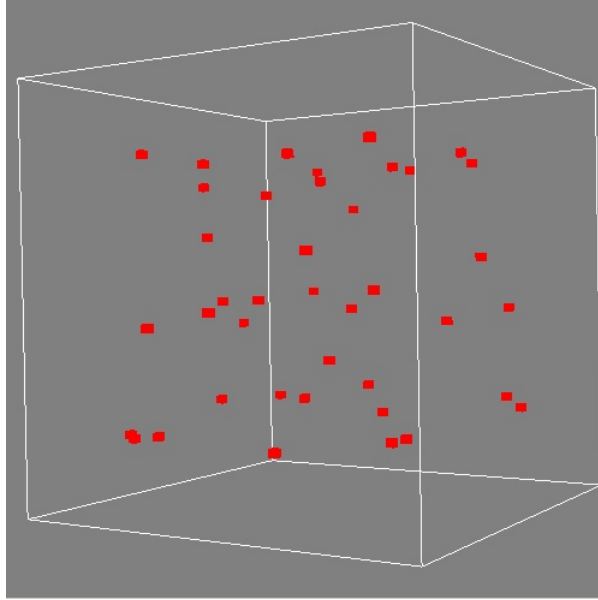


Figure 4.2: A population of 40 predators distributed over the environment.

other for survival of species.

4.5.5 Reproduction process

The fifth gene of the predator is used to represent their capability of reproduction. Depending on its binary value a predator in the simulation will or will not be able to reproduce. The reproduction process for predators is similar to prey species. As the learning capability of predators do not have any genetic representation, only the mobility behavior and reproduction capability takes effect in the reproductive process from one generation to another.

There are two control parameters that effect the reproduction of predator species. The first is “*Reproduction Age Limit*”. This is the minimum age a predator has to reach before starting to involve itself for reproduction. This parameter has been set to 500 iterations during the result and analysis section of the thesis. The second parameter is “*Reproduction Interval*” which has been varied in the simulation from 1000 to 3000 iteration depending on the population of palatable prey species. This is a very important parameter for the simulation as it determines the overall predator population and its rate of increment. Depending on this value we can control the rate of predation on prey species, which on the other hand controls the rate of mimetic behavior of the overall prey population.

Algorithm 4.7 Algorithm for reproduction of the Predator species

for each step in time **do**

capableToReproduce \leftarrow true or false depending on the 5th bit of the genome, and maturity on reproduction age.

if *capableToReproduce* == *true* and *currentCellPopulationOfPredators* > 1 **then**

anotherPredator \leftarrow Select random predator from same cell.

if *anotherPredator* is alive and *capableToReproduce* and also reached reproduction maturity age **then**

Perform genetic cross over and mutation to create new genome.

Create new predator with new genome and zero memory, and release into environment.

Record reproduction time for both predators.

end if

end if

end for

When the above two conditions are met, meaning the predator reaches its age for reproduction and also crosses its age interval, it randomly select another predator residing in the same cell. If this random predator is capable to reproduce depending on its 5th gene, then with a random single point crossover and mutation a new predator species is born which also resides on the same cell and initializes with zero memory configuration. Similarly when the new born predator reaches its maturity of reproductive age and if it is capable of reproduction then the process iterates itself.

4.6 Conclusion

This model has been designed from a practical point of view; to come up with efficient results and achieve the main objective, *evolution of mimicry*. Creation and transformation of different mimicry ring and also the dynamics of it has been integrated to achieve interesting results. This model can also be considered as a complex adaptive system similar to Holland's work on Echo [Hol96]. The seven basics of a complex adaptive system which are: Aggregation, Tagging, Nonlinearity, Flow, Diversity, Internal Models and Building blocks [Hol96] are present in this model. Individual components of this model such as the different

Parameter	Value	Description
Minimum Memory Size	2 to 6	Minimum number of patterns stored in memory before predators start making intelligent decisions.
Maximum Memory Size	10	Maximum number of patterns to be stored in memory. Limited to reduce processing time.
Hopfield Maximum Iterations	20	Maximum number of iterations for Hopfield Network to recognize a pattern. Usually the network reaches a steady state before that. But this restriction is to avoid infinite loop in case the network never reaches a steady state.
Attack Age	500	minimum age a predator needs to reach to be able to attack prey species.
Attack Interval	100	Interval of time which needs to pass before a predators attacks its next prey.
Genome Mutation rate	0.3	Mutation rate for the 5 bit genome of the predators representing their mobility and pattern recognition capability.
Reproduction Age Limit	500	Minimum age a predator needs to reach before engaging in reproduction.
Reproduction Interval	1000 to 3000	Interval of time a predator needs to pass between two reproduction process.
Demise Age	2000 to 7000	Age at which a predator is considered as dead.

Table 4.9: Parameters to control predator population and pattern recognition capability.

types of agents and their properties can be considered as *building blocks*. Each prey species are *tagged* with individual pattern and palatability with which predators recognize them. We are providing different properties, behaviors and goals to the agents but setting them free in the environment to observe their *aggregate* behavior, resulting in *non-linear* or unpredictable outcome. The model has its *flow* as it progresses in time. Also there is *diversity* of prey species in the environment (4.4.2).

To explain internal model for a complex adaptive system Holland has used specifically the example of evolution of mimicry [Hol96]. He choose to use the term ‘Internal Model’ to refer to the mechanism for anticipation. Predators in this model anticipate the bitter taste of prey, by making intelligent decisions for consuming them. This mechanism for anticipation can be considered as the property of *internal model* for this specific study.

Chapter 5

The Results

“The most incomprehensible thing about the world is that it is comprehensible.”

- Albert Einstein

5.1 Introduction

It is always a challenge to extract and analyze data from an Artificial Life simulation. Data and analysis in this simulation has been concentrated on evaluating whether evolution of mimicry has taken place. This evaluation can be made with the number of different rings that has been created and the size of each of those rings along with the population of palatable and unpalatable species. Also it can be established whether Batesian Mimicry and Mullerian Mimicry have taken effect by analyzing the data set of these populations.

5.2 Mimicry Ring Reports

The mimicry ring reports consist entirely of the population of prey species categorized according to pattern and palatability. Data is stored at every instance of time of the simulation keeping an interval of 10 iterations. As the number of rings that get generated reaches as many as 50 or more, and all the population of ring do not last for the entire simulation, so while storing data we have taken the most populous of the surviving 8 rings to plot. Parameters are mentioned in table 5.1.

Parameter	Value	Description
Mimicry Ring Hamming Distance	10 % of the Pattern Size	If the Hamming distance between the pattern of the model and the mimic is 10 % the size of the pattern then they are considered within the same ring.
Number of Rings to report	8	This value is the number of most populous rings that are included in the report.

Table 5.1: Parameters to mimicry ring report.

5.3 Initial configuration with two prey species

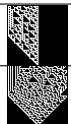

	Prey configuration		Predator configuration	
Population	Rule110 (Palatable)		108	10
	Rule30 (Unpalatable)		108	
Reproduction	Age Limit	100		500
	Interval	1000		1200
Mutation Rate	Pattern	0.05		0.3
	Genome	0.5		
Demise Age	2000		2500	
Minimum Attack Age			500	
Memory Configuration			Minimum	2
			Maximum	10

Table 5.2: Agent configuration of 2 prey species

The set of parameters in table 5.2 were carefully selected to be the initial condition for this run of the simulation. This test has been done with two sets of prey species with very different Cellular Automata pattern and with opposite palatability and equal population. To control reproduction of the prey species their age limit has been set to 100 iterations into the time the species were alive. And the reproduction interval was set to 1000 iterations.

Pattern mutation rate has been set to a minimal level of 0.05 as by increasing this

variable it is possible to increase the size of the number of mimicry rings present in the simulation. The genome mutation rate controls the rate at which genome of the child prey species will deviate from their parents. As mentioned earlier the genome mutation rate has been separated from the pattern mutation rate to bring more control to the number of mimicry rings generated.

Prey demise age has been kept to 2000 iterations while predator demise age is set to 2500. But in the later experiments predator demise age has been increased to 5000 iterations. Predators in this simulation generate selection pressure for the evolution of mimicry. So the longer a predator is present in the simulation it will be making intelligent decisions in term of selecting which prey species to consume and which one to avoid. But with the current rate of demise for predator we were able to create successful mimetic population of prey species as we will see in the analysis in the following results.

Initial population of predator species has been set to 10 which is in accordance with the prey population in the simulation. The reason for such low number of predator is, unlike prey species which are consumed by predators, there is no cause for the predator species to die except their natural cause of death, that is to reach their demise age. So predator population can explode very easily. That is why their population is controlled in a restrictive manner with the help of high reproduction age limit and reproduction age interval.

The memory configuration size for predators is a very interesting parameter. The minimum memory size is directly associated with the number of prey species with which we initiate the simulation. Otherwise evolution of mimicry is not observed. As mentioned in table 4.9 the minimum memory size is the number of prey species predator would consume before starting to make decisive consumption of prey species. After the initial birth of a predator and when the minimum attack age is crossed, it starts consuming prey species present in its vicinity without making any judgment. At this point its memory size is zero. As long as the minimum memory size is not reached predators will blindly consume prey species and insert their CA pattern and associated palatability into its Hopfield memory bank. No later than the minimum memory size has been reached, predators will start making intelligent decision about consuming its prey species. When catching a prey if its memory tells that it is palatable, it will consume that prey. Otherwise if memory recognizes it to be unpalatable predator will certainly let it go.

Now as we know the behavior of Hopfield memory, a recognition result will always be

achieved depending on the similarity of the patterns stored in memory. So when minimum memory size has been reached the predator will always make a decision based on the similarity of the prey pattern currently captured and the patterns stored in memory.

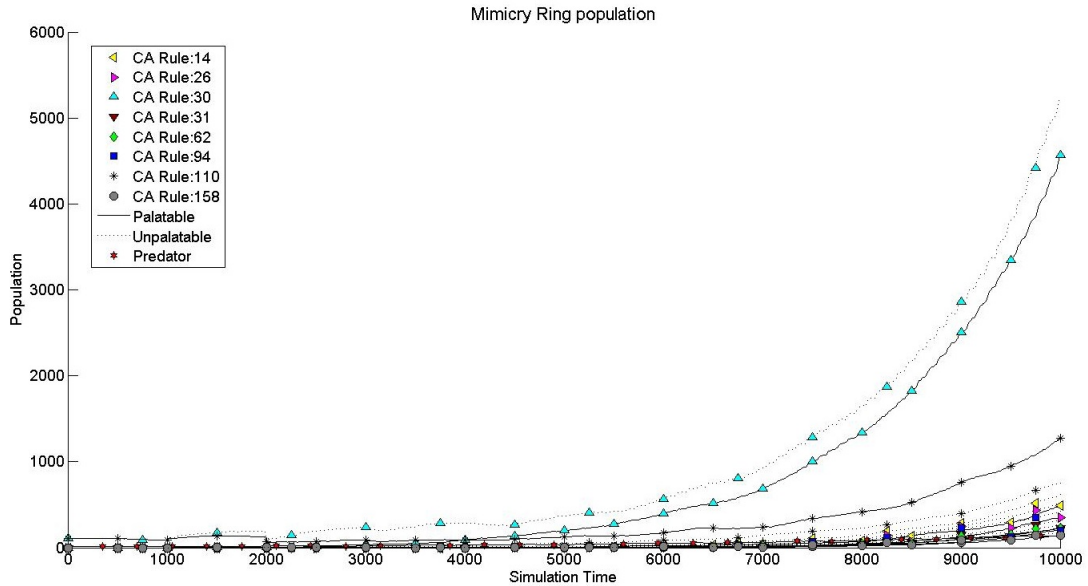


Figure 5.1: Population distribution of mimicry rings, initialized with 2 prey species, 10k iterations

The plot in Figure 5.1 is simulation time verses prey population after running it for 10000 iterations. With the initial configuration in the above table we can observe that multiple rings of prey population have been created. Two prey species are considered to be in a ring if their CA pattern have a Hamming distance within 10 bits. This parameter is mentioned in table 5.1. Population of palatable species has been represented with line curve while population of unpalatable species has been presented with dotted curve. Different signs of squares, triangles and diamonds have been used to distinguish between species of prey population. The simulation was initiated with two prey species having CA rule of 30 and 110 and being palatable and unpalatable consecutively.

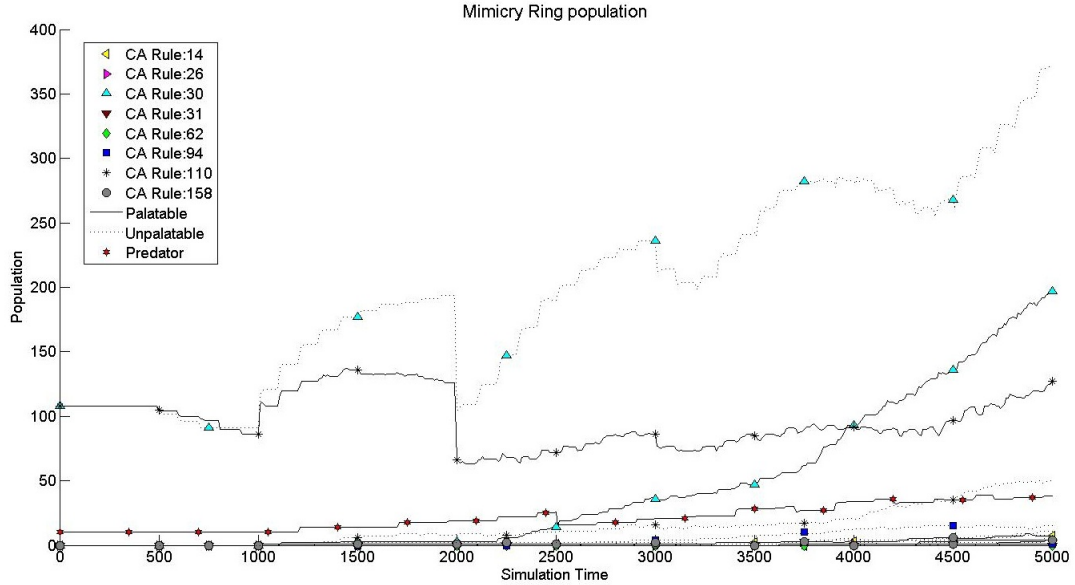


Figure 5.2: Population distribution of mimicry rings, initialized with 2 prey species, 5k iterations.

The plot of figure 5.2 is from the same experiment as of figure 5.1, but only up to 5000 iterations. By closely observing the graph of figure 5.2, we could see the population of two species have started dropping at around time 500 when the initial predator population reaches the maturity for consuming prey species (effect of the parameter ‘Minimum Attack Age’ from table 5.2). At around time 1000 the prey population starts reproducing as the population increases (controlled by ‘Reproduction Age Limit’ from table 5.2). At the same time different other species of prey gets to be born with mutated CA patterns. There is a straight drop of prey population at time 2000, when all the prey species with which the simulation was initialized come to its demise age (controlled by ‘Prey Demise Age’ at table 5.2). Over time the population of CA Rule 30 dominates the population (figure 5.1) as most predators recognize it as unpalatable. Similarly a palatable population of CA Rule 30 or within the same ring of palatable species starts rising, while at one point overlaps the population of CA Rule 110 (Time: 4000 approx.). CA Rule 110 was initialized as a set of palatable species.

We can observe from the above result that the evolution of mimicry has taken effect. A population of mimics were successfully able to exceed the population of other prey species, and the reason being, avoidance by predators of prey pattern similar to unpalatable ones. We can conclude that Batesian mimicry has taken effect in the simulation.

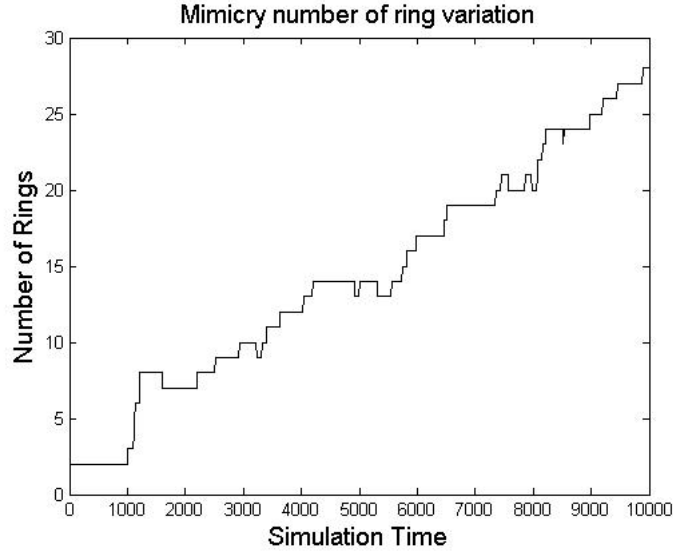


Figure 5.3: Number of mimicry rings, initialized with 2 prey species.

Observing from figure 5.3 the number of rings in this simulation makes a slow increase from 2 at the initial configuration to 27 rings at the end of 10000 iterations. A small change in CA genetic representation can have a very large effect in terms of the phenotype of the pattern with which the prey is represented. For example if we take a look at the set of pattern genotype with very different phenotype in table 5.3.

CA Rule	60 \equiv 00111100	61 \equiv 00111101	62 \equiv 00111110
Pattern			

Table 5.3: Difference in prey pattern genotype and phenotype

All the patterns in table 5.3 have a genetic bit difference of 1. So by a single mutation there can be three different set of phenotype for a child organism from its parent. This is largely the reason for the increased number of mimicry rings created in the simulation. Only the 8 most populous rings are presented in the graphs with population verses simulation time.

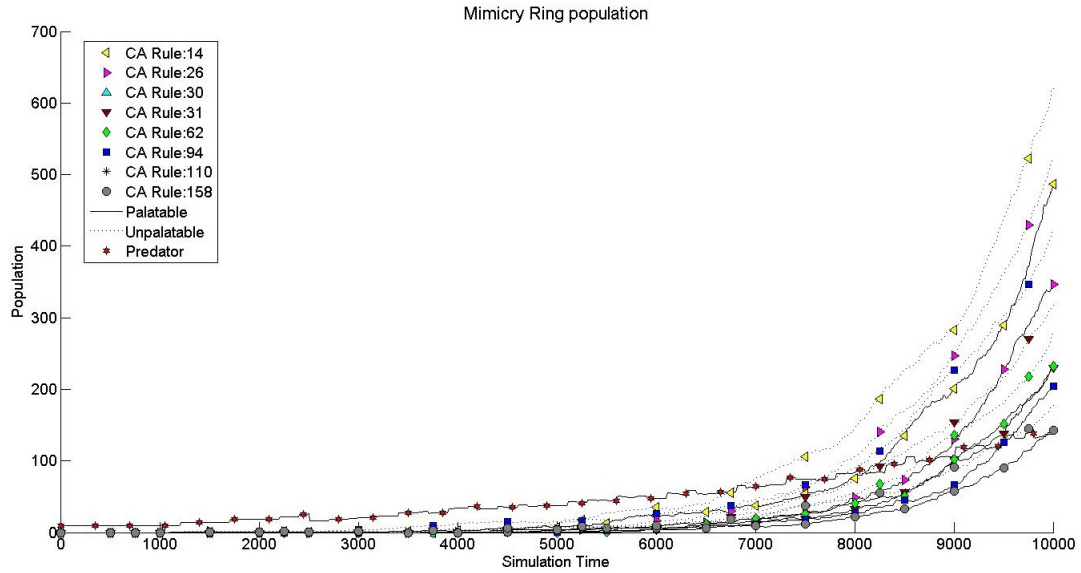


Figure 5.4: Population distribution of generated mimicry rings, initialized with 2 prey species.

Figure 5.4 is the same experimental data of figure 5.1 but with only the newly generated ring of prey species. The initialized dominant population of CA rule 110 and 30, both palatable and unpalatable has been eliminated from this graph to observe the newly created rings. In this plot, among the population of new patterns the unpalatable ones dominate (CA Rule 14, 26, 94). Also the palatable counterpart of the unpalatable population is increasing their number. Species with *un-mimicked* palatable pattern is very few in number.

The screen shot in figure 5.5 is for one instance of time of the simulation. The red agents are predators while the black agents with different textured patterns are the prey species. According to their behavior most of the prey species are flocking together in a group while also being chased by the predators, whose sole purpose is to consume prey species.

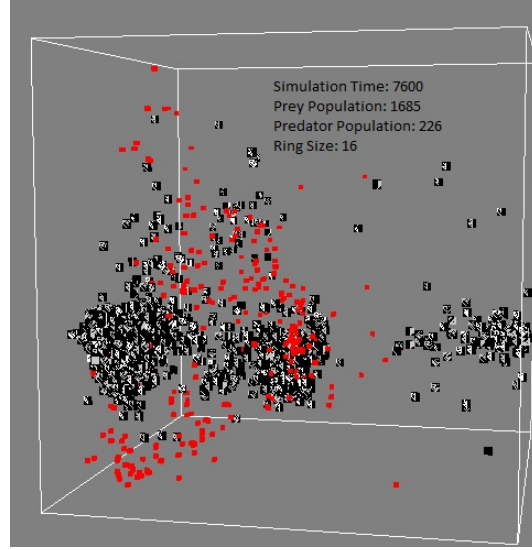


Figure 5.5: Graphical representation of the model, simulation time: 7600.

5.4 Initial configuration with four prey species





	Prey configuration		Predator configuration	
Population	Rule110 (Unpalatable)		50	10
	Rule30 (Palatable)		50	
	Rule55 (Unpalatable)		50	
	Rule190 (Palatable)		50	
Reproduction	Age Limit	100	500	
	Interval	1000	1400	
Mutation Rate	Pattern	0.05	0.3	
	Genome	0.5		
Demise Age	2000		2500	
Minimum Attack Age			500	
Memory Configuration			Minimum	4
			Maximum	10

Table 5.4: Agent configuration of 4 prey species

This run of the simulation has been initialized with four prey species with very different CA pattern configuration. The predator reproduction interval has been increased to 1400, while the minimum memory size has been increased to 4 instead of 2 as the predator is expected to memorize four different species of prey before starting to make intelligent decision about consuming them. The rest of the parameters remain the same as from the experiment in section 5.3.

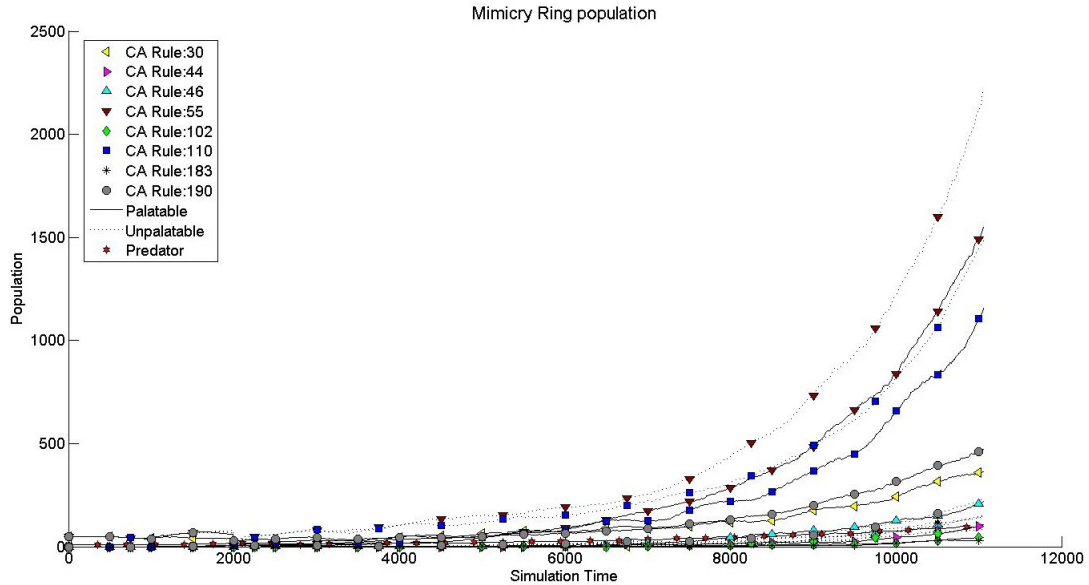


Figure 5.6: Population distribution of mimicry rings, initialized with 4 prey species.

By observing the graph in figure 5.6 we can see the two rings of unpalatable species (CA Rule 55, 110) which was put at the initial state of the simulation has dominated after 10000 iterations. CA rule 55 has taken over all other species while CA 110 is following it (represented with dotted curve). Also the palatable prey species with similar patterns to CA rule 55 and 110 are taking full advantage of their deceiving pattern and increasing their population (represented with line curve). The two palatable set of patterns CA Rule 190 and 30 are unable to dominate the population. Different other mimicry rings are present in the simulation as we can observe from the graph of number of rings verses simulation time in figure 5.7. The total number of rings reached somewhere near 37 while most of them have comparatively low representative population. Batesian mimicry is in full effect in this simulated environment.

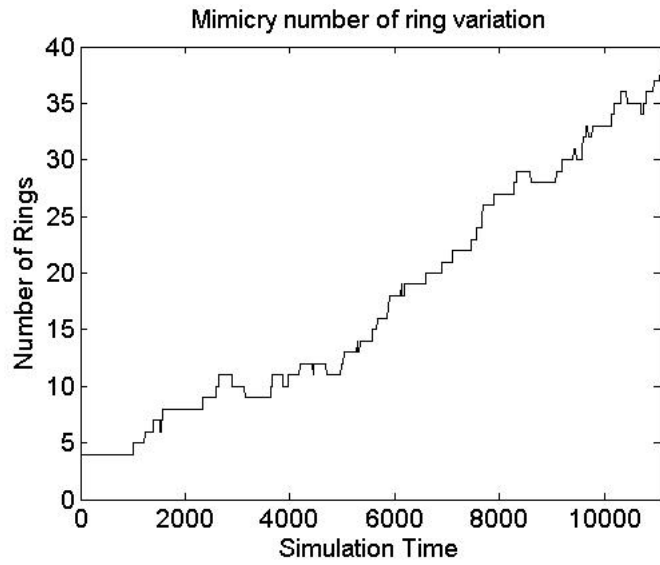


Figure 5.7: Number of mimicry rings, initialized with 4 prey species.

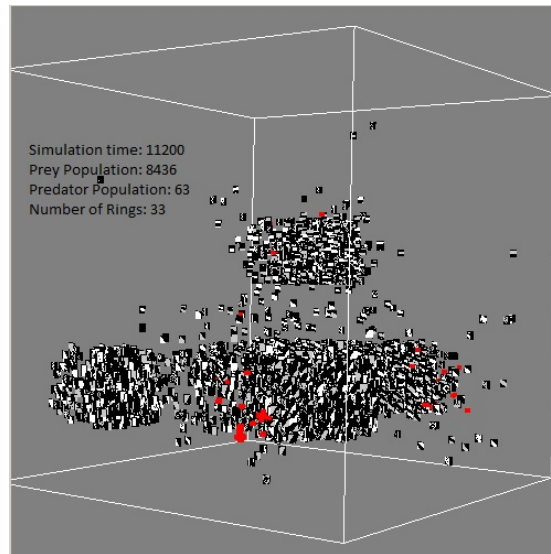


Figure 5.8: Graphical representation of the model initialized with 4 prey species, simulation time: 11200.

5.5 Increased initial population with four prey species





	Prey configuration		Predator configuration	
Population	Rule110 (Palatable)		150	20
	Rule30 (Palatable)		150	
	Rule55 (Unpalatable)		150	
	Rule190 (Unpalatable)		150	
Reproduction	Age Limit	100		500
	Interval	1000		2500
Mutation Rate	Pattern	0.05		0.3
	Genome	0.5		
Demise Age	2000		7000	
Minimum Attack Age			500	
Memory Configuration			Minimum	4
			Maximum	10

Table 5.5: Agent configuration of 4 prey species with increased population

In this run of the simulation some significant parameters have been updated. First one is the initial population of prey species. It has been increased from 50 to 150 for each of the prey species, while increasing the overall population from 200 to 600. Secondly, predator population has been increased from 10 to 20. Thirdly, predator demise age has been increased from 2500 to 7000 while their reproduction interval has been increased from 1400 to 2500. Increasing predator demise age can have an interesting effect, as the longer a predator is present in a simulation, the longer it will be making intelligent decision in terms of consuming its prey and there will be increased selection pressure from the entire population of predators.

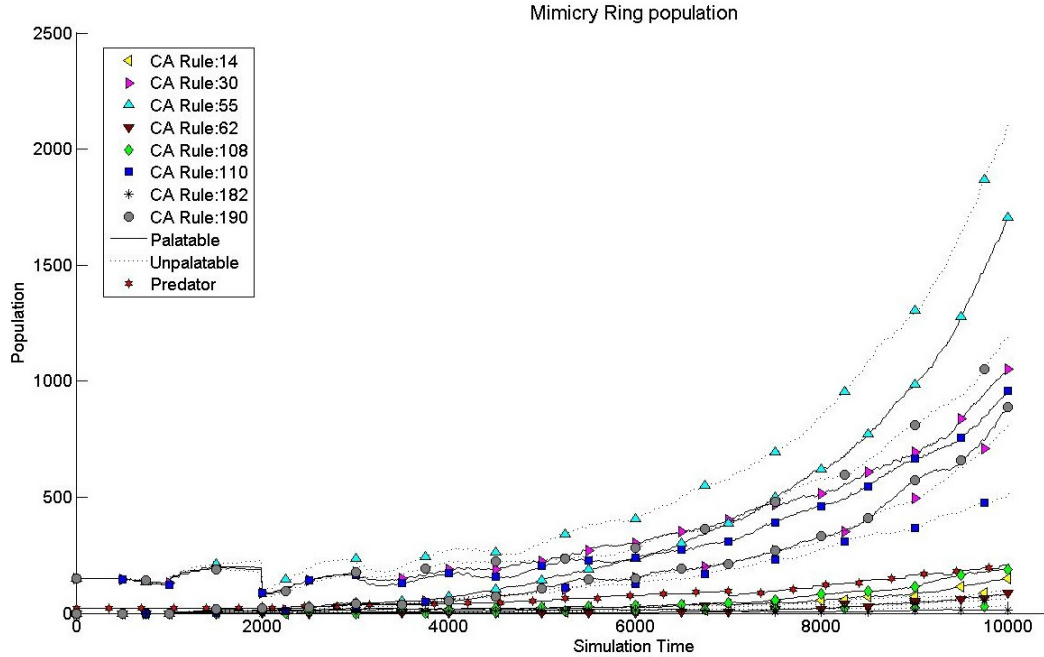


Figure 5.9: Population distribution of mimicry rings, initialized with 4 prey species with increased population.

Some very interesting diversity of mimicry rings can be observed in the simulation for updating these parameters. The unpalatable rings and their deceptive counterparts (palatable prey with similar pattern) still have their dominance. We can observe CA Rule 55 and 190 are the highest in number of population. Interestingly enough CA Rule 30 and 110 which started as palatable prey species has also taken dominance in the simulation, following 55 and 190. The unpalatable counter part of CA Rule 30 and 110 are also following the palatable population.

The total number of rings have increased much in this run compared to the previous one. To summarize the results of this experiment, we can observe, increased diversity of prey population and much more interaction between predator and prey.

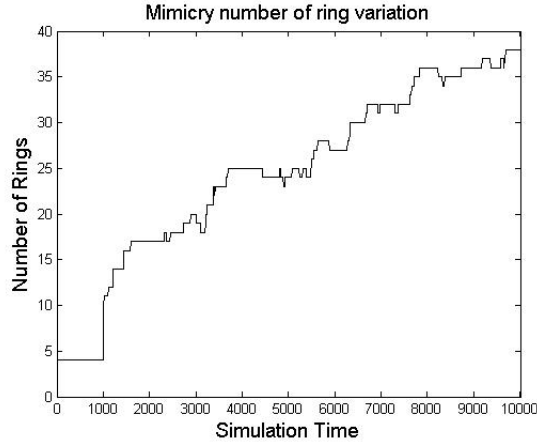


Figure 5.10: Number of mimicry rings, initialized with 4 prey species with increased population.

5.6 Initial population with six prey species

	Prey configuration		Predator configuration	
Population	Rule110 (Palatable)		150	30
	Rule30 (Unpalatable)		150	
	Rule55 (Palatable)		150	
	Rule190 (Unpalatable)		150	
	Rule57 (Palatable)		150	
	Rule105 (Unpalatable)		150	
Reproduction	Age Limit	100	500	
	Interval	1000	2000	
Mutation Rate	Pattern	0.05	0.3	
	Genome	0.5		
Demise Age	2000		7000	
Minimum Attack Age			500	
Memory Configuration			Minimum	6
			Maximum	10

Table 5.6: Agent configuration of 6 prey species.

To evaluate the simulation at a more complex level we increase the prey population to 900, consisting of 6 different species with very different pattern configuration. To boost predator-prey interaction we also increase the number of predator population to 30. Predator reproduction interval has been set comparatively lower in this simulation to 2000 iterations. As the initial population of prey species is very high, over time we are expecting even larger number of prey species. If predator population does not increase at similar rate there might be a prey population explosion were predator will have no effect on providing selection pressure for the evolution of mimicry. So that is why predator population control parameter has been set to this level. For memory configuration, the minimum number is set to 6, meaning the predator will have to store 6 different prey pattern configuration before starting to make intelligent decision about consuming them. As previously mentioned this number is always set in accordance with the initial number of different prey configuration.

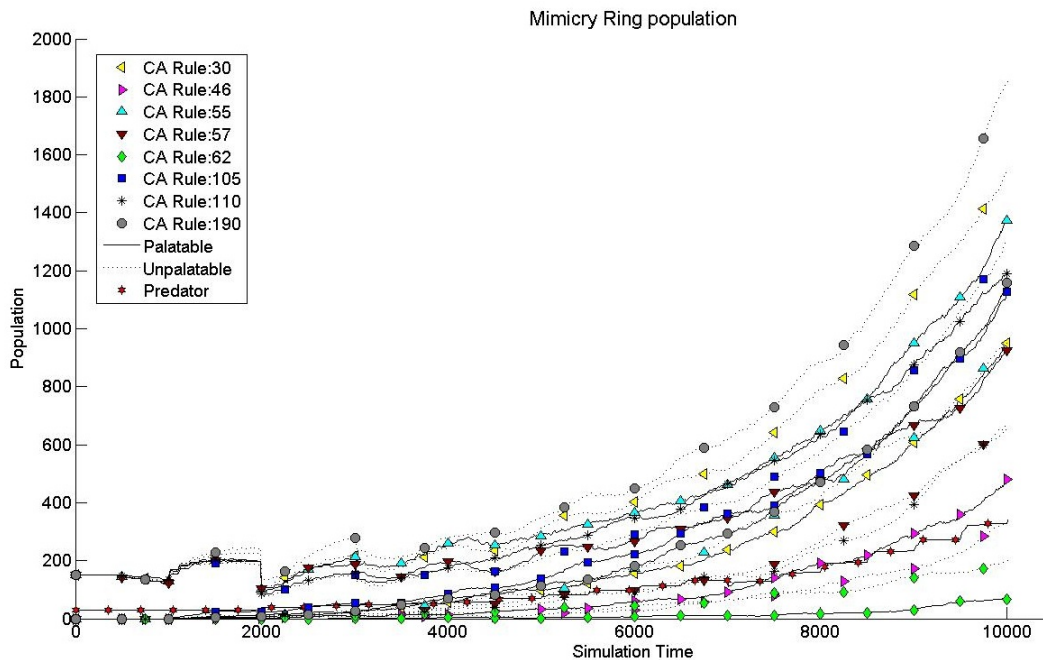


Figure 5.11: Population distribution of mimicry rings, initialized with 6 prey species.

With the above set of parameters we receive the plot of figure 5.11. An enormous diversity of species can be observed from this experiment. In addition to the six prey species with which the simulation starts, the total number of mimicry rings reach nearly 50. Observing the top 8 as presented in the plot of figure 5.11, effects of Batesian mimicry is obvious. Unpalatable CA Rule 190 and 30 dominates the population. Similarly unpalatable

CA Rule 105 is also dominant. But after 10000 iterations CA Rule 55 seems to have higher population than 105 even though 55 is a *palatable* set of species. This anomaly could be the reason for storing large number of patterns in the predator Hopfiled memory. As we know that the number of patterns stored in Hopfiled network is inversely proportional to its recognition rate. In this simulation memory size is controlled by the ‘Maximum Memory Configuration’ which is set to 10 in table 5.6. In this experiment, the value of ‘Minimum Memory Configuration’ is 6, which is rather a large number. So there could be possibility of wrong recognition by Hopfield memory. Then again this phenomenon is common if we consider behavior of predators in nature. It is definitely hard for a single predator to memorize multiple poisonous patterns and because of which Mullerian mimicry takes an effect as mentioned in section 3.3.

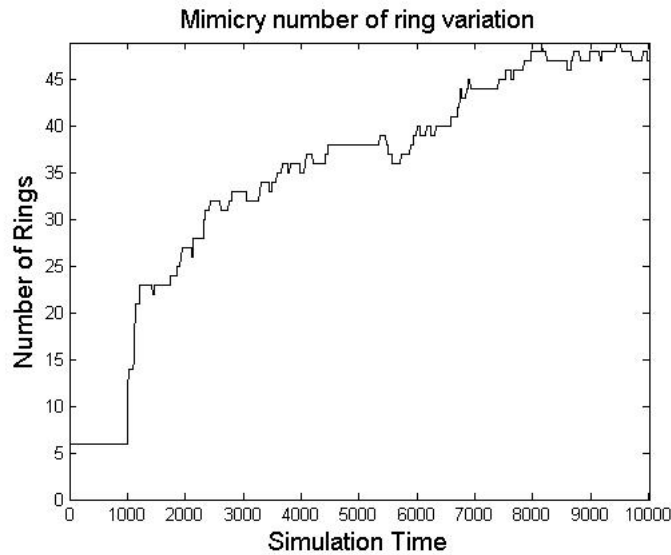


Figure 5.12: Number of mimicry rings, initialized with 6 prey species.

5.7 Initial configuration with only unpalatable species





	Prey configuration		Predator configuration	
Population	Rule110 (Unpalatable)		150	20
	Rule30 (Unpalatable)		150	
	Rule55 (Unpalatable)		150	
	Rule190 (Unpalatable)		150	
Reproduction	Age Limit	100	500	
	Interval	1000	2000	
Mutation Rate	Pattern	0.05	0.3	
	Genome	0.5		
Demise Age	2000		5000	
Minimum Attack Age			500	
Memory Configuration			Minimum	4
			Maximum	10

Table 5.7: Agent configuration of 4 prey species all unpalatable.

To further observe the effects of mimicry ring we initialize the simulation with all unpalatable prey species. After finding different anomalous behavior of predator recognition capability in section 5.6 for increased number of diverse prey species; for this experiment the initial number of species have been reduced from six to four. As explained earlier the minimum memory configuration is also set to four in accordance to the initial number of prey species. Rest of the parameters remain quite unchanged.

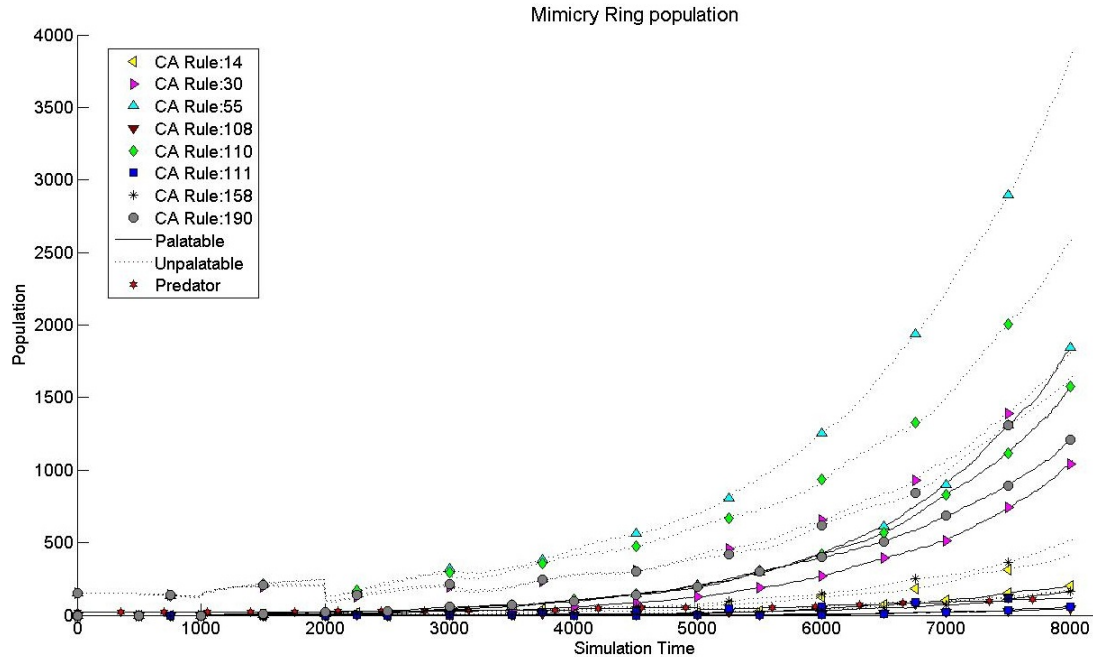


Figure 5.13: Population distribution of mimicry rings, initialized with 4 prey species all unpalatable.

The results according to figure 5.13 are much expected. The population of unpalatable species have prevailed. After nearly 8000 iterations we can see unpalatable species of CA rule 55, 110, 30 and 190 have prevailed with the most population. All of their palatable counter parts are also increasing their population deceiving the predators. We observe the effect of the simulation up to 8000 iterations instead of 10000 as the total number of prey species increases at an enormous rate in this simulation within very short period of time. As most of the species are unpalatable, predators mostly do not get the opportunity to consume, causing a much lower prey death rate.

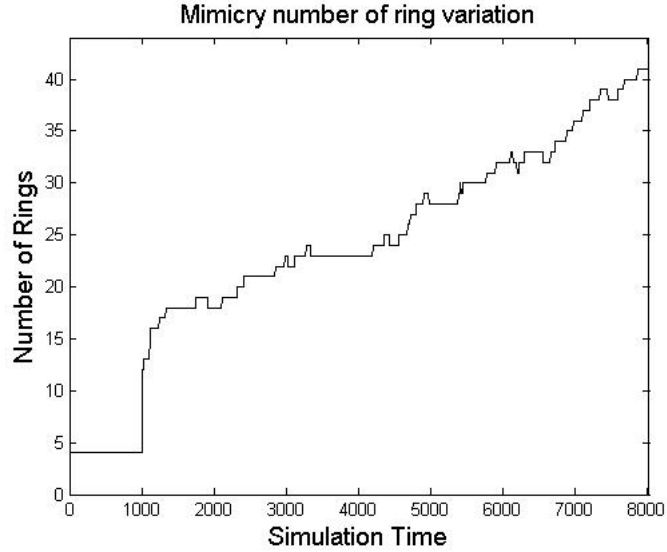


Figure 5.14: Number of mimicry rings, initialized with 4 prey species all unpalatable.

This experiment is an ideal scenario for observing Mullerian mimicry. As mentioned in section 3.3, Mullerian mimicry occurs between multiple species of unpalatable prey population. From the second work of Franks and Noble [FN03] mentioned in section 4.2.1, we note that multiple Mullerian mimicry rings are expected to converge into one large ring through the evolutionary process of punctuated equilibrium. But in this experiment as the predator’s ‘Minimum Memory Configuration’ is set to four, all predators have the capability to recognize four prey patterns before starting to make intelligent decision of consuming them. By setting ‘Minimum Memory Configuration’ to one, also increasing ‘Predator Demise Age’ to 7000 and decreasing predator’s ‘Reproduction Age Interval’ to 1500, we get the following plot.

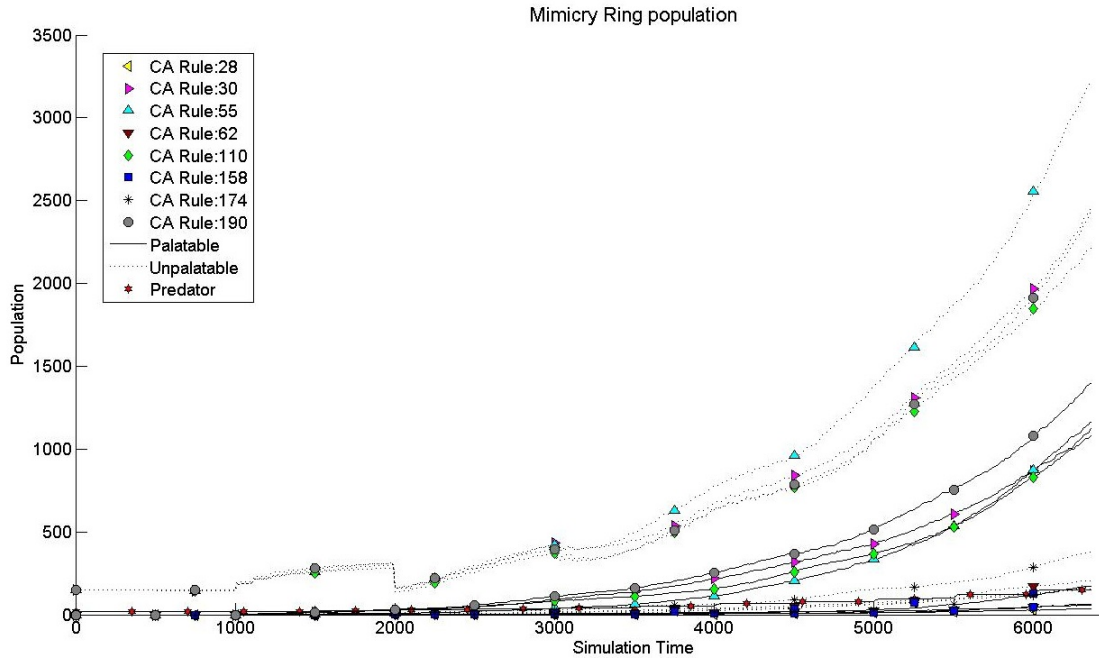


Figure 5.15: Population distribution of mimicry rings, initialized with 4 prey species all unpalatable. Minimum Memory size reduced to one.

Running the simulation for 6000 iterations there was no sign for all prey population to converge into one large ring. As it can be observed in figure 5.15 all four unpalatable prey population have a very dominant presence in the simulation. Even though predator minimum memory has been reduced to only one pattern, different population of predators become familiar with different prey patterns, which results in the existence of multiple Mullerian mimicry ring instead of a single one.

5.8 Initial configuration with only palatable species





	Prey configuration		Predator configuration	
Population	Rule110 (Palatable)		150	20
	Rule30 (Palatable)		150	
	Rule55 (Palatable)		150	
	Rule190 (Palatable)		150	
Reproduction	Age Limit	100		500
	Interval	1000		2000
Mutation Rate	Pattern	0.05		0.3
	Genome	0.5		
Demise Age	2000		5000	
Minimum Attack Age			500	
Memory Configuration			Minimum	4
			Maximum	10

Table 5.8: Agent configuration of 4 prey species all palatable.

The parameters for this simulation has been set exactly the same as in table 5.7 except all the species are palatable at this point. All predator configuration has also been set to exactly same as before.

The results for the configuration in table 5.8 can be observed in figure 5.16 where all population of prey species have reached its demise at nearly 7000 iterations. By analyzing details we can see, around simulation time 500 all population of species come to a steady downfall as at this point initial predator species reach its' age for consumption. Around simulation time 1000 all population of prey species start increasing steadily because they reach their maturity level of reproduction. At iteration time 2000 there is a sudden drop of prey population because the initial population of prey have reached its demise age at this point. By this time the population of unpalatable species mutated from the palatable ones start increasing, and slowly around time 7000 and onwards the entire population of prey come to its demise. Predator population, from the beginning of the simulation has taken a very dominant effect. As all the species are palatable, predator's consumption and

reproduction rate is extremely high and it keep increasing in a geometric rate.

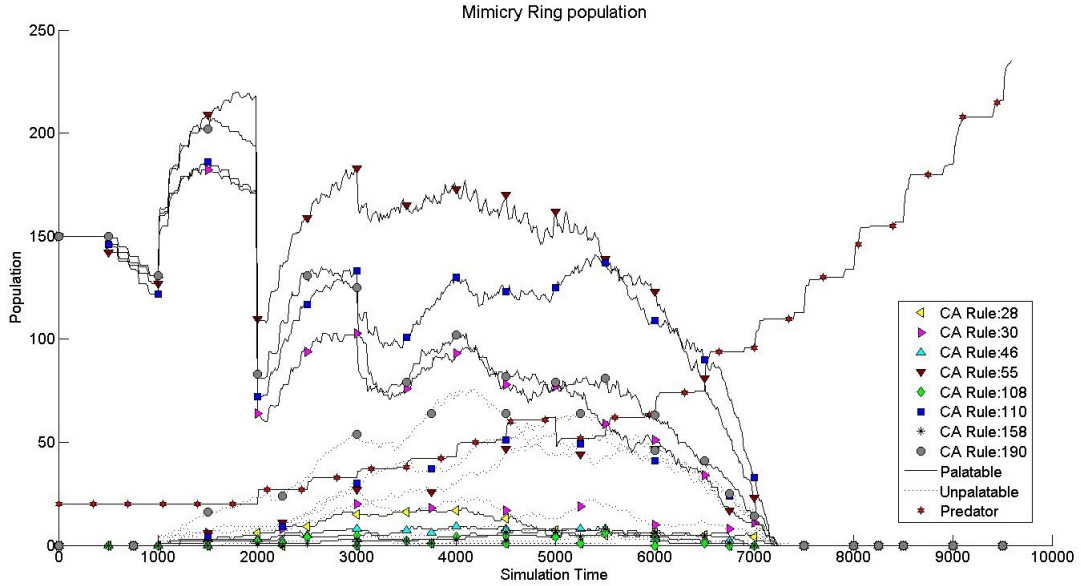


Figure 5.16: Population distribution of mimicry rings, initialized with 4 prey species all palatable.

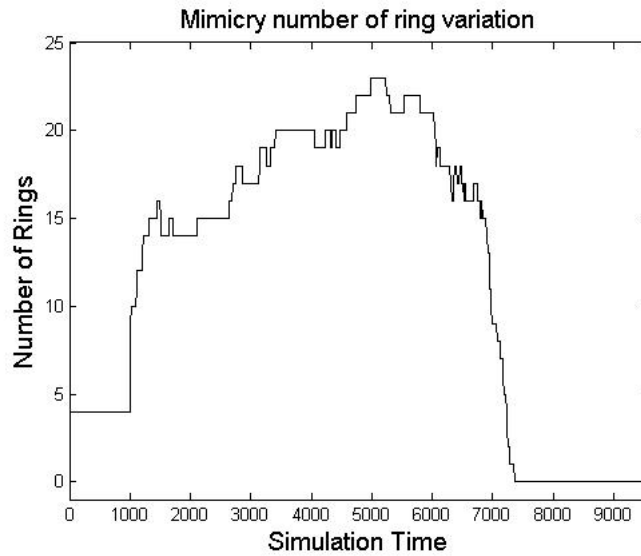


Figure 5.17: Number of mimicry rings, initialized with 4 prey species all palatable.

Observation of mimicry rings also tell us about the results we have seen in the plot of figure 5.16. The number of rings keep increasing but comes to a sudden drop around simulation time 7000 and goes to zero.

5.9 Initial configuration with single prey species

Until now all experiments have been initialized with multiple prey species. The following set of experiments have been initialized with only a single prey species, for both cases of unpalatable and palatable.

5.9.1 Unpalatable


	Prey configuration		Predator configuration	
Population	Rule30 (Unpalatable)		216	10
Reproduction	Age Limit	100		500
	Interval	1000		2500
Mutation Rate	Pattern	0.05		0.3
	Genome	0.5		
Demise Age	2000		7000	
Minimum Attack Age			500	
Memory Configuration			Minimum	2
			Maximum	10

Table 5.9: Agent configuration of 1 prey species unpalatable.

This configuration has only one unpalatable species, one in each of the 216 cells, equally distributed all over the environment. Also have a set of 10 predators. Their minimum memory configuration has been set to 2 as any value below 2 will not help the Hopfield Network Memory to differentiate between two patterns.

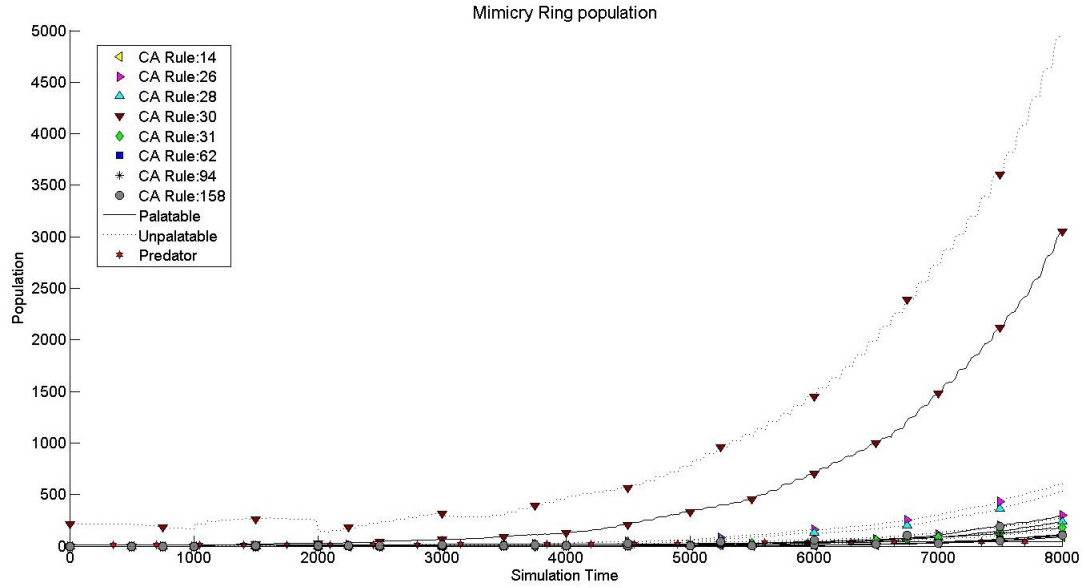


Figure 5.18: Population distribution of mimicry rings, initialized with 1 prey species unpalatable.

From figure 5.18, it can be observed, the initial set of unpalatable species is dominant as expected. The palatable population mimicking that set is following it. A bunch of other rings with majority of unpalatable species have been reproduced through mutation. Mainly because the initial population started with a bunch of unpalatable species, so their reproduced generation have similar behavior as well.

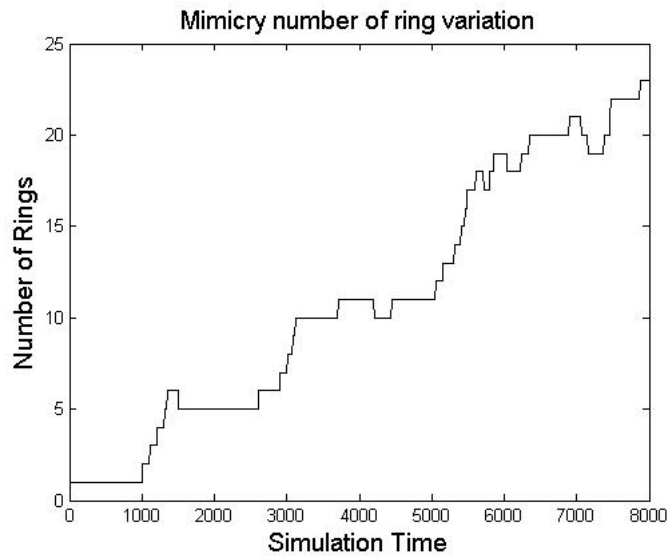


Figure 5.19: Number of mimicry rings, initialized with 1 prey species unpalatable.

5.9.2 Palatable


	Prey configuration		Predator configuration	
Population	Rule30 (Palatable)	 216	8	
Reproduction	Age Limit	100	500	
	Interval	1000	2500	
Mutation Rate	Pattern	0.05	0.3	
	Genome	0.5		
Demise Age	2000		7000	
Minimum Attack Age			500	
Memory Configuration			Minimum	2
			Maximum	10

Table 5.10: Agent configuration of 1 prey species palatable.

This configuration is set exactly as according to section 5.9.1 but with a single set of palatable species. The population of predators have been reduced from 10 to 8 to avoid elimination of all prey species as it happened in case of section 5.8.

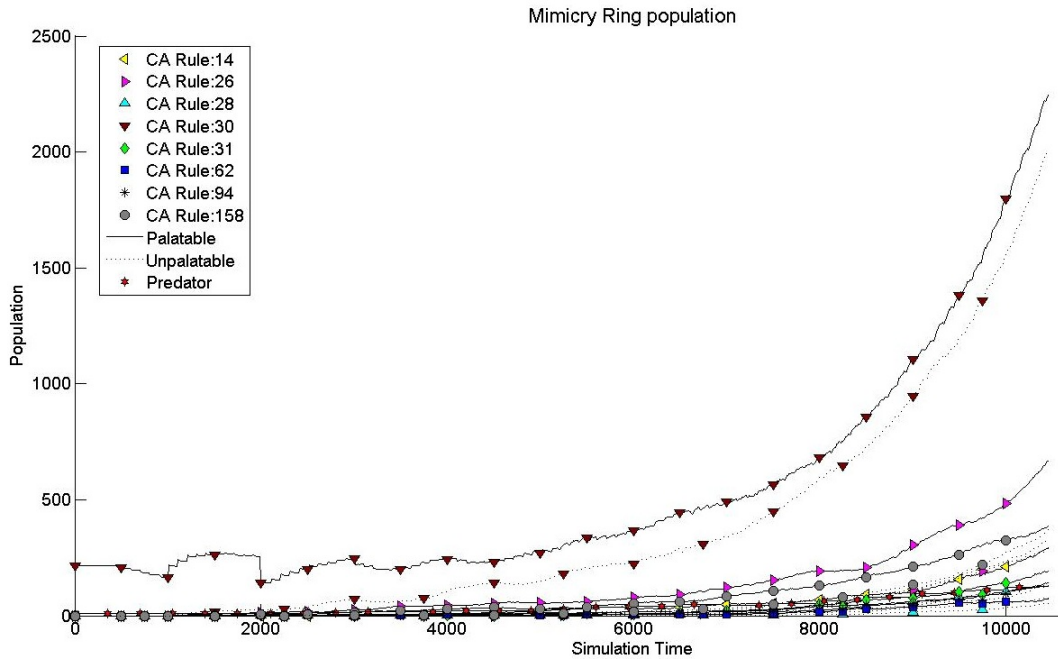


Figure 5.20: Population distribution of mimicry rings, initialized with 1 prey species palatable.

From the initial palatable population we see a bunch of palatable rings created. Major difference from section 5.9.1 is the total number of prey population created. For the unpalatable species in section 5.9.1, total prey population reach above 12 thousand. But for palatable species total population reach nearly above 5000 prey species, as when the preys are palatable there is more consumption from predators.

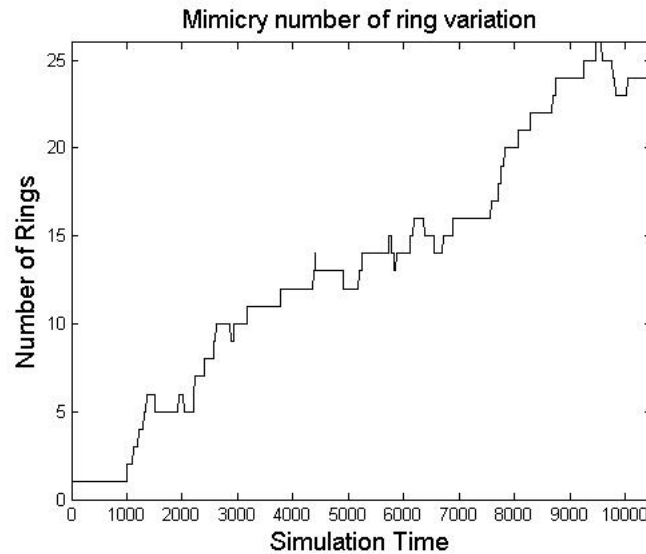


Figure 5.21: Number of mimicry rings, initialized with 1 palatable prey species.

5.10 Analysis of Batesian Mimicry

For all possible initial conditions, Batesian mimicry has taken effect. It can be observed that for every ring of unpalatable species there is an existence of the palatable ring racing to reach the population count of its unpalatable counterpart. Also when the initial population starts with a set of only palatable species we can observe that the total population vanishes within very short period of time (section 5.8). This effect can be explained with the fact that the palatable population does not have any models to mimic, and it reaches extinction. So from this analysis it can be concluded that the model under discussion have successfully simulated the evolution of Batesian mimicry.

5.11 Analysis of Mullerian Mimicry

Effects of Mullerian mimicry can be observed best for the experiment in section 5.7. For that case, we initialized the model with 4 rings of unpalatable species with no palatable ones and after nearly 10K iterations, all of the initial unpalatable rings have survived with dominance. The cause of this behavior can be explained by the minimum number of patterns that each predator can store in memory, which was set to four. So this parameter was reduced to one to observe whether it is possible to converge all different unpalatable rings into one large ring, when predators are capable to memorizing only a single pattern. But as it turned out, the phenomena of “a single large ring” does not occur because different predators recognize different patterns resulting in multiple divergent Mullerian mimicry rings. It can be concluded that our results are consistent with those of Franks and Noble (section 4.2.1), that multiple Mullerian mimics do not converge into one large ring.

5.12 Conclusion

Analysis of the results tell us that we have successfully been able to simulate the evolution of mimicry. In addition to that, this model provides a more accurate simulation of the fascinating natural process of mimicry rings. Not only does this model simulates mimicry with the initialized population but also it provides possibility of creating diverse new rings and their shift in population. This model also verifies the theory of Turner in explaining the evolution of mimicry with punctuated equilibrium (section 4.4.4) [Tur88].

Chapter 6

Future Work: Application

The goal of this thesis is to investigate whether the evolution of mimicry can be used in a problem solving scenario similar to evolutionary programming or genetic algorithm as it is used to optimize solutions using mathematical functions. Predator-prey co-evolution has also been used for solving interesting problems [Hil90], giving better result than conventional methodologies.

The biggest challenge faced while searching for the appropriate problem to be optimized is the idea of associating palatability with CA pattern with which the prey species has been represented. If we consider the CA pattern to be a certain solution to a problem then the set of 2D CA pattern among which the prey species vary could be considered as the solution set of the problem. And if we consider that the predator species are responsible for selecting the appropriate pattern based on their palatability, then we can consider palatability as the criteria for selection of the pattern (or the evaluation/fitness function for the case of Evolutionary Programming/Genetic Algorithm). If the predator associates the CA pattern with palatability then it does not make any sense of having a mimic in the solution space as mimics will have the same pattern with opposite palatability. So using the evolution of mimicry to solve an optimization problem is futile if we try conventional problems which already can be solved with the help of evolutionary programming or genetic algorithm.

So the kind of problems we should be looking for applying the evolution of mimicry, are the ones which cannot be solved with conventional evolutionary methods. We should be looking for a problem where the idea of deception can be useful in term of solving it. An interesting analogy is with anti-virus and virus software. If we consider anti-virus to

be predator and virus (mimic) to be prey species, then the objective of a virus would be to look like a regular virus free program (model) to get pass an anti-virus software. In this way by mimicking a regular program a virus can get pass any anti-virus software available in the market. To successfully implement this software we need to create a virus program which can evolve itself to pretend like a regular program (virus free). The environment will be the operating system itself where all the programs reside. It is worth noting the fact that this objective is malicious for the computer industry, which makes it less significant to pursue.

Chapter 7

Conclusion

If we consider the field of Artificial Life as a tool for biological inquiry then the model presented in this thesis was successful in terms of providing good insight in terms of understanding better the evolutionary process of mimicry as it happens in nature. As mentioned in section 5.12 this model gives us verification of the theory of Turner in explaining the evolution of mimicry with punctuated equilibrium. Also it enforces findings from the work of Franks and Noble that multiple Mullerian mimics do not converge into one large ring.

If we consider Artificial Life as the study where we simulate natural processes to further extend capabilities in the field of computer science, then also this thesis is a success in making an appropriate emulation of the evolutionary process of mimicry. From the sections of 5.10 and 5.11 we can conclude that the complex behavior of Batesian and Mullerian mimicry can be simulated with this model.

But considering Artificial Life as a subject where we essentially learn from nature, its behaviors, and use that knowledge to find better solution to existing problems in the field of computer science then this thesis has not been completely successful. We have not been able to find the appropriate problem solving scenario for which the natural process of mimicry can be applied.

Humans came to understand the concept of biological evolution and its significance when it was first proposed by Charles Darwin in 1859 in his book *On the Origin of Species* [Dar59]. Nearly 200 years later Alan Turing was the first who thought over creating machines which are capable of evolving itself (Section 2.2.1). Eventually in 1975 John Holland was able to make the most applicable use of the process of evolution to solve problems in computer

science by his invention of Genetic Algorithms. The time for this invention was appropriate as the field of computer science was in its burgeoning state. Over the next couple of decades computers became one of most useful and important machinery in human history. Genetic Algorithms became popular as well because of increased capability of computers in term of speed and complexity.

Even though we do not have any problem solving application for the evolutionary process of mimicry at this point, it does not imply that this study will not be useful in the future to give us solution to an important problem in computer science or any other field for that matter. The simulation presented in this thesis, itself, consists of building blocks of knowledge such as Hopfield Network and Cellular Automata. Maybe to use mimicry to solve a problem, other diverse building blocks are necessary which has not yet been invented.

The objective of any research is to look for *unexplored* territory to find new *knowledge*, *understanding* or *explanation*. From the research presented in this thesis we certainly get new *understanding* of the capability of computer science to emulate natures behavior. We also get to *explore* a new model for the evolutionary process of mimicry. In addition to that, the model provides us with better *explanation* and *understanding* of nature, while enforcing some existing *knowledge* with greater support (section 5.11).

References

- [And89] Philip W. Anderson. *The Economy As an Evolving Complex System (Santa Fe Institute Studies in the Sciences of Complexity Proceedings)*. Addison Wesley Publishing Company, January 1989.
- [Axe84] Robert M. Axelrod. *The evolution of cooperation*. Basic Books, New York, 1984.
- [Bak87] James E. Baker. Reducing bias and inefficiency in the selection algorithm. In *Proceedings of the Second International Conference on Genetic Algorithms on Genetic algorithms and their application*, pages 14–21, Hillsdale, NJ, USA, 1987. L. Erlbaum Associates Inc.
- [Bat62] Henry Walter Bates. Contributions to an bisect fauna of the amazon valley. *Transactions of the Linnean Society of London*, 23:495–566, 1862.
- [Bre62] H. J. Bremermann. Optimization through evolution and recombination. *Self-Organizing Systems*, pages 93–106, 1962.
- [Bro88] L. P. Brower. *Mimicry and the Evolutionary Process*. University of Chicago Press, Chicago, 1988.
- [Cot57] H. B. Cott. *Adaptive Coloration in Animals*. Methuen, London, 1957.
- [Dar59] Charles Darwin. *On the Origin of Species*,. New York, Literary Classics, 1859. <http://www.biodiversitylibrary.org/bibliography/46292>.
- [Daw90] Richard Dawkins. *The Selfish Gene*. Oxford University Press, September 1990.
- [Daw96] Richard Dawkins. *The Blind Watchmaker: Why the Evidence of Evolution Reveals a Universe without Design*. W.W. Norton & Co., September 1996.

- [EG72] N. Eldredge and S. J. Gould. Punctuated equilibria: an alternative to phyletic gradualism. *Models In Paleobiology*, pages 82–115, 1972.
- [ES03] A. E. Eiben and J. E. Smith. *Introduction to Evolutionary Computing*. Springer, New York, 2003.
- [FN02] Daniel W. Franks and Jason Noble. Conditions for the evolution of mimicry. In *ICSAB: Proceedings of the seventh international conference on simulation of adaptive behavior on From animals to animats*, pages 353–354, Cambridge, MA, USA, 2002. MIT Press.
- [FN03] Daniel W. Franks and Jason Noble. The origins of mimicry rings. In *ICAL 2003: Proceedings of the eighth international conference on Artificial life*, pages 186–191, Cambridge, MA, USA, 2003. MIT Press.
- [FN04] Daniel W. Franks and Jason Noble. Batesian mimics influence mimicry ring evolution. In *Proceedings of the Royal Society B: Biological Sciences*, pages 191–196, 2004.
- [For64] E. B. Ford. *Ecological genetics*. Methuen, London, 1964.
- [FOW66] L. J. Fogel, A. J. Owens, and M. J. Walsh. *Artificial Intelligence Through Simulated Evolution*. John Wiley & Sons, Inc., New York, 1966.
- [GCS⁺03] Peter Grogono, GuoRong Chen, JunFeng Song, Tao Yang, and Lei Zhao. Laws and life. In *ASC 2003: Proceedings of the 7th IASTED Conference on Artificial Intelligence and Soft Computing*, pages 158–163. International Association of Science and Technology for Development, 2003.
- [GE77] Stephen Jay Gould and Niles Eldredge. Punctuated equilibria: The tempo and mode of evolution reconsidered. *Paleobiology*, 3(2):pp. 115–151, 1977.
- [GL85] D. E. Goldberg and R. Lingle. Alleles, loci, and the traveling salesman problem. In *Proceedings of the 1st international conference on Genetic algorithms and Their applications*, pages 154–159, Hillsdale, New Jersey, 1985. Lawrence Erlbaum.
- [Heb49] D.O. Hebb. *The Organization of Behavior*. Wiley, New York, 1949.

- [Hil90] Daniel W. Hillis. Co-evolving parasites improve simulated evolution as an optimization procedure. *Phys. D*, 42(1-3):228–234, 1990.
- [HJF97] Peter T. Hraber, Terry Jones, and Stephanie Forrest. The ecology of echo. *Artificial Life*, 3:165–190, 1997.
- [Hol92] John H. Holland. *Adaptation in natural and artificial systems*. MIT Press, Cambridge, MA, USA, 1992.
- [Hol96] John H. Holland. *Hidden order: how adaptation builds complexity*. Basic Books, 1996.
- [Hop82] J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the United States of America*, 79(8):2554–2558, April 1982.
- [Huh88] James E. Huheey. Supplement: Mimicry and the evolutionary process. *The American Naturalist*, 131:S22–S41, 1988.
- [Koz92] John R. Koza. *Genetic programming: on the programming of computers by means of natural selection*. MIT Press, 1992.
- [MMS89] Ramon Marimon, Ellen R. McGrattan, and Thomas Sargent. Money as a medium of exchange in an economy with artificially intelligent agents. Working papers, Stanford - Hoover Institution, 1989.
- [Mul79] F. Muller. Ituna and thyridia; a remarkable case of mimicry in butterflies. *Trans. Entomol. Soc. Lond.*, pages xx–xxix, 1879.
- [Nic27] A. J. Nicholson. *A new theory of mimicry in insects*. Royal Zoological Society of New South Wales, 1927.
- [Per99] Alan S. Perelson. *Two theoretical problems in immunology: AIDS and epitopes*, pages 185–203. Perseus Books, Cambridge, MA, USA, 1999.
- [Pou90] E. B. Poulton. *The Colours of Animals, Their Meaning and Use, Especially Considered in the Case of Insects*. D. Appleton and Company, New York, 1890.

- [Pou12] E. B. Poulton. *Darwin and Bergson on the interpretation of evolution*. Bedrock, 1912.
- [RB91] David B. Ritland and Lincoln P. Brower. The viceroy butterfly is not a batesian mimic. *Nature*, 350:497–498, 1991.
- [Rec73] Ingo Rechenberg. *Evolutionsstrategie Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Fromman-Holzboog, Verlag, Stuttgart, 1973.
- [She72] P. M. Sheppard. Some aspects of the geography, genetics, and taxonomy of a butterfly. *Taxonomy and geography*, pages 135–152, 1972.
- [She02] Thomas N Sherratt. The coevolution of warning signals. In *Proceedings of The Royal Society on Biological Sciences*, pages 741–746, 2002.
- [Syw89] Gilbert Sywerda. Uniform crossover in genetic algorithms. In *Proceedings of the third international conference on Genetic algorithms*, pages 2–9, San Francisco, CA, USA, 1989. Morgan Kaufmann Publishers Inc.
- [TJ93] Charles Taylor and David Jefferson. Artificial life as a tool for biological inquiry. *Artif. Life*, 1:1–13, January 1993.
- [TKE84] John R. G. Turner, E.P. Kearney, and L. S. Exton. Mimicry and the monte carlo predator: the palatability spectrum and the origins of mimicry. *Biological journal of the Linnean Society*, 23:247–268, 1984.
- [TS96] John R. G. Turner and Michael P. Speed. Learning and memory in mimicry. i. simulations of laboratory experiments. *Philosophical Transactions: Biological Sciences*, 351(1344):1157–1170, 1996.
- [Tur48] A. M. Turing. Intelligent machinery. *Machine Intelligence* 7, 1948.
- [Tur88] John R. G. Turner. The evolution of mimicry: A solution to the problem of punctuated equilibrium. *The American Naturalist*, 131:S42–S66, 1988.
- [VN66] John Von Neumann. *Theory of Self-Reproducing Automata*. University of Illinois Press, Champaign, IL, USA, 1966.

[Wic68] Wolfgang Wickler. *Mimicry in plants and animals*. McGraw-Hill Book Company, Newyork Toronto, 1968.

Acronyms

AL Artificial Life. 10

CA Cellular Automata. 26

CAS Complex Adaptive System. 10

EA Evolutionary Algorithm. 5

EC Evolutionary Computation. 5

EP Evolutionary Programming. 5

ES Evolutionary Strategies. 5

FormAL Formal Artificial Life. 30

GA Genetic Algorithm. 5

GP Genetic Programming. 5

RNA Ribonucleic Acid. 10