ON MUTATION AND CROSSOVER IN THE THEORY OF EVOLUTIONARY ALGORITHMS

by

James Neal Richter

A dissertation submitted in partial fulfillment
of the requirements for the degree

of

Doctor of Philosophy

in

Computer Science

MONTANA STATE UNIVERSITY
Bozeman, Montana

April, 2010

APPROVAL

of a dissertation submitted by

James Neal Richter

This dissertation has been read by each member of the dissertation committee and has been found to be satisfactory regarding content, English usage, format, citations, bibliographic style, and consistency, and is ready for submission to the Division of Graduate Education.

Dr. John T. Paxton

Approved for the Department of Computer Science

Dr. John T. Paxton

Approved for the Division of Graduate Education

Dr. Carl A. Fox

STATEMENT OF PERMISSION TO USE

James Neal Richter

April, 2010

DEDICATION

This work is dedicated to my father. I miss you every day Dad.

James Franklin Richter (February 24, 1957 - January 14, 2006).

I also dedicate this work to my mothers Maureen, Francile and Geneal, my lovely wife Stephanie and our four children, James Zachary, Kathleen, Brooklyn and Marcus.

ACKNOWLEDGEMENTS

Dr. John Paxton has been my academic advisor since 1994 when I began undergrad studies in Computer Science. His encouragement and advice over the years have been invaluable.

Dr. Alden Wright of the University of Montana acted as the external co-chair of the dissertation committee. His mentorship, expertise and collaboration was instrumental in following this research path.

Since I began PhD studies I've gotten married, lost a father, gained four children, had three different jobs, helped reboot and sell a software company then joined the buyer's software company to tackle bigger challenges. As a result this work was often left idle in drawer. I would like to thank John and Alden for their encouragement and patience through all that.

Thomas Jansen, Michael Vose, Jon Rowe, Richard Watson, Lothar Schmitt, and the late Ingo Wegener are also acknowledged for their encouragement and advice. I would also like to express gratitude to the participants and organizers of the Theory of Evolutionary Algorithms conference series at Schloss Dagstuhl International Conference and Research Center for Computer Science, where the seeds of this research were germinated.

The Departments of Computer Science and Mathematics have my gratitude for the education and support provided. Thanks also go out to my industry co-workers and supervisors at RightNow, Others Online and the Rubicon Project. Thanks also to Michael Guerini, who helped edit and beat this text into better shape.

TABLE OF CONTENTS

TABLE OF CONTENTS – CONTINUED

TABLE OF CONTENTS – CONTINUED

TABLE OF CONTENTS – CONTINUED

TABLE OF CONTENTS – CONTINUED

LIST OF TABLES

Table Page

LIST OF FIGURES

LIST OF FIGURES – CONTINUED

LIST OF FIGURES – CONTINUED

Figure Page

LIST OF ALGORITHMS

ABSTRACT

The Evolutionary Algorithm is a population-based metaheuristic optimization algorithm. The EA employs mutation, crossover and selection operators inspired by biological evolution. It is commonly applied to find exact or approximate solutions to combinatorial search and optimization problems.

This dissertation describes a series of theoretical and experimental studies on a variety of evolutionary algorithms and models of those algorithms. The effects of the crossover and mutation operators are analyzed. Multiple examples of deceptive fitness functions are given where the crossover operator is shown or proven to be detrimental to the speedy optimization of a function. While other research monographs have shown the benefits of crossover on various fitness functions, this is one of the few (or only) doing the inverse.

A background literature review is given of both population genetics and evolutionary computation with a focus on results and opinions on the relative merits of crossover and mutation. Next, a family of new fitness functions is introduced and proven to be difficult for crossover to optimize. This is followed by the construction and evaluation of executable theoretical models of EAs in order to explore the effects of parameterized mutation and crossover. These models link the EA to the Metropolis-Hastings algorithm. Dynamical systems analysis is performed on models of EAs to explore their attributes and fixed points. Additional crossover deceptive functions are shown and analyzed to examine the movement of fixed points under changing parameters. Finally, a set of online adaptive parameter experiments with common fitness functions is presented.

CHAPTER 1
INTRODUCTION

This dissertation addresses fundamental attributes of the crossover and mutation operators of the evolutionary algorithm (EA). The evolutionary algorithm is an umbrella term for a set of biologically inspired optimization algorithms including the well-known genetic algorithm. The EA is frequently applied to problems where special purpose algorithms do not exist or are hard to implement. Since the crossover-enabled members of the EA family were first proposed, it has been postulated that the crossover operator is the strongest operator as it gives the EA the ability to recombine members of the population.

Without crossover, it was thought that the EA would function like a parallel hill-climbing algorithm and as such would be unable to optimize certain kinds of functions. For many years this proved true in part. Various researchers showed families of functions designed to exploit the crossover operator. Attempts to show crossover deceptive functions were limited and often did not result in a substantial difference in optimization performance.

However, when the No Free Lunch (NFL) theorems [1] were introduced, they conclusively showed that no general statement of the superiority of crossover-enabled optimization algorithms was possible. As a side effect, the NFL implies that it is possible to identify functions where either operator is detrimental to optimization in less than exponential time. Can examples be found showing rigorously provable substantive performance penalties when using crossover?

Additionally, the casting of the EA into abstract mathematical models opened up a new avenue of research. Once a dynamic systems approach is used to model the EA, many questions about the meta-properties of this system arise. The starting point of such analysis is often to find the stationary points (or fixed points) of the system and examine how those points behave under changing algorithm parameters. Can such an analysis show interesting properties of the objective function that govern the dynamics of the EA?

This work seeks to answer some of the above postulated questions. Explicit families of functions are designed to accentuate differences between the two operators and a range of analyses are conducted to explore the basic attributes of mutation versus crossover on carefully constructed functions. The intent of this exploration is to demonstrate crossover-deceptive functions.

1.1 Document Organization

This document is organized into the following sections:

- Introduction to EAs,
- Historical review of evolution and models of evolution,
- Review of theoretical models of EAs and associated results,
- A family of functions proven to be difficult for GAs with crossover to optimize,
- An exploration of the stationary points of mutation-only EAs,
- A study of bistability in GAs
- A study of stationary points of the GA
- Experimental Studies of EAs,
- Conclusions and future ideas.

Framework	Mutation-only EAs	Crossover-enabled GAs
Time Complexity Analysis	Chapters 5,6	Chapters 5,6
Dynamical Systems Analysis	Chapters 7,8,10	Chapters 9,10
Empirical Analysis	Chapters 9,11	Chapters 9,11

Figure 1.1: Dissertation Framework

1.2 Framework and Contributions

Each of the above chapters, excluding the literature reviews, addresses a sub-field of EA theory and adds new knowledge to that topic. The contributions are illustrated in Figure 1.1 within a framework and repeated at the top of each chapter. This work uses three main families of analysis techniques, time complexity analysis, dynamical systems analysis and empirical analysis. The methodology and variables of study are listed below.

Analysis Methodology:

- *Time complexity analysis:* Mutation-only Evolutionary Algorithms and Crossover-enabled Genetic Algorithms are analyzed for for small and large populations,
- *Dynamical systems analysis:* Mutation-only EAs and Crossover-enabled GAs are analyzed with an infinite population model,
- *Empirical analysis:* EAs and GAs are studied via experiments with running algorithms.

Variables of Study:

- Deception in crossover-based GAs,
- Stationary points of EA and GA models under varying parameters,
- Effects of adaptive parameter schemes.

CHAPTER 2

INTRODUCTION TO EVOLUTIONARY ALGORITHMS

> The Darwinian process of continued interplay of a random and a selective process is not intermediate between pure chance and pure determinism, but in its consequences qualitatively utterly different from either.
>
> Sewall Wright [2]

2.1 Evolutionary Algorithms

The term Evolutionary Algorithm (EA) encompasses a family of algorithms that mimic the Darwinian evolutionary process, mainly as a method of heuristic search for optimization problems. The essential commonalities of the algorithms and their linkage to neo-Darwinian evolution are shown in Table 2.1.

As it is currently known, the field is sub-divided into four major families shown in Table 2.2. The exact details in each family differ in important ways and are not equivalent in general. Each member of the EA family has a wide number of variants described in the literature and as such, the below generalization should be accepted

Table 2.1: Comparison of neo-Darwinian and Evolutionary Computation.

neo-Darwinian Evolution		Evolutionary Computation
Organism	↔	Candidate Solution
Population of organisms	↔	Population of candidates.
Environment	↔	Problem
Environmental fitness	↔	Solution quality
Survival of the fittest	↔	Differential selection
Genetic inheritance	↔	Genetic inheritance

Table 2.2: The Family of Evolutionary Algorithms.

Evolutionary Strategies	Introduced by Ingo Rechenberg in the 1960's, designed to optimize real-valued systems.
Genetic Algorithms	Introduced by John Holland in the 1960's designed to optimize systems with a bit-string and real-valued representations.
Evolutionary Programming	Introduced by Lawrence Fogel in the 1960's designed to evolve finite state machines.
Genetic Programming	Introduced by John Koza in the 1990's, designed to evolve the parse-trees of functional computer languages such as LISP.

with care.

Even this compartmentalization is not terribly strict in practice, particularly between the GA and ES as there are binary representations of the latter and real-valued genetics for the former. See De Jong 2002 for a unified background on EAs [3]. A major algorithm not fitting into this division is the *estimation of distribution* (EDA) of Pelikan and co-workers [4]. A core novelty of the EDA is that it dispenses with directly utilizing a population of candidate solutions and uses a probability distribution instead, sampling from it then subjecting it to manipulation via the evolutionary process.

A alternate division of the families may be segregated into those that either utilize an executable or static representation. Genetic Programming (GP) and its predecessor Evolutionary Programming, generally evolve a genome of computer code evaluated for fitness by execution. This representation is by its nature variable in length. The GA and ES traditionally use a fixed length binary or numeric representation that must be evaluated by an external function.

There exist many specialized sub-fields with EA variants of the above families. They are applied to specific problems such as evolable computer [5] hardware or less related general explorations of simulated or artificial life [6]. While these areas and the historical development of Genetic Programming and Evolutionary Programming are rich and fascinating, the focus of this work will be a review of the origin and development of theory involving genetically interpreted bit strings. See [7, 8, 9] for details on Genetic Programming and [10, 11] for Evolutionary Programming introductions.

There also exist many related algorithms that don't fit into the divisions above yet are grouped into the EA family. Dorigo's *ant colony optimization* (ACO) technique mimics the foraging and pheromone activity of insects [12]. While ACO uses a population, that population acts upon a shared solution rather than each member containing a population. A closely related technique called *partical swarm optimization* (PSO) [13] wherein a population of solution particles with best-so-far solution memory move about the solution space and transmit information about solutions to neighboring particles. *Artificial immune systems* (AIS) are another group of algorithms sometimes grouped with EAs. The AIS algorithms attempt to mimic attributes observed in biological immune systems to solve problems [14].

The EA also belongs within the wider family of *stochastic local search* (SLS) algorithms. They are commonly applied to both *decision problems* and *optimization problems*. See the excellent text by Hoos and Stützle [15] for more detail on SLS.

2.1.1 The Simple Genetic Algorithm

The canonical version of the GA operates on a binary bit string with three operators - selection, mutation and crossover. A fitness (objective) function is supplied that maps the binary representation to a real number. The goal of the GA is to optimize the fitness function by exploring the binary search space such that the function is either minimized or maximized.

$$f(x) : \{0, 1\}^n \to \mathbb{R}+$$

These types of fitness functions are referred to as pseudo-Boolean functions and are an area of rich study [16]. The GA is frequently applied in practice to problems that are poorly understood, have many inter-dependencies, or are otherwise too complex to optimize directly. Once an encoding of the problem into a pseudo-Boolean function is accomplished, the GA can be applied to generate-and-test candidate solutions via a simulation of the evolutionary process.

Algorithm 2.1: The Genetic Algorithm

```
Create a randomly generated population of $m$ binary strings of length $n$,
Evaluate the fitness of each individual string via fitness function $g(x)$,
Select pairs of parent solutions from the population,
For each pair repeat:
    With probability $p_c$ apply the crossover operator to the parents to produce a child string,
    With probability $p_m$ independently mutate each bit of the child string.
    Insert the child string into the new population,
Replace the old with the new population,
Loop to Evaluation step above.
```

A minor difference between the above algorithm and the one introduced by Holland [17] is that his original one-point crossover operator produced two children and both were added to the population. The formulation has been simplified to single-child for better analysis and usage of more general crossover operators. The mutation and crossover operators are almost always stochastic operators, and the selection operator has many forms, some of them are also stochastic. The exact operational mechanics of the algorithm vary in implementation and effect. Details on variants are deferred until later for specific analysis.

2.2 Pre-History of Evolutionary Algorithms

The story of Evolutionary Algorithms is one of parallel and independent discoveries. The synthesis of Darwinism and computational science was fragmented in its early development. The fragmentation was compounded by slow communications and delayed recognition of ideas.

Perhaps the first to propose computer simulated evolution was Alan Turing. In 1948, he proposed a "genetical or evolutionary search by which a combination of genes is looked for, the criterion being the survival value."[18] Turing is not known to have detailed how such an algorithm would work specifically. In later papers he hinted at possible ideas taken liberally from evolutionary theory [19].

While population geneticists were modeling evolution decades before, the earliest effort to simulate genetic evolution via binary strings dates back to at least 1957 and the work of A.S. Frasier of Australia. A binary string of length n represented alleles, with a specified phenotype mapping based upon the allele values. A given binary string population of size P is established and pairs of strings are selected for recombination with a generalized n-point recombination algorithm where each loci on the string would have a probability of being a break point. A new population of size P' is produced in each generation that may initially be larger that the parent population and the top P fitness individuals then completely replace the parent generation [20, 21]. Fraser and coworkers continued to publish papers including such concepts as gene linkage and epistasis based on many computational experiments. However, they generally made no attempts to develop formal theory about their explorations. While Fraser and Burnell eventually published the second 'prehistoric' book in the EA field [22], their contributions would not be discovered until later [23].

In 1958, Hans-Joachim Bremermann [24, 25] (seemingly independently) created a very similar model of binary string evolution complete with sexual and asexual reproduction, mutation and selection fitness schemes. Bremermann also developed theorems showing that for monotonic functions where fitness is based on the number of bits in the string, the optimal mutation rate is near $1/n$ and specifically $1–(m/n)^{1/(n–m)}$. These results would be rediscovered by Heinz Mühlenbein [26] three decades later. Bremermann would go on to write eight papers by 1973 in this topic area. He described multi-parent crossover, scaled mutation, linear and nonlinear function optimization. [23, 27]. In the view of Fogel and Anderson [28] there was little to distinguish Holland's late 1960's GA from Bremermann's algorithms as of 1962.

G.J. Friedman [29], G.E.P. Box [30], and W.W. Bledsoe [27] also explored digital simulations of evolution over binary strings in the late 1950s and 1960s. Lawrence Fogel was an early figure in EAs. He invented Evolutionary Programming [31], a technique for evolving finite state machines and finding optimal parameters of linear equations.

Two other individuals stand out as both inventors of early evolutionary computation techniques as well as developers of strong theoretical results. Ingo Rechenberg and Hans-Paul Schwefel created the Evolutionary Strategies algorithm for evolving real-numbered representations of optimization problems [32, 33]. While not initially using binary representations or recombination, their work eventually utilized both. These two researchers also established long running research programs in Germany to explore the fundamental attributes of how evolutionary algorithms function.

The reader should consult David Fogel's labor-of-love reconstruction of this "pre-history" in his book *Evolutionary Computation: The Fossil Record* [34]. Ironically, the fundamental attribute of the early development of the family of EAs was their repeated independent origination, development and eventual cross-pollination once the different groups discovered one another.

2.2.1 Birth of Genetic Algorithms

While clearly the ground work had been laid by evolutionary science for over a century, John Holland's 1975 publication of *Adaptation in Natural and Artificial Systems* [35] has become a pivotal event. It was Holland who coined the term Genetic Algorithm that has become synonymous with evolutionary computation. Holland defined his algorithm in terms of chromosomes, genes and alleles thus following the neo-Darwinian orthodoxy closely. At the time, Holland's focus on the importance of the recombination of genes via crossover was considered novel. This differentiated the GA from other random heuristic search techniques such as the Metropolis Algorithm [36], randomized hill-climbers and various heuristic search methods of the operations research community [37, 38].

Ken De Jong, a student of Holland's, published his 1975 dissertation concentrating on the usage of the GA on optimization problems. He introduced what is still used as an introductory test-suite of functions for evaluating the GA [39]. Holland had many other PhD students that helped push GAs into wider use and interest. John Koza, David Goldberg, Stephanie Forrest, Melanie Mitchell and Annie Wu are a few of the more well published of Holland's PhD students in the GA literature. Their contributions defy brief summaries and are detailed in later chapters.

The focus on recombination has become a defining characteristic of GAs. Unfortunately, early success bred an over-confidence in the optimization skill of GAs. Holland himself made grand claims stating that the GA "is all but immune to some of the difficulties of false peaks, discontinuities, high dimensionality, etc - that commonly attend complex problems"[35]. This tendency to be a bit hyperbolic about the GA's prowess is not all that rare, one only has to examine GA literature from the 90s to see volumes of work describing this idea [40]. Interestingly, this over-confidence afflicted descriptions of evolutionary theory itself at the time, more on this later.

2.2.2 Synthesis of Evolutionary Computation

Beginning in the late 1980s and early 1990s researchers began meeting occasionally at conferences to present research and share ideas. The biannual conferences *International Conference on Genetic Algorithms (ICGA)* [41] and *Parallel Problem Solving from Nature (PPSN)* [42] were formed in 1985 and 1990 respectively. The annual *IEEE Conference on Evolutionary Computation (ICEC)* was formed in 1994. The ICGA and ICEC conferences merged with GP and EP conferences in the late 1999s, forming the annual GECCO [43] and CEC [44] conferences.

At the initial PPSN conference the term *Evolutionary Computation* was proposed to encompass all dialects of EA research. Additionally the term *Evolutionary Algorithm* was agreed upon as the generic form of the family of algorithms. An example formulation is shown below. Major EAs that have not been hybridized with other algorithms can fit within this definition.

Evolutionary Algorithm

```
1. INITIALIZE population with random candidate solutions.
2. EVALUATE each candidate.
3. REPEAT UNTIL TERMINATION CONDITION:
4.     SELECT parent candidates. (Reproductive selection)
5.     RECOMBINE AND/OR MUTATE parent candidates into child candidates.
```

6. *EVALUATE* child candidates.
7. *SELECT* candidates for next generation. *(Survival selection)*

2.2.3 Population Size

A crucial initial choice for a GA practitioner is population size. The basic thinking explains that a population should be chosen in relation to the string length of the genome. Too big of a population leads to a waste of computation time with many undifferentiated function evaluations. A too small population size can result in inadequate coverage of the search space.

Goldberg and co-authors were among the first to tackle analysis of this question in a series of papers [45, 46, 47, 48]. Initially, their conclusions were that population size should increase as an exponential function of the chromosome string length. Later results showed a linear dependence as satisfactory, though all conclusions are dependent on fitness functions of a particular type. Early empirical work from Grefenstette [49] and Alander [50] both showed a linear relationship as sufficient.

Reeves [38] gives an interesting result for the minimum necessary population size based upon the simple idea that at least one instance of each possible allele value (zero and one bits) in the population for all loci. This would ensure sufficient initial diversity such that the recombination operator can reach any point in the search space without depending on mutation to discover a missing configuration. Reeves derives an $O(\log n)$ formula for minimum population size to meet this criteria. Cvetkovic and Mühlenbein [51] estimate a similar $O(\sqrt{n}\log n)$ recommendation based upon empirical work on a monotonic fitness function. Recent work by Gao [52] details an analytical result of a linear lower bound on population size.

Suffice it to say that a large population $p >> n$ is generally not required, though given the GAs propensity for premature convergence, especially in multi-modal fitness functions, an argument based upon getting sufficient allele frequencies in the initial population is somewhat suspect in general. Maintaining the allele diversity via a 'large enough' population size may be a more intuitive argument, though both ideas assume that diversity is desirable to have and maintain. This assumption is widely held [40, 3].

The next three sections are brief summaries of the major operators of evolution as seen by the EA community. The reader should examine introductory books for more discussion and examples [11, 53, 54].

2.2.4 The Selection Operators

The EA has two distinct classes of selection operators, selection for reproduction and selection for survival. It is common for the context of the discussion to dictate which operator is being addressed. Both operators have a basis in the evolutionary theories of biology.

Survival selection is the mechanism of choosing the next generation from populations of parent and child individuals. The classic generational method has the child population completely replacing the parent population. De Jong terms this survival disparity as the *generation gap* [39]. In the elitist GA, a certain number or percentage of the best individuals are allowed to transfer without modification and displace the worst individuals of child population. The steady-state GA fully merges the parent and child population and selects the best m as the next generation.

One can easily see that this establishes a continuum of parental survival strategies. The generational GA has zero elitism, meaning no parental survival. The steady-state GA assumes complete parental survival unless a child displaces some member of the parent population. This should not be denoted as elitist, as the term's common connotation conflicts with an entire population surviving. Mitchell refers to the steady-state technique as 'steady-state selection' [53].

The most well known reproductive selection scheme is fitness proportionate, also termed roulette-wheel selection. Parents are chosen with replacement much like the action of a spinning roulette wheel where the divisions of the circle are determined proportionate to the fitness of the individuals. Scaled selection, or sigma selection, is generally the roulette-wheel method with a scheme to normalize the weights according to fitness statistics of the population. The basic theme is to hold 'selection pressure' (the propensity of highly fit individuals to get breeding opportunities) constant throughout the run of the EA. This is an attempt to prevent early domination of highly-fit individuals and to increase late selection pressure.

In ranking selection, the probability to choose is based upon a simple ranking of the population by fitness value. The mapping of rank to a selection probability is arbitrary with linear and exponential schemes common. Again the intent is to prevent domination of breeding opportunities by highly fit members [53].

Tournament selection may be the second most well known method. A random subset of the population is chosen. These are ranked according to fitness with the top two chosen as parents. Alternatively the top ranked parent is chosen and a second sampling from the population is done for the second parent. Boltzman selection is similar to the scaled schemes where scaling is given according to a schedule during the run. In the initial set of generations, the probability of selection may be quite uniform and the scaling is increased to further bias towards highly fit individuals as generations progress. Boltzman schemes have many commonalities to the temperature schedules of simulated annealing [55].

2.2.5 The Mutation Operator

The mutation operator is perhaps the most elementary operator in the EA. It is inspired by the basic mutation observed in biological genetics due to various transcription effects and spontaneous changes in chromosomes observed in early evolutionary studies. The most common mutation scheme for binary genomes is to use a bit-flip probability applied against each bit independently. Thus the actual number of bits changed in an individual is not fixed. A typical mutation rate is $1/n$, giving an expected single bit flip yet allowing for variability. Others have used a rate of $1/mn$, giving an expected single bit in the entire population to be flipped in a given generation. Other mutation schemes are needed for non-binary encoding and it is suggested that further information is available to the reader in introductory evolutionary computation textbooks. [11, 53, 54].

A full discussion of the merits of mutation and the inter-community debates on its relative power are deferred until later.

2.2.6 The Crossover Operator

Crossover is the most complex of the EA operators and is sometimes referred to as the recombination operator in the literature. Crossover does depend on a reproduction selection mechanism of choosing at least two parent strings. It is inspired by the genetic recombination observed during meiosis in sexual reproduction. Note that the specifics of biological crossover between diploid organisms and the crossover operator of the EA are quite different.

The general idea is that genetic information from two or more parents should be combined in some way as to approximately mimic the effects of breeding in natural populations. The original standard form of recombination proposed by Holland is

Table 2.3: One Point Crossover.

Parent 1:	0110|1110
Parent 2:	1010|0011
Child:	0110|0011

called one-point crossover and was analyzed previously [39]. In one-point crossover, a random point $p \in [0, n-1]$ of the genome is chosen. The genome of parent a from loci 0 to p is copied to the child genome. The child genome is completed with the values from loci $p+1$ to $n-1$ of parent b. Holland's original formulation was to produce two children where the previous scheme was reversed for the second child. This has generally been simplified in theoretical analysis of EAs, though the equivalence of effect upon the population is not assured. Below is a simple diagram of one-point crossover with the cross-point denoted by '|'.

Other options include generalized x-point crossover where x cross points are chosen. Multiple points $x_i \in [0, n-1]$ are chosen without replacement. Between each cross point one or the other of the parent's genome is copied and alternated for the next crosspoint. Two-point crossover has been found to work well in some settings as it tends to account for any end-point bias from one-point where a child string is guaranteed to have one end point for each parent [53].

The logical extension of the x-point operator is referred to as uniform crossover. A bit mask of length n is chosen via a typical 50/50 probability of setting each bit to zero or one. For each locus of the bitmask, if the bit is one then parent A's allele value is copied to the child string. Parent B's value is used for a bitmask value of zero. It is not necessary to create such a bitmask in code, a simple 'coin flip' at each loci is sufficient. However, the bitmask explanation is important in analyzing uniform crossover behavior in theory.

Recombination operators with more than two parents have been used. These schemes vary, yet a simple method is that whenever a bitmask or cross-points dictate a choice between parent A versus B, this action is extended to choose between a list of x parents. Another recombination operator studied in EA theory is called 'gene pool' crossover. [56] In this scheme, the allele value of the child for a given loci is sampled from the allele distribution of the entire population at that loci.

Many authors have stated that crossover is the major contributor to individual and population genetic variation. Again, a full discussion is deferred.

2.3 Common Difficulties

Evolutionary algorithms suffer from a few basic aliments in practice. The EA is generally explained as an algorithm that attempts to balance the exploration of the search space with the exploitation of said space. A so called *premature convergence* condition happens when the population ceases to explore and exploits a local area that is genetically similar. The negative connotation of this early convergence is due to the assumption that early implies that the population has not found the global optima or some other 'good' solution. Restarting the exploration mechanism can prove difficult, yet ideally the final population has individuals on many local minima (or maxima) of the search space with the best individual belonging to the optimal global solution.

Another difficulty is that the algorithm does not necessarily adapt to the unique conditions of the search space. If one imagines that the GA is climbing a gradient to a local optimum, then the GA would do well to choose an appropriate step-size to properly climb the gradient. For real-valued representations, this step size refers to the mutation rate of each real-number. Different areas of the search space may have different best step-sizes. Thus, the set parameters of the GA determining step-size may be ill suited to the fitness function at hand.

The non-elitist GA can also suffer from a lack of persistence, meaning solutions found in a given generation may be lost in subsequent generations. While steady-state and elitist strategies may overcome this issue at first glance, there is no principled way of arguing this in general. Note that this lack of persistence is not universally a problem for fitness functions. The fitness landscape itself may be modular and the eventual loss of a given genetic attribute may not lead to an immediate fitness penalty, yet this activity may prevent the algorithm from combining that attribute with others in the future. Examples also exist where a non-elitist GA performs better than an elitist version [57].

A final criticism of EAs is the seeming inability of the algorithm to lend insight into the nature of the problem space or properly utilize human knowledge of a problem space. While this critique has much truth, there are many examples of hybrid EAs incorporating human engineering knowledge and judgement [58, 59] as well as studies of evolved mechanical structures that were later human analyzed for utility [60].

Many specializations of the GA have been made by adding operators or changing aspects of the main operators. The specializations can be helpful in practice, yet frustratingly most of these results are far from generalizable. A typical EA conference has been accused by Stephens [61] of having a proliferation of papers with titles like “A New Genetic Algorithm for the Multi-Resource Traveling Gravedigger Problem with Variable Coffin Size.” Given the propensity of copy-cat usages of algorithms these specializations may actually be harmful when applied blindly.

2.4 Operators and Parameter Setting

A central question for practitioners is how to set the various parameters of the GA. There are very few principled ways to set them a priori, and many practitioners perform multiple runs and attempt to manually choose good parameter settings. Rather than do this, some have proposed creating adaptive or parameterless EAs.

One example of such a scheme is mutation adaptation. A simple scheme is to change the mutation rate according to a deterministic schedule. Other schemes include adaptation of the population size, mutation and/or crossover rates according to some progress or quality heuristic. The methods are varied and a reasonable collection of techniques are reviewed [11, 62, 63]. A further niche group of researchers implementing such a scheme with fuzzy logic rule based systems is also reviewed by this author in [64].

Another general operator is diversity control of the population. These methods fall into two camps, implicit and explicit. They attempt to implement biological knowledge and breeding wisdom on the importance of genetic diversity for healthy populations. Implicit mechanisms include such techniques as incest prohibition [65] and other restrictive mating schemes.

Another idea is utilizing sub-populations where mating and evolution happens within a segregated population with slow migration of individuals. This can implicitly ensure population diversity by slowing down any potential dominance of the population by a handful of good solutions and allow semi-independent directions of evolution. An example of a combined sub-population and restrictive mating scheme is the Cellular GA [66]. Individuals are placed on a lattice of virtual points independent of the search space. Individuals can only mate with neighboring solutions.

Child solutions replace one of the parents in the lattice. Again, the general idea is to slow down the spread of highly fit individuals, thus implicitly ensuring diversity.

Explicit diversity maintenance mechanisms seek to actively manage diversity. Simple schemes include genotype and phenotype duplicate inhibition. Genotype duplicate inhibition is the restriction of the population from containing multiple individuals that are too similar according to some bit-string similarity of the genomes. The phenotype version works similarly with the restriction applied via a similarity metric on the value of the fitness function or other element of the solution.

More complex schemes are fitness sharing and deterministic crowding [67]. In fitness sharing the reproductive fitness of individuals is the true fitness discounted by the number of individuals that are too similar according to a sharing function measuring genotype similarity. The effect is that groups of genetically similar individuals have roughly the same chance of being selected for reproduction as does a lone genetically distinct individual of similar fitness. The deterministic crowding scheme sets up direct competition of parents and children. If the child string is of higher fitness than a parent, it directly replaces that parent string in the population.

The success of the measures summarized above can be difficult to measure a-priori for a given fitness landscape. In general they can be successful in landscapes with many scattered sub-optima or ones with many ‘basins and barriers’ along the way to good optima. While many are quite intuitive, it is not possible to conclusively state that they should always be used. Practitioners should as a matter of course attempt various types of enhancements to the GA to try and improve solution quality. They should also be wary of developing strong preferences for GA modifications as changes of problem type may defeat the implicit assumptions made by the enhancements and optional operators.

2.5 Relation to Metaheuristic Optimization

The EA and GA fit within the class of metaheuristic optimization algorithms. The family inclues the Metropolis-Hastings algorithm, simulated annealing, ant colony algorithms, differential evolution and other recently invented algorithms. See Luke's book on Metaheuristics [68] and the Wikipedia entry [69] for an excellent timeline on the family of algorithms.

2.6 How do EAs really work?

As shall be seen in the next chapter, explanations for the process of evolution have a long and complex history and EA theory is no different. The subject of how EAs work is an area of much speculation and some focused study. Holland speculated about how the GA might be working in his introductory work [17, 70]. Bremermann proposed theorems to highlight expected behavior. Nearly every major EA conference has a theory track. The *Foundations of Genetic Algorithms (FOGA)* [71] (first meeting in 1990) and the Dagstuhl seminar series [72] (first meeting in 2000) have been major influences to the EA theory community.

One could simply ignore this question and accept that the EA is widely useful in solving problems. A conservative estimate of the number of EA publications would be 50,000. The vast majority of those are application papers, instances of the GA's mostly successful usage. Unfortunately, many justifications of the EA's abilities are examples of *post hoc* arguments and/or 'just-so stories' [40]. This criticism can also be applied to the study of evolution in general [73]. However, the central question of how EAs work is fairly open. Major questions are:

- What quality of solution should one expect from an EA?

- How long will the EA take to find a good solution?
- Why was the EA able or unable to find the optimal solution for function X?
- Can the EA be proven to find the optimal solution for a class of functions Y?
- What genetic representation is best for a class of functions Y?
- What are the general dynamics of the EA in terms of progress to the optimal points?
- How does a practitioner go about choosing operators and parameter settings?
- What is the dependence of operators on each other?
- What attributes of incremental solutions did the EA combine to arrive at a solution?
- What are the roles of selection, crossover and mutation in general?

It is quite likely that most of these questions will remain open for the foreseeable future and answered only in fragments. This work will address only a few of these questions for carefully constructed fitness functions to illustrate a hypothesis or explore an idea.

CHAPTER 3

HISTORICAL REVIEW OF EVOLUTION AND EVOLUTIONARY THEORIES

> This preservation of favorable variations and the rejection of injurious variations, I call Natural Selection.
>
> Charles Darwin - On the Origin of Species

3.1 Theory of Evolution

In 1858 Charles Darwin and Alfred Russel Wallace published a joint paper, *On the Tendency of Species to Form Varieties ...* [74], outlining a theory of Natural Selection. This was followed up in 1859 by Charles Darwin's seminal book *On the Origin of Species* [75], which some argue is one the most influential books published so far by humanity. The core premise of Darwinian theory is that individual species in biological systems undergo a constant process of competition for resources, survival, reproduction and adaptation to their environment. In summary, his central ideas are [75, 76]:

- Species generally have more reproductive potential than survival potential.
- Resources like food and water are limited, but mostly stable over a given time period.
- Due to these limits, populations tend to remain generally constant in size.
- The above results in a competitive struggle for survival and resources between individuals.
- Individuals with superior traits or abilities to capture resources tend to outcompete others.

- Survival and successful reproduction by superior individuals generally results in a passage of the traits and abilities to progeny.
- In species that reproduce sexually there is greater variation in traits and abilities in the population.
- The long term effect of this process is a gradual adaptation of a population over time into highly fit optimizers of an environment.
- Periodically, sub-populations of species adapt in different directions, evolving distinct traits and abilities that eventually result in dissimilar species.

The following review of the history of evolutionary theories will outline the major high points as they relate to mutation, crossover and selection. Books by Ernst Mayr [76], Peter Bowler [77] and Janet Brown [78] extensively document the history of biological thought during the Darwinian period. In the words of Felsenstein [79], histories like this are purposely "vague and impressionistic". There is not room to recount full details. The curious reader should consult original sources. [1]

3.2 Historical Anticipation and Reception of Darwinism

Portions of Darwinism had been anticipated as far back as the ancient Greeks, Romans and Chinese in the first millennium B.C. [77]. In the Enlightenment era of the 18th century, Maupertuis, Buffon and others, including Darwin's own grandfather Erasmus, proposed the idea that groups of similar animals, including humans, had

---

[1] This review mimics the framework and time line of several Wikipedia articles on the subject [80, 81, 82, 83, 84]. Much more detail is added from original and modern sources as well as various survey articles on individual topics.

common ancestors. These theorists also proposed the idea of gradual accumulated changes creating new races and species.

The early 19th century had many important developments prior to the publication of Darwin's theories. In 1809, Jean-Baptiste Lamarck put forth his theories of evolution [85], summarized as follows [86]:

- Spontaneous Generation - Simple organisms were being spontaneously generated continuously.
- Use and disuse - Organisms develop useful and lose unuseful characteristics.
- Complexifying force - An intrinsic force propelling organisms towards more complexity of characteristics.
- Adaptive Force - An intrinsic force propelling organisms to depart from complexification and adapt to their environment.
- Inheritance of acquired traits - Organisms pass on the acquired characteristics due to use-disuse to their descendents.

Robert Chambers anonymously published the *Vestiges of the Natural History of Creation* in 1844 [87]. The book was widely read and very controversial. It put forth his synthesized theory of *transmutation*, asserting that all present forms of life evolved via a process that began with the formation of the solar system, continued to global geology, sea and land plants, fish, reptiles, birds, mammals and finally humans. He notably used examples from the fossil record and geology to reinforce his ideas. Chambers' theory also explored the similarity of mammalian embryos early in their development, i.e. comparative embryology, and postulated that branching of development similarities corresponded with the gradual evolution and speciation of those organisms.

The book spurred a public debate in the United Kingdom and elsewhere pitting those who believed in transmutation of species against, among others, those who saw the present state of the animal kingdom as fixed and species as immutable as well as those who objected on religions and philosophical grounds. [77]. Darwin himself notes in Origin that the publication of the Vestiges "[...] has done excellent service in this country in calling attention to the subject, in removing prejudice, and in thus preparing the ground for the reception of analogous views."

Much of the next hundred years following the publication of Origin was spent analyzing and critiquing Darwin's ideas as well as the controversies it created. A recent accounting of that history is available in Brown [78]. A full recitation of the opposing views is outside the scope of this review. The next section will contain only debates and responses within the biological literature.

3.3 Early Interpretations

While Darwin's book was convincing to the biology community, it was not particularly effective at arguing that the primary driver of change is natural selection. Many contemporaries embraced the progressive idea of improvement and how it could be test with observational studies. The naturalist community was among his main champions, particularly gravitating to the idea of *common ancestry.* However, many scientists were deeply sceptical of natural selection being the mechanism of change. There were fierce debates about what drove evolutionary change [77].

Alternate late 19th century pre-evolutionary ideas were *othogenesis* and *saltations.* Othogenesis postulated that there existed some inherent non-physical force driving organisms towards perfection and was advocated by paleontologists like H. F. Osborne and Lamarck himself [76].

The theory of saltations states that evolution can proceed in discontinuous jumps and went against the theory of "Natura non facit saltus", (Latin for "nature does not make (sudden) jumps"), dating back to at least Leibniz (approx 1765). Earlier versions of the maxim date as far back as Albertus Magnus in the 13th century [88]. Darwin embraced this ethos of gradualism and viewed that natural selection acts by the "preservation and accumulation of infinitesimally small inherited modifications". Thomas H. Huxely, who came to be known as "Darwin's Bulldog" for his support of Darwin's ideas, nevertheless took Darwin to task for too heavily refuting the idea of discontinuity [89]. Later, William Bateson also objected to the gradualist idea and advocated discontinuous variation [90].

August Weismann is notable during this period for developing a theory of inheritance and strongly supporting the Darwinian notions of strong natural selection. In particular, he developed the theory of germplasm, wherein the inheritance of traits happens only in special germ cells. Examples of germ cells are the sperm and egg. He argued that the remaining cells of the body, somatic cells, do not contribute to inheritance and as such any acquired traits of these cells could not pass to future generations, negating the central idea of Lamarck. This became known as the Weismann Barrier and survives to this day, though recently, holes are being poked in it. More on this at the conclusion of the chapter.

Weismann also argued for the role of recombination and the selective advantage of sex to multiply genetic variability. In his view, the random variation due to mutation was not sufficient to explain the adaptations present in life, the variability must be guided by selection. He also strongly advocated the idea of amphimixis (sexual recombination) to spread advantageous variations, and was among the first to put forth the process of 'crossing over' due to his knowledge of cytology (the study of cells). Weismann was also a confirmed gradualist and uncompromising advocate of the selectionist theory [76, 91, 92].

Francis Galton was also important to this period. He was an advocate of hard heredity and rejected the inheritance of acquired traits. His law of ancestral heredity was based upon the idea of populations and their genetic variation. He observed that the traits of a population stayed constant in mean value across generations [76]. This idea of regression towards the mean became very important in later population genetics and statistics at large. During the course of this work he also conceived of the ideas of standard deviation, correlation and the normal distribution [93].

3.4 Genetics and The Chromosome

In 1866, Gregor Johann Mendel published a paper on plant hybridization [94]. In it he worked out the basic laws of heredity based upon an extensive hybridization study of pea plants. This work was largely ignored until Hugo de Vries and Carl Correns rediscovered it in 1900 while doing similar hybridization studies. Bateson translated Mendel's paper into English and set to work himself, coining the term *gene* to describe the particulate unit of inheritance as well as the term *genetics* [76]. Wilhelm Waldeyer had in 1888 described the thread-like structures present in the nucleus of cells as *chromosomes*, largely due to the fact that they readily absorbed a colored stain during cytology studies. Prior to the Mendelian rediscovery, there was no ability to link the chromosomes to hereditary traits [95].

Breeding experiments carried out by Wilhelm Johannsen in 1903 on pure-line bean plants demonstrated that variations in plant size were not separable under human directed selection and breeding because they continued to follow the Normal curve [96]. This caused significant issues with Darwin's basic mechanism of evolutionary change, creeping variation and fluctuation of quantitative traits organized by selection [97]. This lead to the hypothesis that some missing element was at play, causing Johannsen to coin the terms *genotype* and *phenotype* to separate genetic variations from quantitative environmental variations.

The rediscovery of Mendel's laws and subsequent work divided itself into two groups, those supporting the study of genotypic heredity and those supporting the study of phenotypic variation via statistical techniques, biometrics as pioneered by Galton [76]. The genotypic Mendelians and phenotypic naturalists formed the two camps of evolutionists that persisted for the subsequent two decades. The work of Kellogg describes the various combinations of genotypic/phenotypic, gradualist/saltationist and hard/soft inheritance held by evolutionists up to 1907 [98].

3.5 Mutationism

The mutationist view of evolution was expressed in the works of early geneticists William Bateson, Thomas Hunt Morgan, Reginald Punnett, Wilhelm Johannsen and Hugo de Vries. Mutationism argued that changes in the genome's particulates are the core mechanistic idea. The resulting variants are then subject to selection via differential survival. Variants may be both genotypic and phenotypic manifestations. Bateson and de Vries posited that selection in populations served only to find, choose and isolate representatives of existing types. Mutations and the resulting altered development is an internal creative process, whereas selection is the external process. This duality, argued for by Morgan in 1903 [99], contributes to the discontinuity, directionality and creativity of evolution [97].

The program of the mutationists was to study the laws of Mendel, the mechanisms of heredity and attributes of the spread of mutations, essentially to work out discretized genetics. Morgan's famous studies of *Drosophila melanogaster*, the common fruit fly, illustrated many details of Mendelian-chromosome theory. By carefully breeding and counting mutant characteristics of many thousands of fruit fly generations, Morgan and colleagues were able to map out important factors of the fly's chromosome including recessive genes affecting eye color.

Herbert S. Jennings undertook a series of extensions of Johannsen's pure-line research both experimentally and mathematically. He generalized the results of sexually bred pure-line populations under selection. Jennings concluded that via crossbreeding and selection it was possible to isolate attributes or mix them in indefinite combinations and determined that it was not possible to produce anything not already present in the population via selection - save for when rare mutations occur.

The mutationist debate with the biometrics camp continued. Led by Karl Pearson and Walter Weldon, they continued to develop statistical techniques, including the chi-square test and coefficients of correlation, and extended the work of Galton further into the study of the naturalist attributes of populations. Around this time Pearson made the extraordinary claim that sexual reproduction within the Mendelian system would produce little inheritable variation. Among others, J. Arthur Harris analyzed the results of the pure line experiments with statistics and called the data and conclusions of many studies into question, though he remained receptive to the pure line theory itself. Interestingly, Galton was claimed by both schools while staying mostly neutral or slightly leaning towards the mutationists in some areas.

During this first decade of debate from 1900 to 1910, G. Udny Yule published three papers proposing the compatibility of Mendel's laws and the Darwinian laws of ancestral heredity. He noted that the basic conflict between the two was predicated on the assumptions of hybridization versus interracial heredity and that these assumptions were orthogonal. By moving away from the assumed complete dominance of certain traits under hybridization and factoring in the effects of the environment one could show mathematically that the laws were consistent. Yule rebutted arguments of Bateson, Pearson and Johannsen during this time. Unfortunately his compatibility arguments were buried in the personal aspects of the debate and his prescient ideas would have to wait a decade or two to resurface within the revolution of population genetics [89, 100].

It is also notable that the biology community readily embraced the pure line work and conclusions of Johannsen. The citations of criticism was rare and many treated the data and results as if they proved the pure line theory.

This debate is well covered in Provine's 1971 book [89]. Provine asserts that the debate was likely exacerbated by the egos and personal issues between the proponents of both camps. He also notes that among the experimentalists like Johannsen, there seemed to be many examples of confirmation bias because researchers were often looking for data to support their theories.

3.6 Selection, Recombination and Linkage

Without recognizing it, Pearson's arguments on recombination pointed out a large potential for heritable variation. Pearson was blinded by the debate and stopped at his conclusions that Mendelian heredity produced no variation. During the second decade of debate the differences began to narrow as the mutationists saw experimentally that selection and recombination could produce much variation.

Around this time H. Nilsson-Ehle began work on oats and wheat, conducting a program of crossings of varieties. He repeatedly found the linkages between multiple Mendelian factors and violations of Mendelian breeding ratios. The combinations produced demonstrated that in some cases three or more independent factors determined phenotypic attributes of kernel color in wheat. He calculated, based upon observations, that in a case of ten factors with sufficient diversity in the breeding population, 60,000 different possible phenotypes could be derived, each with a different genotype. Given this work on genetic recombination he concluded that in small populations many cases of new phenotypes, by supposed mutation, could in fact be attributed to the rare recombinations of genetic material present. Nilsson-Ehle also concluded that the real purpose of sexual recombination is to increase the possibilities for genetic mixing and that the resulting variation contributed to selective survival [101, 102].

William Castle, an embryologist, began as a supporter of Bateson against the biometricans and published a paper on Mendelian heredity. After a scathing review on his mathematical methods by Pearson and subsequent work on the inheritance of coat color in rats, Castle converted. He and others began to reason that the conclusions of the pure line theory against selection were not so sound under conditions of higher population diversity and cross-breeding. Starting in 1907 and continuing for a decade, he conducted a program of rat and guinea-pig breeding with Hansford MacCurdy similar to Morgan's work on flies. Castle designed experiments to upend the pure line results and indeed showed that it was possible to produce new 'types' with cross-breeding and selection [89]. He had essentially selected beyond the range of initial variation in hooded rats.

Edward East adopted a similar methodology of Nilsson-Ehle that he applied to corn plants. Initially he wanted to demonstrate multi-factor inheritance, or linkage, in maize corn experiments. As Castle's work became known, East began to see the full possibilities for recombination as he followed Nilsson-Ehle's program within a single cross-breeding population. His work confirmed that there was a Mendelian basis for apparently continuous variations in phenotypic attributes [103].

As the conclusions of previous pure-line theories were questioned, Jennings undertook a new series of experiments. He showed that in fact selection could change pure lines of a simple protozoa. Jennings put forth the idea that unseen recombination was responsible for the new success of selection. In 1917 he wrote that recombination and selection were central to understanding evolution. He was converted, and published an influential book on mathematical genetics [104].

At this time Morgan's work was leading in new directions as well. He was beginning to map out the linkages between eye color genes in his flies. Morgan also noted that a new mutation would not automatically spread through a population. Mutations may be advantageous, neutral or deleterious. Ernst Mayr famously characterized these views as theories of "lucky mutants" and "bean bag genetics" [89]. A central contribution of Morgan at this time was the idea that genetic linkage was impacted by the geometry of crossover. The farther apart on the chromosome during the cross-over process the more likely linked genes could be disrupted, thus impacting their correlation frequency in the populations.

Morgan and his team were primarily responsible for producing both evidence and the basic idea that the chromosome was the physical medium of inheritance. They postulated that genes were lined up along the chromosome like beads and that the observed crossing-over of the chromosome during meiosis was the major mechanism of recombination [95]. It is interesting to note that Morgan was initially a great skeptic of the genes-on-chromosomes theory. His conversion was a sign of the times [106].

As late as 1918 East felt compelled to restate the results of Nilsson-Ehle on the importance of recombination. Provine postulates that had Nilsson-Ehle'a early work in 1909 been published into English, the significance of sexual recombination in producing new variation may have become apparent to the wider community sooner [89].

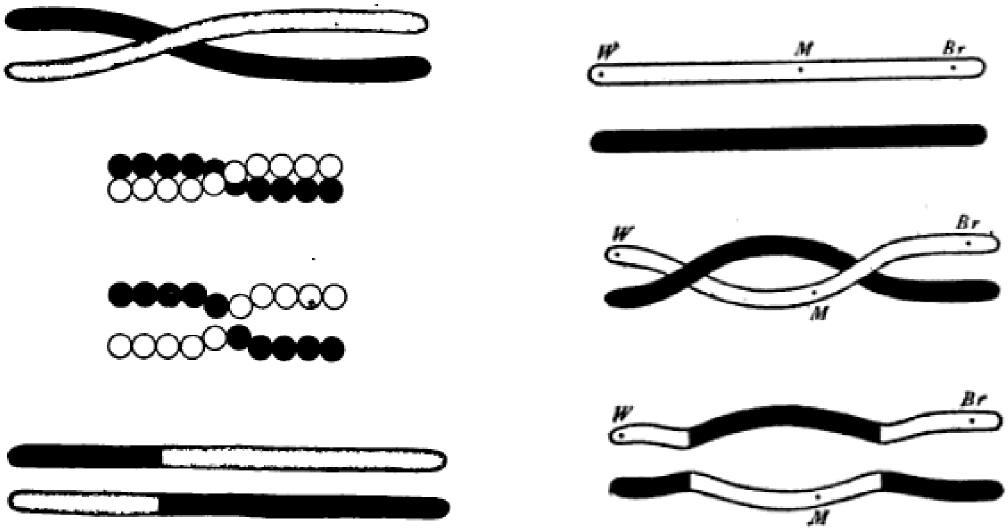


Fig. 64. Scheme to illustrate a method of crossing over of the chromosomes.

Fig. 65. Scheme to illustrate double crossing over.

Figure 3.1: Diagram of crossover from Morgan's 1916 *A critique of the theory of evolution* [105]. *Graphically reformatted from original.*

Fig. 68. Scheme to indicate that when the members of a pair of chromosomes cross (at a-b) the region on each side is protected inverselv to the distance from a-b.

Figure 3.2: Diagram of crossover position effect from Morgan's 1916 *A critique of the theory of evolution* [105]. *Graphically reformatted from original.*

3.7 Population Genetics

Prior to 1918, three important developments would set the stage for a mathematical revolution of genetics based upon populations and the idea of a *gene pool.* First was the co-discovery of the Hardy-Weinberg equilibrium principle of allele frequencies. While Mendel had experimentally worked out the stable ratios of genes in his hybrid bean populations, he published no attempt to derive any general formula for predicting those stable ratios. Yule and Castle also worked in this area under the conditions of random mating and set up a general principle of equilibrium for a single locus with two alleles [89].

Pearson attempted to generalize this to n loci, yet only under uniform initial conditions. G. H. Hardy was introduced to the problem by Punnett and quickly derived an equilibrium solution [107]. This solution was anticipated six months earlier by Wilhem Weinberg [108], but unknown or ignored for years afterward until the law was finally named the Hardy-Weinberg law. Modeling the frequencies of alleles in genes would become a major feature of population genetics.

Jennings and Raymond Pearl worked separately on the problems of analyzing inbreeding under Mendelian inheritance. Under certain assumptions they were able to work out a formula that fit their data by trial and error. This method was too difficult for some of the complex mating schemes of breeders, and the effort was stalled. The quantitative analysis of inbreeding would become the second important development in the birth of population genetics.

The third development is the analysis of selection. R. C. Punnett wanted to understand the speed at which new alleles would spread or be destroyed in a population under selection. Working with H. T. J. Norton, a mathematician, they derived a survival table of a recessive/dominant gene structure under different selective advantages and disadvantages in 1915. When combined with Morgan's previous work the analysis gained wide usage. It was used to show that rare but slightly deleterious recessives could not easily be expunged from a population, while also demonstrating that modest selective advantage could quickly change gene frequencies in only a few generations [89].

As 1920 neared, many began viewing Mendelism and Darwinism as complementary [89, 109]. Howard Warren published a mathematical argument that Mendelism and Darwinian selection were compatible. Yule's previous papers were being examined, and eventually Weinberg's work would cease being ignored. Yet it took the triumvirate of R. A. Fisher, Sewall Wright and J. B. S. Haldane to reinvent evolutionary thinking and bring mathematical rigor to bear on the three above areas of development.

Ronald A. Fisher initially worked with Pearson and other biometricians, yet as he wanted to synthesize Darwinism, Mendelism and biometry he was not well received. In 1918, his seminal paper interpreting Mendelian inheritance with the tools of biometry was published [110]. He examined the previous claims of Pearson and showed via (his) new analysis of variance techniques that there was no conflict between measured correlations in phenotypic attributes and Mendelian inheritance. Fisher also explored the epistatic linkage of genes and showed that it could confound analysis of genetic similarity via biometric methods. This was demonstrated as true for environmental factors as well and that the effects of environment versus genetic dominance could be separated. Examining data on human inheritance of traits from the biometrists, he looked at the statistical effects of assortative mating, loci linkage and genic interaction. His model showed that for large populations these effects were negligible. The central conclusion of the paper was that discrete Mendelian factors could in fact determine the continuously varying phenotype attributes of organisms, such as height in humans. This conclusion disproved the basic thesis of Pearson and the biometricians [89, 111].

Fisher went on to examine the interaction of selection, mutation, gene extinction, assortative mating and dominance. Building a model of this process, a task that the Mutationist community of Bateson shied away from, he quantitatively analyzed Mendelian heredity. Combining selective advantage with the Hardy-Weinberg equilibrium model he worked out selective conditions for homozygote versus heterozygote dominance in a large population. His models showed that a small mutation rate could be effective at preserving genetic variation in a large population. He followed up the theoretical results by working with others on experimentally verifying or amplifying his models. Fisher's central idea of evolution was that within large populations with inherently high variance, natural selection tended to act deterministically on single genes. It was also his conviction that small selection pressures over a long period of time within large populations were crucial for species change. His 1930 book, *The Genetical Theory of Natural Selection* [112], would button up his seminal ideas on population genetics. It is important to note as well that he spawned many new developments and sub-disciplines in statistics while working with genetics [111].

Sewall Wright began his work on evolution with an intensive study on the interaction of a system of genes controlling coat color and patterns within Castle's laboratory. This work on genetic interaction systems would influence much of his future thinking on what was important in evolution. He assisted with the results of Castle by selecting attributes in hooded rats beyond their initial levels of variability. While in Castle's lab, Wright assisted with calculations for the ratios of homozygotes under inbreeding. This eventually turned into a well known paper and easy to use model on the coefficients of inbreeding in 1922. Wright also used the Hardy-Weinberg model to calculate genotype frequencies as a method of hypothesis testing in a series of papers on color inheritance in mammals, though at the time he did not know of the publications of Hardy or Weinberg. He also invented a method of path analysis, the use of directed dependencies amongst conditional variables, to describe and analyze systems of breeding. Using this technique Wright was able to determine that in a population of a highly inbred stock of guinea pigs it was environment and not heredity that explained the variability in coat color and markings. He also published a series of influential papers on mating systems that analyzed the effects of assortative mating, selection and the genetic composition of a population of inbred stock. He later applied these same modeling techniques to inbreeding of a variety of cattle in 1923 [113].

Returning to the analysis of guinea pig mating that had been going on under his direction since 1915, Wright showed that some stocks of inbred families lost much vigor and had issues with birth weight, litter size and birth mortality. Some families did not degrade in vigor and were similar to the control group. He noted also that when inbred stocks were crossed, they recovered much of their fitness. Wright explained the differences as follows. While inbreeding quickly produced homozygosis, the differentiation between families resulted from the random fixations of genetic factors originating with the heterozygous source stock. Crossing disrupted those fixations.

Wright was working for the U.S. Department of Agriculture and used these results and his models to develop a set of recommendations on combining crossbreeding, inbreeding and selection for the general improvement of livestock. By inbreeding certain lines, one could observe the hereditary differences between lines and thus more effectively choose candidates for crossing. The desired attributes of the lines would likely remain in the hybrids while the decline in vigor of the inbred line would not transfer due to crossing.

These results were both highly practical and very important in evolutionary thinking. Inbreeding would fix large networks of gene interaction systems, allowing more effective artificial selection. To Wright this obviously meant that selecting at the level of interaction systems was far more important than selection of individual gene effects. As he developed a more general theory of this he became convinced that genetic drift during inbreeding was a more important effect than Fisher had recognized. Fisher was of the opinion that natural selection was not very sensitive to genetic linkage and acted to increase fitness by single gene substitutions in large populations. Wright was more of the opinion that natural selection operated most effectively on small populations, inducing fixation of some interaction systems and random drift in others, producing a faster adaptation than a large population. Wright also concluded that Fisher's ideas of dominance were flawed since single genes could effect multiple aspects of an organism's phenotype, small selection pressures were ineffective in driving change [114].

While Wright stated that his different methods could arrive at the same gene frequency results as Fisher, he differed sharply on the interpretation of what was happening in the evolutionary process. His *shifting balance theory*, involving random drift, interdeme and intrademe selection, became a major principle of thought in evolution [113].

John B. S. Haldane first looked at genetics via an examination of third-party data on mice heredity. After his review of other work, he conducted his own breeding experiments on mice and rats, and published an early paper on gene linkage in mammals. After service in WWII, he set to work on deriving models for estimating the errors in existing gene linkage data as well as corrective methods for experimentally derived chromosome maps formed from linkage counting. His major early work in the field concerned the development of a general mathematical theory of selection.

His papers progressed from models with broad simplifying assumptions to ones with varying mating schemes and inbreeding. Most of the models, like Fisher's, dealt with an infinite population size. Similar to Fisher, he also developed models for the survival of mutations under various selection pressures and the balance between the two processes. In contrast to Fisher, he noted that single mutations could cause high selection pressure as well as the important effects of migration in large populations.

In later work he explored gene linkage and relevant equilibrium effects. Haldane's models of gene frequencies under selection are widely used in genetics textbooks and his illuminating work on the intensity of selection and the resulting pheonotypic change would be fundamental ideas in population genetics. He also anticipated later work from Wright on the meta-stable states of equilibrium under selection. Haldane's series of ten papers, "A Mathematical Theory of Natural and Artificial Selection", and later book re-examining the basic Darwinian theory, *The Causes of Evolution* [115], became a central part of the coming evolutionary synthesis.

His most famous two papers dealt with the costs of selection and effect of variation on fitness. He showed that the variation of a population is dependent on the overall mutation rate, which is now known as mutation load. The other paper concluded that the fixation of a gene in a population under selection depends more on its initial frequencies than the coefficient of selection. His basic work and ideas would have a large effect on the later resurgence of mutationism as a source of variance in the 1970s. According to James Crow [116], Haldane's status as third among the triumvirate is due via his breadth of interests and influence among his contemporaries. Unlike Wright and Fisher, he did not make any path-breaking accomplishments in evolution.

Provine states that the three had an interesting set of symmetrical views on the similarity of each other's work [89]. Essentially, each of them thought that the other two more similar to each other than to himself. Wright disagreed with both on the importance of large populations, yet agreed with Haldane on the effects of drift as being important. Fisher though both underestimated the importance of small selection pressures acting over long time periods. Haldane saw Wright and Fisher's work as being more self-similar than his own efforts and disagreed with Fisher's ideas on genetic dominance.

3.8 Modern Evolutionary Synthesis

The state of evolutionary research in the late 1930s was one of growing consensus. The table was set for illustrating that consensus via group definition. One flaw of the work of Fisher, Wright and Haldane was that the results were too mathematical and not understood by the wider biological and naturalist community. As told by Mayr [76], the rise of new leaders was a precondition of any synthesis or consensus between the population geneticists and the naturalists. It was necessary for a young group of scientists to be interested in applying population genetics to the population problems naturalists cared about. New leaders among the naturalists had to learn that the genetic interpretations of the younger Mendelians were no longer opposed to natural selection as a primary stance. They would revisit Darwin's gradualism.

Theodosius Dobzhansky had been a worker in Morgan's fruit fly lab and understood the interface of population genetics with actual populations. He had also been influenced by obscure work in the 1920s from the Russian Sergei Chetverikov. Chetverikov had followed the work of the West, conducted inbreeding experiments and arrived at many of the conclusions of Fisher, Wright and Haldane ten years earlier. Unfortunately his works were unknown for years outside of Russia. His student Nikolay Timofeeff-Ressovsky, who was more well known after leaving Russia for Germany, put forth an early analysis of mutation as the source of genetic variability in geographically dispersed populations [117].

In 1937, Dobzhansky published *Genetics and the Origin of Species* [118]. This book aimed to bridge the gap and present conclusions of the population genetics work in an accessible format. The main points of emphasis were that sub-populations with distinct genetics were important and that the variability of real world populations was far greater than previously realized. Dobzhansky presented the theory that natural selection acted to drive evolutionary change and maintain genetic diversity. Mayr [76] and others have noted that the book marked the official beginning of the *modern evolutionary synthesis*.

Ernst Mayr contributed to the synthesis by defining the species concept, a reproductively isolated population of interbreeding organisms. Under such conditions the action of natural selection upon small inbreeding populations in addition to the ecological factors at play conspired to create new species.

The resulting synthesis has four central tenets [76]. The first is that natural selection is the primary engine of change. The second asserts that evolutionary change is gradual, with variation driven by mutation and recombination. The third is that one can explain all known evolutionary phenomena in a manner compatible with known genetic mechanisms and the naturalist's observed evidence. The fourth concerns the species concept above and the primacy of population thinking, the genetic diversity of a population dominates its evolutionary direction. Julian Huxely summarized this and coined the phrase *evolutionary synthesis* in his 1942 book *Evolution: The Modern Synthesis* [119].

George Simpson buttoned up any remaining issues from paleontology by arguing that the steady linear progressions paleontologists constructed from the fossil record, including descriptions of the horse, did not stand up to scrutiny. Published in *Tempo and Mode in Evolution* [120], he applied the principles of Wright's shifting balance theory to show varying rates of change in evolution were possible. Simpson also broke away from precise genetic interpretations. He applied the modern synthesis to the problems of macro-evolution, showing a concordance with general genetic principles. By departing from a definition of evolution as a change in gene frequencies, he made compatible the new synthesis and the well studied and organized field of paleontology.

The concept of the gene pool played an important role as variational aspects of the process worked to maintain genetic diversity. It is worth nothing that while the new synthesis could be viewed as a reconciliation, Ernst Mayr and others saw the synthesis as a final victory over mutationism [76].

3.9 Chromosomes and DNA

While a full review of DNA development and function is outside the scope of this work, the research of a few groups stands out in creating the final linkage between genetic theory and biological fact.

Cyril Darlington played an important role in pre-synthesis work on the chromosome and some say he essentially invented the major theoretical cytology links to genetics [95]. While the work of Morgan tied the observed crossing over in cytology to observed genetic linkage effects, the hypothesis was not yet demonstrable. Ten years later, in the mid 1920s, cytology was an ancillary science, no longer central to evolutionists. Darlington set out to change that and prove the link between chromosomes and genetics. He deduced the ‘laws’ of observable chromosome behavior, chromosomes pair up end-to-end in a bundle called a *chiasmata* and that the chiasmata observed during meiosis is always a consequence of genetic crossover. Mayr calls Darlington the greatest early contributor to the understanding of recombination and its evolutionary importance [121]. Note that while Darlington laid much of the idea work in showing the linkage between genetic crossing over and chiasmata, it was

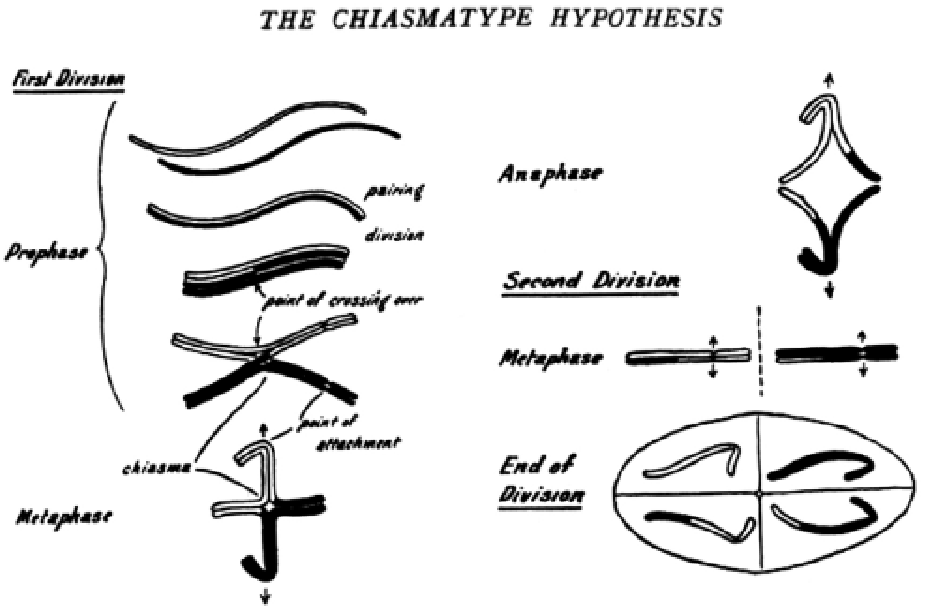


Figure 3.3: The Chiasmatype Hypothesis from Darlington's 1932 *Recent Advances in Cytology* [123]. *Graphically reformatted from original.*

Harriet Creighton and Barbara McClintock [122] who first demonstrated the physical linkage [95].

More striking at the time were his arguments that heredity itself was subject to evolution. The processes of meiosis and chiasmata as well as the entire reproductive process is genetically controlled and as such are themselves selected upon. This implied a more complex process of evolution than was initially thought, and that evolution itself evolved the process of forming simple chromosomes, then diploid chromosomes and the mechanical processes of genetic mixing. According to Harman [95], Darlington was left out of later credit for his contributions to the synthesis. He effectively developed a complete general theory of how evolution worked through chromosomes with a strong emphasis on recombination for the production of complexity [124]. Cytologists disliked his books. His idea that the genes-on-chromosomes theory was itself subject to the evolution hypothesis was met with uncertainly by geneticists. Both Haldane and Dobzhansky came to appreciate his contribution.

The link between chromosomes and DNA came into focus when Avery, MacLeod and McCarty [125] showed in 1944 that DNA was the hereditary material within the chromosome. Finally in 1953 Watson and Crick [126] showed the function and structure of DNA to finally complete the linkage of genes, heredity and biological systems.

3.10 Neutral Theory of Evolution

The victory over the mutationists turned out to be short lived. In 1968 Motoo Kimura proposed the influential *neutral theory of molecular evolution* [127, 128]. The theory states that random drift of selectively neutral mutants causes the vast majority of evolutionary changes at the molecular level. Kimura discovered that when genomes within species are compared, the bulk of the observed genetic diversity is neutral, giving no selective advantage. The cause of this diversity in Kimura's theory was the random genetic drift acting upon these neutral regions.

Kimura argued that while phenotypic change was dominated by natural selection, at the genetic level the "great majority of evolutionary changes are caused by random drift of selectively neutral mutants" [129]. Jack King and Thomas Jukes followed up this theory with the provocative paper "Non-Darwinian Evolution" [130]. The paper took a strong tone and signaled a break with the standard modern synthesis. They examined the amino acid sequence differences between various species and noted that they differed to an extent unnecessary relative to common function. They surmised that point mutations and genetic drift account for these differences, concurring with the ideas of Kimura. Kimura later published an influential textbook about the theory [131].

In 1973, Kimura's student Tomoko Ohta showed a 'nearly neutral theory' of molecular evolution [132, 133]. She observed that the non-coding regions of DNA diverged much faster than coding regions, on the order of the inverse of population and number of generations. The majority of the mutations driving divergence were 'slightly deleterious'. Ohta later updated her model to add both slightly positive and negative advantage mutations as well as selection itself, unlike the strictly neutral model. The express purpose of the model is to explore the interaction of random genetic drift and selection. Her conclusions were that in large populations selection dominates the evolution, yet in small populations drift takes over. Ohta's models predict that small populations have more rapid evolution than large populations, whereas the traditional selection theories without drift predict the opposite conclusions.

These ideas and the provocative titles sparked some debate [134, 135, 136], essentially reigniting certain old mutation-selection arguments within the new molecular world of genetics under the heading "neutralist-selectionist debate". As the controversy around Ohta's theory continued, it was noted in a published debate by Kreitman [137], Ohta [138] and Dover [139], that both neutral and selectionist theories could be used to predict observed data by adjusting the various parameters of the models. Dover argued that this makes it not a general theory at all. Furthermore, he argued, the models do not include a myriad of real observed effects such as genetic hitchhiking, hot/cold recombination spots, differential mutation rates and balancing selection.

Kreitman observes that the debate around deterministic versus stochastic forces in evolution follows a very predictable pattern. First, a new set of empirical data is acquired, which is then used to support some particular theoretical model. The alternative model is quickly updated to account for the new data and the cycle repeats. Thus far neither model was able to correctly distinguish between competing hypotheses given the inherent flexibility. Kreitman does conclude that the neutral models continue to serve a valuable function by providing a mechanism to test for the presence of selection.

It can be demonstrated that drift processes are able to fix genetic codes within a population, and that these codes may be either advantageous, disadvantageous or neutral. Furthermore, population size matters a great deal. If the population size is less than the reciprocal of the selective advantage of a specific gene, then random drift processes will determine the fates of the distribution of gene values. This simple observation has been confirmed with the ever expanding data available to geneticists. Thus, the strictly neutral theory has become the *null hypothesis* of selection and other models of genetic evolution [129, 140].

Given a set of data and a candidate hypothesis or new model of selection acting on a sequence of genes, the first action one must take is to reject the null hypothesis that drift can account for the observed effects. If it can be shown that the selection hypothesis is predicted to speed up fixation of advantageous genes and slow down fixation for disadvantageous genes relative to the effects of drift, then one can reject the null hypothesis. The neutral theory is also the basis for the *molecular clock technique*, which evolutionary biologists use to estimate the time passed since divergence of species from a common ancestor. Comprehensive treatments on the current views of this modern neo-mutationalism are available in articles by Nei [141, 142] and Stoltzfus [97].

Finally, it is important to note that neither the neutral theories nor many of the selection theories involved in the debates attempt to account for the effects of crossover in changing gene value frequencies in a population. This is a major shortcoming of these models, though as will be seen later there are obvious reasons why recombination makes full theoretical analysis difficult.

3.11 Views on Recombination

During the lead up to the modern synthesis and after, selection came to dominate thought. It was assigned as the primary operator of evolution. Mutation was explicitly downplayed by the advocates of the synthesis. Recombination, while having some prominence, was not given anything near equal billing to selection as the driving force of change. However the ubiquity of sex and recombination is a very well studied problem in evolutionary biology [79, 143, 144].

In biological systems, recombination can refer to any process that produces a reassortment of genetic material, such as sexual reproduction (blending of maternal and paternal genes), Medelian reassortment of chromosomes, crossing over during meiosis and the action of repair of double strand DNA breaks. Viruses can exhibit two types during replication, independent assortment of unlinked and the recombination of incompletely linked genes.

Weismann was the first advocate of the idea that sexual reproduction provides variation upon which selection acts [92]. The principle is that recombination does not increase fitness directly, it serves to increase the variance of fitness of the population. This variance is acted upon by selection to produce higher mean fitness down the line of generations [144].

In 1913 Morgan's book *Heredity and Sex* [145] was published, and in it he called out Weismann's principle as one that opened up many new ideas. Morgan argued that sex and recombination act to both increase the vigor and weakness of a population by allowing vigorous and weak combinations of traits to propagate in the population. While he argued that recombination would not produce anything new, the process of evolution is concerned with the “appearance and maintenance” of new characters. Morgan did not attempt to explain the origin of sex, yet emphasized that recombination would result in the building of new combinations of traits and result in increased variability within the population.

Herman J. Muller was one of the first to offer an explanation of the ubiquity of sex in biology [146]. With Fisher, Muller argued that the advantage of sexual recombination is that adaptive mutations occurring in distinct lineages can be combined within a new lineage. Without recombination, the lineages compete until one wins out or gene pool equilibrium is reached. The selective advantage would ultimately and indirectly select for genes that promote sex and recombination. This is known as the Fisher-Muller hypothesis [147].

Muller worked out the quantitative theory of this hypothesis in two papers decades later [148, 149]. He also argued that recombination alone could be responsible for the combining of non-advantageous gene values into combinations resulting in positive advantages. Muller stated that every population contains such deleterious mutations. This fact is not a problem in sexual populations as recombination has the ability to remove such mutations from subsequent generations. Asexual populations do not have this capability and deleterious mutations build up over time in a step wise fashion and there is no turning back (without unlikely reverse mutations).

This concept is known today as Muller’s Rachet, and the effect is most pronounced in small populations and of course varies with mutation rate. The stochastic effects of Muller’s ideas would be worked out later and spawn many studies in theoretical genetics. Muller also postulated that most speciation comes in the form of geographically isolated gene pools. These gene pools would evolve different genetic mechanisms for accomplishing the same goal. Yet in hybrids the different geneotype to phenotype mapping would result in discordant structures. Known as Dobzhansky-Muller theory it is partially confirmed by experimental hybrids of related species [150, 147].

Wright's seminal 1932 paper [151] on the comparative roles of mutation, selection, inbreeding and cross breeding in evolution was among the first to quantitatively analyze the subject from a population genetics perspective. Among other insights, he discussed the possibility-space of combinations of genes. When correlated with fitness this would form a fitness landscape upon which evolution acts. A population would move up the nearest peak of this landscape via the action of selection of highly fit individuals. See Figure 3.4 for Wright's depiction of a landscape. He notes that with each additional loci added to the analysis, the number of such peaks and valleys magnify. To be effective on such a field, there "must be some trial and error process on a grand scale" and that the population must not be under the strict control of natural selection to be effective.

When a large population with freely interbreeding members is evolving in this environment, the population ends up operating on only one peak in the landscape. Movement in general is slow, migrating to another peak would require many simultaneous mutations and is very unlikely. This is in contrast to small populations where inbreeding can force fixation of genes which could move a population down the peak's gradient. When a large population is divided into many smaller ones, each breeding freely within the sub populations and occasionally cross-breeding, a very different dynamic emerges. The more rapid movement of small populations and the effects of cross breeding speed up the overall trial and error process of evolution. See Figure 3.5 for a depiction of movement on a landscape. While the paper has continued to be influential within population genetics, the effect of his ideas in the field of computer simulations of evolution was pronounced. More on this later.

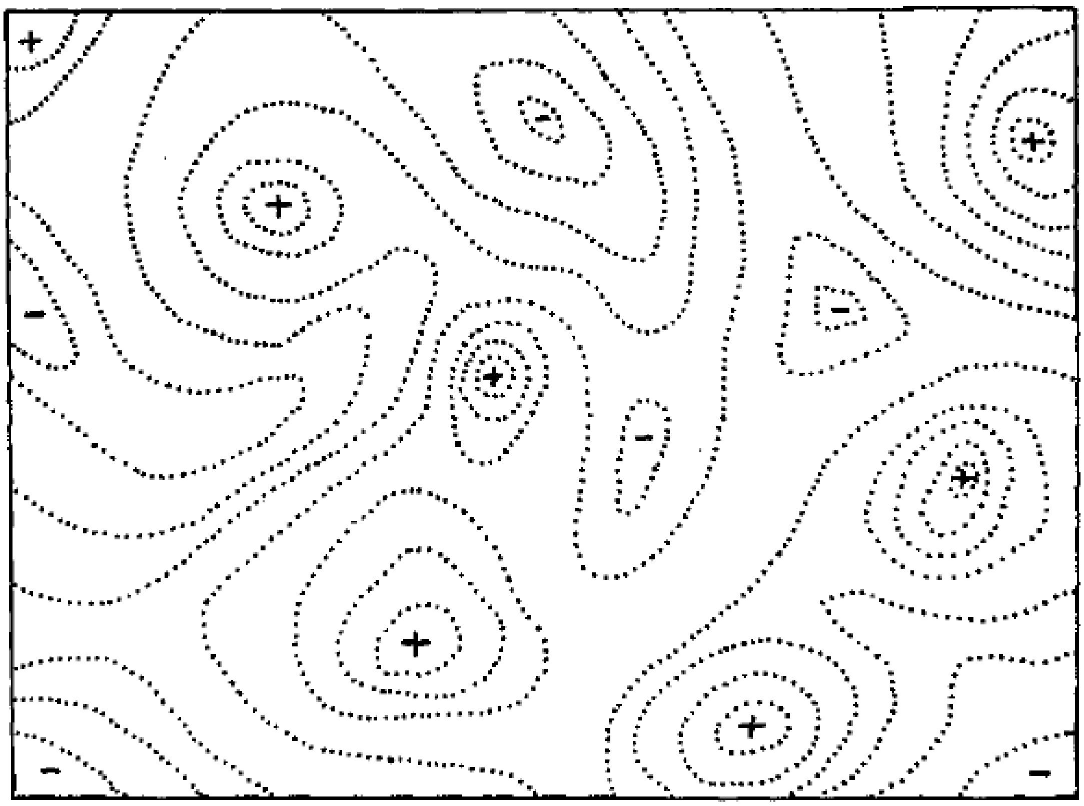
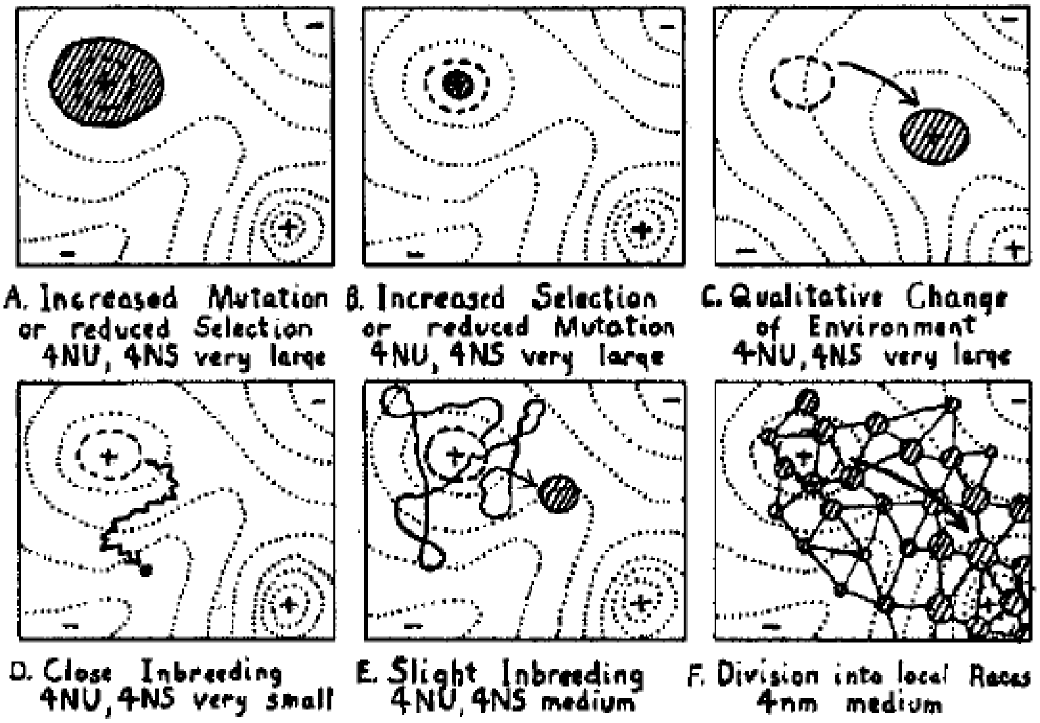


FIGURE 2.—Diagrammatic representation of the field of gene combinations in two dimensions instead of many thousands. Dotted lines represent contours with respect to adaptiveness.

Figure 3.4: Wright's fitness landscape.
From *The Roles of Mutation, Inbreeding, Crossbreeding and Selection in Evolution* [151].



A. Increased Mutation or reduced Selection 4NU, 4NS very large
B. Increased Selection or reduced Mutation 4NU, 4NS very large
C. Qualitative Change of Environment 4NU, 4NS very large
D. Close Inbreeding 4NU, 4NS very small
E. Slight Inbreeding 4NU, 4NS medium
F. Division into local Races 4nm medium

FIGURE 4.—Field of gene combinations occupied by a population within the general field of possible combinations. Type of history under specified conditions indicated by relation to initial field (heavy broken contour) and arrow.

Figure 3.5: Movement on Wright's fitness landscape.
From *The Roles of Mutation, Inbreeding, Crossbreeding and Selection in Evolution* [151].

During the lead up to the modern synthesis, the explorations of experimentalists and discoveries of gene linkages had a profound effect on the thoughts around the benefits of recombination and sexual reproduction. Of course the naturalists already knew of the ubiquity of sex in nature and this delay in consensus was a primary reason for the controversies pre-synthesis. In 1942 Huxely [119] summed up the new consensus on recombination. While stating that mutation is the ultimate source of the truly new in evolution, recombination of genes accounts for the great majority of differences between parents and offspring, and is the source of some gene linked novelty when isolated populations cross-breed.

Simpson concurred with this general assessment [152]. He did carve out an exception stating that recombination and selection, without mutation, were capable of carrying a population far beyond its former range of variation via utilization of the total pool of genetic material within population. Simpson goes on to say that without a variable mutation rate simultaneous mutations are too rare of an event and "such a process [mutation] has played no part whatever in evolution".

Mayr was also not so gracious in his assessments of mutation [109]. He saw mutation as a minimal force during the hostilities of the neutralist-selection debate of the 60s and beyond. Yet, by 2001 Mayr states that evolution is a *two-step dance* of variation by recombination and mutation followed by an ordering by selection [153]. While continuing to endorse the view that recombination is primary, Mayr's two-step dance seems indistinguishable to this author of many early abstract views on the evolutionary such as Morgan's 'duality' theory of 1903 [99]. His theory was stated many years prior to Morgan's conversion to recombination advocacy and acceptance of genes-on-chromosome facts.

In 1965 James Crow and Kimura published a paper with a model of sexual versus asexual populations [154]. The model contained no genetic drift and had a finite population. Their conclusions showed that a population undergoing sexual recombination would more rapidly combine advantageous mutations, and that the advantage scaled with population size and rate of beneficial mutation and low selective advantage. This conclusion confirmed Muller's Ratchet. However, if the mutations were individually deleterious yet collectively advantageous, i.e. epistatic, then recombination was a disadvantage. [2]

John Maynard Smith viewed those conclusions as wrong and responded to this with his own model and a counter example. His assumptions were slightly different, modeling an infinite population and (unlike Crow-Kimura) allowing mutations to re-occur without limit. Maynard Smith used a derivation from the equilibrium principle and concluded that recombination offered no advantage over mutation [155]. He did not address the epistatic claims. Crow and Kimura [156] responded that Maynard Smith's counter-example assumed a particular distribution of gene frequencies and that Muller's Ratchet still holds in general. Maynard Smith conceded this point then created a new model using drift analysis and concluded that recombination would in fact accelerate evolution. He also offered up a hypothesis that sexual reproduction is most valuable when populations that have been evolving individually suddenly merge, or when the environment changes enabling faster adaptation [157].

Hill and Robertson, in 1966, cite an effect combining drift, selection and recombination [158]. In a finite population undergoing recombination and selection, random linkage disequilibrium will be created. This disequilibrium may or may not be advantageous, yet the creation itself slows down evolutionary progress. Recombination breaks the disequilibrium, which then allows selection to act to propagate advantageous loci values in the population. Importantly, this effect appears to happen

---

[2] While the EA community does not widely cite this result, its ideas are influential and were either taken directly or rediscovered.

independently of the fitness linkage between loci [79]. Recombination acts to achieve a higher average equilibrium fitness.

Another important effect of recombination referred to as *hitchhiking* was first discussed in 1974 by Maynard Smith and Haigh [159]. Due to the inverse proximity bias effect outlined by Morgan [105], mutant genes that are neutral or weakly deleterious yet are near beneficial mutant genes may spread through the gene pool via this proximity. This results in a change in the expected gene frequencies due to selection alone or the expected frequency of that gene were it in a different location. A recent survey paper on hitchhiking by Barton is recommended reading [160].

Felsenstein surveyed recombination effects [79] and cites several instances of debate and additional models by other researchers. The results are mixed with some concluding that recombination is beneficial and some not. Felsenstein points out that initial conditions matter a great deal. If the initial number of beneficial mutations is high enough, then recombination serves to preserve diversity, otherwise it confers little advantage.

Felsenstein summarized that when a finite population is assumed in general models, the population benefited from recombination. Models that assumed infinite populations found no benefit. An exception occurs with epistasis and linkage disequilibrium in infinite population modes where there is an advantage to recombination. Felsenstein also conducted one of the first computer studies on recombination and confirms the above summary as well as the basic Fisher-Muller hypothesis.

More recently, Otto and Barton in 1997 constructed an artificial model consisting of multiple gene loci and an extra gene that contributed nothing to fitness yet modified the recombination rates between the other loci. The hypothesis is that the system will evolve a preference for stronger recombination as it will allow breaking of linkage, subsequent speedup in evolution and a higher equilibrium fitness. Otto and Barton derive analytic expressions as well as conduct simulations validating this hypothesis. They conclude that there is always a selection for increased recombination rates, and that a little recombination seems to be enough to achieve the breaking of linkage. This holds even in the face of epistasis between loci where individually some loci values are negative, yet in concert they are positive.

The above slice of history is but one example of the many debates on the utility of sex and recombination. In nature, sexual recombination has many real costs, it is slow, can endanger the organisms involved, spread disease or parasites etc. Many previous theoretical models taking these and other costs into account tend to predict that sexual recombination should be rare and useful only in restricted circumstances. This obviously flies in the face of observed biology [161].

Models predict that sex should have no advantage in populations with allele frequencies at the selection-expected frequencies, meaning no linkage disequilibrium. Additional variation within a population undergoing recombination is not always favorable, it can have a negative selective effect. Recombination itself is also essentially a long-term bet and not necessarily a short-term benefit as it can disrupt a fit genome from one participant.

Otto and Lenormand recently addressed this paradox and the issues mentioned [161]. They note that a main issue with population genetics models is that those that do not consider stochastic effects are the easiest to analyze yet are known to predict different outcomes that harder to analyze stochastic models. Otto and Lenormand then demonstrate that it is possible to construct a model incorporating genetic drift that will evolve towards greater utilization of sexual recombination, even in the face of positive epistasis.

3.12 Punctuated Equilibria

In 1972 Eldredge and Gould introduced their theory of *punctuated equilibria* [162] wherein they argued that the fossil record shows evidence for long periods of stasis with intermittent bursts of evolutionary change. This was a radical idea in evolutionary science at the time and spurred much debate on the points large and small of their theory. The debate with Richard Dawkins would be long term [163], and Daniel Dennet accused Gould of being too conservative in advocating the idea [164]. Gould responded among other places in [165] arguing that a form of Darwinian Fundamentalism was emerging to reassert the primacy of gradualism and selection in evolution.

To an extent this idea was anticipated by Goldschmidt [166] and Simpson [120] and Gould credits both [167, 168]. Goldschmidt termed his idea “hopeful monsters”. He postulated that minor gradual changes of microevolution could not create the macroevolutionary changes observed in nature. Simpson’s theory of quantum evolution argued that the rate of genetic change of population groups varied widely and leaned on Wright’s random genetic drift ideas. He argued that evolution would proceed in *inadaptive phases* where genetic drift moved sub-populations about Wright’s adaptive fitness landscape, followed by an *adaptive phase* driving the sub-population to a fitness peak.

3.13 Statistical versus Dynamical Theories

There are two common conceptions of evolutionary theory when modeling it. One is the dynamical theory of forces, where natural selection, recombination, mutation, random drift and migration act upon a population thus changing their trait or gene frequencies. Loosely speaking the dynamical theories seek to model the exact nature of evolution, much as a physical model of atomic particles would seek to model the exact dynamics of their interaction. The second theory on modeling is one of statistics. This conception is much like the statistical mechanics models of thermodynamics, where for instance probability theory is used to model the aggregate behavior of a large number of atomic particles. The statistical view models the effects (rather than the causes) of selection, drift, mutation, etc.

These alternate models are often presented in both introductory population genetics texts such as Hartl [169] and graduate versions like Ewens [170] with little controversy. However, within the philosophy of science community a recent debate on the comparative correctness of these interpretations of evolution is summarized in [171, 172, 173]. [3]

3.14 Overconfidence in Evolutionary Theory

Many observers continue to remark on the seeming over-confidence that has afflicted contemporary descriptions of evolutionary theory. Gould and Lewontin [73] critique the so called *adaptionist programme* as a collection of post-hoc speculative tales that fail to separate origins versus present utility of a particular trait.

Reid [174] and Løvtrup [175] both review evolutionary theory from a skeptical perspective and illuminate many examples of problematic “heuristic logic”, general overconfidence in neo-Darwinism as well as many examples of shortcomings in the structures of the theory. Shapiro [176] cites four areas of modern genetics where the tenets of gradualism in evolution failed to explain many of the interconnected genome structures found via modern gene sequencing. He also notes that cells have

---

[3] To this author they read as a strange and irrelevant side-show to the science itself.

multiple error-correcting structures present to prevent the kind of random mutations that gradualism asserts as the source of innovation. Mochizuki et al. have recently shown small sequences of RNA function as error correctors during conjugation of protozoa [177].

Taken in total, the neo-Darwinian gradualist framework of evolution is problematic for the theory of evolution. Not only is the overall probability of a sequence of isolated adaptive mutations very small, new genetic evidence of error correction implies that many mutations that may be small steps towards an adaptive change might be discarded. Cynically, large changes may have to wait on Goldschmidt's hopeful monsters for populations to make substantial genetic jumps. Obviously in the long sweep of evolutionary time beneficial changes will take place, one only has to look at the fossil record to see this. Yet, it is obvious that the theory, possibly the crowning achievement of human science, needs additional work to be more explanatory and accurately encapsulate what actually happens in detail within biological evolution.

3.15 Current Status of the Synthesis

Reverse transcription, retrogenes and epigenetics are forcing a revision of the modern synthesis. Essentially, there is mounting evidence of much faster genome change being passed in one generation. While details of these recent developments are beyond the scope of this work, it appears that a new Lamarckian framework is being discovered in biology. The reader should consult Steel et al. [91], Jablonka and Lamb [178] and consult ongoing developments in epigenetics.

3.16 Relevance to Evolutionary Algorithms

The reader might wonder what the relevance of this history to evolutionary algorithms and its theory. As we'll see in the following chapter with a review of EA theory there are of course many similarities between the topics. This author's opinion is that one can better understand the emergence of the EA, claims made about its efficacy for problem solving, debates about EA operators and the various competing models of EAs if they are viewed in the context of the larger work on evolution. I view the EA as a branch-point in the larger development of evolutionary theory. Time and again the EA literature invokes the larger evolutionary theory results to motivate aspects of study, argue points of view and derive mathematical models of artificial evolution.

CHAPTER 4

REVIEW OF EVOLUTIONARY ALGORITHM THEORY

> Variation in fact is Evolution. The readiest way then of studying the problem of Evolution is to study the facts of Variation.
>
> William Bateson [90]

4.1 Introduction to EA Theory

The following sections will give a brief overview of the Theory of Evolutionary Algorithms. EA theory has a fairly active, but small, research group. The initial *Foundations of Genetic Algorithms* in 1990 was among the first times an organized gathering of EA theory researchers was held. The community is focused primarily on quantitative and qualitative illuminations of how EAs function, the effects of EA operators and the strengths or limitations of the EA on particular functions. Much of the roots of EA theory come from the wider evolution theory literature, particularly population genetics.

4.1.1 Theoretical Population Genetics

Much of theoretical population genetics applies to evolutionary algorithms in general. A typical introduction to the topic would encompass equilibrium distributions, haploid versus diploid genotype model differences, random genetic drift, population growth, basic mutation and recombination models and Fischer's Fundamental Theorem of Natural Selection. More sophisticated models combine mutation and selection, analyze different selection models, examine epistasis, linkage disequilibrium as well as incorporate recombination into various models.

The comprehensive *The Mathematical Theory of Selection, Recombination and Mutation* by Reinhard Bürger [179] is an excellent review of this area. It covers many of the models and controversial ideas in the development of the theories of biological evolution (see previous chapter). Alternative references are the concise introduction by Gillespie [180] and the introductory textbook by Ewens [170]. The interested reader is encouraged to review the literature for far more detail than is presented below.

The Hardy-Weinberg law gives the relative allele value frequencies for diploid genes under the conditions of random mating excluding forces such as selection, mutation, migration and drift for very large (or infinite) populations. Recall that diploid genes are composed of two alleles at each loci whereas haploid genes have only one allele per loci. Assume binary allele values and let p be the frequency of bit value 1, q be the frequency of bit value 0 and $p + q = 1$. Hardy-Weinberg states:

$$freq(11) = p^2,\ freq(01, 10) = 2pq,\ freq(00) = q^2 \tag{4.1}$$

The effect of Hardy-Weinberg is that sexual shuffling of the genes does not affect the frequencies of values in the population at large. Note that the conditions for Hardy-Weinberg are broken under biased selection, small populations, genetic drift, population migration or nonrandom mating situations. These effects are known to happen in actual biologic populations. Yet Hardy-Weinberg equilibrium can be used as a baseline upon which to measure the effects of these factors, analyzing the deviation caused by random genetic drift for instance.

An equilibrium state is known as the fixed-point or stationary-point of a dynamic system. When the system is in this state it stays there. Stable equilibrium states are those that under small perturbations of state will return to the fixed-point. The size of the perturbation under which this is true is dependent on the so called *basin of attraction* of the fixed point. Unstable fixed-points are those that under small perturbations will not return to the fixed point. Saddle-points (a type of unstable fixed point) are those that have attracting and repelling regions depending on the direction of the perturbations. Other types of equilibrium states are possible as well. For a more complete discussion of discrete and continuous dynamic systems see [181, 182].

Altenberg [183] notes that in evolutionary computation the semantic meaning of fitness changed as it emigrated from population genetics. In genetics, genotypic fitness or value in general refers to the rate at which a given type is sampled to contribute to the subsequent population. As such the fitness coefficient "lumps together all the disparate influences from different traits, intraspecific competition, and environmental interaction that produce it" [183]. This includes the effects of selection. This contrasts to the meaning in EC, where fitness corresponds to the values given by the objective function(s) used in the EA. In this chapter the semantic meaning will be explicitly called out as necessary.

The notation in Table 4.1 is used for the remainder of the population genetics material. Note that population genetic notation varies and Altenberg's version is closer to that of EA literature than for instance Bürger's notation. Let the term *chromosome* be interpreted a specific instance of a sequence of allele values for the genetic loci, ie the genotype. Treat it as synonymous with the EC concept of an individual instance of the search space, a binary string.

The 'genotypic value' of an individual is the value of the effect of that individual's genes tied to a particular trait being measured. For Bürger fitness w is a

Table 4.1: Population Genetics Notation of Altenberg [183] and Bürger [179]

Altenberg	
$p(\mathbf{x})$	frequency of chromosome $\mathbf{x}$ in the current population,
$p(\mathbf{x})'$	frequency of chromosome $\mathbf{x}$ in the next population,
$p(x_i)$	frequency of allele value x_i in a population,
$T(\mathbf{x} \leftarrow \mathbf{y}, \mathbf{z})$	The transmission function defining the probability that offspring genotype $\mathbf{x}$ produced by parents $\mathbf{y}$ and $\mathbf{z}$ as a result of the application of genetic operators,
$w(\mathbf{x})$	fitness of chromosome, $\mathbf{x}$
$\overline{w} = \sum_x w(x)p(x)$	Mean fitness of the population,
Bürger	
G	genotypic value of an individual,
$\overline{G}$	mean genotypic value in the population,
γ	selective breeding value of an individual,
$\Delta\pi$	allele frequency changes due to natural selection,
$Cov_A(G, w)$	additive covariance of genotypic value and fitness,

joint measure of reproductive success and survival [1]. This contrasts with Altenberg's re-interpretation of it as standard EC fitness. Let $Cov_A(G, w)$, additive genetic covariance, be defined as below.

$$Cov_A(G, w) = 2\overline{w}\gamma^T \Delta\pi \tag{4.2}$$

Here the canonical GA is defined in population genetics terms. It comprises the iteration of selection, random mating and offspring production against the current population forming the next. An important note here is that the model is a so-called *infinite population model*, meaning that the model operates on a population distribution over all possible individual genetic representations. This is common in population genetics [179] and will be a crucial attribute of some EC models.

$$p(\mathbf{x})' = \sum_{y,z \in S} T(\mathbf{x} \leftarrow \mathbf{y}, \mathbf{z}) \frac{w(\mathbf{y})w(\mathbf{z})}{\overline{w}^2} p(\mathbf{y})p(\mathbf{z}) \tag{4.3}$$

This equation defines a dynamical system and was first put in this form by Lewontin [184] and Slatkin [185] in 1970, though Alternber's notation is used here. This model is well cited in the population genetics literature. It was independently derived within the EC community by Vose [186] and Vose and Liepens [187]. Full details on that model are deferred. Stephens and Waelbroeck [188, 189] have also derived another version of this system. First a model of a population evolving without genetic operators such that $T(\mathbf{x} \leftarrow \mathbf{y}, \mathbf{z})$ is defined. Note that δ is the Kronecker delta.

$$T(\mathbf{x} \leftarrow \mathbf{y}, \mathbf{z}) = \left[\delta(\mathbf{x}, \mathbf{y}) + \delta(\mathbf{x}, \mathbf{z})\right]/2 \tag{4.4}$$

---

[1] Bürger [179] actually uses W, usage of w is a concession to Altenberg's notation.

Fisher's so called Fundamental Theorem of Natural Selection (FTNS) was first detailed in 1930 [112]. The FNTS is worth mentioning here as it is so elementary that it applies to evolutionary computation in general under proportional selection. The version below is presented in Altenberg's notation.

Theorem 1 (Fundamental Theorem of Natural Selection)**.** *The rate of increase of the mean population fitness via the process of natural selection is equal to the population's genetic variance in fitness at that time. [112, 183, 179].*

$$\Delta\overline{w} = \overline{w}\, Var\left(w(\mathbf{x})/\overline{w}\right) \tag{4.5}$$

Proof. See Price [190] or Altenberg [183]. □

Note that it has no recombination or mutation operator in use, only proportionate selection operator acts upon the population. Typical of Fischer's results, it was both influential and open to interpretation. Fisher gave no specific equation or proof and only hinted at a derivation of it while noting the "ease of its interpretation". It would take thirty years for the first formal proofs of the theorem under general dynamic models of evolution. Crow [191] reviews the debates spawned and states that it "has had an exegesis worthy of the most diligent Talmudic scholars." Note that Fisher did not agree with a general view that evolution is directed to maximize the mean fitness of the population [179].

Ewens developed formal interpretations and proved the result under various discrete models in a series of papers [192, 193, 194]. Kimura developed a refinement under continuous-time models where evolution moves the population upon a fitness gradient

[195]. Price's famous 1972 paper breaks down Fisher's description and assumptions to arrive at a formal interpretation followed by a proof.

In conjunction with the Hardy-Weinberg model, the FNTS implies that under selection the mean fitness of a population is non-decreasing when the population is not at an equilibrium state. Before discussing conditions under which the FNTS is not true, a few more results and concepts need illumination.

Geiringer's Theorem of 1944 [196] applies to situations where recombination is applied repeatedly under random mating and without mutation. It states that the expected proportion of a genotype $\mathbf{x}$ in a population will approach the product of the individual gene loci frequencies in the initial population. Let $p(\mathbf{x})^{(t)}$ be the proportion of a given genotype in a population at time t and let $p(\mathbf{x_i})^{(0)}$ be the proportion of a given allele value in a genotype of length n loci in the initial population.

Theorem 2 (Geiringer's Theorem)**.** *For repeated applications of the recombination operator and random selection, the expected proportion of a genotype S in a population will approach the product of the individual gene loci frequencies in the initial population.*

$$\lim_{t\to\infty} p(\mathbf{x})^{(t)} = \prod_{i}^{n} p(\mathbf{x_i})^{(0)} \tag{4.6}$$

Proof. See Geiringer 1944 [196]. □

This condition is called linkage equilibrium or Robbin's proportions and is an important concept in EC and population genetics. Genes at two different loci are in linkage equilibrium when their joint probability is equal to the product of their independent probabilities within the population. Assuming binary alleles, let p_{ij} be the frequency of (for instance) the gene-pair 01, where either of the two genes may exist at any loci i, j in the genome. Let p_{x_i} be the frequency of some allele value x_i in the entire population. Linkage equilibrium is simply stated as:

$$p_{x_{ij}} = p_{x_i} p_{x_j} \tag{4.7}$$

When this condition is false, then the population is said to be in linkage disequilibrium. Note when the only operator is recombination and the population is in linkage equilibrium it is also by definition at an equilibrium state of the system. Absent other factors, the recombination operator's effect is to decrease linkage disequilibrium. Within the EC literature Booker [197] notes that the conclusion for EAs is that the speed that a given recombination operator pushes a population towards linkage equilibrium is an important differentiator among algorithms.

When selection (ie non-random mating) is added there can exist equilibrium states that are not in linkage equilibrium. Recombination also may introduce periodic orbits, or limit cycles, where the state is not a stationary distribution of the population but a sequence of distributions that is connected in a cycle [179]. Akin proved they exist for continuous time systems [198] and Hastings showed they exist for discrete models [199].

In EC, Wright and co-authors showed cyclic and chaotic behavior for binary GAs [200, 201]. Chakraborty and Mühlenbein [202] cover applications of Geiringer to EC. Poli, Stephens, Wright and Rowe [203] as well as Mitavskiy and Rowe [204, 205] extend it for various applications to EAs.

Thus Fisher's FTNS is not true in general for models including recombination. When the population is in linkage disequilibrium, it can not be said that fitness is non-decreasing. In addition, when equilibrium states exist that are asymptotically stable, such as those where mutation is used, there must exist trajectories to that state where fitness decreases [179].

The FTNS predicts the rate of change of mean population fitness, the variance of which is driven to zero under selection. Selection acts indirectly on individual quantitative characters via their fitness. Thus population genetics is often interested in the rate of change of an individual character under selection.

Robertson's 1966 Secondary Theorem of Natural Selection gives this rate [206, 207]. Let G be the genotypic value of a characteristic.

Theorem 3 (Secondary Theorem of Natural Selection)**.** *The rate of change of an individual characteristic is equal to the additive covariance between the genetic value of the character and its fitness. [179, 208].*

$$\Delta\overline{G} = Cov_A(G, w)/\overline{w} \tag{4.8}$$

Proof. See Robertson 1966 [206]. □

While the above are influential results, both are special cases of a more powerful theorem by Price [209]. Price's theorem shows that the selective mechanism directing population change is the covariance between traits from an offspring and its parents. As noted by Rice [210], the mathematical formulation is "an exact characterization of a relationship that must hold between phenotype, fitness, selection, and evolution." Let ΔG be the change in genotypic value from one generation to the next.

Theorem 4 (Price's Theorem)**.** *The one generation change in the frequency of an individual characteristic is equal to the additive covariance between the genetic value* *of the character and its fitness modified by the expected genetic changes from parent to offspring [179, 208].*

$$\Delta\overline{G} = Cov_A(G, w)/\overline{w} + E\left[w\Delta G\right]/\overline{w} \tag{4.9}$$

Proof. See Price 1972 [190]. □

Epistasis is another concept important in both population genetics and the theory of EC. Bürger [179] defines it negatively, saying that the absence of epistasis is when a multi-locus genotype's contribution to some character (ie fitness) is representable as a function of independent single-loci characters. Anything else is epistasis.

Thus genetic interaction and epistasis are different facets of the same idea. One example in biology are modifier genes. If a gene at loci i modifies the phenotypic behavior of a gene at loci j, gene i is said to modify gene j. The interaction here is epistasic in that it affects the equilibrium distribution of the two genes. Another more basic example is where the combination allele values at i and j interact to give greater differential fitness (positive or negative) than other combinations of allele values at those loci, thus their contributions to fitness are not independent of one another.

Genetic drift and fixation are common elements of an introduction to population genetics. Genetic drift was introduced as a concept in the previous chapter. A common model of drift is Wright-Fisher [169]. Given a single allele with binary values (A and B in pop. genetics), and a population of N reproducing individuals let p be the frequency of value A in the population and q be the frequency of B. For diploid populations, the probability of obtaining k copies of A in the next generation is:

$$\binom{2N}{k} p^k q^{2N-k} \tag{4.10}$$

The next logical question to ask is what is the waiting time for a particular value to be fixated or lost completely in the population via the effects of drift? The following formulas give the answers [169]. Note that for haploid populations like the EA, replace $2N$ with N as their is only one gene per individual. The effects of drift are inversely proportional to population size, the bigger the population the less it drifts.

$$\bar{T}_{Fixed} = \frac{4N(1-p)ln(1-p)}{p} \approx 4N \tag{4.11}$$

$$\bar{T}_{Lost} = \frac{-4Np}{1-p}\ln(p) \approx 2N\ln(2N) \tag{4.12}$$

Even a modest treatment of population genetics is far beyond the scope of this work. These models can tend to make different assumptions than those relevant to EAs because they are attempting to model real biology. The true concordance between EA theory and mathematical population genetics is largely unexplored, though much of EA theory borrows heavily from this discipline. The work of Lee Altenberg is a notable instance of a population geneticist working in evolutionary computation [211, 212, 213].

4.1.2 Reductionist Approaches

Reductionist approaches in population genetics are common, where a component of evolution (selection, recombination, mutation, etc) is removed and the system analyzed. Some EA theory work takes this approach. Goldberg and Deb [214] produced analytical results for takeover-time. Takeover time is defined as the number of generations needed for a highly fit string to takeover the population under various types of selection. This is similar to the fixation-time analysis of population genetics [215], the computation of the expected generation count until a novel mutation is fixed in the entire population.

Thierens and Goldberg analyzed the mixing and convergence time of recombination [216, 217] as well as convergence models under various selection schemes [218]. An interesting result is that the mixing time, or time to recombine and propagate novel building blocks, must be less than the takeover-time to ensure proper trials and combinations of those blocks.

Bäck analyzed selection pressure of populations under various schemes [219]. In [220], Rudolph utilized Markov chains to analyze convergence times of non-elitist selection schemes. Mühlenbein and Schlierkamp-Voosen did an analysis of different selection schemes and the respective convergence effects with recombination and mutation [221]. Blickle and Thiele compared a set of selection schemes in [222] and analyzed tournament selection [223].

Mahfoud examined effects of genetic drift under niching and sharing diversity schemes [224]. Asoh and Mühlenbein did a similar analysis of drift without selection [225]. Hesser and Männer [226] and later Bäck [227] sought optimal mutation rates for specific situations. Tate and Smith investigated convergence and fixation of alleles and the relationship with mutation rates [228].

In general, the reductionist models of EAs have much in common with the reductionist population genetics models from biology, though those models tend to be tailored or constrained to answer questions useful to biologists [179, 229, 180].

Analysis of the representation, or encoding of the problem to a real or binary string, that the GA operators acts upon has been an area of reductionist study unique to EAs. The book by Rothlauf [230] is a comprehensive introduction to this area.

One example of this area is new work by Rowe, Vose and Wright [231]. They introduced formal theory for looking at the compatibility of a given representation and the genetic operators applied. This is of some importance as many commonly applied GA operators are not properly invariant w.r.t the underlying problem representation. As such, a simple reordering of the meaning of each allele can result in different algorithm dynamics.

4.1.3 Schema Theorem and Building Block Hypothesis

After formulating the classical GA, Holland crafted the first attempt to explain the behavior and reasons for success of the GA called the *Schema Theorem* [17]. First a definition of *schema* is needed.

Definition of a Schema. *A binary schema is a subset of the binary string space* 2^l *where all strings share a set of defined values.*

Example: *For four bit strings, the binary string space is* $2^4 = 16$ *and the schema* $(\mathit{1{*}{*}1})$ *defines the subset* $\{(\mathit{1\ 0\ 0\ 1}), (\mathit{1\ 0\ 1\ 1}), (\mathit{1\ 1\ 0\ 1}), (\mathit{1\ 1\ 1\ 1})\}$

Schemata (plural of schema) define hyperplanes in the l (used synon) dimensional space of the bit-string. [2] Let $\mathcal{S}$ be a given schema and $N(\mathcal{S}, t)$ be the frequency of that schema in the population. Let $\eta(\mathcal{S}, t)$ be the probability that the GA destroys a given schema, and let $\sigma(\mathcal{S}, t)$ be the probability of the GA creating the schema. A generic version of the theorem is below.

Theorem 5 (The Schema Theorem)**.**

$$E[N(\mathcal{S}, t+1)] \geq 1 - \eta(\mathcal{S}, t)\sigma(\mathcal{S}, t)N(\mathcal{S}, t) \tag{4.13}$$

---

[2] The symbols l, n, and d are used synonymously in this work to reference the length of the EA bit-string.

Proof. See Holland 1975 [17]. □

As a proven theorem it is true, interpreting it theorem has been a source of controversy. Holland's concept of *implicit parallelism* argues that the GA is evaluating a large number of short schemata (relative to the length of the genome) across the entire population's genome with a fairly small set of trials. The central idea of implicit parallelism is that the operators of proportional selection, mutation and crossover are processing the schemata independently. Schemata with sufficiently above average fitness will increase in frequency and increase the expected fitness of the next population. This assumes that the propensity to create such "highly fit" schemata is greater than the GAs propensity to destroy them. Altenberg has shown that the schema theorem is a special case of Price's theorem [183].

Goldberg's *Building Block Hypothesis* (BBH) [232] is related to Holland's work. Goldberg argues that the GA's *modus operandi* is a situation where short low-order (fewer defined bits) and highly-fit schemata, or *building blocks*, combine through crossover to progressively construct better solutions. Thornton [233] among others have pointed out that the BBH seems at variance with the Schema theorem due to its expectation of acquiring longer schemata in late generations. They point out that long sequences of alleles in the genome are more likely to be destroyed by crossover than shorter sequences.

One common complaint is that the schema theorem is a kind of tautology [234] and moreover is not predictive of GA behavior. It's important to note that there is a distinction between the theorem itself and GA analysis and theory based upon its ideas.

The schema theory and BBH are often critiqued in the same papers and many objections to the basic BBH have surfaced. Vose, Radliffe, Rudolph, Grefenstette and many others have offered their critiques. Reeves and Rowe [40] devote a sizeable chapter detailing the debate of both ideas. They claim the schema based theory was "comprehensively eviscerated" by a series of papers via various counter-examples as well as proving that it does not hold for some operators and fitness landscapes. Mitchell [53], and Rudolph [235] also give their versions of events. Notably, Vose [236] showed that minor changes in mutation rate can lead to wildly divergent results, a classic demonstration of the butterfly effect in dynamical systems.

In this author's view the critiques while valid, are somewhat unfair. It has been said that Price's equation itself is a mathematical tautology whose power lies in its ability to illuminate the effects of the dependence between the variables of evolution [237]. In that sense the Schema Theorem may be a useful artifice.

Both the Schema Theorem and the BBH are good tools for thinking about how the GA might work. The original forms do not predict either the dynamical or limit behavior of EAs, yet these ideas still remain very intuitive. Altenberg and Radcliffe [238, 239] have put the schema theorem into better perspective. Watson has reinvigorated the basic concepts of the BBH via his *compositional evolution* [240]. In work with Jansen [241] a reconstructed Royal Road was proven to need building blocks exploited with crossover to optimize the function. Riccardo Poli has contributed reformulations of the schema theorem [242, 243].

Jones [244] compared crossover to a more disruptive macromutation operator in hillclimbing conditions. His macromutation operator consists of "headless chicken crossover", wherein rather than crossing-over the two parents to produce two child strings it instead performs recombination against two random individuals (one for each parent) to produce new child strings. This GA variant is then compared to a more standard GA on various functions. For functions with a clear intuitive building block present, the random crossover is not productive. However for functions with no clear building blocks present, the randomized crossover's performance in indistinguishable. When applied to a fully deceptive function, the randomized crossover beats standard crossover. Jones suggests that the obvious conclusion is that crossover may only be really useful for the 'macromutations' it performs when building blocks are absent. He derives a new hill-climber algorithm with a macromutation operator.

Recently, Stephens and co-workers produced results deriving more exact versions of the Schema Theorem and BBH (yet quite different from the originals) that integrate compatibly with other EA theory work [188, 245, 189, 246]. An example usage of these techniques is Lozano et al. [247] where they studied the optimal sizes of building blocks.

There remain many unknowns about how relevant the intuitive explanation of GAs provided by the Schema Theorem and the BBH are to real-world problems.

4.1.4 Markov Models

A popular analytical tool in analyzing the EA is the Markov model that is used in both reductionist and integrated approaches to analyze expected behaviors of EAs. A Markov model, or Markov chain, is a description of a random/stochastic process where each possible configuration of the process, or state, has a probability of transitioning to some other state. A crucial aspect of the model is the Markov property which dictates that the transition probabilities depend only on the current state and not any prior state history.

Markov models enable predictions to be made about the *expected* next state of the process given knowledge of the current state. The number of states are finite and the probabilities of transitioning to a given state in the next step is only dependent on the current state, not any prior states. Markov chains have two types of states, transient and recurrent. Transient states are those with a non-zero probability of never being visited again. Recurrent states are those with a zero never-return probability. A special case of recurrent states is the absorbing state, one with a zero probability for leaving that state.

An equation can be derived which exactly models the GA, given a bit-string chromosome of 2^l possible strings (or states) and a population of one string. Once additional strings are added to the model, crossover can be included. The number of states in an l bit-string Markov GA model with m individuals is defined in Equation 4.14 [40].

$$\binom{2^l + m - 1}{m} \tag{4.14}$$

As an example, a GA with 10 bits and 10 individuals results in $3.0 * 10^{23}$ states. This number rapidly climbs to computationally unmanageable levels as m and l increase. Nevertheless, these reductionist-type models allow granular performance prediction of simple GAs. An example of a GA Markov model for a 2-bit genome is given in Table 4.2 and Figure 4.1.4. This example defines the Markov transition matrix for a GA with a mutation rate of 0.1, no crossover and a single individual population (ie. no selection). The subsequent graphic is a state graph of the Markov chain.

Table 4.2: Example transition matrix of a Markov model.

	`00`	`01`	`10`	`11`
`00`	0.81	0.09	0.09	0.01
`01`	0.09	0.81	0.01	0.09
`10`	0.09	0.01	0.81	0.09
`11`	0.01	0.09	0.09	0.81

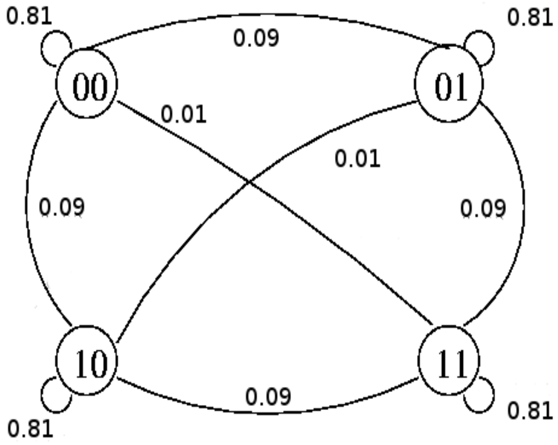


Figure 4.1: Example transition graph of a Markov model.

A very early work on mathematical models of GAs by Bridges and Goldberg [248] models the effects of crossover and selection. Other early uses of Markov chains in EAs are by Eiben et al. [249]; Nix and Vose [250]; and Davis and Principe [251]. Rudolph [235] also uses Markov models of very simple mutation-selection GAs showing various convergence properties. Reeves and Rowe [40] detail Markov models of GAs with selection, mutation and crossover.

4.1.5 Infinite Population and Dynamical Systems Models

The development of infinite population models of evolutionary computation was an important milestone. They are not approximation models, rather they are exact models capable of capturing all dynamics of EC. In this author's view it brought EC theory up to a comparable level as the mathematical theory of evolution.

An infinite population model (IPM) represents the population as a distribution over all possible individuals (all combinations of allele values). For a binary representation this is the total 2^n chromosome space. One iteration of an IPM maps the given population distribution to the exact population distribution of the next generation.

IPMs on biological haploid and diploid genomes have been around for some time and are quite complex [179].

Michael Vose was an early critic of the usefulness of the Schema Theorem. While working with Liepins on more exact models of schema disruption [252] they introduced the first infinite population model of GAs [187]. Further work by Vose [253, 254, 255] was capped off by a full accounting of a new dynamical systems model of GAs in his densely mathematical book *The Simple Genetic Algorithm* [256]. Reeves and Rowe [40] and Rowe [257] give a more gentle, example based introduction to the infinite population models of Vose and co-authors.

When crossover is included, this model is nonlinear and not directly solvable. The tools of nonlinear mathematics from texts like Strogatz [182] and Brin and Stuck [181] can be brought to bear on the Vose infinite population model, however there are many unanswered questions about the Vose model. Note that as a result of the development of the IPM, the similar-looking Markov models of GAs are now considered extensions of rather than distinct from the IPM.

As previously noted, Altenberg adapted elementary population genetics models and introduced them to the EC community in [183]. These so-called *transmission function* models were extended by van Kemenade et al. [258]. They showed models with tournament selection, elitism, steady-state populations as well as the Breeder GA. Population flow diagrams were produced for various deceptive fitness, much like those done when learning dynamical systems from texts such as Strogatz.

Stephens and Waelbroeck [188, 189] introduced another exact IPM and used it to analyze the schemata evolution and building blocks. Stephens and co-workers also developed a "coarse grained" version of their exact model [259, 260, 261], more on this later. Stephens and Poli used these models to research recombination operators [262, 263].

Any references here to these models by their primary author's last name (Vose, Stephens, Altenberg) is shorthand and not intended to slight any of their numerous co-workers. All of these models are exact dynamical systems models utilizing infinite populations, and the various authors have not given the models proper names. Their direct concordance with infinite population models of population genetics is largely unexplored territory. More detail on Vose's model is given later.

4.1.6 Statistical Mechanics

In physics and related disciplines an approach called statistical mechanics is used to model the macroscopic behavior of a complex system of many interacting parts. This contrasts with microscopic approaches where all elements of the system are modeled, for instance a standard markov model of EAs.

The approach has been used by multiple camps within the EA community to model the several variables of the system that most characterize its behavior. Prügel-Bennet, Shapiro, Rattray and Rogers have teamed up on a series of papers on utilizing the method [264, 265, 266, 267, 268].

An example is as follows. Choose fitness as the macroscopic features of analysis and let it be defined as some probability density function representing the mean fitness over all possible runs of the GA. The next step is to derive a set of equations giving the moments of the mean distribution. Using these moments, cumulants of this feature can be predicted as a function of time. For instance, the mean, variance and skew of fitness can be predicted using the derived equations. In this instance it can be thought of as an infinite population model, yet it only models the feature chosen and cannot be used to predict other features, such as the allele value probability for a set of genes for instance. In contrast to the infinite population model, these models are approximations.

A Ph.D. thesis by van Nimwegen used this approach to explore the dynamics of evolution [269]. The particular focus was on epochal dynamics, the observed behavior of punctuated equilibrium in evolutionary processes. He and co-workers looked at the effects of population size and mutation rates on the dynamics of EAs, forming models to predict the most efficient population sizes and mutation rates to solve particular functions as well as the dependence of those parameters on the required number of function evaluations to reach a global optima. van Nimwegen used dynamic systems techniques in concert with statistical mechanics to predict the existence of so called *neutral networks*, areas of the fitness landscape that populations would get stuck in for a time before escaping to higher fitness regions. These neutral networks and the drift within them do bear a similarity in spirit to the drift analysis of Kimura [127]. He derived equations showing the existence of *meta stable* states that influence the GA and passage times through these states.

Unfortunately while this area of study makes good intuitive sense as a tool in understanding GA behavior, it is quite under-utilized outside a small set of authors within the EA community.

4.1.7 Time Complexity Analysis

A major branch of EA theory is analysis of specific EAs and their time complexity to solve optimization problems. This area was recently surveyed by Oliveto et al. [270]. An early use of the new Markov chain models of EAs led to a seminal set of convergence results by Rudolph [235]. A stochastic optimization algorithm is said to converge when it finds the global optimum and holds this solution for all future steps. Ideally, this is done in finite time, preferably in polynomial time to the size of the population size, binary string length or other attribute of the fitness function.

Rudolph showed, via Markov chains, that GAs using selection, mutation and optionally crossover would not converge to global solutions by this definition. Elitist GAs would achieve success. Rudolph also developed criteria for convergence implying that specific analysis would only be required when those criteria are not met. While his results for the first time showed that it was not necessary to use a full Markov chain analysis, simple proofs are possible, the results were so general as to not give specific expected times for the 'first hit' of the solution. As a result, the only upper-bounds possible are the same as for random-search.

Beginning in 1998, Droste, Jansen and Wegener [271, 272] studied the so called $(1+1)$ EA, a single candidate mutation-only elitist algorithm, and produced the first proofs of minimum probabilistic waiting times for simple pseudo-boolean functions. In 1999 [273] and 2001 [274] this type of analysis was extended by Jansen and Wegener to the $(\mu + 1)$ EAs having a population greater than 1.

Results have continued to progress and many tools have been developed [270]. Tail inequalities such as Chernoff bounds and other skills common to randomized algorithm analysis are a necessary common thread [275]. Other techniques include drift analysis [276], potential functions [272, 277], and family-trees [277].

The basic premise of this avenue of research is threefold, first to allow EAs to be analyzed like any other randomized algorithm and thus be accepted more widely as useful computational tools. Secondly to compare and contrast different types of EAs against a common framework. Thirdly, to examine and analyze the EA's behavior against typical structures found in many optimization problems. The approach has also proven quite useful for proving and disproving various conjectures of the EA community, for example, the ability of crossover to optimize some structures much faster than mutation alone. These topics are covered in more detail later.

4.1.8 No Free Lunch Theorems

An important contribution to the theory of EAs and optimization in general is Wolpert and Macready's *No Free Lunch* theorem [278].

No Free Lunch for Optimization. *Given any two optimization algorithms, their performance is exactly equal over the total space of all possible fitness functions to be optimized.*

Reeves and Rowe [40] detail the implications and subsequent research on NFL. At first glance this means that any algorithm is just as good as any other in general. There is much discussion in the GA literature over the effect of NFL. Some argue that it is nearly irrelevant since any practical function worth optimizing contains some structure that a search algorithm can exploit.

The NFL theorem would certainly prohibit any result such as "my adaptive GA with customized operators is provably better than your old rusty classic GA". In practice this prohibition is likely ignorable, since it is demonstrable that some GAs perform better on a class of fitness functions than others. However, practitioners must be careful not to extrapolate the results of their empirical study to hold for all known fitness functions. The NFL shuts the door on the idea that there is a universally superior GA (or superior SLS algorithm) out there to be discovered.

Wolpert also showed a version of the NFL for supervised machine learning in general [279]. In supervised learning, a training set is used to learn a function. This function is validated against a held-out test-set, and the general goal is to minimize the error or misclassification rate. When this classic situation is applied in a noise-free scenario, the NFL theorem applies. This means that one cannot make a priori quality distinctions between the average general performance of two learning algorithms.

Recently Wolpert and MacReady [280] have shown that under co-evolutionary optimization there *are* free lunches. Under a self-play scenario where a set of players generate a champion and subsequently this champion is pitted against antagonists in a multi-player game, it can be shown that some learning algorithms perform better than others on average. The distinction here is that there is no objective function, rather candidate solutions are judged on their performance against other candidates. See [281, 282, 283] for details on co-evolutionary algorithms as even a modest presentation is beyond the scope of this work.

Notably, Droste et al. prove that for *black-box optimization* the NFL does not hold [284]. The black-box situation is one where the class of functions to be considered is restricted to those with short descriptions and can be evaluated in polynomial time. Whitley and Rowe [285] have illustrated situations where the NFL does occur in fitness functions permuted with the grey code. Rowe, Vose and Wright have more recently reinterpreted the NFL [286] as a set-theoretic construct in order to separate the application of the NFL from the symmetries of the theorem itself.

4.2 Crossover versus Mutation

Fogel and Atmar [287] study EAs applied to solving systems of linear equations and conclude that that crossover is of minor importance, mutation alone is sufficient for finding solutions. Eshelman and Schaffer [288] argue via schema processing methods that crossover is effective on mildly deceptive problems.

4.2.1 Hard Functions for Mutation based EAs

This section highlights past research on fitness functions hard to optimize with mutation alone, while the addition of crossover allows effective optimization. First proposed by Mitchell et al. [289], the well known Royal Road (RR) class of fitness functions were designed to demonstrate that the crossover operator in genetic algorithms is essential to optimizing a class of fitness functions. Jones [290] has a complete description of the Royal Roads and its components.

Mitchell et al. also show that for an idealized GA ignoring the effects of hitchhiking, the expected optimization time is $O(2^k \log(n/k))$. Somewhat unexpectedly, follow up experimental studies by Forrest and Mitchell [291, 292] show that some random mutation hill-climbers outperform GAs with crossover. This prompted the same authors to define an open problem [293].

- Define a family of functions and prove that genetic algorithms are essentially better than evolutionary algorithms without crossover.

Prügel-Bennett [294] follows this question by presenting a fitness gradient with a series of barriers. He demonstrates with asymptotic approximations and experiments that a hill-climber is mildly outperformed by GAs without crossover and greatly outperformed by a full GA. Jansen and Wegener [295] proved that the expected optimization time of the well known (1+1) EA on the classic Royal Road function is $O(2^k(n/k)\log(n/k))$ where n is the string length, k is the length of sub elements of the string and $1/n$ is the mutation rate.

The concatenated trap functions of Deb and Goldberg [296] consist of many concatenations of smaller fully deceptive fitness functions. In a fully deceptive function, all points in the space other than the optima give local advice to go in the opposite direction of the optima. They form a composite function of many non-overlapping deceptive functions concatenated together.

Mutation fails to optimize the composite function in reasonable time, while the crossover operator builds short-order sequences of highly fit bits and recombines these sequences to successfully optimize the function. This building block process [296], is the historical rationale for claiming crossover superiority on some functions. Note that if the size of the trap is constant or n^{ε} where $\varepsilon \in (0,1)$ then it has been suggested that these conclusions are false.[3]

While both the concatenated traps and the Royal Road functions were intended to demonstrate the necessity for crossover, the subsequent evidence of hypothesis failure for the RR resulted in an explanation that the functions were separable and non-deceptive. These attributes allowed non-crossover EAs and RMHCs to optimize RR. Addressing this issue, Watson [297, 298] created a hierarchical fitness function called HIFF (hierarchical if and only if) where sub-blocks are interdependent and non-separable as well as being deceptive to the mutation operator. Non-crossover EAs require expected exponential time to optimize the HIFF. Later Dietzfelbinger et al. [299] asymptotically analyzed a recombinative hill-climber on the HIFF function and showed an expected time complexity of $\Theta(n \log n)$.

In [300] Jansen and Wegener introduced $\text{JUMP}_{m,n}$, a fitness function with a mutation frustrating structure. The central feature is a fitness canyon separating a false optimum from the global optimum with an m-bit gap of low fitness ($m < n$). The waiting time for a steady state GA (with uniform crossover) to optimize $\text{JUMP}_{m,n}$ is $O(n^2 \log n)$ steps. For successful optimization, a steady state hill-climber that accepts no fitness decreases [like the (μ+1) EA] must simultaneously mutate m bits to cross the canyon. The waiting time for this event is $O(n^m)$, leading to a polynomially large difference in optimization times.

Jansen and Wegener state [295] that the well known $(1+1)$ EA solves the classic Royal Road function in expected $O(2^k (n/k) \log(n/k))$ steps where n is the string

---

[3] By an anonymous reviewer of the subset of Chapter 5 appearing in PPSN X

length, k is the length of sub-elements of the string and $1/n$ is the mutation rate. This contrasts with the order n/k speed up of the idealized GA's $O(2^k \log(n/k))$ time. As mentioned above, experimental results also showed that various RMHCs, including the $(1 + 1)$ EA, are faster than standard GAs.

Jansen and Wegener [295] continued by introducing Real Royal Road (RRR) functions for both 1-point and uniform crossover. For these functions a steady state GA versus equivalent non-crossover EA are analyzed to produce rigorous time complexity bounds. The GA solves the RRR functions in polynomial expected time, while the EAs take expected exponential time. Their results use a non-constant population of size $s(n)$ that is allowed to vary at a rate different than the string length n.

Storch and Wegener [301] created additional Real Royal Roads for both uniform and one-point crossover for constant population size. They used a population size of two, the smallest population where a crossover operator can function. Using the steady state $(2 + 1)$ GA they proved expected polynomial optimization time, while the $(2 + 1)$ EA takes expected exponential time.

A valid critique of the Real Royal Roads is that they are artificial functions constructed to meet a narrow challenge. Other work has been done on more natural functions. Fischer and Wegener [302] show a specialized GA does far better than the EA for both types of crossover. However, the results are mixed for a standard GA. If a correctly chosen λ is picked, the $(1 + \lambda)$ EA performs well compared to typical GAs. Another result from Sudholt [303] on the Ising family shows polynomial time complexity for the GA versus required expected exponential time of the EA.

Horn et al. [304] proposed a path of 1-bit mutations, or Hamming path, called the Long Path. It was shown in Horn et al. [304] and in Rudolph [235] that commonly known hill-climbing algorithms searching a one-bit neighborhood will take exponential time to optimize a long path of length n. Subsequently Rudolph proved that the $(1{+}1)$

EA, which can flip multiple bits in a search step, can optimize long paths in $O(n^3)$ time.

Garnier and Kallel [305] followed up with a statistical study of the expected first hitting times of Rudolph's variant, the *long k path*. For any k that is not a constant relative to the length of the bit string n, the expected waiting time of a simple mutation only $(1+1)$ EA is exponential.

Recently, Rosenblueth and Stephens [306] have detailed a method of analysis for determining the utility of recombination for a given fitness function. The analysis uses both Stephen's *Building Block Basis* [307, 61] and the Walsh basis [308] methods. They examined two functions, counting-ones (CO) with no epistasis and the needle-in-a-haystack (NEEDLE) with maximal epistasis. Via their methodology it was shown that recombination is useful for the CO, but not for the NEEDLE function. Interestingly when compared in terms of genotype distributions, seeing the benefit or harm of recombination was difficult. However when using the BBB or Walsh methodology the distinction became very clear.

Burjorjee has recently illuminated a new class of problems important to computational genetics (the biological kind!) and demonstrated the effectiveness of the GA on them as well as outlining an alternative view of describing the GAs competency and effectiveness on problems [309].

4.2.2 Hard Functions for Crossover based EAs

Functions hard for crossover yet easier for mutation alone, though rare in EA/GA literature, are not completely new. In Poli et al. [310] the authors introduced a function called OneMix and experimentally showed that it is deceptive to crossover. The fitness function has a distinctive sawtooth pattern with a high frequency oscillation in the fitness space moving alternately from low to high fitness on one side of the unitation fitness space towards the global optimum. The other half of the space has a smooth gradient ascending to a local optimum. The sawtooth area is an instance of a 2-longpath and requires flipping two bits to jump the low fitness areas on the gradient.

They introduce a *low pass filtering theory* for GAs and argue that the crossover operator is unable to discern the high frequency fitness switching and thus the crossover operator over a large population acts as a low-pass filter. The low pass filter averages out the differences between low and high areas of the sawtooth pattern, thus tricking a GA into perceiving a gradient with a much lower slope than the slope drawn only through the high-fitness 2-longpath.

They analyze the function on an infinite population model and validate the model with run-time experiments. The filtering effect of each genetic operator is also postulated with crossover having the most effect. Poli et al. [310] make no effort to use run-time analysis methods to prove any rigorous bounds on the GA versus EA on OneMix.

As an aside, I speculate that the OneMix result described above can be rigorously proven to be true via run time analysis. I also conjecture that small population GAs, such as the (2+1) GA will not be effectively deceived by the sawtooth pattern in the fitness landscape. Additionally this author believes that if a trivial diversity mechanism were added (ensuring that individuals in the population are some distance away from each other in the fitness landscape) then a small population GA will solve the OneMix operator in polynomial time with an exponentially low failure rate.

4.3 Genetic and Population Diversity

There is much accumulated knowledge from the biological community on the importance of genetic and population diversity. There are passive attempts like Davis' no-duplicates policy [311]. Incest prohibition schemes have been invented by Eshelman and Schaffer [312] and Craighurst and Martin [65]. Less passive attempts like the adaptive mutation schemes of Arnone et al. [313], Xu and Vukovich [314], and Lee and Takagi [315] seek to maintain diversity by monitoring it and responding by altering the mutation rate of the EA.

While the importance of diversity is often stressed in introductory texts [54, 53], from a theoretical point of view this topic has been neglected. Crowding and niching techniques have been studied by DeJong [39] and Mahfoud [316]. Schaffer et al. studied how incest prohibition affects genetic drift and allele value loss in the population [317].

Recently in a series of papers [318, 319, 320] Friedrich and co-authors analyzed the effects of different explicit diversity maintenance schemes on the time complexity of the EA solving specific problems. They illustrate multiple techniques and fitness functions where diversity maintenance is required for polynomial time solutions with EAs using both crossover and mutation.

4.4 Detailed Review of Relevant Theory

As noted before, critiques of the schema analysis were that it could not explain time complexity, convergence or dynamical behavior of evolutionary algorithms. The following sections will provide an overview of these areas of EA theory and highlight relevant material for later use. The reader is encouraged to consult the surveys Yao

[321], Rudolph [322], Eiben and Rudolph [323], Beyer et al. [324] and Oliveto at al. [270] for time complexity work. For the dynamical systems work see introductions by Reeves and Rowe [40] and Rowe [325] for more background and detail.

4.5 Detailed Review of Time Complexity Analysis

Among the earliest looks at convergence and time complexity properties of EAs is the dissertation of T. E. Davis linking the convergence theory of simulated annealing to genetic algorithms [326]. The work of A. E. Eiben et al. [249] was among the first to use Markov chains to analyze GA convergence. Hart and Belew [327] looked at problem hardness and GAs, showing that arbitrary functions are difficult to optimize. Other early work included H. G. Beyer's asymptotic results for various Evolutionary Strategies algorithms [328, 329, 330]. [4]

Günter Rudolph's late 1990s papers [322, 331, 332] and dissertation, *Convergence Properties of Evolutionary Algorithms* [235], are considered by many to be touchstones in the time complexity and convergence analysis of binary EAs. Given a definition of convergence stating that an EA should find the optimal solution in finite time and hold that solution in the population for all future steps, Rudolph showed that classical GAs with proportional selection, mutation and crossover do not converge. Elitist variants will converge by this definition. His methods included extensive use of Markov chains, and showed that one need not derive an exact Markov model for each EA to characterize convergence behavior. Rudolph also introduced general conditions of convergence and non-convergence. Roughly stated, if the conditions are such that the Markov chain is ergodic then a visitation of the global optimum in finite time is

---

[4] In general this work will use GA with this notation to denote the use of crossover, while the EA will denote mutation-only. The reader should interpret by context as needed.

assured. In addition, if an elitist mechanism of selecting the best individuals from both the children and parents are used then convergence in finite time is assured.

The University of Dortmund has produced a prodigious amount of EA researchers and papers. Beginning with Hans-Paul Schwefel's ES group in the mid 1980s (Rudolph belonged to this group) and continuing with Ingo Wegener's EA group through 2008, their collective influence has shaped much of the notation in EA theory. In keeping with that notation n is used as a synonym of l, the number of bits in the genome string. Unless otherwise stated the standard mutation rate of $p_m = 1/n$ is assumed [227].

Wegener laid out methods for analysis of EAs on pseudo-Boolean functions [333, 334]. The objectives of this analysis are to illustrate the behavior of EAs on typical structures of functions, show examples of extreme behavior, disprove some widely held conjectures on EAs, as well as provide a basis for comparison of different EAs. All pseudo-Boolean fitness functions are of the type $f(x) : \{0,1\}^n \Rightarrow \mathbb{R}$. Let X_f be the random variable denoting the time for the solution to be found by the EA. In general, the outcomes of this type of analysis are an estimation of $E(X_f)$ in the best, average and worst case and an analysis of the success probability distribution $Pr(X_f \leq t)$. Wegener was a strong advocate of utilizing standard methods of randomized algorithm analysis and commonly gave talks at GA conferences on this point. See the textbook by Motwani and Raghavan [275], older papers by Rabin [335] and Karp [336] and Wegener's influential *blue book* [16].

In the classic notation of the ES community, aspects of survival selection are nearly encapsulated in a simple symbolic notation, $(\mu/\rho \overset{+}{,} \lambda)$. In the $(\mu/\rho + \lambda)$ ES, the population size is μ and λ is the number of children produced. The number of parents chosen to produce a single child is ρ and when $\rho = 2$ it implies two parent recombination and is generally omitted. When greater than two it assumes *multire-*

combination takes place, inspired by similarities with protozoan genetic recombination [337]. The plus-sign (”+”) implies a steady-state survival replacement, with the best μ individuals of the combined $\mu + \lambda$ chosen. The generational survival selection is denoted by a comma rather than plus-sign, as in the (μ, λ) ES where $\lambda \geq \mu$ and the best μ are kept. The ES community does not have symbols to denote the specific reproductive selection scheme used and instances of the above schemes can be found in the literature.

4.5.1 The $(\mu + \lambda)$ EA

Below is the so-called $(\mu + \lambda)$ EA. It starts with μ individuals in the current population, creates λ children and picks the best μ from the $\mu + \lambda$ for the next generation. This is an elitist EA, a non-elitist version is the (μ, λ) EA where the next generation is the best μ of the λ children. The crossover operator is excluded for now, yet it is easy to see how adding it makes this algorithm the classic GA. The specific method of randomized parent selection may be changed as well. In ES nomenclature, ’selection’ generally refers to selecting the next population from the current and child populations [337]. In GAs, selection generally refers to the method of choosing parents. Context will generally provide clarification and when necessary it will be referenced here as *parental selection* and *survivor selection*.

Algorithm 4.2: The $(\mu + \lambda)$ EA

1. Initialization: Choose μ individuals $x \in \{0, 1\}^n$ uniformly at random.
2. Mutation: Create λ new individuals by choosing x randomly from the parent population and flipping each bit with probability p_m.
3. (Survivor) Selection: create the new population by choosing the best μ individuals from the $\mu + \lambda$ set.
4. Repeat the last two steps until some stopping criteria is found.

The classic GA with crossover would be notated as (N, N) GA, yet note that parental selection operator is random selection. The simplest form of the above is the $(1+1)$ EA, while the $(1, 1)$ EA is simply the classic random-walk of stochastic process theory. The simplest elitist GA is the $(2+1)$ GA since a population of two is required. The steady-state GA of DeJong [39] would be notated $(N+1)$ GA, while a fully elitist version is the $(N+N)$ GA. The notation's usefulness decreases for describing other mechanisms of elitism and the massive variety of GAs in common use, however it neatly bridges the GA and ES camps from a nomenclature perspective.

4.5.2 The $(1+1)$ EA

Algorithm 4.3 is the $(1+1)$ EA and its properties are summarized extensively in Rudolph [235, Chapter 5]. It is the simplest example of both an elitist and steady-state EA and some might call it *The Fundamental Evolutionary Algorithm.*

Algorithm 4.3: The $(1+1)$ EA

```
1. Choose mutation rate $p_m \epsilon (0, 1/2]$
2. Choose $x \quad \epsilon \quad \{0, 1\}^n$ uniformly at random.
3. Create $y$ by flipping each bit of $x$ independently with $p_m$
4. If $f(y) \geq f(x)$, set $x := y$
5. Continue at line 3
```

Rudolph proved that the $(1+1)$ EA would optimize the classic function ONEMAX in average $O(n \log n)$ time [235]. ONEMAX is defined below.

$$ONEMAX(x) = \sum_{i=1}^{n} x_i \tag{4.15}$$

Droste, Thomas Jansen and Ingo Wegener have teamed up on a great number of papers analyzing the $(1+1)$ EA and derivatives on functions like ONEMAX [272, 338, 339, 340, 341]. The goal is generally to apply rigorous methods of algorithmic analysis against elementary functions to prove various attributes of EAs.

Let the function BIN be defined as the real number representation of a bitstring as if it were binary encoded. The same authors also showed that the $(1+1)$ EA would optimize the function BIN in average $O(n \log n)$ time, and extended this result to all boolean functions with separable inputs [272]. They were among the first to identify and prove that functions exist for which the $(1+1)$ EA finds the global optimum in $\Theta(n^n)$ time [341]. This function DISTANCE is defined below and is a quadratic parabola with a minimum where $n/2$ bits are one, and two maxima at the all-zeros and all-ones string. The all-zeros string's fitness value is slightly higher than the other maxima.

$$DISTANCE(x) = \left(\sum_{i=1}^{n} x_i - \left(\frac{n}{2} + \frac{1}{3} \right) \right)^2 \tag{4.16}$$

Continuing this thread Droste [341] also analyzed the LEADINGONES function, defined below, and first presented by Rudolph [235]. They show it can be optimized by the $(1+1)$ EA in $\Theta(n^2)$ steps.

$$LEADINGONES(x) := \sum_{i=1}^{n} \prod_{j=1}^{i} x_j \tag{4.17}$$

In [338] they prove that EAs without crossover will require expected exponential time to solve fitness functions derived from MAXSAT, a family of well known NP-hard problems. They also claim that adding crossover results in the same outcome.

The Metropolis Algorithm [36] and Simulated Annealing [55] are classic optimizations algorithms. In their most basic form applied to binary string representations the aforementioned algorithms differ from the above $(1+1)$ EA in that only one randomly chosen bit is flipped in each generation. The second difference is the acceptance criteria, both may accept downward fitness moves with a given probability. In simulated annealing, this *acceptance of worsenings* criteria happens on a "temperature schedule". Both are considered local search algorithms, though various restart mechanisms can alter this characterization. Hoos and Stützle [15] contains comprehensive details on these and other related algorithms.

4.5.3 Metropolis Algorithms and Simulated Annealing Hillclimbers

Droste et al. [340] show how the (1+1) EA algorithm can be modified to resemble Metropolis algorithms and give various convergence and complexity results. Below is a $(1+1)EA$ with Metropolis selection added. The function $\alpha : \mathbb{N} \Rightarrow [1;\infty)$ is the *selection schedule.* When $\alpha(t)$ is constant, this algorithm is identical to the Metropolis algorithm. When this schedule is non-constant, the algorithm is of the Simulated Annealing type. Note that this variant of the $(1+1)$ EA uses a mutation operator of Hamming distance one, ie mutating one bit only.

Algorithm 4.4: The Metropolis and Simulated Annealing Hillclimber

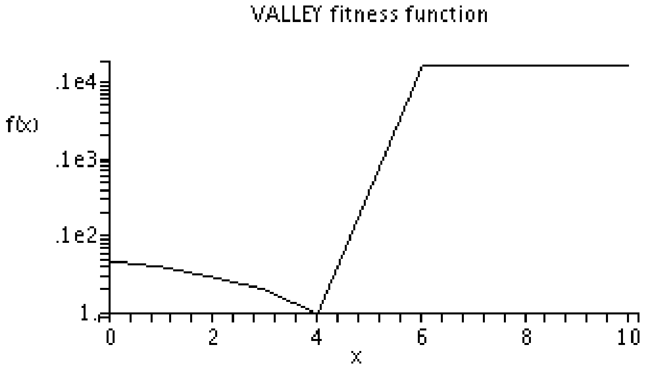
```
1. Set $t := 1$. Choose $x \epsilon \{0,1\}^n$ uniformly at random.
2. Create $y$ by flipping one bit of $x$ randomly.
3. With probability $\min\{1, \alpha(t)^{f(y)-f(x)}\}$ set $x := y$.
4. Set $t := t + 1$. Continue at line 2.
```

The analysis is restricted to symmetric functions and a Markov chain with $n+1$ states. Symmetric fitness functions (also called unitation) are those that depend only on the number of ones on the genome, this allows a reduction in the state space of the model. The following is a fitness function, the plot of the function, and a set of convergence bounds on for Algorithm 4.4.

Definition 6. *The function VALLEY:* $f(x) : \{0,1\}^n \Rightarrow \mathbb{R}$ *is defined as (without loss of generality* n *is even):*

$$VALLEY := \begin{pmatrix} n/2 - ||x||_1 & for\ ||x||_1 \leq n/2, \\ 7n^2 \ln(n) - n/2 + ||x||_1 & for\ ||x||_1 > n/2, \end{pmatrix} \tag{4.18}$$

where $||x||_1$ *is the number of ones in* x*.*


VALLEY fitness function

Theorem 7. *The expected number of steps until Algorithm 4.4 with constant* $\alpha(t)$ *reaches the maximum of VALLEY for the first time is*

$$\Omega\left(\left(\sqrt{\frac{\alpha}{4}}\right)^n + \left(\frac{1}{\alpha} + 1\right)^n\right) = \Omega(1.179^n) \tag{4.19}$$

for all choices of $\alpha \epsilon [1;\infty)$*.*

Theorem 8. *With probability* $1 - \mathcal{O}(n^{-n})$ *the number of steps until Algorithm 4.4 with selection schedule*

$$\alpha(t) := 1 + \frac{1}{s(n)} \tag{4.20}$$

reaches the maximum of VALLEY for the first time is $1-\mathcal{O}(n\cdot s(n))$ *for any polynomial* s *with* $s(n) \geq 2en^4 \log n$. *Furthermore, the expected number of steps until this happens is* $\mathcal{O}(n \cdot s(n))$, *if one sets* $\alpha(t) := 1$ *for* $t > 2^n$.

They continue and present a version of the $(1 + 1)$ EA with a cyclic mutation rate schedule and prove that it is able to optimize a specially structured function in polynomial time. They show that the standard version of the EA requires superpolynomial time as well as showing better worst-case performance in general for some functions. Interestingly, the standard (1+1) EA will provably outperform Algorithm 4.5 on some trap functions, while Algorithm 4.5 outperforms the (1+1) EA on functions with fitness cliffs.

Algorithm 4.5: The (1+1) EA with Dynamic-Cyclic Mutation

```
1. Choose $x \epsilon \{0,1\}^n$ uniformly at random.
2. Choose mutation rate $p_m := 1/n$.
3. Create $y$ by flipping each bit of $x$ independently with $p_m$.
4. If $f(y) \geq f(x)$, set $x := y$.
5. $p_m := 2p_m$. If $p_m > 1/2$, set $p_m := 1/n$.
6. Continue at line 3.
```

Jansen and Wegener [339] introduce yet another variant of the algorithm with a slightly different dynamic mutation schedule. They introduce a fairly complex fitness function with a fitness barrier where a jump is required as well as following a specific Hamming path in the binary space. These artifices make analysis easier. They demonstrate that the standard algorithm with an optimal mutation rate will require $O(n^{2.361})$ time, while the standard $p_m = 1/n$ rate requires superpolynomial time. The dynamic rate algorithm succeeds in $O(n^2 \ln n)$ steps. This general approach to constructing fitness functions will foreshadow future papers and fitness functions designed to differentiate mutation-only and crossover enabled EAs.

Recently, Jansen and Wegener [342] compare the (1+1) EA directly with simulated annealing (SA) and Metropolis (MA). They show that for a subclass of unitation functions, those depending only on the numbers of ones in the bitstring, that both the MA and SA can optimize in $O(n \ln n)$ time. Also shown are functions where the EA and SA differ exponentially in expected time. They derive conditions on the cooling schedule for various time bounds. There are examples shown in [342] where the behavior of the EA and SA/MA differ as a direct result of the limit that the latter have on flipping only one bit per generation. Any fitness barrier more than one bit wide will require that the MA/SA accept a worsening move to make progress across the barrier. The EA can flip multiple bits and thus cross the barrier faster.

Jansen and Wegener next tackle the question of fitness plateaus [343]. Note that the classic $(1+1)$ EA will only accept a child string if its fitness is greater than or equal to the parent. What of a trivial modification requiring strict fitness superiority? It is easy to see that this would pose a problem on fitness functions with so-called fitness plateaus. These are Hamming neighbor areas of flat fitness values across which the algorithm must traverse to find a higher fitness region. They take the opposite tack and define a family of functions SPT_n where the classic version of the algorithm requires exponential time but the version not accepting neutral moves optimizes in $O(n^3 \ln n)$. This function again contains a path that the algorithms are channeled to and adds traps that the classic algorithm follows. This result is extended with multiple paths and traps with the same differentiation holding.

Wegener and Witt present an analysis of the $(1+1)$ EA and multi-start variants on quadratic pseudo-boolean functions [344]. These are polynomials of degree 2, a class of functions that contain NP-hard problems. A multi-start variant is one where $p(n)$ independent runs of the $(1+1) EA$ take place. Their analysis is complete in that they present examples of this class that give a complete picture of success and failure of the two algorithms. For the subclass of quadratic functions with only non-negative coefficients, the $(1+1)$ EA can optimize in expected polynomial $O(n^3)$ time. However for the subclass of quadratic functions with negative coefficients, the algorithm takes expected exponential time. This subclass is effectively optimized by a multi-start variant with expected $O(p(n)n \ln n)$ steps where $p(n) = \omega(n \ln n)$. Next, a version of the $TRAP_n$ function is presented which is reduced to be in the quadratic class. For this function both algorithms will take exponential time or require exponential restarts to achieve success. Finally they show that if the quadratic function is separable, i.e. it can be represented as the sum of quadratic functions on smaller domains, then the $(1+1)$ EA will not take as exponential number of steps to optimize.

The above is not an exhaustive list of all papers dealing with the $(1+1)$ EA, there are many more papers available. The material presented was chosen to support later chapters of this work and to highlight past progress leading up to the addition of populations and crossover to the analysis.

Obviously the $(1+1)$ EA lacks a population in the traditional sense and some might argue that it is closer to the family of hill-climber algorithms than the GA, yet it has served a valuable purpose in illustrating methods of analysis and outcomes that can provide intuitive and rigorous results for population based EAs. Note that without a population $p > 1$, the EA cannot utilize any form of the crossover operator.
5

4.5.4 Comparing 1 vs μ Population EAs

Comparing a single individual EA to population based EAs requires some care. Counts of fitness function evaluations are a more correct comparison metric than

---

[5] The headless-chicken crossover (ie macro-mutation) of Jones would be an exception to this rule, though unfortunately it is under studied.

generation counts. One must also make an effort to include any restart strategies of the algorithms.

Jansen and Wegener [274] present a class of functions where the (1 + 1) EA takes super-polynomial time to optimize, yet a $(\mu+1)$ EA with fitness proportionate parental selection and reverse fitness proportionate truncation of the extra member will take polynomial time to optimize. The $JUMP_{k,s}$ functions are defined to have a k-bit barrier next to the optimal all-ones bitstring and the fitness value itself is scaled to weaken proportionate selections effects.

Jansen, De Jong and Wegener [345] examine the extension of the $(1+\lambda)$ EA on LEADINGONES and ONEMAX. The algorithm is not presented here, yet it is sufficient to imagine a version of Algorithm 4.2 with $\mu = 1$. For LEADINGONES, the algorithm is proven to take $\Theta(n^2 + n\lambda)$ function evaluations, and $O(n \log n + n\lambda)$ for ONEMAX where $\lambda = O(\log n)$.

Jägersküpper and Storch [57] compared the plus (elitist) and comma (generational) survivor selection strategies of the $(1+\lambda)$ EA. For large λ they formalize and prove the conjecture that the two algorithms are not significantly different in the mechanism used for searching the solution space. They then restrict the comma strategy to $\lambda = O(\ln n)$, and prove the comma strategy superior to the plus strategy with any λ for a function derived from ONEMAX with a fitness cliff inserted.

$$CLIFF(x) = \left\{ \begin{array}{ll} ONEMAX(x) - \lfloor n/3 \rfloor & if\ |x| \geq n - \lfloor n/3 \rfloor \\ ONEMAX(x) & if\ |x| < n - \lfloor n/3 \rfloor \end{array} \right\} \tag{4.21}$$

Witt [277] looks at the steady-state mutation-only $(\mu+1)$ EA for $\mu = poly(n)$. For LEADINGONES it is shown that the algorithm takes $O(\mu n \log n + n^2)$ function evaluations, and for ONEMAX it takes $O(\mu n + n \log n)$ evaluations. Witt also introduces a fairly complex function not reproduced here where the $(1+1)$ EA takes exponential evaluations yet the $(\mu + 1)$ EA with conditions on μ takes $O(\mu n)$ evaluations. While there have been many papers experimentally examining the long held belief in GA lore that populations are required for efficient optimization of some functions, Witt was among the first to prove a specific case.

Witt [346, 347] follows up with a similar result comparing the steady state EA of Jansen [274] (with population size ≥ 1) to the $(1 + 1)$ EA showing an exponential difference in time complexity favoring the EA, thus improving upon the prior Jansen-Wegener result.

He and Yao [348] prove, for a set of elementary fitness functions, that the $(N+N)$ EA will increase the probability of success and decrease the expected waiting time over the $(1 + 1)$ EA.

4.5.5 Analyzing Crossover

Crossover has been more challenging to analyze and get rigorous time complexity bounds. The references below are highlights of results on this topic and more detail is presented later in this work.

For the first time Jansen and Wegener [300, 349] presented a function where crossover was rigorously proven to be necessary. The function $JUMP_n$ consists of an n-bit barrier in ONEMAX with only the all-ones string on the other side of the barrier. This function has no scaling of the fitness value. They show the $(1+1)$ EA to take $\Theta(n^m + n \log n)$ steps to optimize the function. They then introduce a full steady-state GA with the modification such that a no-duplicates child policy is enforced. For simplification of the analysis, the parental selection algorithms are uniform-random and as in any $(\mu + 1)$ GA the survival selection algorithm is a truncation of the worst fitness individual. This GA was proven to take small degree polynomial expected optimization time. If the simple no-duplicates diversity technique is removed then optimization time becomes much worse.

Jansen and Wegener [350, 295] derive a so-called ‘Real Royal Road’ for which a steady-state GA with both uniform and one-point crossover outperforms any mutation-only EA. These algorithms both have variable sized populations and the GAs require only polynomial optimization time versus expected exponential time for the EAs. Note that a trivial change is made to convert this to a no recombination EA.

Algorithm 4.6: The Steady State GA

1 Select independently and randomly the s(n) individuals of the initial population.

2 With probability $p_c(n)$ go to Step 3’ and with the remaining probability of 1 - $p_c(n)$ go to Step 3”. *Steps 3’ and 3” are mutually exclusive.*

3’ Choose two parents x and y from the current population. Let z^* be the result of uniform crossover applied to x and y and let z be the result of mutation applied to z^*.

3” Choose one parent x from the current population. Let z be the result of mutation applied to x.

4 If the fitness of z is smaller than the fitness of the worst individual of the current population, go to Step 2. Otherwise, add z to the population. Let W be the multi-set of individuals in the enlarged population which all have the worst fitness and let W' be the set of those individuals in W which have the largest number of copies in W. Eliminate randomly one element in W from the current population. Go to Step 2.

Algorithm 4.7: The Steady State EA

- Set probability $p_c(n) = 0$ and run Algorithm 4.6.

Storch and Wegener [351, 352] complement the previous result by simplification of the scenario to the smallest and simplest possible population based EA and GA. These algorithms are presented below and note that the uniform probabilities in the search steps are arbitrary. The EA requires exponential time to optimize, while the GA is polynomially efficient.

Algorithm 4.8: The (2+1) EA

1. *Initialization*: Randomly chose two different individuals $x, y \epsilon \{0,1\}^n$.
2. *Search*: Produce an individual z,
   - with probability 1/2, z is created by mutate(x),
   - with probability 1/2, z is created by mutate(y),
3. *Selection*: Create the new population P.
   - If $z = x$ or $z = y$, then $P := \{x, y\}$
   - Otherwise, let $a \epsilon \{x, y, z\}$ be randomly chosen among individuals with the worst f-value. Then $P := \{x, y, z\} - \{a\}$

Algorithm 4.9: The (2+1) GA

1. *Initialization*: Randomly chose two different individuals $x, y \epsilon \{0,1\}^n$.
2. *Search*: Produce an individual z,
   - with probability 1/3, z is created by mutate(x),
   - with probability 1/3, z is created by mutate(y),
   - with probability 1/3, z is created by mutate((crossover(x, y)).
3. *Selection*: Create the new population P.
   - If $z = x$ or $z = y$, then $P := \{x, y\}$
   - Otherwise, let $a \epsilon \{x, y, z\}$ be randomly chosen among individuals with the worst f-value. Then $P := \{x, y, z :\} - \{a\}$

Dietzfelbinger et al. [299] introduce a recombinative hill-climber derived from the gene-invariant GA of Culbertson [353]. They call this algorithm the (1+1) GA where the individual is crossed-over with its bitwise compliment, and no mutation is used. It is shown that the expected optimization time equals $\Theta(n \log n)$.

Doerr et al. [354] compared standard mutation (calling it global) to the local mutation operator of flipping exactly one bit at random. They argued that in general it is difficult to derive principles to extend results from the easier to analyze local mutation to the general global mutation. They also show a fitness function where the local operator exponentially outperforms the global operator.

Oliveto et al. [355] show another alternative mutation scheme, rank-based mutation, where individuals are mutated more aggressively based upon the weakness of their fitness rank in the population. They show various results for different fitness functions proving rank-based mutation is superior one some and provably worse on other functions.

Neumann et al. [356] investigate fitness proportionate selection and show via drift analysis that the EA with a logarithmic population size (in n) cannot optimize ONEMAX in polynomial time with high probability. However if the fitness function is scaled with a coefficient, the EA can then optimize effectively.

Drift analysis in deriving the time complexity of EAs has grown in usage. While powerful, it is fairly complex and verbose to write. Recently, Oliveto and Witt introduce a simplified version of drift analysis and demonstrate its use in [357].

Watson and Jansen [241] derive a building block fitness function related to the HIFF and show a GA with two-point crossover can solve in average polynomial time versus single and multi-population EAs that require exponential time. This rigorous result is noteworthy as the function is the first of the explicit building block type to be an instance of the BBH hypothesis.

For domain specific and less artificial fitness functions a series of results have been released. In Fischer and Wegener [302] and in Sudholt [303] the Ising model, statistical mechanics model of magnetic fields, is analyzed and crossover enabled EAs are shown to be of clear benefit.

Graph problems have also been given attention on provable bounds for the EA. Neumann [358] and Doerr et al. [359] introduced an alternative mutation schemes involving jumps on the graph for solving the *Eulerian cycle* problem faster than standard mutation.

Horoba [360] analyzed a simple EA on the *multiobjective shortest path* graph problem, which is NP-Hard. He showed the EA to be a fully polynomial-time randomized approximation scheme (FPRAS) for it. Doerr et al. [361] show that for the well known (and real world) *all pairs shortest path* graph problem, using crossover has a provable modest advantage over mutation alone.

4.6 Detailed Review of the Infinite Population Model

The intent of the Infinite Population Model (IPM) model is to allow a mathematical analysis of the Simple GA. Vose covers the model in great detail in [256] including SGAs with proportionate, truncation and tournament selection. For review purposes, define the Simple GA as an EA with proportionate selection, bitwise mutation and several standard crossover operators. First the review will be restricted to mutation-selection GAs with no crossover, elitism or other advanced techniques. Later crossover will be introduced.

The IPM is a discrete dynamic systems model or *map*. Maps translate discrete-time inputs to discrete outputs and are generally constructed by iterating continuous mathematical functions. An example with chaotic behavior is the well-known Logistic Map [182].

$$f(x) := \lambda x(1-x)\ \{\text{where } x, \lambda \epsilon \mathbb{R};\ \lambda > 0;\} \tag{4.22}$$

In addition, the model is a deterministic dynamical system - there is no random component to the mapping of inputs to outputs. The inputs and outputs, ie populations, are represented as probability distribution vectors over the search space. Iterating the model maps the current population vector to the next population vector. The population's probability distribution is simply a vector of proportions with dimension s, where s is the size of the search space. Each entry in the vector is the proportion of members in the global population that are represented by a given chromosome in the search space. This representation allows utilization of techniques and theorems from the mathematical theory of dynamic systems to analyze the GA.

Let Λ be the simplex containing all distributions over the binary search space and let $\mathcal{G}$ be the GA-map.

$$\mathcal{G} : \Lambda \rightarrow \Lambda \tag{4.23}$$

For a 2-bit genome, a possible population vector is $\vec{\mathfrak{p}} = (0.1, 0.2, 0.5, 0.2)$. This could represent a population of 10 individuals, 1 copy of `00`, 2 copies of `01`, 5 copies of `10`, and 2 copies of `11`. Note that the population vectors have the simplex property such that each component of the vector is in the range $[0, 1]$, and the sum of the components equals 1.

$$\Lambda := \left\{ (x_0, \ldots, x_{s-1}) : \sum_{i=0}^{s-1} x_i = 1 \quad \ni \quad x_i \in [0,1]\,,\ x \in \mathbb{R} \right\} \tag{4.24}$$

Note that fixed-size populations form a subset of the simplex called a *lattice.* Some simplex population vectors are not representable with finite populations. An example is any vector with an element like 2/3 where the cardinality of the vector is not a multiple of three for a population size such as 10. Finite populations move from lattice point to lattice point in the simplex. The smaller the population size, the sparser the lattice points are in the simplex.

The $\mathcal{G}$-map is composed of the three operators of the simple GA, fitness proportionate selection $\mathcal{F}$, mutation operator $\mathcal{U}$ and the crossover operator $\mathcal{C}$. The $\mathcal{G}$-map is expressed symbolically as follows: vn

$$\mathcal{G}(p) = \mathcal{C} \circ \mathcal{U} \circ \mathcal{F} \tag{4.25}$$

$\mathcal{F}$ and $\mathcal{U}$ are operators implemented as 2-D matrices of the cardinality of the search space. $\mathcal{C}$ is a quadratic operator composed of a stack of 2-D matrices $\mathcal{C}_k$.

Iterating the map simulates the trajectory of the GA (in the limit of infinite population size), where the next population vector becomes the input to the next generation of the GA. This forms a sequence of population vectors. $\vec{\mathfrak{p}_1}, \vec{\mathfrak{p}_2}, \vec{\mathfrak{p}_3}, \ldots, \vec{\mathfrak{p}_\mathfrak{t}}$. The sequence is the trajectory of the GA model through the population space.

For a sufficiently large population in a real GA, the model allows accurate predictions of the expected next population and the long term behavior of the population [256, 257]. If the population is small, then the actual populations sampled by the GA may have wide variability compared to the predicted population. If the population is very large, then the actual population distributions produced by the GA should be close to the predicted model for a large number of generations [256].

Using the eigensystem of the mutation-only IPM model, fixed points of the mutation-selection GA can be found. Fixed points are population vectors such that applying the GA to them results in the same population vector. Fixed points are not the optimal points in the fitness landscape, they represent the expected long-run distribution of the population. A fixed point may or may not contain a significant proportion of the global optima of the fitness landscape.

4.6.1 Selection-Mutation Models

The following section summarizes Rowe [257] as it applies to selection-mutation models of GAs. Given a binary genome of length l, the search space of the GA is of size $s = 2^l$. To use the Vose model, represent the population as a vector of proportions of length s, $\vec{\mathfrak{p}} = (p_0, \ldots p_{s-1})$. Each p_i is the proportion of membership in the population by the binary string i.

The mutation-selection infinite population model is created as follows. The mapping operator G , which implements the GA model, is defined by:

$$p(t+1) := G\,(p(t)) = \frac{1}{\overline{f}(p(t))}\; U\; S\; p(t) \tag{4.26}$$

Let $\overline{f}(p)$ be the average fitness of the population p . The s x s mutation matrix U is composed of the probabilities that a chromosome string j will mutate into string i. Note that this U matrix is symmetric. The probability a given bit in the chromosome string mutates to its complement state is q. The s x s selection matrix S is a diagonal matrix consisting of fitness values along the diagonal and zeros elsewhere. Dividing by $\overline{f}(p)$ implements proportionate selection.

$$\begin{aligned} U_{i,j} &= q^h(1-q)^{l-h} & h &= \text{Hamming Distance}(i,j) \\ S_{k,k} &= f(x_k) & k &\,\epsilon\, [0, s-1] \end{aligned} \tag{4.27}$$

The U matrix is commonly referred to as the *mixing matrix.* From the theory these five properties of the $U\;S$ matrix [257] are known.

1. $U\;S$ is an irreducible, positive matrix, all entries are non-negative.

2. Fixed-points of the model are the normalized (so that all elements sum to 1) eigenvectors of $U\ S$.
3. Only one normalized eigenvector is in the simplex (via Perron-Frobenius theorem [257][181]).
4. Eigenvalues of $U\ S$ are the average fitness of the population given by the corresponding eigenvector.
5. The largest eigenvalue corresponds with the lone eigenvector inside the simplex.

These properties allow the computation of the fixed points of the infinite population model for a given fitness function and mutation rate. A normalized vector is one where all elements have been scaled such that they all sum to 1, creating a vector that obeys the simplex property. The lone normalized fixed point is the global attractor of the dynamical system modeling the GA. This fixed point is the expected population distribution of the GA for a sufficiently large number of generations and population size.

Metastable states defined by the model can also be calculated. These states are fixed points of the model that exist outside the representable space of a real population (ie outside the simplex). This effect was also investigated independently of Vose and coworkers by

There can exist fixed points outside, but very near, the simplex. Such vectors do not obey the simplex property and cannot represent the finite population of a running GA. Metastable states near the simplex can create *metastable regions* inside the space that influence the behavior of finite GA populations. Finite populations $\vec{\mathfrak{p}}$ can be attracted to these regions. If the GA population enters such a region, it may typically spend a long time in the region before escaping. These regions were investigated by Vose and co-workers as well as independent illustrations by van Nimwegen and Crutchfield [362].

There is a different kind of metastable region around a fixed point that can not be represented by a finite population. These are points in-between the lattice points formed by finite populations, yet the distance from $\vec{\mathfrak{p}}$ to $G(\vec{\mathfrak{p}})$ is small [363].

Vose argues that these points as well as saddle fixed points are related to *punctuated equilibria* effects observed in running GAs [363].

4.6.2 Course Grained Models

A major difficulty in utilizing the IPM is the large size of the systems for GAs with non-trivial string lengths. While the IPM is an exact model, using it to predict dynamics is challenging due to the computational complexity induced by the state space size.[6] Recently there has been much interest in developing an approximate model of GA dynamics as described by the IPM. Stephens built upon his initial work with Waelbroeck [188] on building-blocks and the *degrees of freedom* inherent in GA models to develop coarse-graining techniques. The core idea of these techniques is that the full dynamical model of the GA is exact and as such contains many microscopic degrees of freedom that are not important to the dynamics predicted.[7]

Details of these techniques applied to the IPM is beyond the scope of this work. The reader is encouraged to consult the results of Stephens and co-workers [259, 260, 261] for more detail. Relevant to the above, Stephens et al. [259] produce a coarse-graining that gives convergence speed to linkage equilibrium under one-point crossover. Rowe, Vose and Wright also have a series of coarse-graining results [366,

---

[6] I speculate that effective versions of the IPM can be coded in the emerging MapReduce [364] framework. Google and others have used this to calculate PageRank [365]. The core of PageRank is nothing more than computing the stationary distribution of a markov matrix via the Perron-Frobenius result on eigenvectors. This is directly applicable to mutation-selection EAs.

[7] These techniques are well used in physics literature. Loosely speaking one might compare this to using a truncated Taylor Series expansion to approximate a function, low-order elements are truncated from the expanded equation.

367] as does Burjorjee [368]. Recently Mitavskiy et al. [369] describe a method of utilizing quotients of the Markov models of GAs to develop coarse grained models.

4.7 Summary

This review is by no means fully comprehensive. It should give the reader a birds-eye view of EA theory. The book by Reeves and Rowe [40] is a fine starting point for deeper study. The edited work by Menon [370] is an excellent tour d'horizon of the frontier of EC theory.

CHAPTER 5

IGNOBLE TRAILS FOR CONSTANT POPULATION SIZE

> Most of the variation of genotypes available for selection in a population is the result of recombination, not of new mutations.
>
> Ernst Mayr [153]

Framework	Mutation-only EAs	Crossover-enabled GAs
Time Complexity Analysis	✓	✓
Dynamical Systems Analysis		
Empirical Analysis		
Methods: Constant population size, Theorem-Proof		

Figure 5.1: Chapter 5 Framework

5.1 Introduction

The chapter examines a small population EA and GA against a new fitness function. Time complexity analysis is performed in a theorem-proof method. Figure 5.1 illustrates the contributions within the framework. [1]

Beginning with the early days of the GA and the Schema Theorem it has often been argued that the crossover operator is the more important genetic operator. The early Royal Road (RR) functions were intended to show where crossover would excel. Mutation-only EAs were subsequently experimentally shown outperform GAs with crossover on these functions, failing to confirm the hypothesis. Recently, several new RRs have been introduced and proven to require expected polynomial optimization

---

[1] An earlier version of this chapter was published as
J. Neal Richter, Alden H. Wright, John Paxton: Ignoble Trails - Where Crossover Is Provably Harmful. PPSN 2008, the 10th International Conference on Parallel Problem Solving From Nature.

time for GAs with crossover, while the mutation-only EAs need exponential time. This chapter does the converse, inverting the recent Royal Road results by showing proofs that GAs with crossover require exponential optimization time on new Ignoble Trail functions while mutation based EAs optimize them efficiently.

First proposed by Mitchell et al. [289], the well known Royal Road class of fitness functions were designed to demonstrate the essential nature of the crossover operator in genetic algorithms for optimizing that class of fitness functions. They also showed that for an idealized GA ignoring the effects of hitchhiking, the expected optimization time is $O(2^k \log(n/k))$. Somewhat unexpectedly, follow up experimental studies by Forrest and Mitchell [291] show that some random mutation hill-climbers outperform GAs with crossover on the Royal Road. This prompted the same authors to define an open problem in [293].

- Define a family of functions and prove that genetic algorithms are essentially better than evolutionary algorithms without crossover.

In [295], Jansen and Wegener proved that the expected optimization time of the well known (1+1) EA on the classic Royal Road function is $O(2^k (n/k) \log(n/k))$ where n is the string length, k is the length of sub elements of the string and $1/n$ is the mutation rate. Recently in EA research there have been several fitness functions built to meet this challenge in a rigorous way and these are discussed in the next section.

The goal of this paper is to do the opposite, provide a fitness function where EAs with mutation alone are provably better at optimization than GAs with crossover:

- Define a family of functions and prove that evolutionary algorithms without crossover are essentially better than genetic algorithms.

We are not alone in seeking this result. Very recently Poli et al. have produced a fitness function called OneMix [310] where crossover is shown experimentally to be not helpful.

5.2 Minimal Population Evolutionary Algorithms

These algorithms are instances of steady-state evolutionary algorithms [311] where the population is not fully replaced at each generation. A no-duplicates policy is also in place, forcing a population of distinct strings.

5.2.1 The Steady-State (2+1) EA

Here the (2+1) EA is restated. It is an instance of the well-known (μ+1) EA, studied among other places in Rudolph [235] and Beyer [337].

Algorithm 4.8: The (2+1) EA

```
1. Initialization: Randomly choose two different individuals $x, y \in \{0,1\}^n$,
2. Search: Produce an individual $z$,
    • with probability 1/2, $z$ is created by mutate($x$),
    • with probability 1/2, $z$ is created by mutate($y$),
3. Selection: Create the new population $P$.
    • If $z = x$ or $z = y$, then $P := \{x, y\}$
    • Otherwise, let $a \in \{x, y, z\}$ be randomly chosen among individuals with the worst $f$-value. Then $P := \{x, y, z\} - \{a\}$.
4. Goto Search.
```

5.2.2 The Steady-State (2+1) GA

Here the simple steady-state GA is redefined from [301] to work on a population size of 2, the smallest population size allowing crossover. Note that the usage of equal probability $\frac{1}{3}$ in the search step is arbitrary. The later results hold for any constant probability ϵ where $\epsilon > 0$.

Algorithm 4.9: The (2+1) GA

```
1. Initialization: Randomly choose two different individuals $x, y \epsilon \{0,1\}^n$,
2. Search: Produce an individual $z$,
    • with probability 1/3, $z$ is created by mutate($x$),
    • with probability 1/3, $z$ is created by mutate($y$),
    • with probability 1/3, $z$ is created by mutate(crossover($x, y$)).
3. Selection: Create the new population $P$.
    • If $z = x$ or $z = y$, then $P := \{x, y\}$
    • Otherwise, let $a \epsilon \{x, y, z\}$ be randomly chosen among individuals with the
      worst $f$-value. Then $P := \{x, y, z :\} - \{a\}$.
4. Goto Search.
```

5.3 Ignoble Trails

A new class of functions, the Ignoble Trails, are defined below. These functions are created for the purpose of rigorously proving that a given mutation based EA outperforms a given crossover based GA on these functions. Like the Real Royal Roads (RRR) and the HIFF functions, these functions are somewhat contrived to serve a specific theoretical purpose. No claim is made here that real world problems can be mapped to these new functions.

5.4 Ignoble Trails for Uniform Crossover

The first function $IT1^u_n(x)$ is a modification of the $R^u_n(x)$ function of [301] for uniform crossover. The symbol u refers to the uniform crossover operator. Most of the details are the same as $R^u_n(x)$ except for the addition of b^{**}. Assume a bit-string length of $n := 6m$, where n and m are even integers. Also note that $\|x\|$ refers to the number of ones in the string, $|x|$ is the length in bits of x, and $H(x, y)$ is the Hamming distance of x and y.

$$IT1_n^u(x) := \begin{cases} 16m & x = b^{**} \\ 15m & x \in T \\ 14m & x = a^* \\ 6m + i & x = a_i \in P_1 \cup P_2 \\ 6m - \|x\| & x \in R := \{0,1\}^n - P - T - \{b^{**}\} \end{cases}$$

The major features of $IT1_n^u(x)$ are as follows. The base fitness of the set R is defined to slope in increasing fitness towards the all zeros string. The path P is a sequence of distinct strings $a_1, ..., a_p$ such that consecutive strings on the path have a Hamming distance of 1. P contains $7m + 1$ total points where $a_i = 0^{n-i}1^i$ for $i \leq 6m$, and $a_i = 1^{n-j}0^j$ for $i = 6m + j$. P is segmented into two subpaths P_1 and P_2.

The P_1 subpath is defined as points $(a_0, ..., a_{5m-1})$ and the P_2 subpath is defined as (a_{5m+1}, a_{7m}). The fitness for all points P_i is $6m + i$, with the single exception that a local optimum is created at point $a^* := a_{5m}$ with fitness $14m$. The other local optimum of P is at the endpoint $a^{**} := a_{7m}$ with fitness value $13m$.

There also exists an area T defined to contain all points $b1^{4m}c$ where the substrings b and c obey $|b| = |c| = m$ and $||b|| = ||c|| = m/2$. In $R_n^u(x)$, T is the target and can be created with high probability with a population of $\{a^*, a^{**}\} := \{0^m1^{5m}, 1^{5m}0^m\}$ via uniform crossover.

The crucial modification to $R_n^u(x)$ is to add a point b^{**} with fitness greater than the region T. We define b^{**} to be $1^m0^k1^{4m-k}0^m$. This point has k bits different than

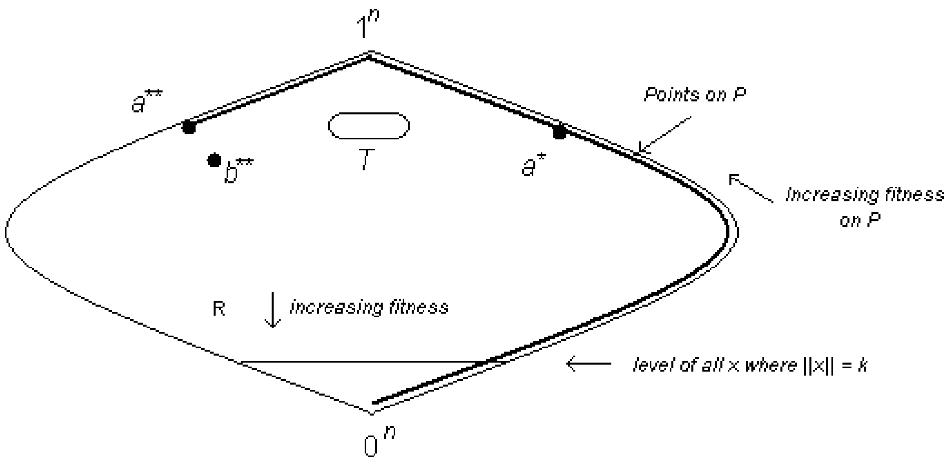


Figure 5.2: Illustration of Ignoble Trail 1

a^{**}, or $H(a^{**}, b^{**}) = k$. Here k is defined to be a constant where $n = 6m$ is chosen so that $3 < k < m/4$.

5.4.1 Behavior of the EA and GA on Ignoble Trail 1

Referring to Figure 5.2, the initial random population of two distinct individuals will begin the process of traveling down R towards the initial point of P, $P_0 := 0^n$. Both algorithms will discover and optimize P unless exceptional luck strikes and $\{T \cup \{b^{**}\}\}$ is discovered first. Since the selection method prohibits duplicate strings, once on path P there is a leading point and a trailing point on P. These points travel up P until such time as a^* is found [there is a probability $\Theta(1/n)$ a^* is skipped]. If a^* is found, the behavior degenerates to mimic the $(1 + 1)$ EA as a^* is fixed in the population and the other string is available for continued optimization of P until a^{**} is found.

Once the population becomes $\{a^*, a^{**}\}$ the behavior of the two algorithms diverges. The EA is very unlikely to discover T via mutation, and is likely to find b^{**} in $O(n^k)$ steps. Conversely the GA is very likely to discover T via crossover before it discovers

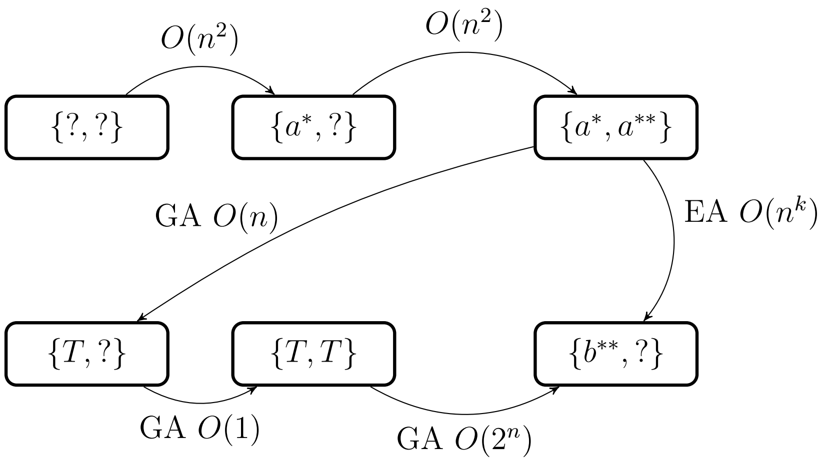


Figure 5.3: Diagram of proofs of Lemmas and Theorems for $IT1^u_n(x)$. The labels on each arc refer to the expected waiting time to transition from state to state for the labeled algorithm.

b^{**}. Once the GA has found T, it will accumulate both individuals in T in short order. The expected waiting time to discover b^{**} from T is exponential. Thus refer to T as the *trap* rather than the *target* of $R^u_n(x)$. Note that crossover is of little assistance in discovering b^{**} from either a^{**} or T.

Figure 5.3 contains a visual representation of the optimization paths that both algorithms will follow with high likelihood. Alternate paths will be covered in detail in the proofs.

5.4.2 Time Complexity Results

Note that the next set of proofs take some arguments from [301] or [295]. The addition of b^{**} requires many additional steps to prove rigorous results, there are many more good and bad events to account for above those from [301].

Lemma 9. *The probability that* $(2+1)$ *EA without crossover and* $(2+1)$ *GA with uniform crossover find a point in* $P_2 \cup T \cup \{b^{**}\}$ *without discovering path* P_1 *within* $O(n^2)$ *steps is at most* $e^{-\Omega(n)}$.

Proof. Recall that k is a constant, and recall that $n = 6m$ is chosen so that $3 < k < m/4$. Let $Q := P_2 \cup T \cup \{b^{**}\}$ and note that all elements of Q have at least $5m - k$ ones. Let R be the set of points not in P with at most $4m$ ones. The probability of initializing a member of the population with more than $4m$ ones is $e^{-\Omega(n)}$ by Chernoff's bound [275]. Since Q is contained in that same set, the same holds for Q. Each point of R has a better Hamming neighbor. The probability of discovering that neighbor via mutation is at least $p = 1/(3en)$. Applying Chernoff bounds, the waiting time for at most $n = 6m$ successful events is $O(n^2)$, and the probability that this waiting time is exceeded is exponentially small. The probability of producing a point in Q from R via mutation is at most $n^{-m+k} = e^{-\Omega(n)}$ by Chernoff's bound. Turning to the crossover operator, the probability of producing a point in Q from two points in R via crossover is $e^{-\Omega(n)}$ by the following argument. Let d be the Hamming distance between the two parent strings r_1 and r_2. Let $s = ||r_1 \wedge r_2||$, thus the expected number of ones is $s + d/2$. Unless $d > m - k$, the child string cannot have at least $5m - k$ ones. Applying Chernoff's bound on the differing bits of the parents, $r_1 \oplus r_2$, the probability to create at least $d/2 + m - k$ ones is $e^{-\Omega(n)}$. As for the joint operator, the probability of producing a point in Q from two points in R via crossover and mutation is $e^{-\Omega(n)}$ as follows. Either crossover produces a point with at least $9m/2 - k$ ones or it does not. In the first case, the probability that crossover produces a point with at least $9m/2 - k$ ones is $e^{-\Omega(n)}$ by the Chernoff bounds on the the bits differing in the parents. In the other case, mutation must go from a point with less than $9m/2 - k$ ones to a point with at least $5m$ ones, and the probability that this happens is $n^{-m/2+k} = e^{-\Omega(n)}$. Applying the union bound, it is seen that the total failure probability is $e^{-\Omega(n)}$. □

Lemma 10. *The* $(2+1)$ *EA will optimize* P *and find* $\{a^{**} \cup b^{**}\}$ *in* $O(n^2)$ *steps with probability* $1 - 2^{-\Omega(n)}$. *The* $(2+1)$ *EA will discover a point in* T *from* P *with probability* $2^{-\Omega(n)}$.

Proof. Beginning from Lemma 9, assume the population contains a point in P_1. Each point on the path P has a better fitness Hamming neighbor, except a^*. The probability of discovering that neighbor via mutation is at least $p = 1/(3en)$. Inverting and substituting provides a waiting time of at most $7m$ (the length of P) successful events of probability p. Applying Chernoff's bound gets the first result above. As for the second result, by the definitions of P and T, the Hamming distance between them is at least $m/2$. The mutation hitting probability is $(1/n)^{m/2}(1-1/n)^{n-m/2}$. However, there are $\binom{m}{m/2}^2$ points in T, so the probability of hitting T is increased by this amount. Bounding the number of points in T via a standard binomial coefficient inequality [2], the result is $\binom{m}{m/2}^2 \leq (2e)^{m/2}$. Thus the probability of hitting T from $p_i \in P$ is bounded by $(1/n)^{m/2}(1-1/n)^{n-m/2}(2e)^{m/2} \leq (2e/n)^{m/2} < 2^{-\Omega(n)}$. $\square$

Theorem 11. *The (2+1) EA will optimize the* $IT1_n^u(x)$ *function in expected* $O(n^k)$ *steps and within* $O(n^k \ln n)$ *steps with probability* $1 - O(1/n)$.

Proof. Referring to Lemma 10, the next step is to establish the expected waiting time to discover b^{**} from a population of $\{a^{**}, p_i \in P\}$. The Hamming distance between a^{**} and b^{**} is defined to be constant k where $n = 6m$ is chosen so that $3 < k < m/4$. Thus the probability of mutating from a^{**} to b^{**} in one step is $p = (1/n)^k(1-1/n)^{n-k}$. This is bounded below by $1/(en^k)$, resulting in an expected waiting time that is bounded above by $en^k = \Theta(n^k)$. Note that this is the best case possibility of finding b^{**} from any point on P as the Hamming distance for all points in P is $H(p_i \in P, b^{**}) \geq k$.

---

[2] $\binom{n}{k} \leq \left(\frac{en}{k}\right)^k$

Applying Chernoff bounds, the probability of finding b^{**} within $2en^k \ln n$ steps is $1 - O(1/n)$. From Lemma 10 it is known that the probability of finding T from any point in P is exponentially small. Thus the probability of finding T before finding b^{**} is also exponentially small. ☐

Lemma 12. *The* $(2+1)$ *GA with uniform crossover will discover a point in* $P_2 \cup T \cup \{a^*\}$ *in* $O(n^2)$ *steps with probability* $1 - 2^{-\Omega(n)}$. *The probability of the* $(2+1)$ *GA with uniform crossover finding* $\{b^{**}\}$ *while searching for* $P_2 \cup T \cup \{a^*\}$ *is* $2^{-\Omega(n)}$.

Proof. Lemma 2 of [301] proves the first part of the result. For the second result, note that b^{**} contains $5m - k$ ones. Recall that k is a constant, and that $n = 6m$ is chosen so that $3 < k < m/4$. It has already been shown that as long as the points in the population contain at least $4m$ ones, the probability of finding b^{**} is exponentially small. The remaining possibility is mutating to b^{**} from a point in the population $p_i \in \{P_1 - a^*\}$ where $4m < i < 5m$. It is easy to see that it is exponentially unlikely that the other point of the population is not in $\{a_i \in P_1 \; i \geq m\}$. The minimum Hamming distance between a point of the population and b^{**} is $2m - k$, so the probability of finding b^{**} by mutation is at most $2^{-\Omega(n)}$. Turning to the crossover operator, recall that $b^{**} = 1^m 0^k 1^{4m-k} 0^m$. Both members of the population are of the form $0^{n-i}1^i$ for $m \leq i < 5m$ so both points have 1s in the last m positions. Thus, it is impossible to cross the two points in the population to produce a point with Hamming distance less than m from b^{**}. ☐

Proposition 13. *With probability* $O(1/n)$, *the* $(2+1)$ *GA will find a point in* $P_2 \cup T \cup \{b^{**}\}$ *before finding* a^*.

Proof. The proof of Theorem 4 of [301] shows this result without reference to b^{**}. The Hamming distance from P_1 to b^{**} is exponential, and thus does not change the result. ☐

Lemma 14. *If the population contains a^*, the $(2+1)$ GA will find a point in T in $O(n^2)$ steps with probability $1 - O(1/n^k)$.*

Proof. Lemma 3 of [301] shows the result (without reference to b^{**}) with probability $1-2^{-\Omega(n)}$. First, consider the possibility that b^{**} is found before T. To start, consider the possibility of finding b^{**} by crossover plus mutation from a population of a^* and any other point $a_i \in P$. For $0 \leq i \leq 5m$ this is exponentially unlikely via the argument given in the proof of Lemma 12. For $5m < i \leq 6m$ this results in crossover on a^* and a_i setting the last m positions to 1. Yet b^{**} has zeros in these positions, so subsequent mutation must flip at least m bits. Finally, if the other point is $a_{6m+j} = 1^{n-j}0^j$ for $0 < j \leq m$, then a^* and a_{6m+j} agree in $k+m-j$ bits different from the corresponding bits of b^{**}. Thus crossover and subsequent mutation of at least those $k+m-j$ bits is required, giving a probability of discovering b^{**} bounded above by $O(1/n^{k+m-j})$. As long as a_i is not a^{**}, a better point on P will be discovered with probability $1/(3en)$. From this and the bounds derived above, it can be seen that either a^{**} or a point of T will be found with probability $1 - O(1/n^k)$.

Now assume the population $\{a^*, a^{**}\}$. The one-step probability of finding b^{**} by either mutation or crossover followed by mutation is $p = O(1/n^k)$ whereas the one-step probability of discovering T was shown to be $q = \Theta(1/n)$ in Lemma 3 of [301] by an application of Sterling's formula [371]. There is a sequence of independent trials until one or the other of these outcomes happens. A probability argument[3] shows that the probability of finding b^{**} over all trials is $p/(p+q) = O(1/n^k)/(O(1/n^k)+O(1/n)) = O(1/n^k)/O(1/n) = O(1/n^{k-1})$. □

---

[3] Given that either event A or B will eventually happen, let $p := Pr[A]$, $q := Pr[B]$ and $r := 1 - p - q$. The probability that A eventually happens is $p/(1-r) = p/(p+q)$.

Lemma 15. *The expected waiting time to hit* b^{**} *from a population* $\{t_i \in T, t_j \in T\}$ *is exponential for the (2+1) GA with uniform crossover.*

Proof. It is possible for a crossover plus mutation operation to get b^{**} from two elements of T. Remember that $b^{**} := 1^m 0^k 1^{4m-k} 0^m$. If the two population elements of T are binary complements of each other in the b and c regions, and if the crossover mask is chosen correctly, crossover could get the first and last m bits of the child to match b^{**}. Then mutation would need to get the k bits of the middle 1^{4m} bits to match b^{**}. The probability of getting the correct crossover mask is 2^{-2m}. Thus the probability of getting the correct mask and the correct mutation is bounded above by $O(2^{-2m})$.

Another possibility would be for crossover to get all but $0 \leq j \leq 2m$ of the first and last m bits correct. These correspond to the substrings b and c from the definition of T, $b1^{4m}c$ where b and c contain exactly half 1s. It is not necessary for these j bits of the crossover mask to be correct, thus the probability of choosing the correct crossover mask is 2^{-2m+j}. Following crossover, mutation must correct $k+j$ bits, with probability $(1/n)^{k+j}(1-1/n)^{n-k-j} \leq (1/n)^{k+j}$. Consequently, the probability of getting the crossover mask right and the correct mutation is $\leq (1/n)^{k+j}(1/2)^{2m-j} \leq (1/2)^{2m+j}$ which is exponentially small. □

Theorem 16. *The (2+1) GA with uniform crossover will need exponential time steps to optimize* $IT1^u_n(x)$ *with probability* $1 - O(1/n)$.

Proof. Beginning from Proposition 13 and Lemma 14 above, assume the population contains a point in T. By the selection method of the GA, once a member of T exists in the population we should only have to wait constant time $O(1)$ for both members of the population to be in T. Once the GA contains two members of T, the probability of crossover plus mutation or mutation alone discovering b^{**} is exponentially unlikely by Lemma 15. Of the various bad events, the probability from Proposition 13 of skipping a^* is maximal at $O(1/n)$. □

5.4.3 Lowering the Failure Probability

A simple tweak can be made to lower the $O(1/n)$ failure probability that the GA takes exponential time. We again follow Storch [301] in introducing a k-bit barrier in P. Here j is used instead of k, as k is taken in the notation of Ignoble Trails.

$$IT2_n^u(x) := \begin{cases} 16m & x = b^{**} \\ 15m & x \in T \\ 14m & x = a^* \\ 6m + i & x = a_i \in P_1 \cup P_2 \\ 6m - \|x\| & x \in R := \{0,1\}^n - P - T - \{b^{**}\} \end{cases}$$

Note that the fitness specifications of $IT2_n^u(x)$ are unchanged from $IT1_n^u(x)$. The definition of P_2 is altered from (a_{5m+1}, a_{7m}) to (a_{5m+j+1}, a_{7m}) where j is some constant $1 \geq j \geq m-1$. This means that the points $(a_{5m+1}, ..., a_{5m+j})$ are now in R and form a j-bit barrier to jump from $P_1 \cup a^*$ to P_2. Take $j = k$ for convenience.

Theorem 17. *The (2+1) EA without crossover will optimize the $IT2_n^u(x)$ function in expected $O(n^k)$ steps and within $O(n^k \ln n)$ steps with probability $1 - O(1/n)$.*

Proof. The methods of Theorem 11 for proving the waiting time to jump the k-bit barrier between a^{**} and b^{**} is trivially reused here to jump the k-bit barrier from a^{**} to P_2. Thus the result of Theorem 11 holds for $IT2_n^u(x)$ as introducing a second k-bit barrier does not alter the bounds on the waiting time. □

Theorem 18. *The (2+1) EA without crossover will optimize the* $IT2_n^u(x)$ *function in expected* $O(n^{k+1})$ *steps with exponentially small failure probability.*

Proof. Again based upon Theorem 11 this is established by an application of Chernoff's bound. $\square$

Theorem 19. *The (2+1) GA with uniform crossover will need exponential time steps to optimize* $IT2_n^u(x)$ *with probability* $1 - O(1/n^k)$.

Proof. The results from Theorem 16 and dependent Propositions and Lemmas transfer with the exception of the bad event probability of skipping a^*. The proof of Theorem 5 of [301] shows the probability of skipping a^* with the same k-bit barrier in P is $O(1/n^k)$ without reference to b^{**}. The Hamming distance from P_1 to b^{**} is exponential, and thus their result transfers. $\square$

Thus the GA's failure probability has been lowered with only a polynomial running time penalty for the EA.

5.5 Ignoble Trails for One-point Crossover

Storch and Wegener [301] present two additional functions $R_n^1(x)$ and $R_n^{1*}(x)$ and show that a (2+1) GA optimizes both in polynomial time while the (2+1) EA requires exponential time to optimize. In this section the goal is to again invert those results with careful modification of the functions. The first function is a modest variant of $R_n^1(x)$ and results are proven for it.

Here Storch's definition of a one-point crossover variant of $R_n^u(x)$ is described. The crossover target T is changed to be the single point 1^n. The path P_2 from a^* to a^{**} is moved as follows. P_2 is defined as the points $(a_{5m+1}, ..., a_{7m})$ where $a_{5m+2i-1} := 1^i 0^{m-i} 1^{5m-i+1} 0^{i-1}$ and $a_{5m+2i} := 1^i 0^{m-i} 1^{5m-i} 0^i$ for $1 \leq i \leq m$. For example the point a_{5m+1} is now defined to be $10^{m-1}1^{5m}$, with Hamming distance 1 from $a^* := a_{5m}$.

All points on P_2 have $5m$ ones except a_{5m+1} which has $5m+1$ ones. As such, they are exponentially far from $T := 1^n$. Note that the new P_2 cuts through T of $IT1^u_n(x)$. It also appears at first glance to cut through b^{**} as well, however this is not the case. Remember that $b^{**} := 1^m 0^k 1^{4m-k} 0^m$ with k defined to be a constant where $n = 6m$ is chosen so that $3 < k < m/4$. Thus b^{**} contains $5m - k$ ones and the Hamming distance between b^{**} and the closest points of P_2 is exactly k.

Next we parallel the modifications to create $IT3^1_n(x)$. Inverting the results of Storch require that both population points accumulate in T, so define T to be $\{0^1 1^{n-1}, 1^n, 1^{n-1}0\}$. For completeness here are the fitness specification (unchanged from the other Ignoble Trails) and a diagram of $IT3^1_n(x)$.

$$IT3^1_n(x) := \begin{cases} 16m & x = b^{**} \\ 15m & x \in T \\ 14m & x = a^* \\ 6m + i & x = a_i \in P_1 \cup P_2 \\ 6m - \|x\| & x \in R := \{0,1\}^n - P - T - \{b^{**}\} \end{cases}$$

5.5.1 Behavior of the EA and GA on Ignoble Trail 3

The heuristic description of the optimization of $IT3^1_n(x)$ is identical to $IT1^u_n(x)$. Figure 5.5 displays the proof digram. The $(2+1)$ EA optimizes R to find P_1 which leads to P_2 and a^{**}. The EA then waits expected $O(n^k)$ steps to jump to b^{**}. The

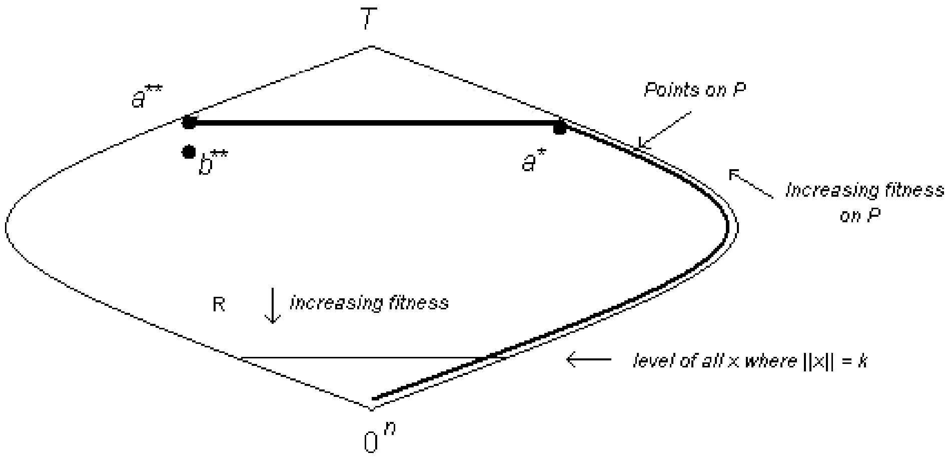


Figure 5.4: Illustration of Ignoble Trail 3

(2 + 1) GA follows the same route and jumps to T via crossover in expected $O(n^2)$ steps.

5.5.2 Time Complexity Results

Theorem 20. *The (2+1) EA without crossover will optimize the* $IT3^1_n(x)$ *function in expected* $O(n^k)$ *steps and within* $O(n^k \ln n)$ *steps with probability* $1 - O(1/n)$.

Proof. The arguments from Theorem 11 and its dependent Lemmas can be reused here without modification. □

Theorem 21. *The (2+1) GA with one-point crossover will need exponential time steps to optimize* $IT3^1_n(x)$ *with probability* $1 - O(1/n)$.

Proof. Begin from Lemma 12 and Proposition 13, which apply unchanged here. Let $T_i \in T$ be $T := 1^n$. It is necessary to find the probability of reaching T_i from the population $\{a^* := 0^m 1^{5m}, a^{**} := 1^{5m} 0^m\}$ via one-point crossover. Any crossover point chosen from the middle $4m$ bits will result in a child at 1^n. The probability of this event is $4m/6m = 2/3$, thus one must wait only an expected constant number of steps for a correctly chosen crossover point. The probability of choosing the only two

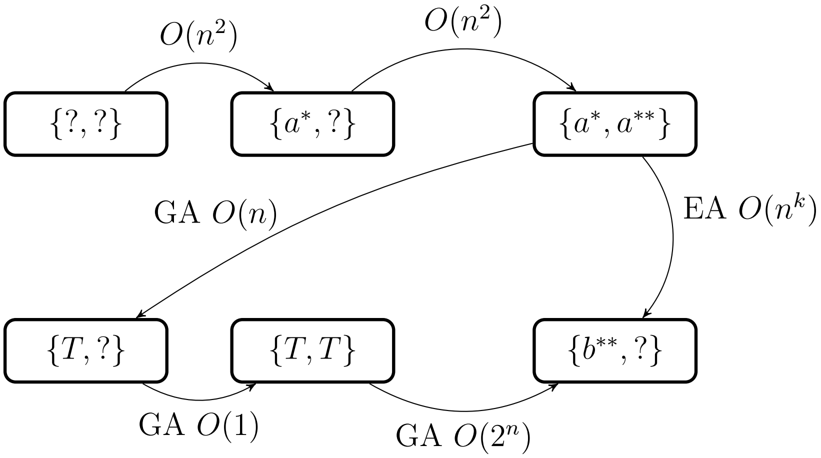


Figure 5.5: Diagram of proofs of Lemmas and Theorems for $IT3^1_n(x)$. The labels on each arc refer to the expected waiting time to transition from state to state for the labeled algorithm.

correct crossover points for producing the remaining two points in T is $2/n$ and this event is ignored. Once a single point in T is found, the probability of discovering either of the other points in T via mutation is $O(1/n)$. Hence after a linear number of steps both population points are in T. It is easy to see from the $O(m)$ Hamming distance from T to b^{**} that exponential time would be required to escape T. The bad-event probability from Lemma 14 of skipping a^* remains unchanged at $O(1/n)$. □

5.5.3 Lowering the Failure Probability

Storch only notes that it is possible to lower the bad-event probability of skipping a^* to $O(1/n^k)$ by again altering P_2 to contain a k-bit barrier between a^* and the first point in P_2. This modification is explicitly performed (as in $IT2^u_n(x)$) to create $IT4^1_n(x)$. However, the result here is kept rigorous.

Let $IT4^1_n(x)$ be identical to $IT3^1_n(x)$ with the exception that P_2 is defined to be the path (a_{5m+j+1}, a_{7m}) where j is some constant taken to be the same as k.

$$IT4^1_n(x) := \begin{cases} 16m & x = b^{**} \\ 15m & x \in T \\ 14m & x = a^{*} \\ 6m + i & x = a_i \in P_1 \cup P_2 \\ 6m - \|x\| & x \in R := \{0,1\}^n - P - T - \{b^{**}\} \end{cases}$$

Theorem 22. *The (2+1) EA without crossover will optimize the* $IT4^1_n(x)$ *function in expected* $O(n^k)$ *steps and within* $O(n^k \ln n)$ *steps with probability* $1 - O(1/n)$.

Proof. The methods of Theorem 11 for proving the waiting time to jump the k-bit barrier between a^{**} and b^{**} is trivially reused here to jump the k-bit barrier from a^{**} to P_2. Thus the result of Theorem 11 holds for $IT4^1_n(x)$ as introducing a second k-bit barrier does not alter the bounds on the waiting time. □

Theorem 23. *The (2+1) GA with one-point crossover will need exponential time steps to optimize* $IT4^1_n(x)$ *with probability* $1 - O(1/n^k)$.

Proof. Similar to Theorem 19, the results from Theorem 16 and its dependents transfer with the exception of the bad event probability of skipping a^*. The proof of Theorem 5 of [301] shows the probability of skipping a^* with the same k-bit barrier in P is $O(1/n^k)$ without reference to b^{**}. The Hamming distance from P_1 to b^{**} is exponential, and thus their result transfers. □

5.6 Closing Remarks and Future Ideas

The above is believed to be the first proven example of a situation where a crossover based GA is expected to be exponentially outperformed by an EA without the crossover operator for both uniform and one-point crossover.

As a conjecture, we speculate that a correctly specified restarting mechanism for $IT1$ and $IT3$ *may* help the GA hit the failure event of skipping a^*. This was explored a bit on paper. The effort was confounded by the fact that the lower-bound on the probability of skipping a^* is quite small, along the lines of $(1/n)^{2m}(1 - 1/n)^{-2m}$, meaning that the restarting technique may not work very well.

These results do not extend to any functions other than those described here, though there are undoubtedly many more of these types of functions to discover. The next chapter will present Ignoble Trail functions to duplicate the outcomes of this work with an arbitrary population EA and GA.

CHAPTER 6

IGNOBLE TRAILS FOR ARBITRARY POPULATION SIZE

> Darwin's "survival of the fittest" was replaced by Kimura's "survival of the luckiest,"
>
> Giorgio Bernardi [372]

Framework	Mutation-only EAs	Crossover-enabled GAs
Time Complexity Analysis	✓	✓
Dynamical Systems Analysis		
Empirical Analysis		
Methods: Finite population size, Theorem-Proof		

Figure 6.1: Chapter 6 Framework

6.1 Introduction

The chapter examines a large population EA and GA against new fitness functions. Time complexity analysis is performed in a theorem-proof method. Figure 6.1 illustrates the contributions within the framework.

Extending the results of Richter et al [373] and following the methods of Jansen and Wegener [295] a new set of Ignoble Trails functions are derived from the Real Royal Road functions. These functions are optimized with arbitrary finite sized populations and proven to take polynomial time for a mutation-only EA versus exponential time for the crossover-enabled GA, inverting the results of Jansen and Wegener.

6.2 Steady State Evolutionary Algorithms

These algorithms are instances of steady-state evolutionary algorithms [311] where the population is not fully replaced at each generation. Let $s(n)$ be the size of the population. First a GA is given with a crossover operator, then an EA without crossover is given.

Algorithm 4.6: The Steady State GA

```
1  Select independently and randomly the s(n) individuals of the initial population.
2  With probability $p_c(n)$ go to Step 3' and with the remaining probability of 1 -
   $p_c(n)$ go to Step 3". Steps 3' and 3" are mutually exclusive.
3' Choose two parents $x$ and $y$ from the current population. Let $z^*$ be the result of
   uniform crossover applied to $x$ and $y$ and let $z$ be the result of mutation applied
   to $z^*$.
3" Choose one parent $x$ from the current population. Let $z$ be the result of muta-
   tion applied to $x$.
4  If the fitness of $z$ is smaller than the fitness of the worst individual of the current
   population, go to Step 2. Otherwise, add $z$ to the population. Let $W$ be the
   multi-set of individuals in the enlarged population which all have the worst
   fitness and let $W'$ be the set of those individuals in $W$ which have the largest
   number of copies in $W$. Eliminate randomly one element in $W$ from the current
   population. Go to Step 2.
```

Algorithm 4.7: The Steady State EA

- Set probability $p_c(n) = 0$ and run Algorithm 4.6.

The above algorithms do not specify the parental selection method of step (3) of the algorithm. It is presumed that if $f(x) \geq f(y)$ then the probability to select x is greater than or equal to the probability to select y. Note also that the truncation selection in step 4 tends to push the lowest fitness class of the population to have a uniform number of duplicates.

Next new fitness functions are derived from the RRR family that target uniform crossover and one point crossover.

6.3 Ignoble Trails for Uniform Crossover

The Real Royal Roads of Jansen and Wegener [295] are as follows. Let n be the string length and $n = 2m = 6j$. The string $x \in \{0,1\}^n$ is split into two equal strings

$\{x', x''\}$ both with length m. The substring x'' is subdivided further into three strings $\{x''_1, x''_2, x''_3\}$, all with length j. The symbol j is used rather than the k notation from [295].

Let x'' be on circle C if $x'' \in \{0^i 1^{m-i}, 1^i 0^{m-i} | 0 \leq i \leq m-1\}$. The points on C form a closed Hamming path of length $2m$ with each point having two neighboring points reachable with a single 1-bit change in x''.

The region T is defined to be any x'' where all substrings x''_1, x''_2, x''_3 contain $\lceil j/2 \rceil$ zeros and $\lfloor j/2 \rfloor$ ones. Let $H(a, B)$ be the Hamming distance between point a and set B where the value of $H(a, B)$ is the minimum distance between a and all points in B.

$$R^*_{n,m}(x', x'') := \begin{cases} n - H(x'', C) & : x' \neq 0^m \text{and } x'' \notin C \\ 2n - H(x', 0^m) & : x'' \in C \\ 0 & : x' = 0^m \text{and } x'' \notin C \cup T \\ 3n & : x' = 0^m \text{and } x'' \in T \end{cases}$$

We propose the fitness function $IT5^u_{n,m}(x', x'')$, a modified form of $R^*_{n,m}(x', x'')$. Let the Hamming ring C be $x'' \in \{0^i 1^{m-i}, 1^{m-i} 0^i | 0 \leq i \leq m-1\}$. Let k be some constant chosen so that $k > 6$. Assume that n is sufficiently large that $k < m/4$. The ring is now divided into two segments $C_1 : x'' \in \{0^i 1^{m-i}, 1^{m-i} 0^i | 0 \leq i \leq m/2\}$ and $C_2 : x'' \in \{0^i 1^{m-i}, 1^{m-i} 0^i | m/2 + k \leq i \leq m-1\}$. Note that C_1 and C_2 have lengths of $n/4$ and $n/4 - 2k$ points respectively. C_1 has flat fitness, while C_2 has a fitness slope to the global optimal point $x'' = 1^m$. Thus, there exists a k bit gap between the segments on each side of the bisected ring C. Let $a^* = 0^{m/2} 1^{m/2} \in C_1$ and $a^{**} = 0^{m/2+k} 1^{m/2-k} \in C_2$, similarly let $b^* = 1^{m/2} 0^{m/2} \in C_1$ and $b^{**} = 1^{m/2+k} 0^{m/2-k} \in C_2$. All points where $x' = 0^m$ and $x'' \neq C_1 \cup C_2 \cup T$ are set to fitness zero.

$$IT5^u_{n,m}(x',x'') := \begin{cases} n - H(x'', C_1) & : x' \neq 0^m \text{and } x'' \notin C_1 \\ 2n - H(x', 0^m) & : x'' \in C_1 \text{and } x' \neq 0^m \\ 2n & : x'' \in C_1 \text{and } x' = 0^m \\ 0 & : x' = 0^m \text{and } x'' \notin C_{1,2} \cup T \cup 1^m \\ 4n & : x' = 0^m \text{and } x'' \in T \\ 5n - H(x', 1^m) & : x' = 0^m \text{and } x'' \in C_2 \end{cases}$$

The numeral 5 signifies the fifth member in the family of Ignoble Trails. While the following section follows the results of [295] it should be noted that their results use expected time complexity. Here all results must be stated with probabilistic bounds and account for multiple failure probabilities due to 'fork in the road' events. Let $p_c(n) \leq 1-\varepsilon$, where $0 < \varepsilon < 1$. While [295] sets $s(n) \geq n$, here it is set to be $s(n) \geq n$ and $s(n) = \Theta(n)$ for reasons to be explained below. The expression of results with $s(n)$ is retained for comparability to [295].

6.3.1 Behavior of the EA and GA on Ignoble Trail 5

The expected optimization path of both algorithms follow various phases. These are detailed below in a diagram of the states each algorithm moves through. Figure 6.2 also represents an outline of the Theorems and Lemmas to follow.

6.3.2 Time Complexity Results

Here phase 1 is to find the half-ring C_1 for segment x'' for both the EA and GA.

Lemma 24. *All individuals in the steady-state EA and GA with uniform crossover have* $x'' \in C_1 \cup T \cup C_2 \cup \{1^m\}$ *within* $O(n^2 s(n) \log s(n))$ *steps with probability* $1-2^{-\Omega(n)}$.

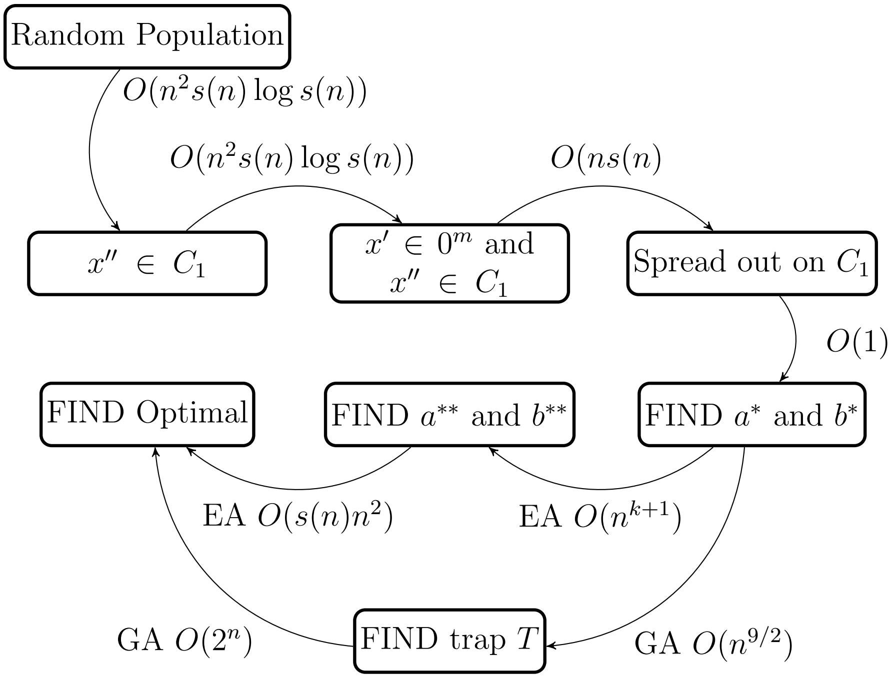


Figure 6.2: Diagram of proofs of Lemmas and Theorems for $IT5_n^u(x)$. The labels on each arc refer to the Chernoff bound waiting time to transition from state to state. Those arcs labeled with EA or GA are the expected paths of the respective algorithm.

Proof. The results of Claim 12 of [295] are adapted. Assume they do not find $x'' \in \{T \cup C_2\}$ and ignore the crossover operator for the GA. Also assume that no individuals exist with $x' \neq 0^m$ as initializations to this region are exponentially unlikely.

For individuals $x = x'x''$ where $x' \neq 0^m$ and $x'' \notin C_1$ the goal is to be replaced with strings where $x'' \in C_1$. Define distance d to be the sum of $H(x'', C_1)$ for all strings $x' \neq 0^m$ and $x'' \notin C_1$. Due to the selection and deletion procedure, this distance is monotonically decreasing during optimization. Distance d is at most $s(n)m$ and decreases for any mutation of x to z where $H(z'', C_1) < H(x'', C_1)$. Given a selection of x where $x \neq 0^m$ and $x'' \notin C_1$, there are $\lceil d/s(n) \rceil$ single bit mutations to decrease d via creation of z. Replication (via selection, neutral mutation and displacement of weaker strings) of any string on or off C_1 also decreases d. Thus the expected waiting time to decrease d is $O(ns(n)/d)$ with a bound of $O(n^2 s(n)/d)$

steps with probability $1 - O(2^{-n})$ via Chernoff. Combining the waiting times results in $O(n^2 s(n) \log(s(n)m)) = O(n^2 s(n) \log s(n))$.

$$\begin{aligned}\sum_{d=1}^{s(n)m} O(n^2 s(n)/d) &= O(n^2 s(n) \sum_{d=1}^{s(n)m} 1/d) \\ &= O(n^2 s(n) \log(s(n)m)) \\ &= O(n^2 s(n) \log s(n))\end{aligned}$$

The failure probability of any one decrease in d establishes the failure probability of the combined waiting time. □

The second phase of optimization is to find $x' = 0^m$ for both algorithms.

Lemma 25. *Starting from* $x'' \in C_1 \cup T \cup C_2 \cup \{1^m\}$, *all individuals in the steady-state EA and GA with uniform crossover have* $x' = 0^m$ *and* $x'' \in C_1 \cup T \cup C_2 \cup \{1^m\}$ *within* $O(n^2 s(n) \log s(n))$ *steps with probability* $1 - 2^{-\Omega(n)}$.

Proof. The results of Claim 13 in [295] are adapted. Assume they do not find $x'' \in \{T \cup C_2 \cup 1^m\}$ and ignore the crossover operator for the GA. Define distance d to be the sum of $H(x', 0^m)$ for all strings $x' \neq 0^m$ and $x'' \in C_1$. Distance d is bounded above by $s(n)m$. Due to the algorithm's selection and deletion operator, this distance is monotonically decreasing during optimization. Any mutation of x'' to z'' where $H(z', O^m) < H(x', 0^m)$ also decreases d. Given a selection of x where $x \neq 0^m$ and $x'' \notin C_1$, there are $\lceil d/s(n) \rceil$ single bit mutations to decrease d via creation of z. Replication (via selection, neutral mutation and displacement of weaker strings) of any string on or off 0^m also decreases d. Thus the expected waiting time to decrease d is $O(ns(n)/d)$. Via Chernoff the bound is $O(n^2 s(n)/d)$ steps with probability $1 - O(2^{-n})$. The remaining steps follow Lemma 24. □

The third phase of optimization is to spread out and discover all points on the half-ring C_1.

Lemma 26. *Algorithm has achieved the state of all population members in* $x' = 0^m$ *and* $x'' \in C_1$*. The waiting time for the steady-state EA and GA with uniform crossover to discover all strings in the set* $x' = 0^m$ *and* $x'' \in C_1$ *at least once in expected* $O(ns(n))$ *steps and within* $O(n^2 s(n))$ *steps with probability* $1 - 2^{-\Omega(n)}$.

Proof. Beginning at the end-state of Lemma 25, Claim 14 of [295] is closely followed. Assume that neither T or C_2 is reached. All points in C_1 have one or two neighboring points of Hamming distance one away. Recall the previous assumption that $s(n) \geq n$. Let x be a point such that it has a Hamming neighbor $\tilde{x} = (\tilde{x}' = 0^m, \tilde{x}'' \in C_1)$ that is not in the current population. Let d be the number of such points $\tilde{x}$ not contained in the population. As in [295], two different bounds can be derived to decrease d to zero. The following ignores the crossover operator as it only serves to increase the probability of reaching the goal. First, the neighbor x can be selected directly and mutated properly to $\tilde{x}$. The probability of this event is $1/(ns(n))$ giving an expected waiting time of $O(ns(n))$. Via Chernoff's bound, an upper bound of $O(n^2 s(n))$ steps with probability $1 - 2^{-\Omega(n)}$ is found. The second scenario is omitted here, see Claim 14 of [295], as the resulting $O\left(n^3 + s(n)^2 \log n\right)$ upper bound is always greater than first scenario's upper bound when $s(n) \geq n$. □

Phase four involves finding the special points a^* and b^* of C_1.

Claim 27. *Since all members of* C_1 *have been discovered at least once, both* a^* *and* b^* *are contained in the population of the EA and GA. Furthermore, the expected number of copies of each is* $O(s(n)/n)$.

Proof. Lemma 26 established discovery of these two points. The truncation-selection methods of the EA and GA specify that if the new child resulting from mutation or crossover + mutation is accepted, a random individual of the worst fitness level with the highest number of replicants is removed from the population. Thus, the flat fitness of C_1 and the fact that $s(n) \geq n$ implies that once a^* and b^* are found, they cannot be displaced by subsequent discoveries of other members of C_1. Given the truncation-selection method and that the length of C_1 is $n/4$ and $s(n)$, the number of copies of both points is expected to be $4s(n)/n = O(s(n)/n)$ after the completion of Lemma 26. □

At this point the behavior of the algorithms diverges. For the EA, the fifth phase is to cross the k-bit gap on the ring C. For the GA, the fifth phase is to discover a point in T, the crossover trap, and accumulate the population to T.

Lemma 28. *Given a population containing a^* and b^*, the steady-state EA discovers at least one of the points a^{**} and b^{**} in expected $O(n^{k+1})$ steps and within $O(n^{k+1} \ln n)$ steps with probability $1 - O(1/n)$.*

Proof. The Hamming distance between a^* and a^{**} is defined to be constant k, and chosen so that $6 < k < m/4$ where $n = 2m$. Thus the probability of mutating from a^* to a^{**} in one step is $p \geq (1/n)^k(1 - 1/n)^{n-k}$. Probability p is bounded below by $1/(en^k)$. If the number of copies of either point is expected to be $4s(n)/n$, then the probability of selecting either point is $1/n$, resulting in a combined probability of $1/(en^{k+1})$. Thus the expected waiting time is bounded above by $O(n^{k+1})$. Via Chernoff bounds, the probability of finding a^{**} within $O(n^{k+1} \ln n)$ steps is $1-O(1/n)$. □

Next one needs to calculate the failure probability of the EA reaching T before it jumps the gap to C_2.

Lemma 29. *Given a population containing a^* and b^*, the steady-state EA will discover T before C_2 with probability $2^{-\Omega(n)}$.*

Proof. The Hamming distance between any point on C_1 and T is $\Theta(n)$, thus this event has exponentially small probability. □

Turning back to the GA, its characteristics for reaching T and C_2 are examined. We must address the concern that some members of the GA population might find the set C_2 during phases 1 to 4. It is shown that in the time that it takes to complete these phases, points of the population have a small probability of jumping the gap from C_1 to C_2.

With the assumption that $s(n) = \Theta(n)$, phases 1 to 4 of the GA will be completed within $O(n^3 \log n) \leq O(n^4)$ steps with probability $1 - 2^{-\Omega(n)}$, so the probability that a point of the GA population will find C_2 during these phases is $O(1/n^3)$.

Lemma 30. *Given a population all within C_1 and containing a^* and b^*, the steady-state GA with crossover will discover T within $O(n^{9/2})$ steps with probability $1 - O(1/n^{k-7/2})$.*

Proof. Importing arguments from Claim 15 of [295], only steps with crossover are considered. Recall that the population is such that all members have $x' = 0^m$, $x'' \in C_1$. Also recall the probability of choosing either a^* or b^* as parents is $4/n$, thus the probability of selecting both is $O(1/n^2)$. Next, uniform crossover is applied on $x = a^*$ and $y = b^*$ to produce $\tilde{z}$, $\tilde{z}' = 0^m$. Given that x'' and y'' are binary compliments, then $\tilde{z}''$ is a random binary string. Recall that T is made up of strings $t'' = (t''_1, t''_2, t''_3)$, where t''_j contains exactly $\lfloor j/2 \rfloor$ ones and $\lceil j/2 \rceil$ zeros. Note that $n = 6j$. The probability that some z''_i of the random string z'' obeys that criteria is $\Theta(j^{-1/2})$ by Sterling's formula [371].

The probability that subsequent mutation does not move z out of T is 1/4. This results in a probability of $\Theta(j^{-3/2})$ that $z'' \in T$. Given that the population is all within C_1 and given its flat fitness, the expected selection probability of a^* or b^* is $\Theta(1/n)$. Thus the combined probability of selecting both points and mutating is $\Theta(n^{-2}j^{-3/2}) = \Theta(n^{-7/2})$, giving an expected waiting time for success of $O(n^{7/2})$. Via Chernoff, success will happen within $O(n^{9/2})$ steps with probability $1 - 2^{-\Omega(n)}$.

There are other possibilities for choosing members of C_1 to create individuals in T, however the above is the most likely possibility. The remaining step is to find the failure probability of finding C_2 before T. The probability that a point of the population will jump the gap from C_1 to C_2 in one step of the algorithm is at most $1/n^k$. It was just shown that once the conditions of claim 27 have been met, the probability that T will be found in one step of the algorithm is $\Theta(n^{-7/2})$. Using a result from probability theory [1] , the probability of finding C_2 before T is bounded above by:

$$\frac{n^{-k}}{n^{-7/2} + n^{-k}} = \frac{n^{7/2}}{n^k + n^{7/2}} = O(1/n^{k-7/2})$$

□

Lemma 31. *The probability of the GA finding C_2 before T steps is $O(1/n^{k-7/2})$.*

Proof. Note that the probability of initialization into C_2 or by mutation into C_2 from outside of C_1 is exponentially small. See the failure probability of Lemma 30. □

Lemma 32. *Given a population containing one element of T, the steady-state GA with crossover will accumulate all population members in T within $O(s(n) \log s(n))$ steps with probability $1 - O(n^{k-7/2})$.*

---

[1] Let $p := Pr[A]$, $q := Pr[B]$ and $r := 1 - p - q$. Given that A or B will eventually happen, the probability that A is first is $p/(1-r) = p/(p+q)$.

Proof. Replication (via selection, neutral mutation and displacement of weaker strings) of any string in T decreases the number of individuals of weaker fitness off of T. Define distance d to be the number of individuals not in T, with a goal of $d = 0$. Due to the GA's selection operator, d is monotonically decreasing during optimization. Distance d may also decrease via any mutation from x'' to $z'' \in T$. Distance d is at most $s(n)$. Summing over the waiting times one gets:

$$\begin{aligned}\sum_{d=1}^{s(n)} O(s(n)/d) &= O(s(n)\sum_{d=1}^{s(n)} 1/d) \\ &= O(s(n)\log s(n))\end{aligned}$$

Recall from Theorem 30 the probability of discovering T before C_2 is bounded by $1 - O(n^{k-7/2})$. □

For the EA, the sixth phase of optimization involves optimizing C_2, the global optimum. The sixth phase of the GA is the attempt to escape the trap T.

Lemma 33. *Given a population containing at least one point in* C_2*, the steady-state EA will find the global optima* $x' = 0^m, x'' = 1^m$ *within* $O(s(n)n^2)$ *steps with probability* $1 - 2^{-\Omega(n)}$.

Proof. Each point on the path C_2 has a better fitness Hamming neighbor. The probability of picking a point in C_2 and then mutating to the better neighbor is at least $p = 1/(3ens(n))$. Note that the length of C_2 is $O(m)$. Inverting and substituting, an expected waiting time of $O(n)$ successful events with probability p is derived. It is not necessary that a majority of population members accumulate into C_2, though this is a side effect of any replication events via the truncation selection process. Applying Chernoff's bound the above result is shown. □

Lemma 34. *Given a population all within T, the steady-state GA will need an exponential number of steps to find the global optimum $x'' = 1^m$ with probability $1-2^{-\Omega(n)}$.*

Proof. It is easy to see from the $O(m)$ Hamming distance from T to all points on C_2, including the optimal 1^m, that exponential time would be required to escape T. Applying Chernoff's bound the above result is shown. $\square$

Theorem 35. *The steady-state GA with uniform crossover will need exponential time steps to optimize $IT5^u_{n,m}(x', x'')$ with probability $1 - O(1/n^{3/2})$.*

Proof. Summing the relevant optimization phases of the GA, note that the waiting time is exponential in n to optimize $IT5^u_{n,m}(x', x'')$. The failure probability of this bound is the smallest of the above failure events. $\square$

Theorem 36. *The steady-state EA will optimize $IT5^u_{n,m}(x', x'')$ in expected $O(n^{k+1})$ steps and within $O(n^{k+1} \ln n)$ steps with probability $1 - O(1/n)$.*

Proof. Summing the relevant optimization phases of the GA, the waiting time is dominated by the waiting times from Lemma 28. The Lemma 28 exponential failure probability of finding T before C_2 does not exceed the Chernoff bound failure probability from Lemma 29. $\square$

It is worth noting the relationship between $s(n)$ and k. The above assumes $s(n) = \Theta(n)$ and $k > 6$. We believe that this is the example of a GA that has been proven to need exponential optimization time versus a mutation only EA taking polynomial time on a given fitness function.

6.4 Ignoble Trails for One-point Crossover

Jansen and Wegener [295] also present a function $R_{n,m}$ and prove that the above GA with one-point crossover optimizes the function in polynomial time while the EA requires exponential running time. The GA for the below fitness functions is Algorithm 4.6 with one-point crossover rather than uniform. The EA used here remains Algorithm 4.7. Let $p_c(n) \leq 1 - \varepsilon$, where $0 < \varepsilon < 1$, and $s(n) \geq m + 1$ and $s(n) = \Theta(n)$.

$$R_{n,m}(x) := \begin{cases} 2n^2 & : \quad x = 1^n \\ n|x| + b(x) & : \quad |x| \leq n - m \\ 0 : & : \quad \mathit{otherwise} \end{cases}$$

Let $n = 6m$, this implies that $n - m = 5/6m$. Also assume that $s(n) \geq m + 1$ and n is even. Let $b(x)$ be the length of the largest contiguous subsequence of ones in the string and let $|x|$ be the number of ones in the entire string. Note that $b(x)$ was intended in [295] to reward subsequences that could become 'building blocks' for crossover. Below we introduce another Ignoble Trail for one-point crossover derived from this function. We add a new region A into the fitness valley of $R_{n,m}(x)$. The region A contains $m - k$ strings and is defined to be of the form $0^i 1^{n-m-k} (01)^{2k} 0^j$ where $i + j = m - k$. Let k be a constant chosen so that $4 < k < m/4$ and note that A has the property $|x| = n - m + k$. Also let the set of points at $|x| \leq n - m$ be denoted by P. Note that the closest points in P to A are points x such that $|x| = n - m$. Let the area where strings have exactly $n - m$ ones and $b(x) = n - m$ be defined as P^*. The region P^* contains m strings. Define a *contig* to be a contiguous sequence of two or more 1s. Contigs do not overlap and are bounded by zeros. Let $q(x) = |1_{iso}|$, the number of isolated ones in the string. The term $q(x)$ is minimized to zero when there are no isolated ones. Thus P^* has the same fitness in $R_{n,m}(x)$ and $IT6^1_{n,m}(x)$.

$$IT6^1_{n,m}(x) := \begin{cases} 3n^2 & : \quad x \in A \\ 2n^2 & : \quad x = 1^n \\ n|x| + b(x) & : \quad |x| \leq n - m \text{ and } q(x) = 0 \\ n - q(x) & : \quad |x| \leq n - m \text{ and } q(x) \neq 0 \\ 0 & : \quad \textit{otherwise} \end{cases}$$

The general effect of the third term, $n|x| + b(x)$, is to reward both the number of ones in the string and the length of the longest contig. The fourth term, $n - q(x)$, actively punishes isolated ones in the string. Note that the third and forth terms are mutually exclusive.

The maximal fitness strings for the third term, via a string with exactly $n - m$ ones within a single contig, is $n^2 - nm + n - m$. The minimal fitness is a string with two contiguous ones, $2n + 2$. The maximal fitness of the fourth term is $n - 1$ via a string with a single one. The minimal fitness of the fourth term is a string with $n/2$ ones, composed entirely of alternating ones and zeros at fitness level $n/2$.

6.4.1 Behavior of the EA and GA on Ignoble Trail 6

Again the optimization process of the EA and GA is broken down into phases. Figure 6.3 displays the proof diagram. The initial phase is to discover at least one individual in P having no more than $n - m$ ones or find the high fitness set $\{1^n\} \cup A$. Note that as in [295] the probability to initialize a population with any members with more than $(2/3)n$ ones is $e^{-\Omega(n)}$ by Chernoff's bound. This means that the initial population has an exponentially small probability to have any members on the unitation line $|x| = n - m$, in A or at 1^n.

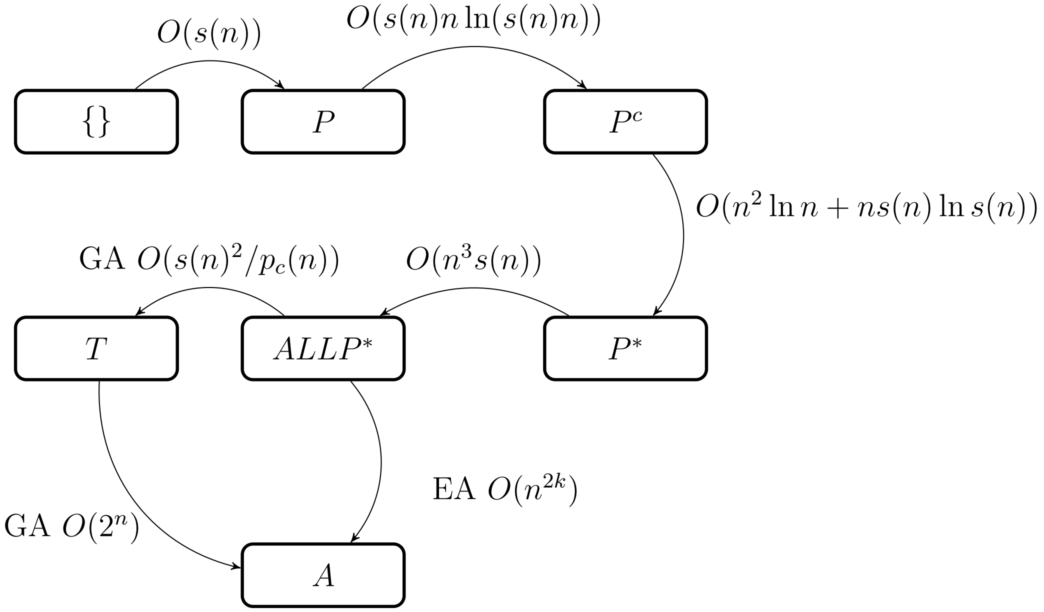


Figure 6.3: Diagram of proofs of Lemmas and Theorems for $IT6^1_{n,m}(x)$. The labels on each arc refer to the expected waiting time to transition from state to state. Those arcs labeled with EA or GA are the expected paths of the respective algorithm.

6.4.2 Time Complexity Results

The first phase of the optimization process is initialization of the population.

Lemma 37. *The steady-state EA and GA will find* P *or* $\{1^n\} \cup A$ *within* $O(s(n))$ *steps with probability* $1 - 2^{-\Omega(n^2)}$.

Proof. Assume that they have not found $\{1^n\}\cup A$. The effects of crossover are ignored as they only serve to increase the probability of finding the goal. The probability of initializing a member of the population with more than $(2/3)n$ ones is $e^{-\Omega(n)} = 2^{-\Omega(n)}$ by Chernoff's bound. The probability to do so for all $s(n)$ of the initialization steps is $s(n)/2^{\Omega(n)} < 2^{-\Omega(n^2)}$. Thus they must wait for $O(s(n))$ steps, at the end of which the population will all be within P with probability $1 - 2^{-\Omega(n^2)}$. □

The next phase is to eliminate all strings with any isolated ones from the population.

Lemma 38. *The steady-state EA and GA will contain a population in P without any strings with isolated ones or is in $\{1^n\} \cup A$ in expected $O(s(n)n\ln(s(n)n))$ steps and within $O(ns(n)^2 \ln s(n))$ with probability $1 - 2^{-\Omega(s(n))}$.*

Proof. Let our distance metric d be the sum $d = \sum_x q(x)$, ie total number of isolated ones in the population. The maximum value of d is $s(n)m$. The first method of decreasing d is by replication of any individual with the maximal $n - q(x)$ fitness value or an individual where $q(x) = 0$. This method works as long as all individuals of the population do not have identical fitness.

Let j_i be the number of individuals with a fitness value greater than i. The probability of choosing one of the higher fitness values is $j_i/s(n)$, with an expected waiting time of $s(n)/j$ steps. By Chernoff's bound a better individual will be chosen within $s(n)^2/j_i$ steps with probability $1 - 2^{-\Omega(s(n))}$. Summing that result over all values of $i \in \{1, \ldots, n - m\}$ and $j \in \{1, \ldots, s(n) - 1\}$ one gets an expectation of $O(ns(n) \ln s(n))$ and a Chernoff bounds of $O(ns(n)^2 \ln s(n))$ with probability $1 - 2^{-\Omega(s(n))}$.

The second method of decreasing d is to flip any isolated one to a zero. There are up to $s(n)(n/2)$ isolated ones. If this problem is treated as the entire population being a string, then the coupon collector's theorem can be used to argue that it takes expected $O(s(n)n\ln(s(n)n))$ time to find a population strings of all zeros. Via Chernoff the bound is $O(s(n)n\ln(s(n)n)+(s(n)n))$ with probability $1-e^{-e^{-c}}$. Adding the bounds together results in

$$s(n)n \ \ln s(n) + s(n)n \ \ln\left(s(n)n\right) = O\left(s(n)n \ \ln\left(s(n)n\right)\right)$$

for the combined expectation and

$$s(n)^2 n \ \ln s(n) + s(n)n \ \ln \left(s(n)n\right) = O\left(ns(n)^2 \ \ln s(n)\right)$$

for the combined Chernoff bound. □

The third phase is to find a population where all individuals have $n - m$ ones in a single contig or are in $\{1^m\} \cup A$. Recall that P^* is defined to be the area where strings have exactly $n - m$ ones, $b(x) = n - m$ and $q(x) = 0$.

Lemma 39. *The steady-state EA and GA will contain a population where each individual is in P^* or is in $\{1^n\} \cup A$ in expected $O(n^2 \ln n + ns(n) \ln s(n))$ steps and within $O(n^3 \ln n + ns(n)^2 \ln s(n))$ with probability $1 - 2^{-\Omega(min(s(n),n))}$.*

Proof. Claim 6 of [295] is closely followed. Ignore crossover as it cannot make the task more difficult. Assuming the region $\{1^n\} \cup A$ is not found, let metric d be defined by $d = \sum_x (b(x))$ for all $s(n)$ population members. The maximum value of d is $s(n)(n - m)$. The first method of increasing d is by replication of any individual with some $b(x)$ value, displacing a lower $b(x)$ value. This method works as long as all individuals of the population do not have identical fitness.

Let j be the number of individuals with a $b(x)$ value greater than i. The probability of choosing one of the higher $b(x)$-values is $j/s(n)$, with an expected waiting time of $O(s(n)/j)$ steps. By Chernoff's bound a better individual will be chosen within $s(n)^2/j$ steps with probability $1 - 2^{-\Omega(s(n))}$.

A second method of increasing d is via enhancing a $b(x)$ value via mutation and simultaneously flip the correct two bits to increase $b(x)$. The neighboring zero of the longest ones-block and any of the remaining $n - m - i$ ones not in the longest one-block must be flipped. Any given two bit mutation has a probability of $(1/n)^2(1-(1/n))^{n-2} \geq 1/(en^2)$, resulting in a combined probability of $(n-m-i)/en^2$.

Summing over the possible i values, the resulting expectation is $O(n^2 \ln n)$ steps with a Chernoff's bound of $O(n^3 \ln n)$ steps with probability $1 - 2^{\Omega(n)}$. The above result is achieved by adding the bounds together. □

The fourth phase consists of finding a population containing at least one copy of all possible individuals with $n - m$ ones and $b(x) = n - m$ or discover $\{1^n\} \cup A$.

Lemma 40. *The steady-state EA and GA will contain a population with at least one copy of all strings in* P^* *or find* $\{1^n\} \cup A$ *in expected* $O(n^3 s(n))$ *steps and within* $O(n^4 s(n))$ *with probability* $1 - 2^{-\Omega(n))}$.

Proof. Claim 7 of [295] is closely followed with the addition of the A region. Again the effects of crossover are ignored and it is assumed that $\{1^n\} \cup A$ is not found. There are $m + 1$ distinct strings with exactly $n - m$ ones and $b(x) = n - m$. These strings are of the form $0^i 1^{n-m} 0^j$ where $i + j = m$. Note that $s(n) \geq m + 1$ must hold for there to be the possibility of the population covering that set of distinct strings. The goal is met if there exists all possible values of i or j in the population. Until then, there is at least one value of i such that it has a neighboring value $i \pm 1$ not existing in the population. The probability to select this individual is $1/s(n)$. Mutating this individual into the missing i value is accomplished with a two bit flip of either the leading or trailing zero of the 1-bit block and flipping the trailing 1 on the opposite side of the 1-block, thus shifting the starting position of the 1-block up or down in i value. As before, two bit mutations are done with probability $1/en^2$. By the truncation method of the EA and GA, a successful discovery of a missing string will displace some other population member with excess duplicates. This process must repeat at most m times to cover the space of goal strings. This results in an expected time of $O(n^2 s(n) m) = O(n^3 s(n))$ steps and by Chernoff's bound of $O(n^4 s(n))$ steps with probability $2^{-\Omega(n))}$. □

A crucial consideration is the characterization of the difficulty of either algorithm not following the optimization path as presented so far. Note that the minimum Hamming distance between P^* and A is $2k$ bits when the point $0^i1^{n-m}0^j$ where $i = j = m/2$ is chosen.

Claim 41. *The probability of the EA or GA discovering A before finding all copies of strings in P^* is $O(1/n^{2k-1})$.*

Proof. Every point of region A contains $2k$ isolated ones and a string of $n - m - k$ consecutive ones. Given that A has more than 2/3 ones, the probability to initialize into this region is exponentially small. Generating any string in the initial population with more than m consecutive ones is also exponentially unlikely.

Finding $x \in A$ during phase 2 is exponentially unlikely. During this phase the fitness function gives no advantage to long strings of consecutive ones. Thus crossover followed by mutation would be exponentially unlikey to generate points with $n-m+k$ consecutive ones.

After phase 2 is completed, any string with isolated ones, except for strings in A, will be rejected by the selection process. Thus, the result of crossover can contain at most 1 isolated one. Generating $x \in A$ via crossover followed by mutation would require $2k - 1$ mutations. This has probability at most $1/n^{2k-1}$. $\square$

At this point the GA has two paths to follow, discover A (success) or 1^n (and be exponentially far from A). Remember A has m strings and is defined to be of the form $0^i1^{n-m-k}(01)^{2k}0^j$ where $i + j = m - k$. Note that constant k is $4 < k \leq m/2$.

Lemma 42. *Given a population with at least one copy of all possible individuals in P^*, the steady-state GA will find 1^n in expected $s(n)^2/p_c(n)$ steps and within $s(n)^3/p_c(n)$ steps with probability $1 - O(1/n^{k-2})$.*

Proof. Claim 8 of [295] is closely followed for this event. From Lemma 40, the population must have at least one individual with pattern $1^{n-m}0^m$ and one with pattern 0^m1^{n-m}. The probability of a crossover event between two such individuals is $p_c/s(n)^2$. Any crossover point such that it is greater than m and less than $n-m$ will result in a child 1^n. The probability of selecting this point is $[(n-2m+1)/(n-1)] \geq 1/3$. Once child 1^n is created via crossover, the probability that the mutation operator does not alter the child is at least $1/4$. The result follows via the expected waiting time of the event's probability and the application of Chernoff's bound to the probability. The remaining failure event is the GA finding A before 1^n. This happens with probability $O(1/n^{k-2})$.

$$\frac{s(n)^2/p_c}{en^{2k}+s(n)^2/p_c} = \frac{s(n)^2/p_c}{en^{2k}+s(n)^2/p_c} = O(1/n^{2k-2})$$

□

At this point the GA has discovered 1^n. Note that this GA has no duplicates prohibition, so the population to become dominated with $s(n)$ copies of 1^n happens in expected $O(s(n))$ time due to the selection operators.

Theorem 43. *The steady-state GA with a population dominated by* 1^n *will take exponential time to find* A *with probability* $1-1/n^{m-k}$.

Proof. Similar to previous arguments, the population will in $O(s(n))$ time collect into 1^n. The Hamming distance from 1^n to A is $m-k$, where k is a constant. Thus, only a probability $1/n^{m-k}$ event will discover A, and there is no possibility that crossover between two individuals 1^n will result in a jump to A. Given the Hamming distance, it is easy to see that exponential time is required to escape from 1^n. □

Theorem 44. *Given a population with at least one copy of all possible individuals in* P^**, the EA will find* A *in expected* $O(n^{2k})$ *steps and within* $O(n^{2k}\ln n)$ *steps with probability* $1 - O(1/n)$*.*

Proof. The minimum Hamming distance between P^* and A is $2k$ bits when the point $0^i 1^{n-m} 0^j$ where $i = j = m/2$. Constant k is chosen so that $4 < k < m/4$. Thus the probability of jumping this gap via mutation in one step is $p \geq (1/n)^{2k}(1-1/n)^{n-2k}$. Probability p is bounded below by $1/(en^{2k})$, thus the expected waiting time is bounded above by $O(n^{2k})$. Via Chernoff bounds, the probability of the event within $en^{2k}\ln n$ steps is $1 - O(1/n)$. □

Theorem 45. *The steady-state EA will optimize* $IT6^{1}_{n,m}(x', x'')$ *in expected* $O(n^{2k})$ *steps and within* $O(n^{2k}\ln n)$ *steps with probability* $1 - O(1/n)$*.*

Proof. The result follows from summing up the expected times of all phrases, applying Chernoff bounds and taking the maximum failure probability. □

Theorem 46. *The steady-state GA with one-point crossover will need exponential time steps to optimize* $IT6^{u}_{n,m}(x', x'')$ *with probability* $1 - O(1/n^{2k})$*.*

Proof. The result follows from summing up the expected times of all phrases and taking the maximum failure probability. □

Bounds for the coupon collector's problem can be found in Motwani and Raghavan [275] as well as Mitzenmacher and Upfal [374]. It is also used to prove bounds for the $(1+1)$ EA solving ONEMAX in Droste et al. [341].

6.5 Closing Remarks and Future Ideas

While this result is similar to the previous Ignoble Trails result, that work started with small populations and was a proving ground to learn and understand time complexity proofs for EAs. Once scaled up to arbitrary size populations more factors must be considered during function construction and proofs. The difficulty in extending Real Royal Road results to Ignoble Trails concerns the many failure modes one must take into account that might prevent the function from being crossover deceptive.

This is believed to be the first proven example of functions where a crossover based GA with variable population size is exponentially outperformed by an EA without the crossover operator. While these function are highly artificial, first results in this hole in the GA theory literature are still worth putting forward as a first step to more natural crossover deceptive functions. These functions are artificially constructed for a purpose, and it is hoped that some generalizations can be learned. In essence, one general method of crossover deception is to construct a forked path structure where crossover is drawn in a direction away from the optimal points.

The key lesson learned from this work and the previous OneMix and Real Royal Road papers is that both the GA and the EA can be deceived with relatively simple structures. Challenges for future work include showing an example of a more real world graph theory problem where the GA fails and the EA succeeds. It is also hypothesized that a function can be constructed where crossover deceptive structures are hierarchically stacked to form a HIFF like function where the GA has a more diabolical challenge. Perhaps the functions from Watson and Jansen [241] may provide a starting point.

CHAPTER 7

UNITATION MODELS FOR THE SIMPLE GA

> In my opinion there is no definable boundary line between variation and mutation and the difference between these two phenomena depends solely on the number larger or smaller of ids which have varied in the same direction.
>
> August Weismann [92]

Framework	Mutation-only EAs	Crossover-enabled GAs
Time Complexity Analysis		
Dynamical Systems Analysis	✓	
Empirical Analysis		
Methods: Infinite population model construction and analysis		

Figure 7.1: Chapter 7 Framework

7.1 Introduction

Departing from time complexity analysis, this chapter examines a dynamical systems model of the EA against a suite of unitation fitness functions. Analysis and visualization of this infinite population model of a mutation-only Simple Genetic Algorithm is presented. The model's fixed-point behavior is analyzed as are the effects of fixed points lying outside of the population space, but near enough to influence behavior of the Simple GA. Figure 7.1 illustrates the contributions within the framework. [1]

---

[1] An earlier version of this chapter was published as
J. Neal Richter, Alden Wright, John Paxton. "Exploration of Population Fixed Points Versus Mutation Rates for Functions of Unitation", Workshop on Evolutionary Computation Theory, GECCO-2004 June 26-30, 2004 Seattle, Washington.

7.2 Dynamical Systems Model

The Vose infinite population model [363] of the Simple GA is a dynamical systems model that represents populations as a vector of proportions. This vector has dimension s, where s is the size of the search space. Each entry in the vector is the proportion of members in the global population that are represented by a given chromosome in the search space. This representation allows utilization of techniques and theorems from the mathematical theory of dynamic systems to analyze the GA.

This work is restricted to examining a class of fitness functions called functions of unitation. These functions establish equivalence classes, allowing a reduction in the dimensionality of the corresponding Vose model. The analysis will also be restricted to mutation-selection GAs with no crossover, elitism or other advanced techniques.

Using the eigensystem of the mutation-only Vose model, fixed points of the mutation-selection GA can be found. Fixed points are population vectors such that applying the GA to them results in the same population vector. Fixed points are not the optimal points in the fitness landscape, they represent the expected long-run distribution of a population for a given GA with a large enough population. A fixed point may or may not contain a significant proportion of the global optima of the fitness landscape.

Fixed points will be calculated over a range of mutation rates for several fitness functions. Metastable states defined by the model will also be calculated. These states are fixed points of the model that exist outside the representable space of a real population. The metastable states near the population space can create metastable regions inside the space and have an effect on the GA under real populations [325].

This chapter is largely an extension of Rowe [375] that further explores the effect of mutation rates on fixed points and metastable states. The intent of this exploration is to lay the groundwork for studying how adaptive mutation schemes can be understood and designed.

7.3 Functions of Unitation

Unitation functions are simplified fitness functions that reduce the search space into a smaller number of equivalence classes. The fitness is defined only by the number of ones in a chromosome $x : \{1, 0\}^d$. This reduction allows easier computation of fixed points. All fitness values are non-negative:

$$u(x) : \{0, 1\}^d \rightarrow \mathbb{R}+ \tag{7.1}$$

An example function is:

$$u(0) = 3 \quad u(1) = 2 \quad u(2) = 1 \quad u(3) = 4 \tag{7.2}$$

This definition allows us to reduce the dimensionality of the infinite proportionality population vector from 2^d to $d+1$. This vector is represented as $\mathbf{p} = \{p_0, ..., p_d\}$ where p_k is the proportion of the population having exactly k ones. Note that this vector is of $d + 1$ dimension as it must have entries for the all zeros case.

The mutation matrix is $\hat{U}$, whose entries are the probabilities that a string with j ones mutates to a string with i ones. $\hat{U}$ is defined as an $(d + 1) \times (d + 1)$ matrix with each entry representing the probability that a bitstring with j ones mutates to a string with i ones. The equation below is the formula from [362, 375] where $\delta_{x,y}$ is the Kronecker delta function and q is the mutation probability. Note that like U, $\hat{U}$ is a column stochastic matrix, each column sums to 1. The matrix below contains a three bit example of $\hat{U}$ where q=1/3.

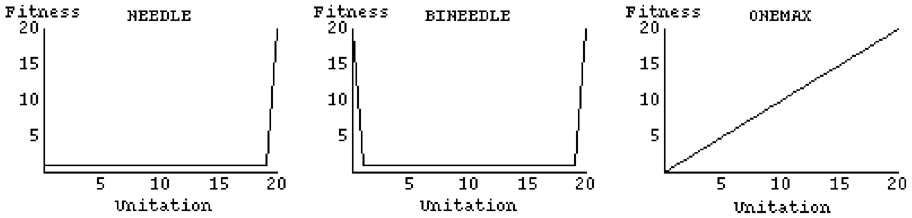


Figure 7.2: NEEDLE, BINEEDLE and ONEMAX fitness functions

$$U_{i,j} = \sum_{k=0}^{d-j} \sum_{l=0}^{j} \delta_{j+k-l,i} \left(\begin{array}{c} n-j \\ k \end{array} \right) \left(\begin{array}{c} j \\ l \end{array} \right) q^{k+l}(1-q)^{n-k-l}$$

$$\text{where } \delta_{j+k-l,i} = \left\{ \begin{array}{ccc} 1 & if & x = y \\ 0 & if & x \neq y \end{array} \right\} \tag{7.3}$$

$$\hat{U} = \begin{pmatrix} .296 & .148 & .074 & .037 \\ .444 & .444 & .333 & .222 \\ .222 & .333 & .444 & .444 \\ .037 & .074 & .148 & .296 \end{pmatrix} \text{ where } q = \frac{1}{3} \tag{7.4}$$

7.3.1 Example Functions of Unitation

The three fitness functions pictured in Figure 7.2 are called NEEDLE, BINEEDLE and ONEMAX, and have been studied before by Rowe [375] and Wright [376] . Here $d = 20$ and $\alpha = 20$ are used for NEEDLE and BINEEDLE.

$$\begin{array}{lcl} NEEDLE & = & \left\{ \begin{array}{ll} 1+\alpha & \text{all ones string} \\ 1 & \text{otherwise} \end{array} \right\} \\ BINEEDLE & = & \left\{ \begin{array}{ll} 1+\alpha & \text{all ones string} \\ 1 & \text{otherwise} \\ 1+\alpha & \text{all zeros string} \end{array} \right\} \\ ONEMAX & = & \text{number of ones in string} \end{array} \tag{7.5}$$

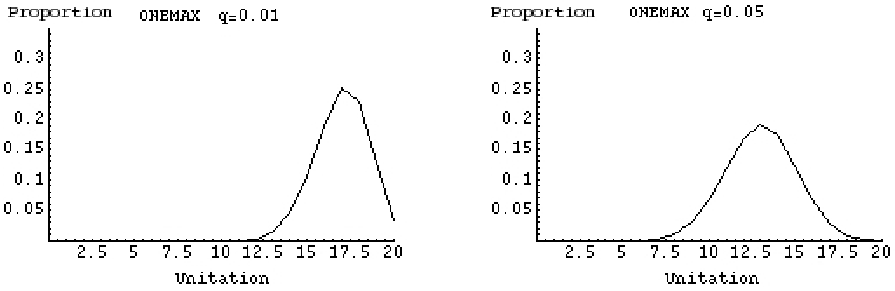


Figure 7.3: ONEMAX fixed point distributions for $q = 0.01$ and $q = 0.05$

7.4 Fixed Points as a Function of Mutation Rate

Figure 7.3 shows two fixed points for the ONEMAX fitness function. These are the normalized leading eigenvector of G with $d = 20$ and mutation rate $q = 0.01$ and $q = 0.05$ respectively. The fixed point shows the long term stable and convergent population of G over the 21 unitation classes. They show, for example, that for $q = 0.01$, 25% of the population will converge to one contain strings with 17 bits of value 1 after a sufficiently large number of generations of the mutation-selection GA have been computed for a large population. Rowe's [375] results as well as those presented in later chapters here suggest 200 generations with on the order of 100 members is more than sufficient for convergence to the fixed point.

Next, a sequence of fixed points were computed for mutation rates in the range $q = [0.01, 0.20]$ and the population distributions were plotted as a 3-dimensional surface. Figure 7.4 shows the ONEMAX fixed points plotted over this mutation rate range. Note that at mutation rates near $q = 0.01$, the population contains a significant proportion of the $f(20) = 20$ optimal mutation class. By $q = 0.05$ there is near zero membership with the fixed point centered around the center $f(10) = 10$ of the unitation classes.

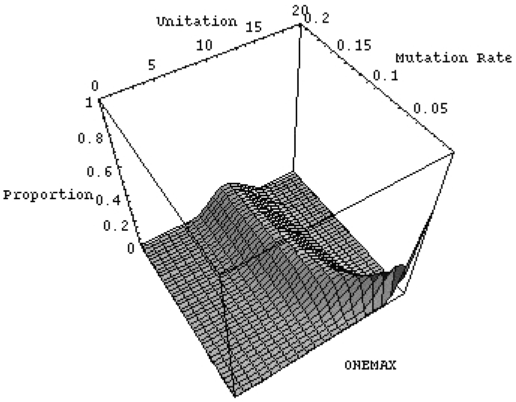


Figure 7.4: ONEMAX fixed point surface

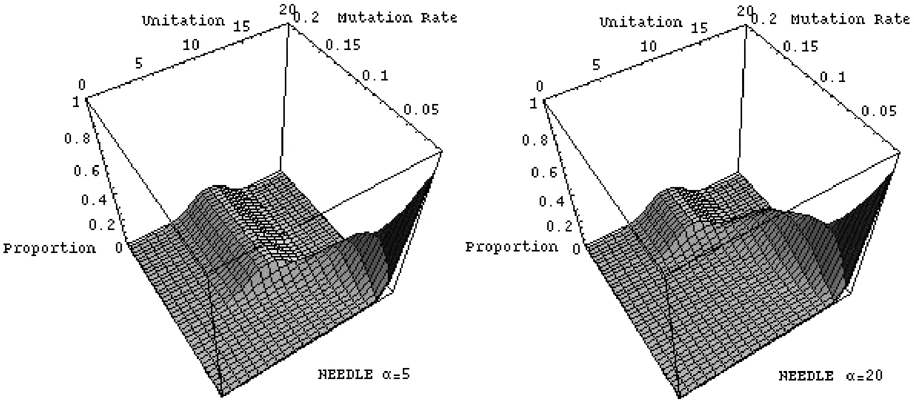


Figure 7.5: NEEDLE fixed point surfaces for $\alpha = 5$ and $\alpha = 20$

7.5 Fixed-point Surfaces for the NEEDLE and BINEEDLE Functions

Figures 7.5 and 7.6 contain the fixed point surfaces of NEEDLE and BINEEDLE for both $\alpha = 5$ and $\alpha = 20$. Figure 7.5 tells us that for low mutation rates, i.e. mutation rates below $q = 0.025$ and $q = 0.05$ respectively, NEEDLE has a significant proportion of the population at the maximum fitness unitation class.

In Figure 7.6 note the dramatic change in population distribution for BINEEDLE that occurs near $q = 0.07$ for $\alpha = 5$. Above this mutation rate the population contains near zero proportional membership by either global optimum. This tells us that

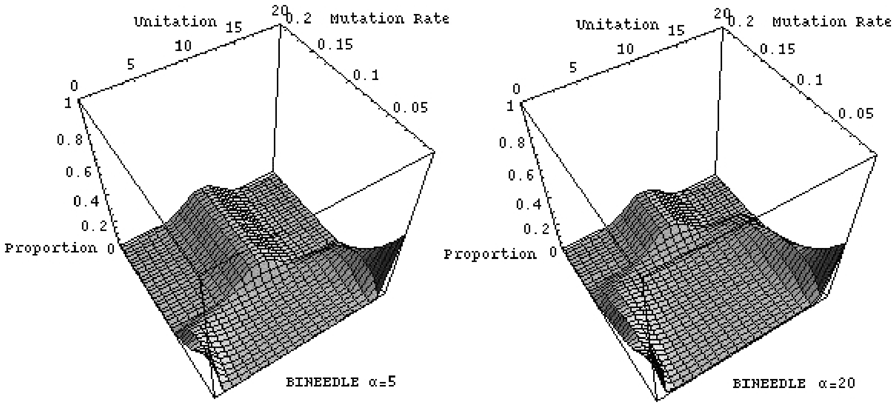


Figure 7.6: BINEEDLE fixed point surfaces for $\alpha = 5$ and $\alpha = 20$

mutation rates below this value are likely very important for good GA performance. For $\alpha = 20$ this phase change occurs near $q = 0.13$ indicating that this version of the fitness function is more tolerant of higher mutation rates. The NEEDLE's fixed-point surface has similar properties. Notice that for mutation rates greater than the critical q values for both functions, the population is centered around the unitation midpoint. At this midpoint the population has equal or near equal representation of ones and zeros.

The phase transitions for NEEDLE and BINEEDLE were studied in [376] and [377], and are shown by the Eigen model [378]. Ochoa and Harvey [377] restate the Eigen model for the GA community and show how the Eigen error thresholds change under finite populations. Finite populations move the phase transition to lower mutation rates.

7.6 Fixed-point Surface for a Fully Deceptive Trap Function

Trap functions are piecewise linear functions that divide the search space into two Hamming space basins [296]. Each basin has an optimal point, one of which is the global optimum. In Deb and Goldberg [296], they set forth a set of conditions

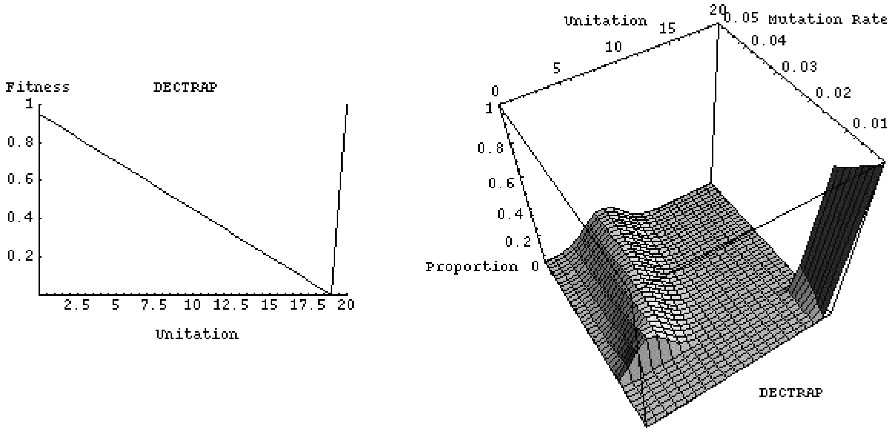


Figure 7.7: Fully deceptive trap functions DECTRAP and fixed point surface

for calling a fitness function *fully deceptive.* A fully deceptive function from [296] is adopted, and refered to as DECTRAP.

$$f(x) = \left\{ \begin{array}{ll} 1 & if u(x) = d \\ 1 - \frac{1+u(x)}{d} & otherwise \end{array} \right\} \tag{7.6}$$

Figure 7.7 illustrates DECTRAP and its fixed-point surface. The trap function is very near a fitness of 1 for the all zeros string, and is fitness 1 for the all ones string. The all zeros basin takes up the majority of the function space.

The fixed-point surface has a drastic phase change at approximately $q = 0.012$. Below this mutation rate a high proportion of the globally optimal string exists. Above this mutation rate the fixed point contains nearly no exemplars of the global optimal. Notice again that as the mutation rate increases, the fixed point moves toward a population centered around the unitation midpoint.

7.7 Fixed-point Surface for Functions with Two Traps

Figure 7.8 illustrates a trap function containing two traps, referred to as 2TRAP. The fixed point surface is very similar to the BINEEDLE surface with a critical

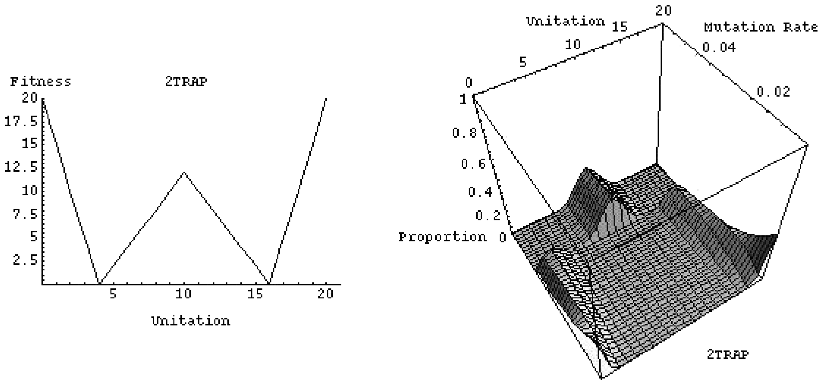


Figure 7.8: Double trap function 2TRAP and fixed point surface

phase change at $q = 0.04$. Note that mutation rates below 0.02 are slightly superior since the fixed points are still centered much closer to the two local optima. As the mutation rate increases from 0.02 to 0.04, the population clusters move away from the maximums. The mutation rates are too high to maintain high membership.

Figure 7.9 shows a deceptive double trap function, or DEC2TRAP. This function is modeled after the fully deceptive function given in [296]. The formula is given here:

$$f(x) = \left\{ \begin{array}{ll} 1 & \text{if } u(x) = d/2 \\ 1 - \frac{1+u(x)}{d/2} & \text{if } u(x) < d/2 \\ \frac{u(x)-d/2-1}{d/2} & \text{if } u(x) > d/2 \end{array} \right\} \tag{7.7}$$

DEC2TRAP's fixed point landscape is very interesting as it has virtually no membership of the $u(x) = 0$ and $u(x) = 20$ high fitness suboptimal points. This result is counter-intuitive at first glance. The regions on either side of the center optimal needle have smooth hills to climb that lead to the local maximums. A practitioner might expect empirical GAs to retain membership in the local optimum, given that low mutation rates make it harder for a population to move outside the basin of either local maximum.

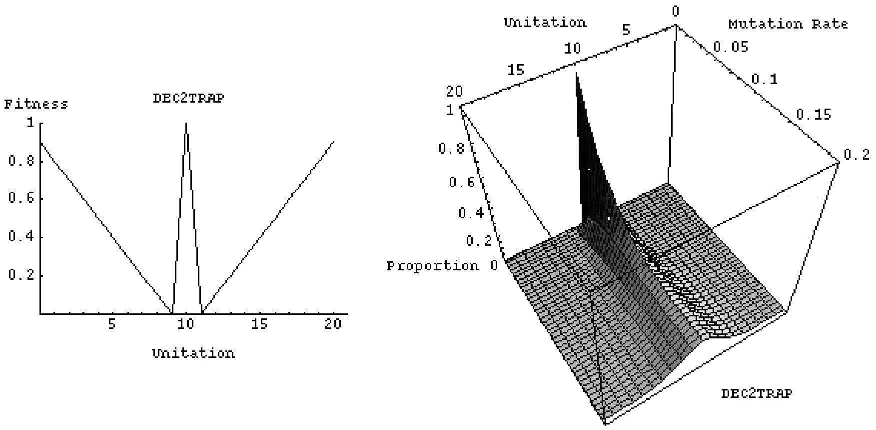


Figure 7.9: Deceptive double trap function DEC2TRAP and fixed point surface

Lack of elitism and other advanced features in the model partially explains the result, as well as the fact that after enough generations have passed, the Simple GA will converge totally into the basins of global maximums. Another reason is that there exist many strings in the $u(x) = 10$ class and only one string in the $u(x) = 0$ and $u(x) = 20$ classes.

7.8 Metastable States

Fixed points have a region around them generally called a *basin of attraction* [182]. Loosely speaking, these are regions where the fixed point exerts influence. The Perron-Frobenius theorem [181] shows that the US matrix has only one fixed point inside the simplex. The other eigenvectors are called metastable states [325].

Rowe [325] defines *metastable regions* as regions inside the simplex near metastable states close to the simplex. The continuity of the IPM implies these regions exist [325]. The question of how these states move around the neighborhood of the simplex as the mutation rate changes is explored next.

For the functions previously described, the sum of the negative components of the eigenvalues is computed. The lowest sum is potentially the closest metastable state to the simplex. Obviously this is not always the case, a true geometric distance will define exactly which of these vectors is closest to the simplex. Computing a geometric distance involves finding the minimum distance from a point to an $n-1$ dimensional hypertetrahedron. While a linear projection would precisely compute this distance, the prior metric is used for now.

Figure 7.10 plots this distance metric for the $\alpha = 5$ NEEDLE and BINEEDLE on a logarithmic scale. Also shown is the population proportion of the $u(x) = 10$ string vs. mutation rate for comparison. Both functions have a similar metastable state movement and visually identical population proportion graphs. Note that the phase changes in all four graphs take place at approximately $q = 0.075$ where the closest metastable state suddenly moves farther away. Figure 7.11 shows the identical graphs for $\alpha = 20$. They show essentially the same effect as $\alpha = 5$ except the phase changes take place at $q = 0.13$.

Figure 7.12 shows the same plots for the three trap functions and ONEMAX. The plots for DECTRAP and DEC2TRAP show similar metastable state movement, with DEC2TRAP's nearest metastable state being approximately twice as far away as DECTRAP for a given q. 2TRAP's metastable plot contains some interesting sharp edges that persisted for various settings from 30 to 40+ digits of precision.

7.9 Closing Remarks and Future Ideas

It is common in GA research papers that a specific mutation rate is chosen with no justification for the particular value. Hopefully this paper will prompt the reader to question the validity of the choice of mutation rate in their next GA project.

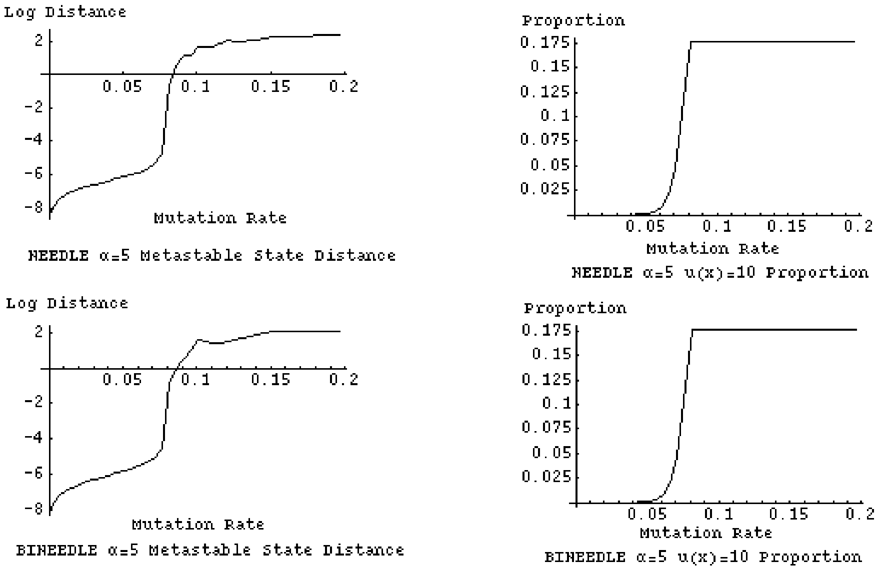


Figure 7.10: Metastable point distances and $u(x) = 10$ proportions for $\alpha = 5$ NEEDLE and BINEEDLE

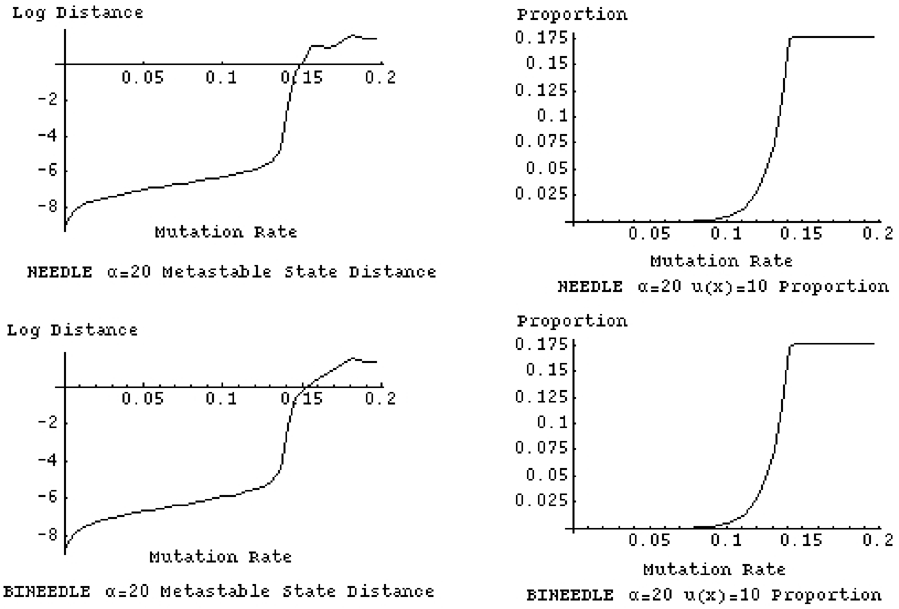
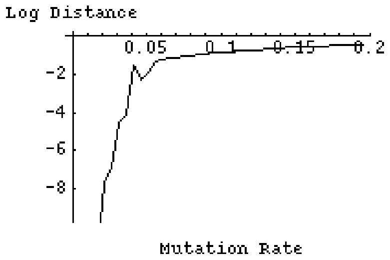
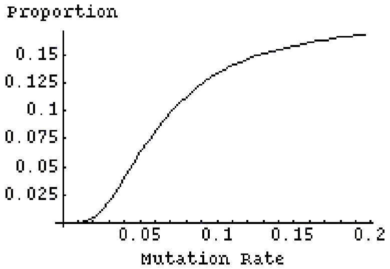


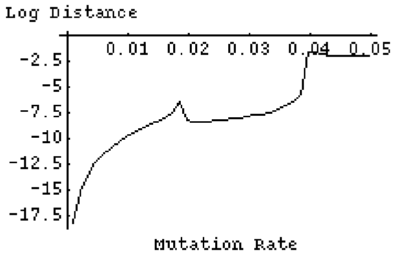
Figure 7.11: Metastable point distances and $u(x) = 10$ proportions for $\alpha = 20$ NEEDLE and BINEEDLE



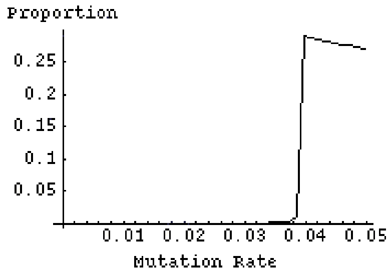
DECTRAP Metastable State Distance



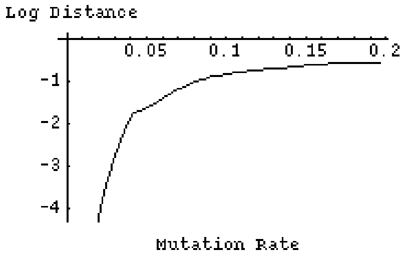
DECTRAP u(x)=10 Proportion



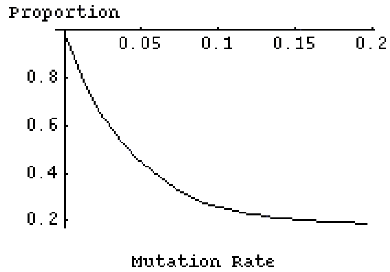
2TRAP Metastable State Distance



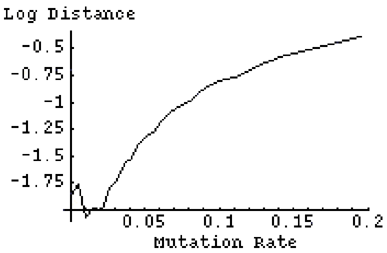
2TRAP u(x)=10 Proportion



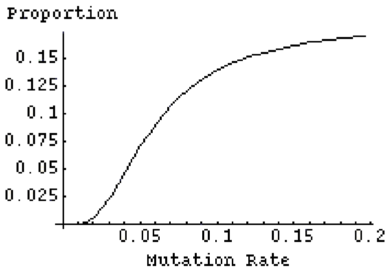
DEC2TRAP Metastable State Distance



DEC2TRAP u(x)=10 Proportion



ONEMAX Metastable State Distance



ONEMAX u(x)=10 Proportion

Figure 7.12: Metastable point distances and $u(x) = 10$ proportions for the trap functions and ONEMAX

Numerical results were shown analyzing fixed-points and metastable state proximity for eight different fitness functions applied to functions of unitation. Unitation functions were chosen as these provide a level of dimensional reduction to ease the tractability of theoretical analysis while still being interesting functions. This analysis does not take into account the crossover operator, and as such no strong conclusions should be generalized to the generic GA. All analysis was done by constructing, simulating and solving the $\mathcal{G}$-map with Mathematica$^{\text{TM}}$.

This type of analysis should aid the understanding of simple mutation-only evolutionary algorithms like the (1+1) EA and related algorithms, as well as dynamic parameter schemes and/or annealing schedules for these algorithms.

CHAPTER 8

UNITATION MODELS FOR THE METROPOLIS EA

> It is selection, and only selection, that directs evolution in directions that are nonrandom.
>
> Richard Dawkins [379]

Framework	Mutation-only EAs	Crossover-enabled GAs
Time Complexity Analysis		
Dynamical Systems Analysis	✓	
Empirical Analysis		
Methods: Infinite population model construction and analysis		

Figure 8.1: Chapter 8 Framework

8.1 Introduction

A infinite population dynamical systems model is derived for a simple elitist non-decreasing EA inspired by the (1 + 1) EA and variants. These algorithms do not utilize a crossover operator. The models are analyzed on several fitness functions of unitation to determine fixed-point movement under varying mutation rates. Figure 8.1 illustrates the contributions within the framework. [1]

---

[1] An earlier version of this chapter was published as
J. Neal Richter, John Paxton, Alden Wright. "EA Models of Population Fixed Points Versus Mutations Rates for Functions of Unitation", GECCO-2005 June 25-29, 2005 Washington, D.C.

8.2 (1+1) EAs AND VARIANTS

First three simple variants of the GA and their associated Markov models are briefly reviewed. Next they are applied to functions of unitation. These algorithms and models have been studied extensively, a few examples are [36, 235, 340].

Algorithm 4.3 is the (1+1) EA and its properties are summarized extensively in [235]. Again, the fitness function is $f(x) : \{0,1\}^d \to \mathbb{R}$.

Algorithm 4.3: The (1+1) EA

```
1. Choose mutation rate $p_m \, \epsilon \, (0, 1/2]$.
2. Choose $x \, \epsilon \, \{0,1\}^n$ uniformly at random.
3. Create $y$ by flipping each bit of $x$ independently with $p_m$.
4. If $f(y) \geq f(x)$, set $x := y$.
5. Continue at line 3,
```

If the restriction in line 4 that the child bitstring y must have a better fitness than the parent bitstring x is removed, the (1,1) EA is created. This algorithm is more precisely denoted a *random walk* and is commonly covered in introductory stochastic modeling texts like [380]. Recall from Equations 4.27 and 7.3 the U and $\hat{U}$ matrices are the Markov transition matrices of the (1,1) EA for standard and unitation (respectively) fitness functions.

Equation 8.1 details how Equation 7.3 is used to construct the (1+1) EA transition matrix. Equation 8.2 is an example of the $\hat{U}^+$ transition matrix for a 3-bit bitstring with $q = 1/3$ with a 3-bit ONEMAX fitness function. Note that these matrices are lower-triangular since they are built in a column stochastic manner.

$$\hat{U}^+_{i,j} = \frac{\mu_{i,j}}{\sum\limits_{k=0}^{l} \mu_{i,k}} \qquad \mu_{i,j} = \left\{ \begin{array}{cc} U_{i,j} & f(j) \leq f(i) \\ 0 & \text{else} \end{array} \right\} \tag{8.1}$$

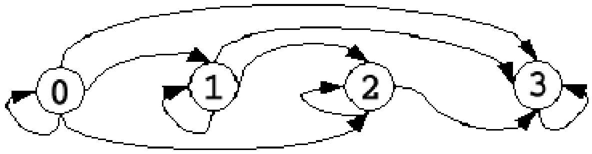


Figure 8.2: (1+1) EA example state diagram.

$$\hat{U}^{+} = \begin{pmatrix} .296 & 0 & 0 & 0 \\ .444 & .522 & 0 & 0 \\ .222 & .391 & .750 & 0 \\ .037 & .087 & .250 & 1 \end{pmatrix} \qquad \text{where } q = \tfrac{1}{3} \tag{8.2}$$

The matrix in Equation 8.2 defined via Equation 8.1 for ONEMAX is an absorbing Markov model. This will occur for any fitness function with a single global optimal, forming an absorbing state at the optimal. Once the transition is made from some bitstring to the optimal string of all ones, all further transitions stay at this string. This is in contrast to Equation 7.4 defined by Equation 7.3, where a transition from any unitation class to any other class is possible. Figure 8.2 is the state diagram for Equation 8.2.

The following algorithm is based upon one detailed and analyzed in [340].

Algorithm 8.10: The (1+1) EA with Metropolis or Annealing Selection

```
1. Choose $p_m \epsilon (0, 1/2]$.
2. Choose $\alpha \epsilon (0, \infty]$.
3. Choose $x \epsilon \{0, 1\}^d$ uniformly at random.
4. Create $y$ by flipping each bit of $x$ independently with $p_m$.
5. If $f(y) \geq f(x)$, set $x := y$.
6. Else set $x := y$ with probability $1/\alpha^{|f(y)-f(x)|}$.
7. Continue at line 4.
```

This algorithm allows selection of child bitstrings with lower fitness values. The probability of selection is in inverse exponential proportion to the drop in fitness. Algorithm 8.10 is an example of the Metropolis algorithm of [36]. The classic Metropolis algorithm restricts line 4 by mutating only one randomly chosen bit of x. The constant α is always greater than 1, usually much greater. Equation 8.3 details the construction of the (1+1) MEA transition matrix and Equation 8.4 is an example matrix for a 3 bit bitstring with $q = 1/3$ and $\alpha = 1024$ for a 3-bit ONEMAX fitness function.

$$\ddot{U}^{+}_{i,j} = \frac{\mu_{i,j}}{\sum\limits_{k=0}^{l} \mu_{i,k}} \qquad \mu_{i,j} = \left\{ \begin{array}{cc} U_{i,j} & f(j) \leq f(i) \\ \frac{1}{\alpha^{|f(j)-f(i)|}} & \text{else} \end{array} \right\} \tag{8.3}$$

$$\ddot{U}^{+} = \begin{pmatrix} .296 & .001 & \varepsilon & \varepsilon^2 \\ .444 & .522 & .001 & \varepsilon \\ .222 & .390 & .750 & .003 \\ .037 & .087 & .250 & .996 \end{pmatrix} \qquad \begin{array}{c} \text{where } q = \frac{1}{3} \\ \text{and } \varepsilon \approx 10^{-6} \end{array} \tag{8.4}$$

The maxtrix $\ddot{U}^{+}$ contains what can be called a probabilistically absorbing state at state 3, see the 4th column of the matrix. The algorithm has a small probability of leaving the state, and if it does will likely return rapidly and stay there for long periods of time.

Algorithm 8.10 can also be converted into a simulated annealing algorithm by having α be a function of time $\alpha(t)$ that decreases as time increases [55, 340].

Both Algorithm 4.3 and 2 are modeled by applying two operators together. The Markov transition generated by each has both a mutation component that is exactly the same as $\hat{U}$. The matrices in Equations 8.2 and 8.4 are really a composition of $\hat{U}P$ where P is a matrix that implements an elitist selection strategy. Here there are two different versions of P, one for each of Equations 8.1 and 8.3. Equation 8.5 implements line 4 of Algorithm 4.3 and Equation 8.6 implements Algorithm 8.10, line 5.

$$\hat{P}_{i,j} = \left\{ \begin{array}{cc} 1 & f(j) \leq f(i) \\ 0 & \text{else} \end{array} \right\} \tag{8.5}$$

$$\ddot{P}_{i,j} = \left\{ \begin{array}{cc} 1 & f(j) \leq f(i) \\ \frac{1}{\alpha^{|f(j)-f(i)|}} & \text{else} \end{array} \right\} \tag{8.6}$$

8.2.1 Modeling a Population based Metropolis EA

Now one has a matrix associated with Algorithm 4.3 to apply the infinite population model selection matrix to via $\hat{U}^{+}S$. This new transition matrix incorporates both the non-decreasing selection of the (1+1) EA and the proportional selection matrix S, and is capable of modeling the behavior of the (μ+μ) EA (see [235]) where μ is large. Unfortunately the Perron-Frobenius theorem cannot be used to analyze this matrix. It is a reducible matrix since $\hat{U}^{+}$ contains an absorbing state at location $\hat{U}^{+}_{d,d}$.

Algorithm 8.11 is an extension of Algorithm 4.3 to support populations greater than 1. It will be referred to here as the (μ+μ) MEA. Conceptually this new multi-population version of Algorithm 4.3 implements a Simple GA with the slight modification that generational replacement is done with a Metropolis style selection scheme.

Algorithm 8.11: The (μ+μ) MEA

```
 1. Choose $p_m \epsilon (0, 1/2]$.
 2. Choose $\alpha \epsilon (0, \infty]$.
 3. Choose $\mu \epsilon (0, \infty]$.
 4. Create a population $P$ of size $\mu$ and initialize each member uniformly at random
    where $P[i] \epsilon \{0, 1\}^d$.
 5. For $i = 1$ to $\mu$ do:
 6.     Select an individual $x$ from $P$ via proportional selection.
 7.     Create $y$ by flipping each bit of $x$ independently with $p_m$.
 8.     If $f(y) \geq f(x)$, set $P_{new}[i] := y$.
 9.     Else set $P_{new}[i] := y$ with probability $1/\alpha^{|f(y)-f(x)|}$.
10.     Else set $P_{new}[i] := x$.
11. Set $P := P_{new}$.
12. Continue at line 5.
```

The '+' notation here is kept to remain consistent with the notation used in [340]. One can make an argument that since the transition matrix of the (1+1) MEA supports going from any string to any other string it should have been referred to in [235] as a variant of a (1,1) EA. The '+' notation is kept to stress the connection with the standard (1+1) EA and use the MEA label to denote the Metropolis modification.

It is important for the reader to realize two things about Algorithm 8.11. The first is that the algorithm has two selection operators. One of them is the Metropolis operator that acts on the choice of replacing the single parent with the mutated child, the other is the population-wide proportional selection operator that selects individuals to be parents for mutation. Equation 8.7 augments Equation 4.26 to supply this elitist operator P. The second thing to realize is that this algorithm does not implement truncation selection as is implied in the usual meaning of the (μ+λ) EA [235].

$$p_{t+1} = G(p_t) = \frac{1}{\mu(p_t)} USPp_t \tag{8.7}$$

8.3 Infinite Population Models

A new model has been constructed that is amenable to applying the Perron-Frobenius theorem as was done in a previous section. An important question is how the new fixed points will compare with those of the no-crossover Simple GA. This new model will be referred to as the (μ+μ) IPMEA (Infinite Population Metropolis EA). More correctly, $\mu = \infty$ since this is an infinite population model. The μ notation is kept to stress the linkage between infinite population models and real GAs with large populations.

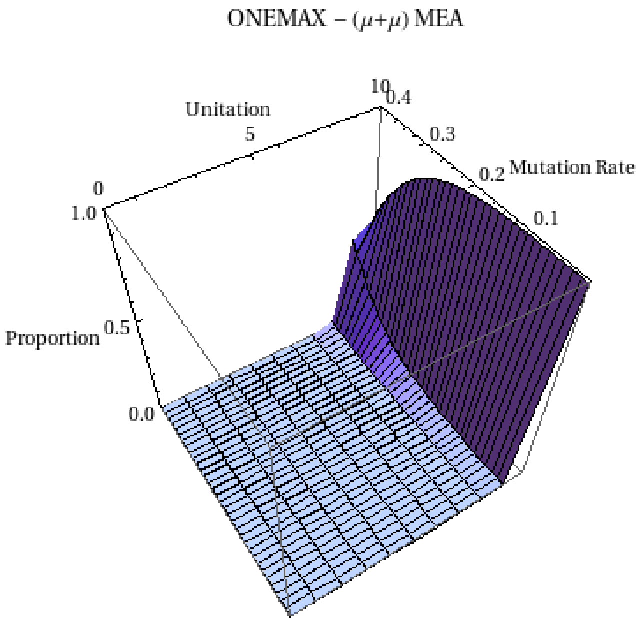


Figure 8.3: ONEMAX Metropolis EA fixed point surface

8.3.1 Metropolis Fixed Points for ONEMAX, NEEDLE and BINEEDLE

Figure 8.3 gives the infinite population model fixed point surface of the $(\mu{+}\mu)$ IPMEA for $q \in [0.01, 0.4]$ and $\alpha = 1024$. Note that the distribution for the EA contains 1-ε proportion of the all-ones string for very small values of q, and this proportion shrinks to about 80% by $q = 0.30$ before rapidly decreasing to 58% at $q = 0.40$. Notice also that the proportion of the second best string, containing d-1 ones, is steadily increasing as the mutation rate grows, growing to about 40% by $q = 0.40$.

It is not easily visible in the figure, however by $q = 0.40$ the proportion of the d-2 ones string grows to nearly 3%. The proportion of each successively less fit unitation class decreases by a factor near 10^{-3}. Peeking back at the matrix in Equation 8.4 one can see the cause. As the mutation rate grows the probability of making a mutation transition from the all-ones string to the $d - 1$ ones string grows as well. Adding proportional selection mitigates this effect to a degree, however it still remains

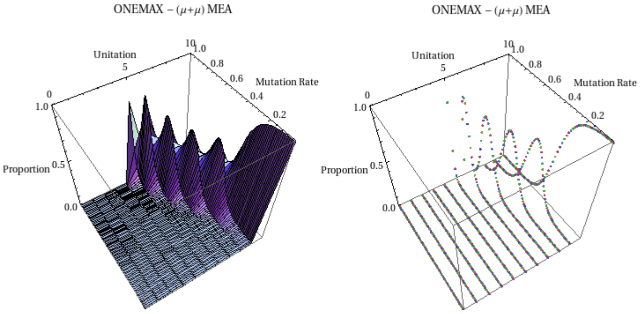


Figure 8.4: ONEMAX (μ+μ) IPMEA over full range of q

true when the transition is made from the (1+1) MEA to the (μ+μ) IPMEA. More importantly, the probability of staying in the $d-$ ones state is fairly high.

This interesting drop off of the optimal proportion at about $q = 0.3$ was curious enough to merit a full surface computation from $q \, \epsilon \, [0.01, 0.99]$ using 80 discrete mutation rates in this range. Normally mutations rates above 0.5 are ignored by the EA community as they tend to result in optimizers that do not do much optimizing. The resulting Figure 8.4 has interesting structure. One interpretation is that at certain points in the mutation range the negative effect of a high mutation rate and positive effect of proportional selection cancel each other and come to a stalemate of two strongly attracting states. Note that the surface plot on the left side implies improperly blends the phase transitions. The point-plot on the right side shows that these phase transitions are discrete at certain high mutation rates.

The phase transitions displayed were not expected, and the intuitive expectation was a gentle curve that sloped down to a high membership in the half ones string while retaining the greatest membership in the all ones string.

Figures 8.5 and 8.6 display the fixed point surfaces for the NEEDLE and BI-NEEDLE functions. Note that the phase transitions displayed in Figures 7.5 and 7.6 are gone. The rising mutation rates were not enough to overcome the effect of proportional selection in favoring copies of the optimal strings, the difference in fitness

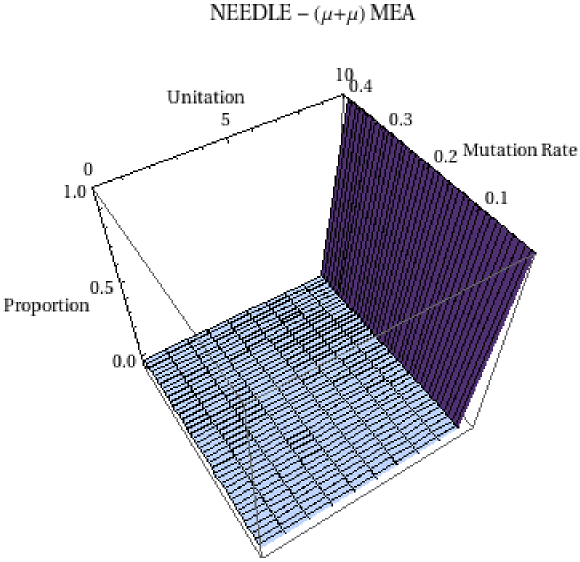


Figure 8.5: NEEDLE Metropolis EA fixed point surface

between the optimum and all other strings is too great. This result is a clear example of a situation where the ergodic (ie non-convergent) IPMEA algorithms are indeed strongly convergent in a practical sense.

8.3.2 Metropolis Fixed Points for the Trap Functions

The next sequence of figures revisits the fitness functions from previous sections. Figure 8.7 gives the fixed point surface for the deceptive trap function DECTRAP. The results for the (μ+μ) IPMEA are interesting if somewhat expected. The results from the NEEDLE apply since the fitness difference between the all-ones string and the $d-1$ ones string is the same. This gives a transition matrix that is dominated at low mutation rates by the action of the Metropolis selection plus the proportional selection operator. Looking back at Figure 7.7, the SGA DECTRAP function showed a phase transition at about $q = 0.05$ moving the bulk of the population from the global fixed point to a population centered most of the way up the basin of the zero string.

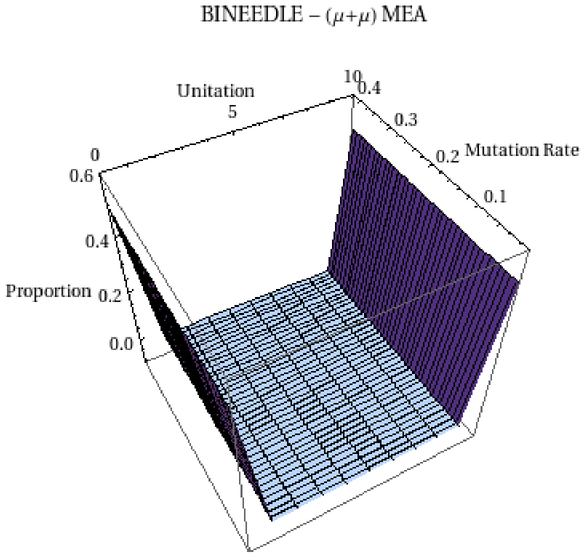


Figure 8.6: BINEEDLE Metropolis EA fixed point surface

Here the phase change is pushed back to about $q = 0.30$ and above that rate the global string still retains a decent proportion in the population.

Figure 8.8 contains the results of the IPMEA algorithm applied to the 2TRAP function. The nearly absorbing states at the all ones and all zeros strings have a very low probability of moving to lower fitness states. The effect of the double selection is too great for the mutation operator to overcome. This is in contrast to Figure 7.8 where there is a strong phase change as the mutation rate is about $q = 0.07$. The elitism of this algorithm eliminated this phase change.

Figure 8.9 displays the surfaces produced by the algorithm on DEC2TRAP. Figure 7.9 showed that the SGA had a slow degradation of the quality of convergence to the global optimum, and it is eliminated with these elitist algorithms. It is also worth noting again that this function is comparatively easy to optimize as a result of the binomial distribution of the unitation classes. If DEC2TRAP were formed by the opposite concatenation of two DECTRAP functions one would expect to see behavior very similar to that of 2TRAP.

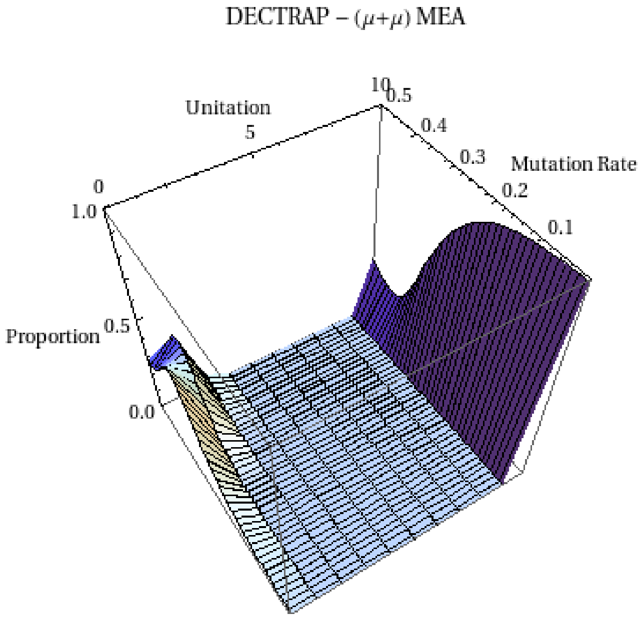


Figure 8.7: DECTRAP Metropolis EA fixed point surface

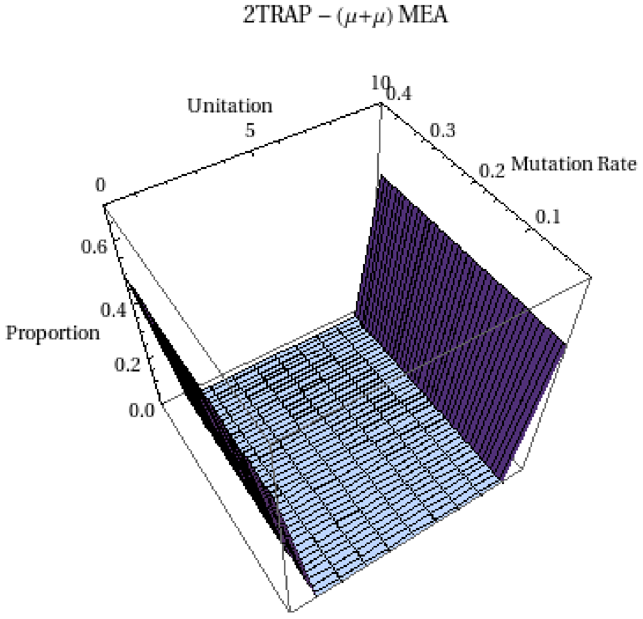


Figure 8.8: 2TRAP Metropolis EA fixed point surface.

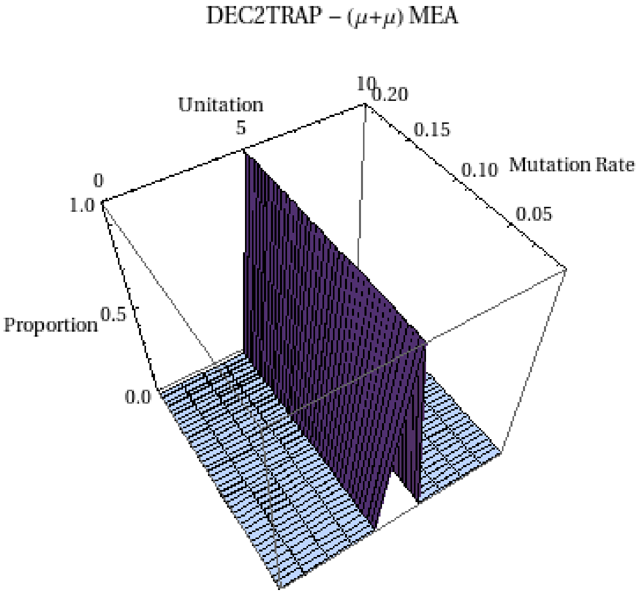


Figure 8.9: DEC2TRAP Metropolis EA fixed point surface

8.4 Empirical Validation

Rowe [325] validated the correctness of this type of analysis by conducting several experiments on real GAs with large populations optimizing functions of unitation. He showed that real populations do converge to the theoretical fixed points predicted by the model. A future work may be devoted to exploring an empirical validation. It will be interesting to study the effect of various population sizes and the convergence to these fixed points.

8.5 Closing Remarks and Future Ideas

Using fixed-points finding techniques of dynamical systems models of GAs, a number of fitness functions were studied to examine how those fixed points move as the mutation rate changes. Additionally, two new infinite population models were built to examine the performance of simple elitist GAs on these same functions. All analysis was done by constructing, simulating and solving the $\mathcal{G}$-map with Mathematica$^{\text{TM}}$.

As with the previous chapter, this work should be a warning regarding the validity of arbitrarily chosen mutation rates. The results shown here even call into question the often cited $1/n$ from Bäck [227]. As seen, this rate of $q = 0.10$ for the functions studied would place the fixed-point dynamics above the phase change observed for several functions. In addition the value of even simple probabilistic forms of elitism is underscored.

One insight from this result is that the Perron-Frobenius fixed point for the $\mu + \mu$ MEA without proportional selection is identical to the stationary distribution for the Markov chain of the corresponding $1 + 1$ MEA. As in the previous chapter, this result does not extend to the full GA with crossover.

:

CHAPTER 9

ON BISTABILITY AND RECOMBINATION

Natural selection is the editor, rather than the composer, of the genetic message.

Jack L. King and Thomas H. Jukes [130]

Framework	Mutation-only EAs	Crossover-enabled GAs
Time Complexity Analysis		
Dynamical Systems Analysis	✓	✓
Empirical Analysis	✓	✓
Methods: Infinite population model construction and analysis, experiments		

Figure 9.1: Chapter 9 Framework

9.1 Introduction

This chapter examines a fitness function showing it to be deceptive to the standard GA while easy to optimize for the EA. A dynamical systems infinite population model is constructed and analyzed. Experiments are also performed. Figure 9.1 illustrates the contributions within the framework. [1]

A major goal of GA theory is to understand the role of crossover in GAs. Here it is shown that there exist situations where crossover can lead to a GA's population failing to converge to the optimum of a single-peak fitness function for a long time. This happens when selection pressure is weak, recombination is strong, and mutation is within a range that depends on the selection pressure. The disruptive aspects of crossover and mutation are responsible for this slowdown. It is conjectured that

---

[1] The chapter is summary of a paper published as Alden Wright, J. Neal Richter "Strong Recombination, Weak Selection, and Mutation" GECCO 2006, Seattle, WA July 2006. Portions reused by permission.

there are *sloping-plateau* functions where a standard GA with strong recombination will require exponentially many fitness evaluations to reach a high membership on the plateau, whereas a mutation-only EA reaches high membership quickly.

A coarse graining of the standard GA infinite population model is given and used in the analysis of examples.

9.2 Background

This chapter considers the maximization of pseudo-boolean functions $f : \Omega \longrightarrow \mathbb{R}_0^+$, where $\Omega = \{0,1\}^\ell$ denotes the space of binary strings of length ℓ. Such a function f is called a *function of unitation* if $f(x)$ depends only on the number of ones in the binary string x. Details of the fitness functions under study are presented next as well as well as highlights of prior results.

9.2.1 Needle Functions

Suzuki and Iwasa [381] investigated the role of crossover in a GA on a needle-in-the-haystack fitness (which they called a Babel-like fitness function). They developed approximate models for the time T_d that it takes a GA that starts with a population consisting of a multiple copies of a single random string to reach a population dominated by the needle string. Note that the needle string dominates the population when it is over half of the population. Their models assume linkage equilibrium[2] and includes finite population effects. They modeled both uniform crossover and a version

---

[2] A population is at linkage equilibrium if the population distribution is determined by the order-1 schemata frequencies. In other words, the frequency of a string is equal to the product of the corresponding bit frequencies. For example, suppose that for 2 bits, the frequency of $0*$ is $1/4$ and the frequency of $*0$ is $1/3$. Then the frequencies of $00, 01, 10, 11$ will be $1/12, 1/6, 1/4, 1/2$ respectively.

of multi-point crossover. They defined the acceleration due to crossover as

$$A^d_{cross} = \frac{T_d|_{\text{without crossover}}}{T_d|_{\text{with crossover}}} \tag{9.1}$$

Suzuki and Iwasa found that for appropriate mutation and crossover rates, the acceleration due to crossover could be large (up to about 11 for string length 12 and 70 for string length 20). However, when the crossover rate was too high, the domination time T_d became very large. There was an interaction between population size and mutation rate to achieve the highest acceleration: as the population size increased, the mutation rate for highest acceleration decreased. Their qualitative conclusions for this type of fitness were:

- "The crossover rate should not be to high nor too low for fast evolution."
- "The mutation rate must be adjusted to a moderate value to enhance evolutionary acceleration due to crossover."
- "To achieve a large acceleration effect by crossover, the order of the advantageous schemata to be created needs to be sufficiently large."

Jansen and Wegener [295, 300, 350] have given carefully constructed fitness functions where a GA with crossover can find the optimal point in polynomial time (with high probability), whereas a mutation based GA without crossover will require exponential time to find the optimum. In these examples, crossover is able to jump a large gap in the fitness by recombining strings on the edge of the gap. Their work showed that crossover is most helpful in overcoming some Hamming boundaries via the use of building blocks within a population.

9.2.2 Bistability and Related Phenomena

Boerlijst et al. [382] introduced bistability in the context of a model of viral recombination. Bistability refers to a dynamical system with two stable fixed points. In the context of evolutionary computation, bistability is the situation where a dynamical systems model of an evolutionary computation algorithm applied to a single-peak fitness landscape has two stable fixed points.

A tractable dynamical systems model of bistability was given in [383, 384] for a genetic algorithm with proportional selection and gene pool crossover. Gene pool crossover is equivalent to an assumption of linkage equilibrium, and is used in some estimation of distribution algorithms such as UMDA [385] and PBIL [386]. In the case of a needle-in-the-haystack fitness function:, the fixed points can be found by solving a single-variable equation, and the stability of fixed points also determined by a single variable equation. Thus, these infinite population model results apply for all string lengths. Gene pool crossover can be used as an approximation to uniform crossover. This work was extended to tournament selection in [387].

In the cases investigated in [376, 383, 384, 387] one of the fixed points will be close to a uniform population consisting of copies of the best individual, and the other fixed point will be close to the population with equal representation of all strings in the search space (the center of the simplex). In practical terms, this can mean that when the GA is started with a random initial population, the algorithm can get *stuck* near the center of the simplex fixed point and take a very long time to move near the fitness peak fixed point. However, when started with a population near the fitness peak or the corresponding fixed point, the population can stay near this fixed point for a long time. Note that a GA with standard non-zero mutation is exactly modeled by an ergodic Markov chain, and thus the GA will eventually visit every possible population.

As described above, Suzuki and Iwasa [381] found that the time T_d to domination went to infinity as the crossover rate increased. The crossover rate at which this happens for uniform crossover, for string length 12, and mutation rate 0.002 was correctly predicted in [376].

9.2.3 An Intuitive Explanation of Bistability

In a GA, selection acts to increase the frequency of more fit individuals. However, a source of variation is needed since selection by itself does not introduce any new kinds of individuals. Mutation and crossover both introduce new kinds of individuals, but they do this at the expense of disrupting some of the more fit individuals. When the bistability phenomenon prevents or slows progress towards the optimum, it is because the disruptive properties of mutation and crossover are overwhelming selection.

In the case of the needle-in-the-haystack fitness function one can be more specific. In this function, all strings have equal fitness except for the all-zeros string which has a higher fitness. Selection will increase the frequency of the all-zeros string which will increase the frequency of the zero alleles.

Crossover does not change the expected allele frequencies. However, crossover does decrease the correlation between the alleles. In other words, crossover moves the population towards linkage equilibrium. Since the all-zeros string represents the correlation between alleles, the frequency of the all-zeros string will be reduced by crossover. However, since the frequency of the zero alleles was increased by selection, even the most extreme crossover (namely gene pool crossover) will not decrease the frequency of the all-zeros string to less than what is was before selection. Thus, if there is no mutation, there will be steady (but possibly slow) progress towards a population consisting entirely of the all-zeros string. The results of [388] show that a no-mutation GA with proportional selection on a single-peak fitness landscape can have only one stable fixed point at the uniform population consisting of copies of the optimal string.

Mutation will drive the allele frequencies towards 1/2. If this mutation pressure overcomes the combined effect of selection and crossover then there will be no progress towards the optimal population.

For a specific example, let us consider a needle-in-the-haystack fitness function where the needle (the all-zeros string) has fitness 6 and all other strings have fitness 1. Let the string length $\ell = 10$. Let C be the center of the simplex population with a weight of $2^{-\ell}$ on every string, and let N be the needle string population with weight 1 on the all-zeros string and 0 weight on all other strings. Let $P = \frac{1}{25}N + \frac{24}{25}C$, and let $Q = \frac{1}{20}N + \frac{19}{20}C$. Figure 9.2 shows the infinite population model average fitness trajectories starting at P and Q. Starting at P, the fitness decreases to the fitness of the center-of-the-simplex fixed point. In other words, the GA is going downhill on the fitness landscape. This is very counter-intuitive. On the other hand, starting at Q, which is a little closer to the needle, fitness first decreases and then increases to the fitness of the needle fixed point.

Prior work suggests several ways to avoid bistability. There is always a range of mutation rates for bistability. Thus, either sufficiently raising or lowering the mutation rate may move the GA out of the bistable range. However, the lowest mutation rate for bistability may be extremely small, and the highest may be impractically large. Increasing the strength of selection may eliminate bistability. For example, if the needle height is raised sufficiently in a GA that uses proportional selection, bistability will be avoided. However, for the needle-in-the-haystack fitness, GAs with binary tournament selection and truncation selection have been shown to have bistability. Sufficiently reducing the crossover rate or changing to a weaker recombination (such as reducing the number of crossover points for multi-point crossover) will avoid bistability.

9.3 Summary of Bistability Results

Wright and Richter [389] extended previous work on bistability. Previous bistability work emphasized the needle-in-the-haystack fitness function. A finite population GA with random initialization requires a population size that is exponential in the string length to be influenced by the needle string. Otherwise, the GA sees only a flat fitness landscape. Thus, it is of interest to understand how bistability scales with string length in a situation where the size of a finite population does not increase exponentially with the string length.

9.3.1 Reduced State Infinite Population Model

A course-graining of the full infinite population model was derived with an argument that trajectories predicted by the reduced model would be compatible with the full model. This course graining necessitated derivations of new equations for the components of the G-map. The mutation operator $\mathcal{U}$ was altered to show transition probabilities between unitation levels as in [375] and previous chapters of this work. The crossover operator $\mathcal{C}$ was similarly reduced in states to establish the transition probabilities that two parents from arbitrary unitation classes would recombine with uniform crossover to produce a child string in a given unitation class.

Once the model was derived it was implemented and used in a set of model iteration studies with results below. As predicted by previous experimental results, the model shows that there are two dominant trajectories, one towards populations

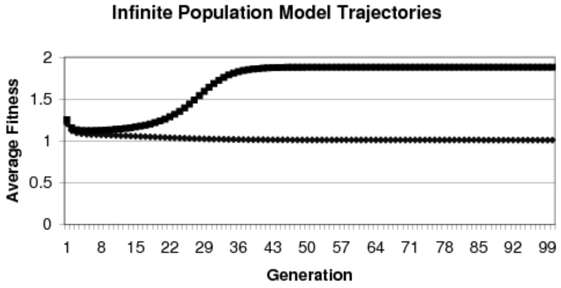


Figure 9.2: Unitation Model Trajectories Needle Fitness, $\ell = 10$

with high membership in the optimal string, and another towards populations in the center of the simplex - meaning they are well mixed with ones and zeros. Note that once the initial population selected a trajectory, there were no phase changes to swap trajectories to the other fixed point.

9.3.2 Limitations of the Model

A standard way to apply a dynamical system model is to find the fixed points then calculate their stability. One can do this with some GA models, then hope that the corresponding finite population GA will spend a lot of time near stable fixed points. There are some reasons why this may not happen.

The behavior of the infinite population model will be influenced by all points in the search space. Thus, if there are points or regions of the search space that a finite population GA is unlikely to sample, then these points or regions may have a substantial influence on the behavior of the infinite population model that is not reflected in the behavior of a typical run of the finite population GA.

An example is the needle-in-the-haystack fitness. Suppose that the string length is at least 20 and the population size is 1000 or less. In the infinite population model, there will be a selective pressure towards the needle string. If the mutation rate is not too high, there will be a stable fixed point near to the uniform population consisting of copies of the needle string. However, a finite population GA will be unlikely to ever sample the needle string in a reasonable number of generations, so the finite population GA will be doing random search in almost all runs.

Genetic drift is another reason why the infinite population model may not predict the behavior of a finite population GA. The infinite population model is a deterministic dynamical system, whereas the finite population GA is a stochastic system. It is expected that the finite population GA will periodically jump from the domain of attraction of one stable fixed point to the domain of attraction of another. For a sufficiently large population size and a sufficiently attracting fixed point, this will happen relatively infrequently, but for a small population size, this may happen more frequently. For a small population size, it is more likely that the next population will be further from the expected population. In fact, theorem 3.5 of [363] shows that the variance of the distance of the next population from the expected next population is proportional to the inverse population size.

Furthermore, there may be unstable fixed points that strongly influence the behavior of both the infinite population model and the finite population GA. The infinite population model will take small steps (i. e. not move very far in each generation) when it is sufficiently close to an unstable fixed point. The same can be true for the finite population GA. For example, a fixed point may be unstable due to the attraction of a high-fitness point or region in the search space that the finite population GA is unlikely to sample, and thus the finite population GA may *stagnate* near the unstable fixed point until it does sample the high fitness point or region.

Despite the above limitations of the infinite population model, it still *usually* (in some imprecise sense) does a good job of predicting the behavior of a GA. Chapter 8 of [363] provides examples of how populations tend to stay close to stable fixed points. Rowe [375] provides a number of examples where the infinite population model gives

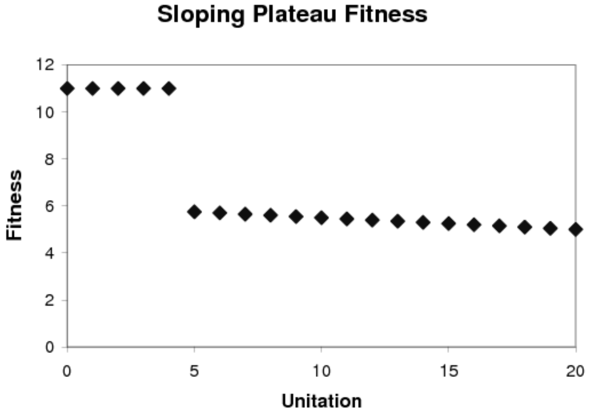


Figure 9.3: Sloping Plateau Fitness, $\ell = 20$, $k = 5$, $a = 5$, $b = 5$

very good explanations of the behavior of a finite population GA. However, it helps to keep the above limitations in mind when applying these models.

9.4 Results

In this section results are presented that illustrate bistability for string lengths of 50 and 100. The *sloping plateau* fitness functions are defined, and results are given using both the unitation model and finite population GA runs.

The sloping plateau functions are defined by

$$P_{a,b,k}(x) = \begin{cases} a+b+1 & \text{if } |x| < k \\ b + (\ell - |x|)/\ell & \text{if } |x| \geq k \end{cases}$$

The function has a plateau of fitness height $a+b+1$ for strings whose unitation is less than k. The plateau is the global optimum. For unitation classes greater than or equal to k, there is a gradual linear slope up to the plateau. The value of b determines the slope: a larger value of b means that the slope up to the plateau is more gradual. The value of a determines the height of the plateau: a larger value of a means that the fitness plateau is higher above the non-plateau points. The sloping plateau fitness for $\ell = 20$, $k = 5$, $a = 5$, and $b = 5$ is plotted in Figure 9.3.

The goal is showing situations where crossover is harmful to the performance of a GA. When there is bistability, there is a stable fixed point near the center of the simplex. The sloping plateau functions are designed to give weak but nonzero selection pressure near the center of the simplex. Once the plateau is reached, then there is reasonably strong selection pressure to stay on the plateau.

More fitness structure on the plateau could have been added, such as an isolated optimum at the end. However, this would not contribute to our discussion of bistability, and it might have led to problems with the correspondence between the infinite population model and finite population behavior.

9.4.1 Model Results for the Sloping Plateau

The results of using the unitation infinite population model on plateau functions are reviewed here. The model was coded in Maple$^{\text{TM}}$. The correctness was checked by comparison to the previously coded full infinite population model.

Bistability for a particular setting of the parameters was checked by iterating the with-crossover model for two different starting populations: The model was started from the center of the simplex and from a population with all weight on the all-zeros string. The model was iterated for up to 2000 generations or until it converged. Here convergence was defined as two successive populations having an absolute value difference in their population fitness less than 10^{-9}. If the final populations from the two starting points were substantially different, bistability was deduced.

For $\ell = 50$, $a = b = 5$, there was bistability for all values of k from 1 to 14. For small values of k, mutation rates for bistability were low, and for large values of k, mutation rates for bistability were high. For example, for $k = 5$, there was bistability

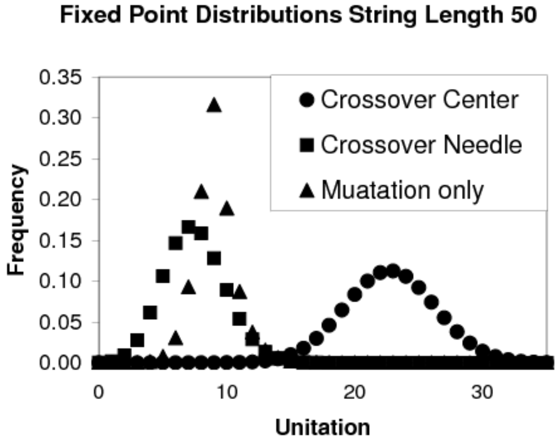


Figure 9.4: Fixed Point Distributions, $\ell = 50$, $k = 10$, $a = 5$, $b = 5$, $\mu = 0.01$

between $1/(8\ell)$ and $1/(2\ell)$, and for $k = 15$ there was bistability between $1/\ell$ and $3/(2\ell)$. The critical mutation rates for bistability decreased as k decreased.

The model without crossover was also run on the same two starting populations for comparison. The Perron-Frobenius theorem implies that there can be only one stable fixed point for the model without crossover, and this is what we observed. The three fixed point distributions for the sloping plateau fitness with $\ell = 50$, $a = b = 5$, $k = 10$, and $\mu = 1/(2 * \ell) = 0.01$ are shown in Figure 9.4.

Boerlijst et al [382] gave results for an approximate plateau fitness function with string length 15. Their results were replicated in [377]. Their approximate plateau function had a fitness of 5 for unitation class 0, 4.8 for unitation class 1, 4.6 for unitation class 2, and 3.5 for the remaining unitation classes. They compared the needle fixed point distribution with mutation rate 0.011 and recombination rate 0.5 with the non-recombination fixed point for the same mutation rate. They found that the needle fixed point had a higher frequency for unitation classes 0 and 1 and lower frequencies for unitation classes 6 and higher. Thus, the with-recombination distribution had less variance (was more compact) than the without-recombination distribution. In Figure 9.4, the with-recombination fixed-point distribution has a higher frequency for the low unitation classes (agreeing with the Boerlijst results), but the variance of the with-recombination distribution has larger variance and is less compact than the without-recombination distribution (which disagrees with the Boerlijst results).

9.4.2 Experimental Results for the Sloping Plateau

The results in this section are for a generational GA that uses proportional (roulette-wheel) selection, standard mutation, and uniform crossover either with crossover rates of 0, 1/2, and 1. Runs were made with mutation rates of $\mu = 1/(8\ell)$, $1/(4\ell)$, $1/(2\ell)$, $3/(4\ell)$, $1/\ell$, and $5/(4\ell)$. The population size for all runs described here was 10,000. Figures 9.5, 9.6, 9.8, and 9.9 all show 1-standard deviation error bars.

Experiments were done for $\ell = 50, 100$, $a = b = 5$. One set of runs was done to determine the number of generations necessary to reach the plateau, and another set was done to count the number of plateau points and determine the average fitness after 200 or 300 generations.

The first set of experiments were designed to find the waiting time for the GA to hit the first plateau (or optimal) point with and without crossover for $\ell = 50, k = 10$, and various mutation rates. The value $k = 10$ was chosen so that an initial random population of size 10,000 would not be likely to contain plateau points, but the GA would not take very long to get to the plateau. Figure 9.5 shows that the with-crossover GA is reaching the plateau much more rapidly than the without-crossover GA. Each result is the average of 250 runs.

A second similar experiment with $\ell = 100, k = 28$ was run as well with very similar results. Figure 9.6 shows again that the uniform crossover operator is beneficial for first hitting time of the plateau. Each result is the average of 576 runs.

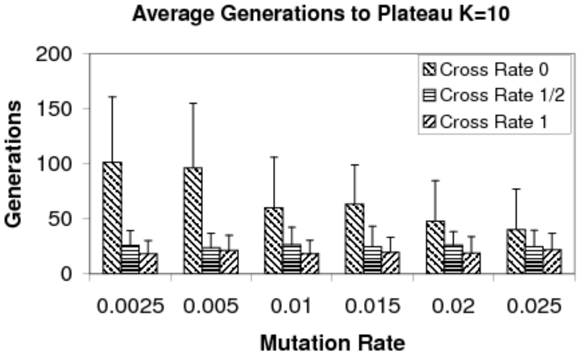


Figure 9.5: Generations to Optimum, $\ell = 50$, $k = 10$

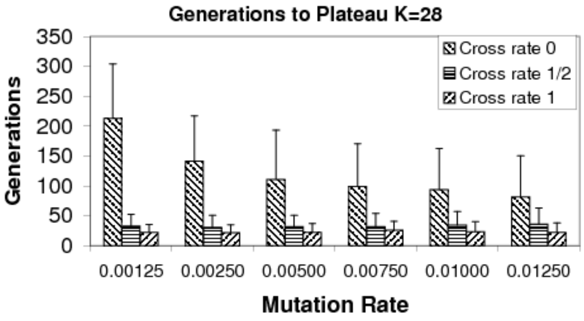


Figure 9.6: Generations to Optimum, $\ell = 100$, $k = 28$

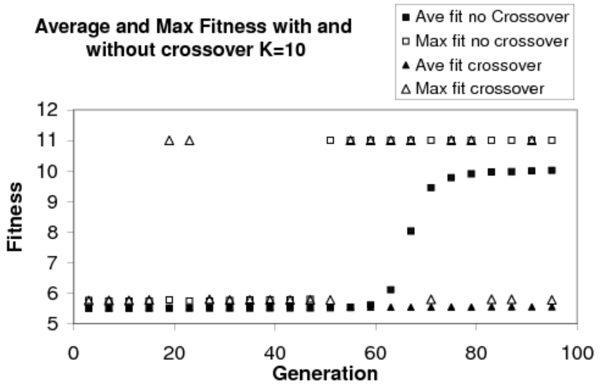


Figure 9.7: GA runs with and without crossover, $\ell = 50$, $k = 10$, population size 10000

The next set of experiments explored the average fitness over a larger number of generations for the previous two plateau functions. Bistability induced by the crossover operator has a strong effect here. The results are very counter-intuitive. The previous section showed a drastically shortened discovery time for optimal strings with crossover. Yet when the GA is allowed to run past that point, crossover inhibits the GA from accumulating many individuals on the plateau in the population. For all runs except those with very small mutation rates relative to $1/\ell$, the mutation-only GA outperformed two GAs with crossover by a significant margin. The GA with very small mutation rates was not bistable. When the GA was bistable, the mutation-only GAs were better able to accumulate highly fit strings on the plateau, producing an increase in the average fitness of the population.

Figure 9.7 demonstrates that this is the typical situation. At generation 19, the with-crossover run first hits the plateau. From then on, there are intermittent copies of plateau strings, but they do not accumulate, so the average fitness stays below 6. The without-crossover GA does not hit the plateau until about generation 50, and then starting at about generation 60, it quickly accumulates plateau strings to bring the average fitness up to around 10. Many more generations could be shown that would look like the last generations on the graph. The with-crossover GA will continue to have intermittent plateau strings and low fitness, while the without-crossover GA will maintain high fitness and many plateau strings.

9.4.3 Bistability and Finite Populations

The large finite population effects of bistability are well illustrated by considering the case where $\ell = 50$ and $k = 10$. The average fitness after 200 generations and various mutation rates is shown in Figure 9.8. Each result is the average of 100 runs, and error bars are shown. In looking at this figure, the reader should keep in mind that the fitness of plateau strings is $a + b + 1 = 11$ and the maximum fitness of non-plateau strings is $b + (\ell - k)/\ell = 5 + (50 - 10)/50 = 5.8$. For crossover rate 1, the four larger mutation rates, namely $1/\ell/2 = 0.010$, $3/\ell/4 = 0.015$, $1/\ell = 0.02$, and $5/\ell/4 = 0.025$ are all infinite-population bistable, and the center-of-simplex fixed point for $1/\ell/4 = 0.005$ is just barely unstable, while only the needle fixed point for $1/\ell/8 = 0.0025$ is stable.

For mutation rate $\mu/2 = 0.005$ the reader might compare to Figure 9.4. The center-of-simplex fixed point distribution in Figure 9.4 has a very small weight on plateau points, and that is what the finite population experiment shows with only 25 of 100 with-crossover runs ending with populations containing plateau points, and for those population with plateau points, there were at most 2 of these points. Figure 9.5 shows that all populations are likely to have hit the plateau by 200 generations, but they cannot maintain any substantial number of points on the plateau. On the other hand, the mutation-only fixed point distribution of Figure 9.4 has a heavy weight on the plateau, and over 85% of these runs ended with populations containing over 1000 plateau strings.

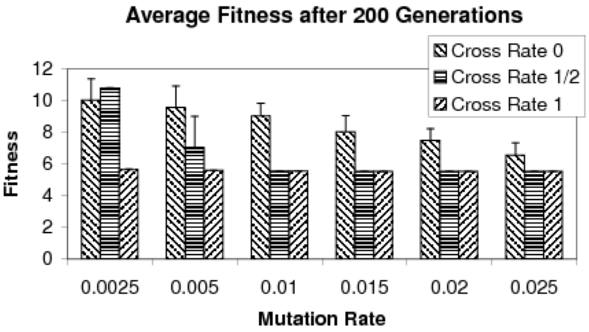


Figure 9.8: Average Fitness, $\ell = 50$, $k = 10$, 200 generations

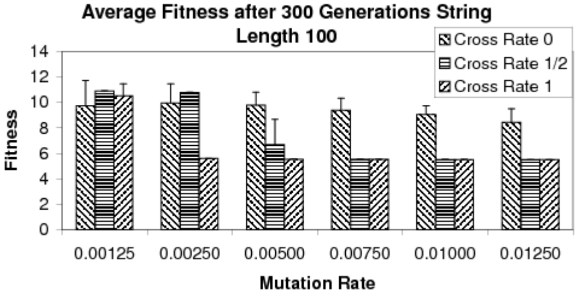


Figure 9.9: Average Fitness, $\ell = 100$, $k = 28$, 300 generations

For the other bistable and nearly bistable mutation rates, the results are similar. For $\mu = 1/\ell/8 = 0.0025$, the GA with crossover rate 1/2 was able to achieve as high an average fitness as the no-crossover GA.

Results for $\ell = 50$, $k = 28$, and 300 generations are shown in Figure 9.9. For mutation rates $\mu = 1/(8\ell)$ and $1/(4\ell)$, the infinite population model was not bistable. For mutation rates $1/(2\ell)$, $3/(4\ell)$, $1/\ell$, and $5/(4\ell)$ the model was bistable. Each result is the average of 66 runs.

9.5 Closing Remarks and Future Ideas

Crossover in a GA can be both very beneficial and very harmful. In this bistable situation, the GA has failed. The sloping plateau fitness functions that were used to demonstrate this are unimodal, simply defined, and not terribly contrived. Bistability occurs in situations with relatively strong recombination, weak selection, and an appropriate level of mutation. Crossover can also accelerate the progress of the GA to wherever it is going.

What is meant by weak selection? If a rank-based selection method had been used on the sloping plateau fitness, the GA would move quickly up the slope and this would probably have eliminated bistability. However, previous results on the needle-in-the-haystack fitness show bistability with binary tournament and truncation selection. Thus, one would expect that these selection methods would give bistability with the plateau fitness. In this case, strength of selection for the infinite population model might be determined by the fraction of the search space that is on the plateau.

There is relatively little work on the bistability effect. While crossover is known to be disruptive in some contexts, much of this is lore and conjecture. Demonstrated, repeatable situations with simple structure are rare. This work does not necessarily extend to multi-modal landscapes, yet it looks likely from basic knowledge of dynamical systems that examples of *backwards evolution* in more complex functions should exist.

CHAPTER 10

MOVEMENT OF VERTEX FIXED POINTS

> The *ultimate* source of all gene variation is mutation. But once a variation exists at an individual loci, recombination becomes the source of virtually endless combinations of alleles.
>
> Bruce Wallace [390]

Framework	Mutation-only EAs	Crossover-enabled GAs
Time Complexity Analysis		
Dynamical Systems Analysis		✓
Empirical Analysis		
Methods: Infinite population model construction and analysis		

Figure 10.1: Chapter 10 Framework

10.1 Introduction

In the mutation-selection (no crossover) Vose model, the Perron-Frobenius theorem [181] dictates that the system has only one fixed-point inside the simplex. However, Vose and Wright [388] show that with a crossover-selection GA (proportional selection and zero mutation) there can exist stable fixed-points at the vertices of the simplex (not all vertex fixed-points are stable/attracting). This work studies the movement of vertex and interior fixed points by varying parameters of the infinite population model. These results were obtained by construction, simulation and analysis of the $\mathcal{G}$-map. Figure 9.1 illustrates the contributions within the framework.

10.2 Vertex Fixed Points for Simple Functions

For the Simple GA a system let n be the string length and $N = 2^n$ be the search space size. Let $n = 2$ and $N = 4$ and define the BINEEDLE function be defined below with $a = 1$. For now assume that the mutation-rate μ is zero.

$$BINEEDLE\ f(x) = \left\{\begin{matrix} 1+a & \text{all ones string} \\ 1 & \text{otherwise} \\ 1+a & \text{all zeros string} \end{matrix}\right\} \tag{10.1}$$

This system produces this set of possible strings and fitness values. The index of this state is given as well, where this is defined as the decimal value of the binary string.

bitstring	*fitness*	*index*
00	2	1
01	1	2
10	1	3
11	2	4

(10.2)

Just as in previous chapters, the GA-map $\mathcal{G}$ on this function operates within a $N-1$ dimensional simplex Λ. The GA-map is

$$p_{t+1} = \mathcal{G}(p_t) = \mathcal{U} \circ \mathcal{C} \circ \mathcal{F}(p_t) \tag{10.3}$$

where the operators are $\mathcal{U}$ for mutation, $\mathcal{C}$ for crossover and $\mathcal{F}$ for selection. For now assume that the $\mathcal{U}$ is the identity map, meaning it has no effect and the muta-

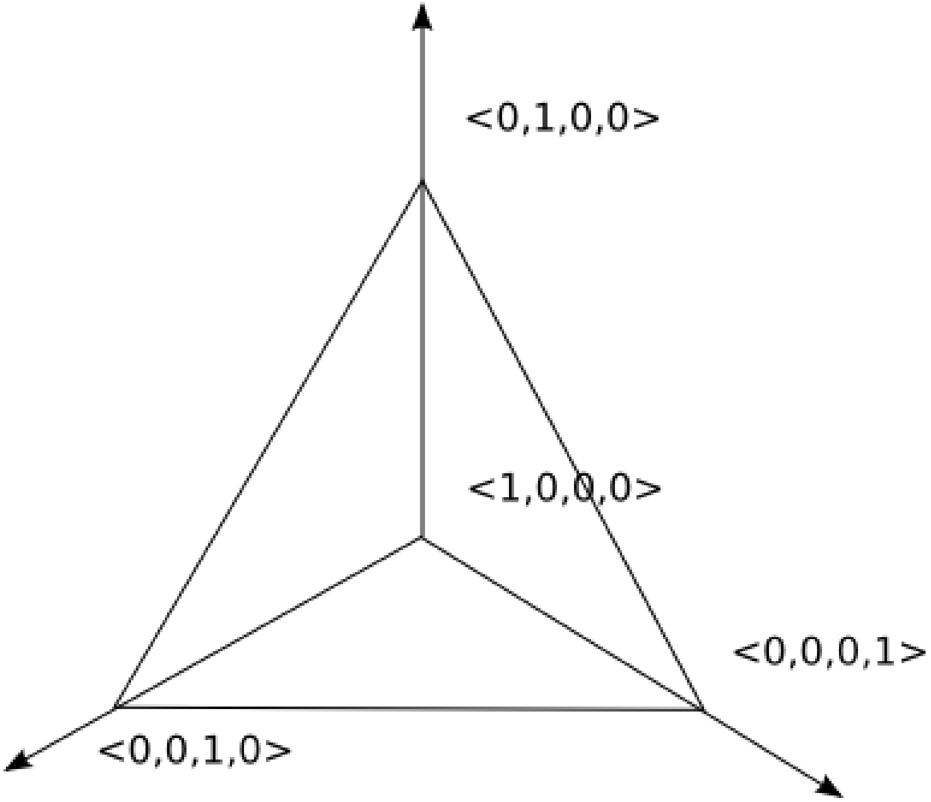


Figure 10.2: 4 dimensional simplex embedded in 3 dimensions

tion rate $\mu = 0$. Iterating $\mathcal{G}$ given some starting population distribution maps that distribution to another distribution.

A population exists on the vertex of that simplex when it contains only copies of one bitstring. For example when the population consists only of copies of 00, it exists at the $< 1, 0, 0, 0 >$ vertex of the simplex. This $3 - D$ simplex is visualized in Figure 10.2.

Vose and Wright [388] detailed the analysis of vertex fixed points. The key finding of their work that the stability of the vertex fixed points can be calculated using a relatively simple formula. Let the population $p_{v1} =< 1, 0, 0, 0 >$ mean that it contains only copies of 00. This relationship holds for p_{v1} and all vertex fixed points:

$$p_{v1} = \mathcal{G}(p_{v1}) \tag{10.4}$$

Each of the other vertexes are fixed points as well as the center of the simplex point $< 1/4, 1/4, 1/4, 1/4 >$. This is easily verifiable by simply iterating the GA map on these points as the starting population distribution.

10.3 Stability Analysis of Fixed Points

The next crucial question is determining the stability of these fixed points. From dynamical systems theory shows there are different types of fixed points. The three types relevant to this discussion are stable, unstable and saddle points. In the language of fixed points (see for example Strogatz [182]) a fixed point x^* is (asymptotically) stable and attracting if trajectories that start near x^* both stay nearby for all time as well as approach x as $t \to \infty$. A fixed point x^* is unstable trajectories starting near x^* do not stay nearby for all time and trajectories are not approaching x^* as $t \to \infty$.

A saddle fixed point x^* is more complicated, they contain stable and unstable manifolds. A saddle fixed point is classified as a sub-type of *unstable* fixed points. It contains an unstable manifold where iterations move away from the fixed point. Saddle points also contain a (typically lower dimensional) stable manifold. This is defined to be the subset of initial conditions x_0 for which $x_t \to x^*$ as $t \to \infty$ when the system is iterated. It is not always possible to represent these stable manifolds within a digital computer.

Saddle points can contain trajectories within the unstable manifold that are *eventually repelling*. In loose language, there exist initial conditions for which iterations

Table 10.1: BINEEDLE vertex fixed points, one-point crossover $\chi = 1.0$ and $\mu = 0$.

population	*eigenvalues*	*type*
$< 1, 0, 0, 0 >$	$[1/2, 1/2, 0, 0]$	*Stable*
$< 0, 1, 0, 0 >$	$[2, 2, 0, 0]$	Saddle
$< 0, 0, 1, 0 >$	$[2, 2, 0, 0]$	Saddle
$< 0, 0, 0, 1 >$	$[1/2, 1/2, 0, 0]$	*Stable*
$< 1/4, 1/4, 1/4, 1/4 >$	$[4/3, 2/3, 0, 0]$	Saddle

move towards the fixed-point for a time, only to then eventually begin to move away from the fixed-point. See van Nimwegen :thesis for instances of this behavior in GAs, he calls them meta-stable regions. See any dynamical systems book for generic examples of these effects.

How can the stability of the fixed points be directly calculated? Assuming the system is differentiable (the $\mathcal{G}$-map is differentiable) the basic procedure is to compute the derivative about the fixed point in question and then compute the eigenvalues of the derivative. For discrete dynamical systems if all eigenvalues are less than one, then the fixed point is stable. If all are greater than 1, then it's an unstable fixed point. If there is a mix of values above and below 1, then the fixed point is a saddle point. The differential of $\mathcal{G}$ is given below. Let F be the $n \times n$ diagonal matrix $F_{i,j} = \delta_{i,j} f(i)$ and let 1^T be an $n \times n$ diagonal matrix of ones that is transposed.

See [388] for derivation and proof.

$$d\mathcal{G}_x = \frac{1}{1^T F x} d\mathcal{M}_{\frac{Fx}{1^t Fx}} F P \;\; where \;\; P = I - x \frac{1^T F}{1^T F x} \tag{10.5}$$

Getting back to the BINEEDLE example above, one can calculate the the stability of the five fixed points in question. For the $\mathcal{G}$-map for one-point crossover (rate $\chi = 1.0$) and mutation $\mu = 0$:

Table 10.2: BINEEDLE vertex fixed points, uniform crossover $\chi = 1.0$ and $\mu = 0$.

population	*eigenvalues*	*type*
$< 1, 0, 0, 0 >$	$[1/2, 1/2, 1/2, 0]$	*Stable*
$< 0, 1, 0, 0 >$	$[2, 2, 1/2, 0]$	Saddle
$< 0, 0, 1, 0 >$	$[2, 2, 1/2, 0]$	Saddle
$< 0, 0, 0, 1 >$	$[1/2, 1/2, 1/2, 0]$	*Stable*
$< 1/4, 1/4, 1/4, 1/4 >$	$[4/3, 2/3, 4/9, 0]$	Saddle

Note the switching of the $< 0, 1, 0, 0 >$ and $< 0, 0, 1, 0 >$ vertices from unstable to saddle points with swapping of crossover methods. If the process is repeated for one-point crossover $\chi = 0.5$, then the results are the same as that of uniform crossover. It is conjectured that the stable manifolds for the saddle vertex fixed points of uniform crossover exist outside the simplex. These outcomes can be confirmed by simply iterating the $\mathcal{G}$.

For the center of the simplex population $< 1/4, 1/4, 1/4, 1/4 >$ the iterates return the same population. However if it is perturbed slightly to $< 0.2499000, 0.250033333, 0.250033333, 0.250033333 >$, then the iterates converge to the $< 0, 0, 0, 1 >$ stable fixed point. For $< 0.250033333, 0.250033333, 0.250033333, 0.2499000 >$ it converges to the $< 1, 0, 0, 0 >$. These are likely portions of the unstable manifolds, though the exact boundaries are not defined here.

The stable manifold contains the point $< 0, 1/2, 1/2, 0 >$ which converges to $< 1/4, 1/4, 1/4, 1/4 >$ in one step. Note that this is the Hardy-Weinberg effect kicking in and taking the population to linkage equilibrium in one step,

This process was repeated completed for string length $n = 4$, $N = 2^4 = 16$ for BINEEDLE with $a = 1$. For both one-point and uniform crossover the only stable fixed points are at the all-ones and all-zeros string vertices. All other vertex fixed points are saddle points. The uniform population fixed point in the center of the simplex is also a saddle point.

If this analysis is done for the NEEDLE function, with fitness 1 for all string values except the all-ones string with fitness 2, then all vertex fixed points are unstable except the uniform population of the all-ones string. The center of the simplex has complex eigenvalues and is not a fixed point here.

How many other fixed points exist in the simplex? It is possible to compute the exact number if one were to take the approach of Wright et al [383] and use the *gene pool* crossover model. This model allows the reduction of the $\mathcal{G}$-map to a solvable system of equations for which the fixed points can be computed by means other than iteration of the system. For $n = 4$ this is a system of 4 equations and unknowns that is numerically solvable. Note that for the gene-pool model it has been shown that a vertex fixed point is stable iff all neighboring vertices 1-bit away have lower fitness. See work by Mühlenbein for more details [391, 385].

Gedeon, Hayes and Swanson (of Montana State University) proved [392] that for a 'typical' mixing operator $\mathcal{M}$ with non-zero mutation the $\mathcal{G}$-map contains finitely many fixed points. Furthermore they argue that one can reasonably assume that for a very large (yet finite) population that the GA has finitely many fixed points.

Vose and Wright [388] have an outstanding conjecture concerning the stability of fixing points in the simplex:

Conjecture 47. *If mutation is zero, the only stable fixed points of* $\mathcal{G}$ *are at the vertices of the simplex* Λ.

The intuitive explanation for the truth of this conjecture is that any population inside the simplex (off of the vertices) must contain membership in multiple elements of the search space. The action of proportional selection against this population will mean in the long term of iterations that the fittest members of the search space will accumulate more membership. All fitness functions are defined on an individual basis, as such they can not be used to specify particular population distributions. However, this argument does not eliminate the possibility of stable fixed points of off vertex points. Wright and Bidwell [200] restated this conjecture in terms of asymptotically stable fixed points.

If one could define a $\mathcal{G}$-map that explicitly promotes a population distribution then a counter example would be possible, yet the algorithm would not be a GA used by Vose and Wright. One idea would be to use *fitness sharing* [67] where the fitness of an individual is a function of both the string and the number of other members of the population at that string. This method of selection explicitly punishes the condition where too many members of a population are at one string relative to its fitness.

$$f_{shared}(x) = \frac{f(x)}{|x|_{population}} \tag{10.6}$$

Conjecture 48. *There exists a fitness-sharing, crossover and no-mutation GA whose* $\mathcal{G}$*-map has stable fixed points in the interior of the simplex* Λ*.*

Going back to the unaltered $\mathcal{G}$-map, examples can be constructed with fitness functions with equal (and maximal) fitness in two or more search space elements where these elements are Hamming neighbors. If a population were to be initialized with equal/uniform membership in only these elements, this situation with the no-mutation GA should devolve into a random walk on that Hamming surface of the simplex. Once a point hit a vertex it would stay there. This is an example of drift as studied in population genetics.

Example fitness function for $n = 4$, $N = 2^4 = 16$:

$$CONCATNEEDLE :< 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1 > \qquad (10.7)$$

For an initial population of $< 1/2, 1/2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 >$, or any population $< X, Y, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 >$ such that $X + Y = 1$, the eigenvalues of the system here have a single entry of 1 and all other entries are less than 1. This means that the surface itself is stable while specific points are not. In this case the surface is a line-segment between the vertexes. Each point of the segment is stable, yet not asymptotically stable.

Conjecture 49. *For particular fitness functions, there exist attracting Hamming surfaces of the simplex for the no-mutation GA with crossover and proportional selection. This constitutes a counter example to the conjecture of Vose-Wright.*

Note that this is just another counter example to the original conjecture, which as Wright and Bidwell point out should have been stated in terms of asymptotic stability.

10.4 Perron-Frobenius Fixed Points

As a reminder, recall from previous chapters that for a GA consisting of proportional selection, mutation and zero crossover the Perron-Frobenius theorem dictates that there can exist only one stable fixed-point in the simplex. Next a set of questions are posed concerning the disparity between the number and location of fixed points on a GA with zero mutation versus one with zero crossover.

10.5 Questions

The dichotomy of many fixed points under crossover-selection GAs and one under mutation-selection GAs, plus basic ideas on continuity from dynamical systems suggests some questions.

Central Questions:

- What happens to the vertex fixed points when mutation is perturbed away from zero by some small ϵ?
- Do they all move inside the simplex? Do some move inside and some move outside? Changes in stability?
- What happens to the Perron-Frobenius fixed point if ϵ crossover is added?
- What happens to the fixed points if the crossover rate were varied to zero at the same time the mutation rate were varied to one?

Note that varying the mutation rate to one here is done only for abstract purposes for completeness, a mutation rate of one has no practical use in optimization.

10.6 Introducing Epsilon Mutation

For the BINEEDLE the a computational study was done to attempt to attempt determine where the vertex fixed points moved to under epsilon mutation. Note solving the full $\mathcal{G}$-map for fixed points is infeasible, see Chapter 7 of Vose [363] for a discussion. He suggests iterating the map to find them.

Here map-iteration experiments were done with initial populations set at either of the two stable vertex fixed points or the simplex-center uniform populations. A range of mutation rates were chosen and one-point crossover is used with $\chi = 1.0$. The results are represented in the table below. Note that the experiment was explicitly done for both stable vertex-populations at all-ones and all-zeros. The symmetry of the fitness function means that results for either hold for both, it's merely an exchange of ones and zeros in the state space.

Let the symbol ϕ represent a number very close to zero, here $|\phi| < 10^{-6}$. As these fixed-points are vectors of length 16, compact abbreviations of the vectors are used as appropriate. For each chosen mutation and initial condition the $\mathcal{G}$-map was iterated until observed convergence or at least 300 iterations. The stability of each observed fixed-point was calculated. For fixed-points marked as saddle points, they converged to the listed point after 500 generations.

Figure 10.3 displays an interpolated version of the bifurcation diagram. Three initial populations were used, one at the all-zeros needle, one at a uniform population, and one with 1/2 the population at the all-zeros and all-ones population.

This bifurcation is an instance of the pitchfork bifurcation, where a stable fixed point bifurcates into two stable fixed points with an unstable fixed point between them. The upper curve of Figure 10.3 is displaying *both* stable fixed points as the y-axis represents fitness and these points have the same fitness by symmetry. The lower curve displays the unstable saddle point's average fitness.

This is an interesting result in that the unstable saddle point has a stable manifold that is converged to. By the symmetry of the BINEEDLE and the initial conditions, these iterations are on the stable manifold and iterations will remain there. This type of direct iterative observation of the lower-dimensional stable manifold is not always possible.

A typical pitchfork bifurcation is given in Figure 10.4 in pictorial form with alpha as the varied parameter.

For simplicity of representing the results, the fixed-point vectors are listed in a second table and referred to in the results table.

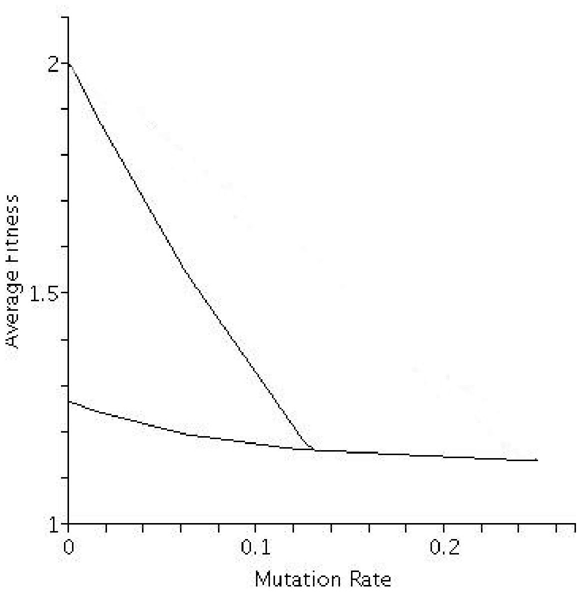


Figure 10.3: Epsilon mutation bifurcation of stable fixed points

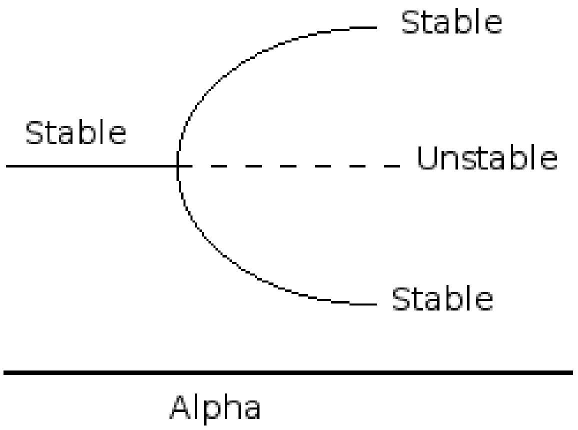


Figure 10.4: Typical pitchfork bifurcation

Table 10.3: BINEEDLE $\mathcal{G}$-map with epsilon mutation results.

μ	*initial population*	*converged f-p*	*f-p fitness*	Stability
1/65535	$< 1, 0, ..., 0, 0 >$	#1 below	2.0	*Stable*
1/65535	$< 1/N, 1/N, ..., 1/N, 1/N >$	#2 below	1.269	Saddle
1/65535	$< 1/2, 0, ..., 0, 1/2 >$	#2 below	1.269	Saddle
1/4096	$< 1, 0, ..., 0, 0 >$	#3 below	1.998	*Stable*
1/4096	$< 1/N, 1/N, ..., 1/N, 1/N >$	#4 below	1.268	Saddle
1/4096	$< 1/2, 0, ..., 0, 1/2 >$	#4 below	1.268	Saddle
1/512	$< 1, 0, ..., 0, 0 >$	#5 below	1.994	*Stable*
1/512	$< 1/N, 1/N, ..., 1/N, 1/N >$	#6 below	1.265	Saddle
1/512	$< 1/2, 0, ..., 0, 1/2 >$	#6 below	1.265	Saddle
1/64	$< 1, 0, ..., 0, 0 >$	#7 below	1.878	*Stable*
1/64	$< 1/N, 1/N, ..., 1/N, 1/N >$	#8 below	1.244	Saddle
1/64	$< 1/2, 0, ..., 0, 1/2 >$	#8 below	1.244	Saddle
1/16	$< 1, 0, ..., 0, 0 >$	#9 below	1.546	*Stable*
1/16	$< 1/N, 1/N, ..., 1/N, 1/N >$	#10 below	1.195	Saddle
1/16	$< 1/2, 0, ..., 0, 1/2 >$	#10 below	1.195	Saddle
1/8	$< 1, 0, ..., 0, 0 >$	#11 below	1.183	*Stable*
1/8	$< 1/N, 1/N, ..., 1/N, 1/N >$	#12 below	1.162	Saddle
1/8	$< 1/2, 0, ..., 0, 1/2 >$	#12 below	1.162	Saddle
1/7.65	$< 1, 0, ..., 0, 0 >$	#13 below	1.160	*Stable*
1/7.65	$< 1/N, 1/N, ..., 1/N, 1/N >$	#13 below	1.160	*Stable*
1/7.65	$< 1/2, 0, ..., 0, 1/2 >$	#13 below	1.160	*Stable*
1/4	$< 1, 0, ..., 0, 0 >$	#14 below	1.137	*Stable*
1/4	$< 1/N, 1/N, ..., 1/N, 1/N >$	#14 below	1.137	*Stable*
1/4	$< 1/2, 0, ..., 0, 1/2 >$	#14 below	1.137	*Stable*

Table 10.4: BINEEDLE mutation rates and fixed-point vectors for epsilon mutation

#	μ	*fixed-point vector*
1	1/65535	$< .99, 0.3e-4, 0.3e-4, \phi, ..., \phi >$
2	1/65535	$< 0.13, 0.076, 0.039, 0.073, 0.039, 0.022, 0.036, 0.076, 0.076, 0.036, 0.022, 0.039, 0.073, 0.039, 0.076, 0.13 >$
3	1/4096	$< 0.9, \phi, \phi, \phi, ..., \phi >$
4	1/4096	$< 0.13, 0.076, 0.039, 0.073, 0.039, 0.022, 0.036, 0.076, 0.076, 0.036, 0.022, 0.039, 0.073, 0.039, 0.076, 0.13 >$
5	1/512	$< 0.98, \phi, \phi, \phi, \phi, \phi, \phi, \phi, \phi, \phi, \phi, \phi, \phi, \phi, \phi, \phi >$
6	1/512	$< 0.13, 0.076, 0.040, 0.072, 0.040, 0.02, 0.036, 0.076, 0.076, 0.036, 0.02, 0.040, 0.072, 0.040, 0.076, 0.13$
7	1/64	$< 0.87, 0.028, 0.028, \phi, 0.028, \phi, \phi, \phi, 0.028, \phi, \phi, \phi, \phi, \phi, \phi, \phi >$
8	1/64	$< 0.12, 0.075, 0.044, 0.070, 0.044, 0.029, 0.03, 0.075, 0.075, 0.03, 0.029, 0.044, 0.070, 0.044, 0.075, 0.12 >$
9	1/16	$< 0.54, 0.087, 0.084, 0.017, 0.084, 0.015, 0.01, \phi, 0.087, 0.014, 0.015, \phi, 0.017, \phi, \phi, \phi >$
10	1/16	$< 0.097, 0.070, 0.052, 0.065, 0.052, 0.042, 0.04, 0.070, 0.070, 0.04, 0.042, 0.052, 0.065, 0.052, 0.070, 0.097 >$
11	1/8	$< 0.14, 0.085, 0.075, 0.060, 0.075, 0.049, 0.052, 0.046, 0.085, 0.050, 0.049, 0.039, 0.060, 0.039, 0.046, 0.040 >$
12	1/8	$< 0.081, 0.067, 0.057, 0.063, 0.057, 0.052, 0.054, 0.067, 0.067, 0.054, 0.052, 0.057, 0.063, 0.057, 0.067, 0.081 >$
13	1/7.65	$< 0.080, 0.066, 0.057, 0.06, 0.057, 0.052, 0.054, 0.066, 0.066, 0.054, 0.052, 0.057, 0.06, 0.057, 0.066, 0.080 >$
14	1/4	$< 0.068, 0.063, 0.061, 0.062, 0.061, 0.059, 0.059, 0.063, 0.063, 0.059, 0.059, 0.061, 0.062, 0.061, 0.063, 0.068 >$

The above results are by definition incomplete, they do not contain all fixed points. Many more initial populations would need to be tried, and unstable (non-saddle) fixed points are not observable via iteration. With small mutation rates, the overall dynamics have not changed much. There still exist three observable fixed points, two stable and one saddle point. Yet it's clearly observable that at approximately mutation rate 1/7.65 something interesting happens, a single stable fixed point splits into three fixed points for mutation rate 1/8.

However, for this $\mathcal{G}$-map one can not be certain of the locations and movement of unobserved fixed points. There could be other stable fixed points in the simplex, though this seems unlikely given the intuitive understanding of a GA with crossover and mutation operating on the BINEEDLE fitness function.

Conjecture 50. *For the GA with crossover and non-zero epsilon mutation, the stable vertex fixed-points of zero mutation GA move inside the simplex and the unstable vertex fixed-points of zero mutation GA move outside the simplex.*

10.7 Introducing Epsilon Crossover

A similar experiment was done with examining the effects of adding epsilon one-point crossover to a fixed mutation rate $\mathcal{G}$-map. The mutation rate is set to $\mu = 1/N = 1/16$, and the crossover rate is varied over a range. Again note that with zero crossover there exists only a single stable fixed point in the interior of the simplex. Figure 10.5 displays an interpolated version of the bifurcation diagram.

These results are very interesting in that they indicate a critical (observable) bifurcation of a single stable fixed into a stable fixed point and a saddle point. This happens at approximately crossover rate 1/128. Over $8,000$ iterations of the $\mathcal{G}$-map

Table 10.5: BINEEDLE $\mathcal{G}$-map with epsilon crossover results.

χ	*initial population*	*converged fixed-point*	*f-p fitness*	Stability
0	$< 1/N, ..., 1/N >$	#1 below	1.576	*Stable*
1/65536	$< 1, 0, ..., 0, 0 >$	#2 below	1.576	*Stable*
1/65536	$< 1/N, ..., 1/N >$	#2 below	1.576	*Stable*
1/65536	$< 1/2, 0, ..., 0, 1/2 >$	#2 below	1.576	*Stable*
1/128	$< 1, 0, ..., 0, 0 >$	#3 below	1.571	*Stable*
1/128	$< 1/N, ..., 1/N >$	#4 below	1.571	Saddle
1/128	$< 1/2, 0, ..., 0, 1/2 >$	#4 below	1.571	Saddle
1/64	$< 1, 0, ..., 0, 0 >$	#5 below	1.571	*Stable*
1/64	$< 1/N, ..., 1/N >$	#6 below	1.566	Saddle
1/64	$< 1/2, 0, ..., 0, 1/2 >$	#6 below	1.566	Saddle
1/16	$< 1, 0, ..., 0, 0 >$	#7 below	1.569	*Stable*
1/16	$< 1/N, ..., 1/N >$	#8 below	1.537	Saddle
1/16	$< 1/2, 0, ..., 0, 1/2 >$	#8 below	1.537	Saddle
1/8	$< 1, 0, ..., 0, 0 >$	#9 below	1.567	*Stable*
1/8	$< 1/N, ..., 1/N >$	#10 below	1.499	Saddle
1/8	$< 1/2, 0, ..., 0, 1/2 >$	#10 below	1.499	Saddle
1/4	$< 1, 0, ..., 0, 0 >$	#11 below	1.563	*Stable*
1/4	$< 1/N, ..., 1/N >$	#12 below	1.430	Saddle
1/4	$< 1/2, 0, ..., 0, 1/2 >$	#12 below	1.430	Saddle
1/2	$< 1, 0, ..., 0, 0 >$	#13 below	1.557	*Stable*
1/2	$< 1/N, ..., 1/N >$	#14 below	1.318	Saddle
1/2	$< 1/2, 0, ..., 0, 1/2 >$	#14 below	1.318	Saddle
3/4	$< 1, 0, ..., 0, 0 >$	#15 below	1.551	*Stable*
3/4	$< 1/N, ..., 1/N >$	#16 below	1.242	Saddle
3/4	$< 1/2, 0, ..., 0, 1/2 >$	#16 below	1.242	Saddle
15/16	$< 1, 0, ..., 0, 0 >$	#17 below	1.547	*Stable*
15/16	$< 1/N, ..., 1/N >$	#18 below	1.205	Saddle
15/16	$< 1/2, 0, ..., 0, 1/2 >$	#18 below	1.205	Saddle
63/64	$< 1, 0, ..., 0, 0 >$	#19 below	1.546	*Stable*
63/64	$< 1/N, ..., 1/N >$	#20 below	1.197	Saddle
63/64	$< 1/2, 0, ..., 0, 1/2 >$	#20 below	1.197	Saddle

Table 10.6: BINEEDLE crossover rates and fixed-point vectors for epsilon crossover

#	χ	*fixed-point vector*
1	0	$< 0.28, 0.041, 0.041, 0.01, 0.041, 0.01, 0.01, 0.041, 0.041, 0.01, 0.01, 0.041, 0.01, 0.041, 0.041, 0.29 >$
2	1/65536	$< 0.28, 0.041, 0.041, 0.01, 0.041, 0.01, 0.01, 0.041, 0.041, 0.01, 0.01, 0.041, 0.01, 0.041, 0.041, 0.29 >$
3	1/128	$< 0.3, 0.05, 0.049, 0.016, 0.049, 0.016, 0.016, 0.032, 0.05, 0.016, 0.016, 0.03, 0.016, 0.03, 0.032, 0.21 >$
4	1/128	$< 0.28, 0.041, 0.041, 0.016, 0.041, 0.016, 0.016, 0.041, 0.041, 0.016, 0.016, 0.041, 0.016, 0.041, 0.041, 0.28 >$
5	1/64	$< 0.49, 0.068, 0.067, 0.01, 0.067, 0.016, 0.016, 0.015, 0.068, , 0., 0.016, 0.014, 0.01, 0.014, 0.015, 0.08 >$
6	1/64	$< 0.28, 0.042, 0.041, 0.017, 0.041, 0.01, 0.016, 0.042, 0.042, 0.016, 0.01, 0.041, 0.017, 0.041, 0.042, 0.28 >$
7	1/16	$< 0.55, 0.077, 0.076, 0.016, 0.076, 0.015, 0.016, 0.0071, 0.077, 0.015, 0.015, 0.0064, 0.016, 0.006, 0.007, 0.01 >$
8	1/16	$< 0.26, 0.045, 0.04, 0.022, 0.04, 0.017, 0.017, 0.045, 0.045, 0.017, 0.017, 0.04, 0.022, 0.04, 0.045, 0.27 >$
9	1/8	$< 0.5, 0.078, 0.077, 0.016, 0.077, 0.01, 0.016, 0.0058, 0.078, 0.015, 0.01, 0.0051, 0.016, 0.0051, 0.0058, 0.01 >$
10	1/8	$< 0.24, 0.049, 0.04, 0.029, 0.04, 0.019, 0.02, 0.049, 0.049, 0.02, 0.019, 0.04, 0.029, 0.04, 0.049, 0.25 >$
11	1/4	$< 0.55, 0.08, 0.079, 0.017, 0.079, 0.015, 0.016, 0.0051, 0.08, 0.015, 0.015, 0.0043, 0.017, 0.0043, \phi, \phi >$
12	1/4	$< 0.2, 0.05, 0.041, 0.04, 0.041, 0.023, 0.02, 0.05, 0.05, 0.02, 0.023, 0.041, 0.04, 0.041, 0.05, 0.21 >$
13	1/2	$< 0.55, 0.08, 0.081, 0.017, 0.081, 0.015, 0.016, 0.0044, 0.08, 0.014, 0.015, 0.0036, 0.017, 0.0036, 0.0044, >$
14	1/2	$< 0.15, 0.065, 0.044, 0.055, 0.044, 0.03, 0.034, 0.065, 0.065, 0.034, 0.03, 0.044, 0.055, 0.044, 0.065, 0.15 >$
15	3/4	$< 0.54, 0., 0., 0.017, 0., 0.015, 0.016, 0.004, 0., 0.014, 0.015, 0.0033, 0.017, 0.0033, 0.004, 0.001 >$
16	3/4	$< 0.12, 0.069, 0.048, 0.062, 0.048, 0.037, 0.042, 0.069, 0.069, 0.042, 0.037, 0.048, 0.062, 0.048, 0.069, 0.12 >$
17	15/16	$< 0.54, 0.08, 0.084, 0.017, 0.084, 0.015, 0.01, 0.0038, 0.08, 0.014, 0.015, 0.0031, 0.017, 0.0031, 0.0038, \phi >$
18	15/16	$< 0.1, 0.07, 0.051, 0.065, 0.051, 0.041, 0.046, 0.07, 0.07, 0.046, 0.041, 0.051, 0.065, 0.051, 0.07, 0.10 >$
19	63/64	$< 0.54, 0.087, 0.084, 0.017, 0.084, 0.015, 0.01, 0.0037, 0.087, 0.014, 0.015, 0.003, 0.017, 0.003, 0.0037, \phi >$
20	63/64	$< 0.09, 0.07, 0.052, 0.065, 0.052, 0.042, 0.047, 0.07, 0.07, 0.047, 0.042, 0.052, 0.065, 0.052, 0.07, 0.098 >$

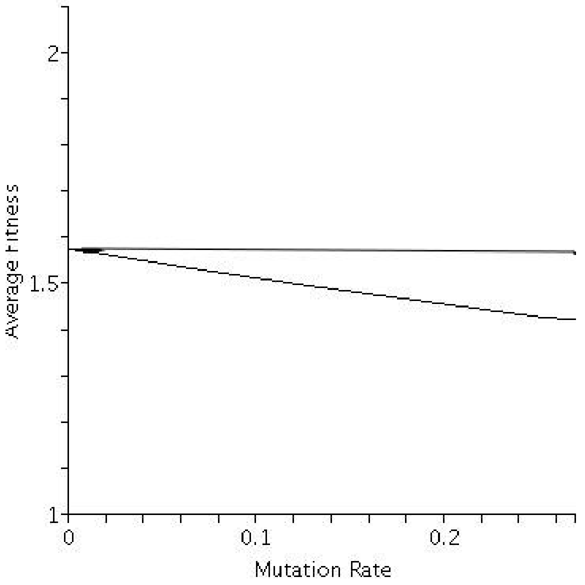


Figure 10.5: Epsilon crossover bifurcation of stable fixed points

were starting from the population uniformly at the all-zeros point. At this point that run was stopped, with the result being row 3 in the above table.

The leading eigenvalue of the derivative at this stopping point was 0.99960, meaning that the trajectory of iteration is very likely close to the stable manifold of the saddle point. The leading eigenvalue of the derivative at the fixed-point in row 4 was exactly 1. Note that these numbers are approximate and at these types of critical points the numerics of computation in binary computers can result in some inaccuracies.

Once the crossover rate grew to 1/64 and above a clear separation of iterative convergence was established between the different starting populations. At crossover rate 63/64 the interior saddle point's population distribution is such that it is approaching uniform frequency of 8% or 1/16. This indicates that the crossover operator is being quite disruptive and has destroyed the GA's ability to maintain high membership in the fittest individuals.

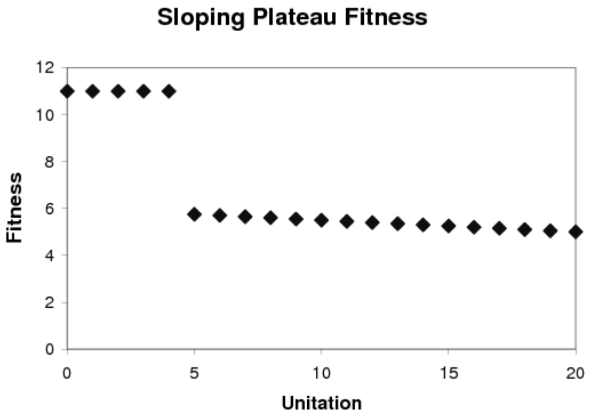


Figure 10.6: Sloping Plateau Fitness, $\ell = 20$, $k = 5$, $a = 5$, $b = 5$

10.8 Revisiting Bistability

The study next revisited the sloping plateau function from Chapter 9. The sloping plateau functions are redefined below. Figure 10.6 duplicates Figure 9.3 with a plateau function with $\ell = 20$, $k = 5$, $a = 5$, $b = 5$. Remember that ℓ and n are interchangeable here to represent string length.

$$P_{a,b,k}(x) = \begin{cases} a + b + 1 & \text{if } |x| < k \\ b + (\ell - |x|)/\ell & \text{if } |x| \geq k \end{cases}$$

The previous chapter looked at the effects of bistability and did some varying of the mutation rate to establish the critical ranges of mutation for which bistability happens. This study chose instead to hold the mutation rate steady and vary the a in a range in an effort to discover a bifurcation point. The parameters of the fitness function were changed as well to $\ell = 40$, $k = 1$, $b = 20$ and $a \in [1, 40]$. This forms a sloping needle and is shown in Figure 10.7.

A key point of this fitness function is that the *floor* area of the function slopes directly to the needle. To a simple hillclimber $(1 + 1)EA$ this function is indistin-

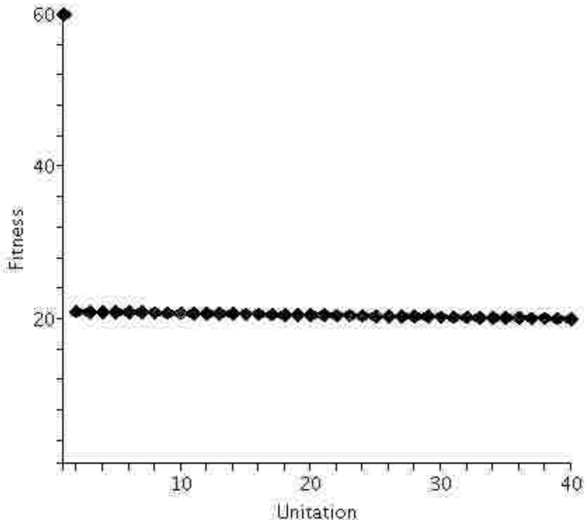


Figure 10.7: Sloping Needle Fitness, $\ell = 40$, $k = 1$, $a = 20$, $b = 20$

guishable from a function like ZEROMAX (the inverse of ONEMAX). In addition it is easily solvable by an non-crossover EA/GA with an arbitrary population size. Just as with the sloping plateau, the function is designed to deceive proportional selection. In general, any EA with a large population will be slower to optimize this function than one with a small population where the effects of 'weak selection' are muted.

The experiments were conducted as follows. At each value of $a \in [1, 40]$ the $\mathcal{G}$-map was iterated to convergence starting from two initial populations. The first is a uniform population while the second consists entirely of members on the needle at the all-zeros string. A mutation rate of $1/3\ell$ was used along with uniform crossover with rate = 1.0.

For values of $a < 21$ both initial population converged to the same center of the simplex point. At $a = 21$ a bifurcation point is reached. The stable fixed point splits into three stable fixed points. Recall that the system has symmetric fixed points near the all-ones and all-zeros strings. For increasing values of a, the average population fitness of these two fixed points climbs until the reaches approximately avg-fitness 41

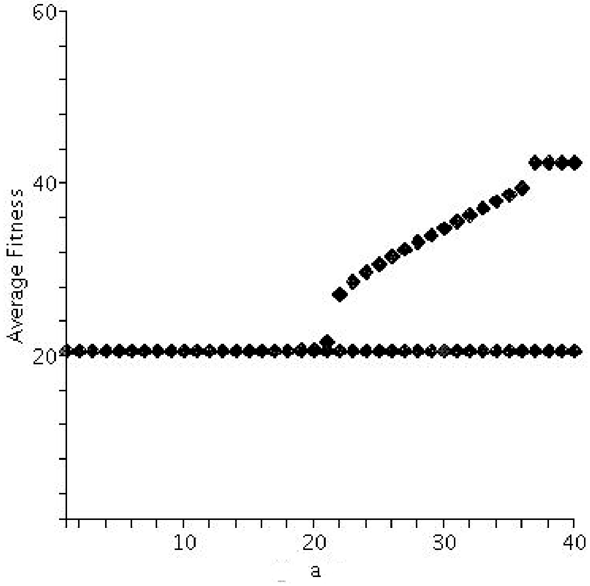


Figure 10.8: Sloping Needle fixed point bifurcation

and levels off. The fixed point for the initial uniform population remains at avg-fitness = 20.5 for all values of a tested.

The resulting bifurcation diagram is pictured below in Figure 10.8. The y-axis represents the average fitness of a fixed point and not the fixed point itself. Recall that visualizing the population distribution movement results in a 2-D surface graph like those seen in prior chapters.

No conclusions can be drawn on the movement of unobserved fixed points, yet presumably other likely unstable fixed points emerged from the bifurcation event as generally in dynamical systems a stable fixed point can not bifurcate into two without an unstable fixed point lying between them in space. See Strogatz [182], Brin and Stuck [181] or Seydel [393] for more information on bifurcations in dynamical systems.

10.9 Closing Remarks and Future Ideas

The results here show multiple computational studies of the infinite population model $\mathcal{G}$-map in Maple$^{\text{TM}}$. The first two were extensions of work on vertex fixed points by Vose and Wright [388]. Key questions were answered on where fixed points moved as mutation or crossover rate changed for the given fitness function. Stability analysis was also conducted. Not all fixed points are traceable in this way and these results say nothing about unobserved fixed points. The BINEEDLE was chosen due to its intuitive ease and optimization success with EAs, yet GAs with strong crossover fail in experiments.

The third study extended the results of Wright and Richter [389] to look at the sensitivity of the needle height in a sloping needle fitness function. This function is also deceptive to a GA with strong crossover due to the effects of weak selection and the disruptive effects of crossover's bit mixing. Interestingly, the height of the needle induces bistability in proportionate selection. Again, no conclusions can be drawn about the unobserved fixed points. A future direction would be to move to gene-pool crossover and reduce the $\mathcal{G}$-map to a system of equations and unknowns. This would allow the computation of all fixed-points and better exploration of the dynamics of high n gene-pool crossover GAs.

This is believed to be one of the few times an analysis has been done on the movement of fixed points of the $\mathcal{G}$-map with crossover.

CHAPTER 11

EXPERIMENTS ON UNITATION, ROYAL ROAD AND LONGPATH FUNCTIONS

> Unless there is an unknown factor tremendously increasing the chance of simultaneous mutations, such a process has played no part whatever in evolution.
>
> George Simpson [152]

Framework	Mutation-only EAs	Crossover-enabled GAs
Time Complexity Analysis		
Dynamical Systems Analysis		
Empirical Analysis	✓	✓
Methods: Experiments, adaptive parameter schemes		

Figure 11.1: Chapter 11 Framework

11.1 Introduction

This chapter applies various parameter adaptation schemes to a running genetic algorithm (not a model) applied to several unitation fitness functions. The effects of the schemes on optimization success and convergence speed are compared experimentally. Figure 9.1 illustrates the contributions within the framework. [1]

The classic GA suffers from the configuration problem: the difficulty of choosing optimal parameter settings. Different setups can have strong effects on the solutions found. Crossover operators, mutation operators, selection operators, and population

---

[1] An earlier version of this chapter was published as
J. Neal Richter, John Paxton. ”Adaptive Evolutionary Algorithms on Unitation, Royal Road and Longpath Functions.” IASTED Computation Intelligence Conference, July, 2005 Calgary, Alberta

size are just a few of the many parameters that are available to be modified or optimized to fit a given fitness function.

GA literature is full of empirical tricks, techniques, and rules of thumb that enable GAs to be optimized to perform better in some way by altering the parameters. May of them are specific to particular problem domains. Some researchers have found interdependencies between operators and formulated helpful heuristics to follow when designing a GA system for a specific domain [53]. However, these techniques are often analyzed on only a narrow set of fitness functions. This paper is a first empirical step in analyzing several parameter adaptive techniques on the unitation class of fitness functions, where fitness is a function of the number of ones in the binary genome.

Many researchers agree that the classic GA has configuration flaws. It can be a robust method, although when set-up improperly that property diminishes. Chief among those pitfalls is the inability of the classic GA to adapt to the changing characteristics of the solution space as the population moves around in it. For example, a GA can have difficulty if the inherent step-size of the algorithm (as defined by the operators) is larger than the optimal step-size of fitness landscape. Hill climbing algorithms can suffer from the same problem.

This paper will concentrate on mutation only Evolutionary Algorithms and use several methods to adapt the mutation rate. The experiments are restricted to binary genomes on the unitation class of fitness functions and one example of each of the Royal Road and LongPath families of functions. Unitation functions are fitness functions defined only by the number of ones in the binary genome.

The Royal Road functions were designed by Holland and coworkers to highlight the building block approach the GA was thought to take in problem solving [53]. Horn et al. designed the LongPath [304] also to highlight the supposed building block explanation of GA function.

The experiments here show a general failure of adaptive methods on simple-looking unitation functions, while the adaptive methods fare well on the well-known and challenging Royal Road problem. The adaptive method also shows improvement on the LongPath problem, which is fairly easy for the basic GA.

11.2 Types of Parameter Adaptation

Hinterding et al. [394] surveyed general GA parameter adaptation. They classify adaptation into four types and four levels. The types are:

- static (unchanging parameters)
- deterministic dynamic (parameters changed with a deterministic function)
- dynamic adaptive (parameters changed with feedback)
- dynamic self-adaptive (adaptation method encoded into chromosome)

The levels are:

- environmental (changing fitness function response to individuals with a heuristic)
- population (any parameters affecting entire population)
- individual (mutation rate, age of individuals)
- component (varying parameters for individual genes)

11.2.1 Deterministic Dynamic

Deterministic mutation schedules are well known to the GA community. They have been used for decades in the Evolutionary Strategies literature [337]. Bäck and Schütz [395] introduced a deterministically decreasing function given in Equation 11.1. This function works on the theory that higher mutation rates in early generations are good for exploration and lower mutation rates in later generations are good for exploiting the local fitness landscape. T is the total number of generations the GA will be run for and n is the length of the chromosome. The mutation rate given by $p_{BS}(t)$ is bounded by $(0, 1/2]$. This function showed good results on hard combinatorial optimization problems. Note that it would be advantageous to floor the function at $\lfloor 1/2n \rfloor$. Mutation rates lower than this are rarely effective.

$$p_{BS}(t) = \left(2 + \frac{n-2}{T-1}t\right)^{-1} \tag{11.1}$$

Droste [340] uses a cyclic mutation operator. The idea of this operator is to try a number of different probabilities in a repeated cycle, giving the GA many chances to use different probabilities during the various natural stages of the GA. The bounds of $p_{Dr}(t)$ are $[1/n, 1/2]$ and the method cycles over $\log n$ different mutation probabilities.

$$\begin{gathered} p_{Dr}(t) = 2p_{Dr}(t-1) \\ \texttt{if}\; p_{Dr}(t) > 1/2,\; \texttt{set}\; p_{Dr}(t) = 1/n \end{gathered} \tag{11.2}$$

11.2.2 Dynamic Adaptive

Thierens [396] introduced two mutation adaptive schemes. The Constant Gain scheme is loosely patterned after the Manhattan learning algorithm. Equation 11.3 contains the mutation update specification. The exploration factor ω and learning factor λ usually have different values ($1 \leq \lambda \leq \omega$). To avoid oscillations in the learning process λ is restricted via $\omega \geq \lambda$. Example values are $\lambda = 1.1$ and $\omega = 1.5$. Thierens also introduced the Declining Adaptive mutation scheme in Equations 11.5 and 11.6. This variant of the first scheme is intended to promote a more aggressive step size while suppressing the wild oscillations that can happen with high learning rate λ.

Thierens' Constant Gain adaptive mutation rule

1. Mutate the current individual (x, p_m) three ways

$$\begin{gathered} \mathcal{M}(x, p_m/\omega) \rightarrow (x_1, p_m/\lambda) \\ \mathcal{M}(x, p_m) \rightarrow (x_2) \\ \mathcal{M}(x, \omega p_m) \rightarrow (x_3, \lambda p_m) \end{gathered} \tag{11.3}$$

2. Select the fittest individual and corresponding new mutation rate

$$\mathcal{MAX}\left\{(x, p_m), (x_1, \lambda p_m), (x_2), (x_3, \lambda p_m)\right\} \tag{11.4}$$

Thierens' Declining Adaptive Mutation Rule

1. Mutate the current individual (x, p_m) three ways

$$\begin{gathered} \mathcal{M}(x, \omega p_m) \rightarrow (x_1, \lambda p_m) \\ \mathcal{M}(x, p_m) \rightarrow (x_2) \\ \mathcal{M}(x, \omega p_m) \rightarrow (x_3, \lambda p_m) \end{gathered} \tag{11.5}$$

2. Decrease the mutation rate of the parent

$$(x, p_m) \rightarrow (x, \gamma p_m) \tag{11.6}$$

3. Select the fittest individual and corresponding new mutation rate

$$\mathcal{MAX}\,\{(x,\gamma p_m),(x_1,\lambda p_m),(x_2),(x_3,\lambda p_m)\} \tag{11.7}$$

The difference between the two schemes is that the second contains no method to increase the mutation rate and that the current mutation rate will decrease unless there is success at the current rate. Factor bounds are as follows: $\lambda > 1$, $\omega > 1$ and $0.9 < \gamma < 1$. Typical settings are $\lambda = 2.0$, $\omega = 2.0$ and $\gamma = 0.95$.

Rechenberg introduced the 1/5 *success rule* for Evolutionary Strategies [337]. The basic idea is to adapt the mutation rate to balance the percentage of fitness-beneficial mutations at 1/5. This rule, shown in Equation 11.8, is applied periodically and not during every generation. A typical value for the learning rate is λ=1.1. Let $\varphi(k)$ be the percentage of successful mutations over x generations and λ be the learning rate.

Rechenberg's 1/5 *Success Rule*

$$\begin{gathered}\texttt{if}\ \varphi(k) < 1/5 \rightarrow (p_m/\lambda)\\ \texttt{if}\ \varphi(k) = 1/5 \rightarrow (p_m)\\ \texttt{if}\ \varphi(k) > 1/5 \rightarrow (\lambda p_m)\end{gathered} \tag{11.8}$$

11.2.3 Dynamic Adaptive with Fuzzy Logic

Shi, et al. [397] introduced a fuzzy logic rule set for adapting the mutation and crossover rate. Below is the rule set associated with the mutation rate. This rule set does require providing fuzzy membership functions for the various metrics and mutation rates. Let $\mathcal{BF}$= best fitness, $\mathcal{UN}$= number of generations since last $\mathcal{BF}$ change, and $\mathcal{MR}$= mutation rate.

Improvement over a GA with fixed parameters was shown using this rule set for evolving classifier systems. For the purposes of this paper, the Sugeno method output

Table 11.1: Shi et al. Fuzzy Adaptive Rule Set for Mutation

IF $\mathcal{BF}$ is low or medium	THEN $\mathcal{MR}$ is low
IF $\mathcal{BF}$ is medium and $\mathcal{UN}$ is low	THEN $\mathcal{MR}$ is low
IF $\mathcal{BF}$ is medium and $\mathcal{UN}$ is medium	THEN $\mathcal{MR}$ is medium
IF $\mathcal{BF}$ is high and $\mathcal{UN}$ is low	THEN $\mathcal{MR}$ is low
IF $\mathcal{BF}$ is high and $\mathcal{UN}$ is medium	THEN $\mathcal{MR}$ is medium
IF $\mathcal{UN}$ is high	THEN $\mathcal{MR}$ is RANDOM(low,medium,high)

functions was used. The Sugeno fuzzy method [398] assigns either constant or linear functions to the output of the fuzzy inferencing method, rather than defining fuzzy membership functions for output. Here this means that three different mutation rates were chosen, one for each fuzzy output. The original Shi et al. rule set used a metric called *variance of fitness* with high, medium and low fuzzy memberships. For this paper that metric was eliminated and three rules were combined to form the last rule in Equation 11.1.

11.3 Fitness Functions

The classes of fitness function used here are unitation, Royal Road and long-path. They are described below along with various specific deceptive functions.

11.3.1 Unitation Functions

Unitation functions are fitness functions where fitness is defined only by the number of ones in a chromosome $x : \{1, 0\}^d$. All fitness values are non-negative:

$$u(x) : \{0, 1\}^d \rightarrow \mathbb{R}+ \tag{11.9}$$

An example function is:

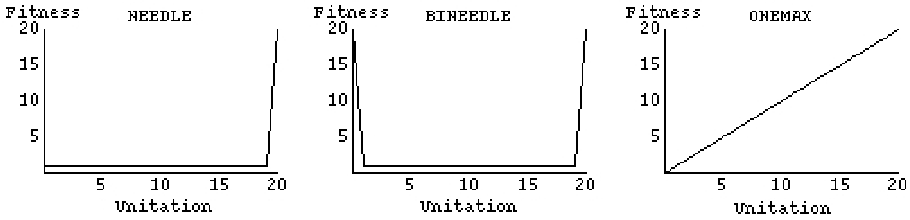


Figure 11.2: NEEDLE, BINEEDLE and ONEMAX fitness functions

$$u(0) = 3 \quad u(1) = 2 \quad u(2) = 1 \quad u(3) = 4 \tag{11.10}$$

This definition allows us to reduce the dimensionality of any analysis from $2n$ to $(n+1)$. This is useful in that theoretical analysis of these functions is computationally easier while still using a function with a complex Hamming landscape.

The three fitness functions given in Equation 11.11 and pictured in Figure 11.2 are called NEEDLE, BINEEDLE and ONEMAX, and have been theoretically studied for fixed parameter simple GAs by Rowe [375], Wright [399] and Richter et al. [400]. The ONEMAX fitness function has been called the *fruit fly* of GA research [396]. Here $n = 10$ and $\alpha = 9$ are used for NEEDLE and BINEEDLE.

$$\begin{aligned}
NEEDLE\ f(x) &= \left\{\begin{array}{cc} 1+\alpha & \text{all ones string} \\ 1 & \text{otherwise} \end{array}\right\} \\
BINEEDLE\ f(x) &= \left\{\begin{array}{cc} 1+\alpha & \text{all ones string} \\ 1 & \text{otherwise} \\ 1+\alpha & \text{all zeros string} \end{array}\right\} \\
ONEMAX\ f(x) &= u(x)
\end{aligned} \tag{11.11}$$

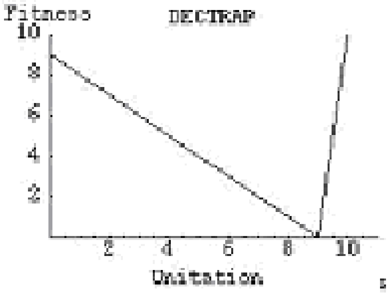


Figure 11.3: Fully deceptive trap function DECTRAP

11.3.2 Deceptive Functions

Trap functions are piecewise linear functions that divide the search space into two Hamming space basins [296]. Each basin has an optimal point, one of which is the global optimum. In Deb and Goldberg [296], a set of conditions for calling a fitness function 'fully deceptive' is given. A fully deceptive function, referred to here as DECTRAP, from [296] is detailed in Equation 11.12.

$$\text{DECTRAP } f(x) = \left\{ \begin{array}{cc} 1 & \text{if } u(x) = n \\ 1 - \frac{1+u(x)}{n} & \text{otherwise} \end{array} \right\} \tag{11.12}$$

Figure 11.3 illustrates DECTRAP. The trap function is set at fitness of 9 for the all zeros string, and fitness 10 for the all ones string. The all zeros basin of attraction takes up the majority of the function space.

Figure 11.4 and Equation 11.13 illustrate a trap function containing two traps, referred to as 2TRAP. This trap was formed in an ad-hoc manner to build a landscape with a large sub-optimal basin at the center of distribution of the unitation classes.

$$\text{2TRAP}$$

$$f(x) = 10 * \left\{ \begin{array}{ll} 1 - \frac{u(x)}{d/5} & \text{if } u(x) < d/5 \\ \frac{u(x)-d/5}{5} & \text{if } d/5 \leq u(x) \leq d/2 \\ \frac{6-u(x)+d/5}{5} & \text{if } d/2 < u(x) \leq 4d/5 \\ \frac{u(x)-4d/5}{2} & \text{if } u(x) > 4d/5 \end{array} \right\} \tag{11.13}$$

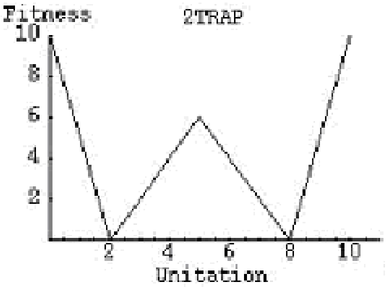


Figure 11.4: Deceptive double trap function 2TRAP

Figure 11.5 and Equation 11.14 show a deceptive double trap function, or DEC2TRAP.

$$\text{DEC2TRAP}$$

$$f(x) = 10 * \left\{ \begin{array}{ll} 1 & \text{if } u(x) = d/2 \\ 1 - \frac{1+u(x)}{d/2} & \text{if } u(x) < d/2 \\ \frac{u(x)-d/2-1}{d/2} & \text{if } u(x) > d/2 \end{array} \right\} \tag{11.14}$$

11.3.3 Royal Road and LongPath Functions

Mitchell et al. [289] crafted a fitness function called the Royal Road which was intended to highlight the building block nature of the GA. The Royal Road is a collection of bit patterns that assigns progressively higher fitness to bit strings that build up long sequences of shorter bit patterns. A modest example for 6-bit strings is

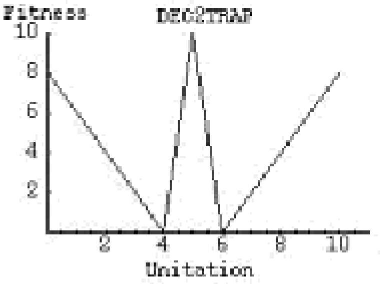


Figure 11.5: Deceptive double trap function DEC2TRAP

Table 11.2: Royal Road Settings

Description	**Symbol**	**Setting**
Number of blocks	2^k	16
Block size (bits)	b	4
Gap size (bits)	g	3
Reward threshold	m^*	2
Per bit reward/penalty	v	0.02
First block bonus	u^*	1
Match increment	u	0.3
Genome size	$2^k(b+g)$	112

given in Equation 11.15. Jones [290] has a complete description of the Royal Roads and its components.

$$\begin{array}{cl} 11****, **11**, ****11 & \text{fitness } f(x) = 2 \\ 1111**, 11**11, **1111 & \text{fitness } f(x) = 4 \\ 111111 & \text{fitness } f(x) = 8 \end{array} \tag{11.15}$$

This paper uses a common reference implementation with settings described in Table 11.2 with a genome size of 112 bits. See the code in Appendix C for more details.

Horn et al. [304] designed a fitness function called *long path.* This function is crafted by choosing a desired path length $k < 2n$. Next a path from a starting bit-string to an ending bit-string is constructed such that it takes k one-bit mutations to follow the path. Each point on the path is Hamming distance one away from each neighboring point. Equations 11.3.3 and 11.16 show a simple example for a 6-bit genome. This paper uses a 9-bit path detailed in Rudolph [235]. The reader is encouraged to examine the code for more details.

$$\begin{aligned}\text{12-path} &= \{000000, 000001, 000011, 000010,\\ &\quad 000110, 000100, 001100, 001000,\\ &\quad 011000, 010000, 110000, 100000\}\end{aligned}$$

$$\text{LONGPATH } f(x) = \left\{\begin{array}{cc} 3 * 2^{(n-1)/2} - 2 - Pos(x) & \text{if } x \text{ is on the path} \\ 3 * 2^{(n-1)/2} - 2 - \|x\|_1 & \text{otherwise} \end{array}\right\} \tag{11.16}$$

Let $Pos(x)$ be the number of the step in the path the string is on. For instance, 000110 is the 4th step in a zero based count for the above.

11.4 Algorithms and Experiments

Thierens [396] applied his two dynamic adaptive schemes, one standard fixed rate and one deterministic scheme to the ONEMAX (or Counting Ones) problem. For these four experiments he used the $(1 + 1)$ EA strategy [340] for the fixed mutation rate and deterministic schemes and the $(1+3)$ EA for his adaptive schemes. Different variants made the comparison more difficult. Algorithm 11.12 gives the $(1 + 3)$ EA.

Algorithm 11.12: The (1+3) EA with optional adaptation

```
1. Choose $p(n) \in (0, 1/2]$.
2. Choose $x \in \{0,1\}^d$ uniformly at random.
3. Create three children $a, b, c$ by copying $x$ then flipping each bit independently
   with $p(n)$ for each copy.
4. Select $x := max\,\{f(x), f(a), f(b), f(c)\}$.
5. Update $p(n)$ according to some scheme. Optional Step
6. Continue at line 3.
```

Algorithm 11.12 will be applied to the five unitation functions given above with each of the seven mutation rate schedules/schemes also described previously. One each of the Royal Road and LongPath functions are also used for the seven schedules. This totals 49 experiments. Each experiment will have 25 trials run. Note that 20-bit versions of the unitation functions are used. The basic shape of each unitation fitness function is the same as presented before.

For Rechenberg's rule, λ=1.1 is used. Constant Gain settings are λ=1.1 and ω=1.5. The Declining Adaptive method used λ=2.0, ω=2.0 and γ=0.95. For the Shi fuzzy rule set, the fuzzy assignments of mutation rate are: high=$4/d$, med=$2/d$, low=$1/d$. These values were derived from those used in source references.

11.5 Experimental Results

Table 11.3 shows the results for the ONEMAX function. For each adaptive method there is an average and standard deviation for the number of fitness function evaluations performed. The *Failed Runs* column gives a count of the number of GA runs that failed to find the optimal solution in 1000 generations.

Table 11.3: ONEMAX results.

Mutation scheme	**Avg number of fitness evals**	**Std dev of fitness evals**	**Failed runs**
static rate	298	159	11
Droste	181	87	-
Bäck-Schütz	815	275	1
Constant Gain	259	223	-
Declining	142	81	-
Rechenberg	99	22	-
Shi	158	83	-

All performed comparably with the exception of Bäck-Schütz, the clear loser. Rechenberg had the best performance with a low average and a small standard deviation. The rest of the methods are generally grouped together within one standard deviation of one another, meaning that they are statistically equivalent methods.

The NEEDLE and DECTRAP results are in Table 11.4. No run of the GA resulted in any scheme finding the optimal points of either fitness landscape. The NEEDLE function is difficulty for many optimization algorithms and generally needs a very high number of mutations to get enough coverage of the $2n$ fitness landscape to find the needle. All of the algorithms were able to ascend to the local maxima of DECTRAP with performance similar to the ONEMAX results. This should also be not surprising since the DECTRAP function's landscape is a near clone of ONEMAX.

The DEC2TRAP results are given in Table 11.5. The various adaptive methods had much difficulty, only the Droste and Bäck-Schütz schemes were successful at frequently finding the global optima, and finding it quickly. The other methods were total failures. While the Declining method found the optimal a four times, I marked their performance as NS (not significant) as these successes were mainly due to lucky population initialization. This result, while initially counter-intuitive, should not be

Table 11.4: NEEDLE and DECTRAP results.

Mutation scheme	**Avg number of fitness evals**	**Std dev of fitness evals**	**Failed runs**
static rate	-	-	25
Droste	-	-	25
Bäck-Schütz	-	-	25
Constant Gain	-	-	25
Declining	-	-	25
Rechenberg	-	-	25
Shi	-	-	25

Table 11.5: DEC2TRAP results.

Mutation scheme	**Avg number of fitness evals**	**Std dev of fitness evals**	**Failed runs**
static rate	-	-	25
Droste	33	37	6
Bäck-Schütz	42	34	-
Constant Gain	-	-	25
Declining	NS	NS	21
Rechenberg	-	-	25
Shi	-	-	25

surprising since only the center unitation class contains the optimal points. While this is clearly the largest unitation class, the fitness function gives local advice to move away from this class. Thus, for a small population only a lucky initialization or mutation event would find the optimal. As can be seen below mutation schemes that either started high or were varied in a cycle were successful.

In comparison to DEC2TRAP, there is a performance reversal with 2TRAP. See Table 11.6 for details. The successful dynamic schemes that worked quickly on DEC2TRAP were total failures on 2TRAP. The reversal continues to an extent with

Table 11.6: 2TRAP results.

Mutation scheme	**Avg number of fitness evals**	**Std dev of fitness evals**	**Failed runs**
static rate	210	65	22
Droste	-	-	25
Bäck-Schütz	-	-	25
Constant Gain	56	26	14
Declining	73	59	10
Rechenberg	41	20	15
Shi	56	9	20

the adaptive methods. These methods found the optimal points at least 10 times each and required a relatively few number of fitness evaluations. The fuzzy method did escape total failure on DEC2TRAP by finding the optimum in 5 of the 25 runs.

Table 11.7 shows the results for the 112-bit Royal Road function. Algorithm 11.12 was run for a maximum of 20,000 generations. All of the adaptive methods except Rechenberg performed well on this function with the Bäck-Schütz scheme the clear winner, as it has the lowest average and the tightest standard deviation. The failure of Rechenberg is so far unexplained, λ=1.1 and λ=2.0 were tried with the same failure result.

Table 11.8 gives the results for the LongPath function. All adaptive schemes faired well on this fairly easy function. The Thierens' Declining rule won out with the lowest average and tighter standard deviation. In both the Royal Road and LongPath functions the static mutation rate was well outperformed by the adaptive schemes.

See Appendix C for details on getting the source code for these experiments.

Table 11.7: Royal Road results.

Mutation scheme	**Avg number of fitness evals**	**Std dev of fitness evals**	**Failed runs**
static rate	-	-	25
Droste	28,312	12,413	6
Bäck-Schütz	18,615	5,251	-
Constant Gain	25,678	12,865	1
Declining	27,548	10,036	-
Rechenberg	-	-	25
Shi	20,340	9,232	-

Table 11.8: LongPath results.

Mutation scheme	**Avg number of fitness evals**	**Std dev of fitness evals**	**Failed runs**
static rate	471	56	-
Droste	157	104	-
Bäck-Schütz	232	165	-
Constant Gain	235	133	-
Declining	129	83	-
Rechenberg	225	134	-
Shi	158	103	-

11.6 Closing Remarks and Future Ideas

There are many methods known in the literature for adapting GA parameters. It is also common to see these methods perform well in those papers. However, when a systematic experiment with a variety of methods is applied to a set of fitness functions displaying basic characteristics, the methods show mixed results.

It is worth noting that combinations of the fitness functions above can be used to construct a wide variety of complex landscapes. The results above can be used as a guide to how a particular adaptive scheme might perform in a certain situation. For instance if a more complex fitness landscape contains hills followed by a flat plateau, an adaptive GA is likely to climb the hill well but get stuck in the plateau.

Of course this problem exists in non parameter-adaptive GAs. However, simply adding adaptive schemes and heuristics does not necessarily cure the GA of the types of problems commonly seen in the analysis of a typical run of the GA. The No Free Lunch Theorem of Wolpert and MacReady [1] is a general proof for GAs that one cannot claim that a GA with fancy operators is provably better than any other GA. Reeves and Rowe [40] detail the NFL Theorem as well as the debate concerning the Building Block Hypothesis. They also note that once an optimization algorithm incorporates domain knowledge of the problem into the operators, the NFL does not apply.

Future work should repeat this analysis on functions with carefully constructed structures. Candidate structures include plateaus, barriers, scaled components, ridges and other structures that are known to pose difficulties. Repeating these experiments for additional difficult structures would help gain intuitions on the strengths of different adaptive schemes.

CHAPTER 12

CONCLUSIONS

> Such a work is never actually finished: one has to declare it finished when one has done all that time and circumstances will allow.
>
> Johann Wolfgang von Goethe, 1786 *Iphigenie auf Tauris*

12.1 Closing Remarks

This work reviewed evolutionary algorithms from base literature and introduced the reader to the history and development of EAs from early ideas in population genetics to modern theoretical models of EAs. This was done to prepare the way for the introduction of several functions that were proven or demonstrated to be difficult or outright deceptive for the crossover operator. While there have been many large works on the benefits of the crossover operator in the EA, this is believed to be the one of first research monographs on the inverse.

The Ignoble Trails family of functions was introduced and proven to be crossover deceptive for both uniform and one-point crossover for several variants of the GA and EA. Both small population and arbitrary population algorithms were analyzed. These are believed to be the first proven examples of functions where a crossover based GA is expected to be exponentially outperformed by an EA without the crossover operator. This was a hole in the current theory of evolutionary algorithms.

Infinite population models of the GA and EA were constructed and analyzed against a suite of fitness functions. Numerical results were shown analyzing fixed-points and metastable state proximity for the mutation-only EA. The results show that choosing a mutation rate can have interesting effects on the dynamics of the algorithm, and that arbitrary choices of mutation rate are unwise.

For the GA, a bistability situation was shown on a crossover deceptive function. This counter-intuitive result for a very rudimentary function demonstrates the disruptive effects and interesting dynamics of the crossover operator under a weak selection operator. In addition, the movement of both vertex fixed-points and the Perron-Frobenius fixed point were examined by varying the mutation and crossover rates. Several examples bifurcation of fixed points were shown.

One insight from these results is that the Perron-Frobenius fixed point for the $\mu + \mu$ MEA without proportional selection is identical to the stationary distribution for the Markov chain of the corresponding $1 + 1$ MEA. The results for the dynamical systems model are among the few on the movement of fixed points of the $\mathcal{G}$-map.

A set of experiments was also done on a set of basic functions using a suite of adaptive parameter schemes that modify the mutation rates. The results were mixed, which should be expected, showing that simply adding adaptive schemes and heuristics does not necessarily cure the GA of the types of problems commonly seen in a typical run.

12.2 Ideas for Future Work

The key lesson learned from this work and the previous OneMix and Real Royal Road papers is that both the GA and the EA can be deceived with relatively simple structures. An open problem would be to follow up on Doerr et al. [361] and produce a reasonable graph problem that where the EA is provably outperformed by a GA using time complexity analysis.

It is also hypothesized that a function can be constructed where crossover deceptive structures are hierarchically stacked to form a HIFF like function where the GA has a more diabolical challenge. The essence of this idea would be to provide a hierarchy of long paths of mutations from low fitness regions ascending to optimal points. If multiple fork-in-the-road structures can be designed that deceive crossover into jumping to dead-ends, then the overall probabilities of GA not being deceived shrink in a multiplicative fashion.

Perhaps the newer HIFF functions from Watson and Jansen [241] may provide an alternate starting point as well. Alden and I think that concatenated trap functions may also be a fruitful starting point. A key goal would be to develop functions exploring both the macro-mutation aspects of crossover as well as its building-block attributes.

Following up on Chapter 10 Alden Wright and I have developed a set of ideas around redoing some of this analysis in the context of gene-pool crossover. Gene pool crossover allows the reduction of the $\mathcal{G}$-map to a set of simpler equations that are directly solvable.

In particular it is believed that this type of approach would be useful for modeling the so-called Estimation of Distribution Algorithm, see Pelikan et al [4], which replaces the use of an actual population with a population distribution that is sampled from. This fits neatly into EA theory which operates on population distributions. It should be possible to construct a variety of functions that show both bistability and crossover deception.

For deeper understanding of the role of crossover, the work of Stephens and coworkers on formalizing a new Schema and Building Block models is an exciting direction [188, 245, 189, 246].

REFERENCES CITED

[1] D. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, 1997.

[2] S. Wright, "Comments," in *Proceedings of the Sixth Congress on Genetics*, P. S. Moorhead and M. M. Kaplan, Eds. Ithaca, New York, 1932: Wistar Institute Press, 1967, pp. 117–120.

[3] K. A. De Jong, *Evolutionary Computation: A Unified Approach.* Cambridge, MA: MIT Press, 2002.

[4] M. Pelikan, K. Sastry, and E. Cantú-Paz, Eds., *Scalable Optimization via Probabilistic Modeling*, ser. Studies in Computational Intelligence. Springer, 2006, vol. 33.

[5] *1st NASA / DoD Workshop on Evolvable Hardware (EH '99), July 19-21, 1999, Pasadena, CA.* IEEE Computer Society, 1999.

[6] C. G. Langton, *Artificial Life: An Overview.* Cambridge, MA: MIT Press, 1995.

[7] J. R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection.* Cambridge, MA, USA: MIT Press, 1992.

[8] J. R. Koza, M. A. Keane, M. J. Streeter, W. Mydlowec, J. Yu, and G. Lanza, *Gentic Programming IV: routine human-competitive machine intellligence.* Kluwer Academic Publishers, 2003.

[9] R. Poli, W. B. Langdon, and N. F. Mcphee, *A field guide to genetic programming*, March 2008.

[10] L. J. Fogel, A. J. Owens, and M. J. Walsh, *Artificial Intelligence through Simulated Evolution.* New York: John Wiley & Sons, 1966.

[11] A. Eiben and J. Smith, *Introduction to Evolutionary Computing.* Springer-Verlag, 2003.

[12] M. Dorigo, "Optimization, learning and natural algorithms," Ph.D. dissertation, Dipartimento di Elettronica, Politecnico di Milano, Milan, Italy, 1992.

[13] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings IEEE International Conf. on Neural Networks.* Perth, Australia: IEEE, 1995, p. 1942–1948.

[14] V. Cutello and G. Nicosia, "An immunological approach to combinatorial optimization problems," in *IBERAMIA*, ser. Lecture Notes in Computer Science, F. J. Garijo, J. C. R. Santos, and M. Toro, Eds., vol. 2527. Springer, 2002, pp. 361–370.

[15] H. Hoos and T. Stützle, *Stochastic Local Search: Foundations & Applications.* San Francisco, CA: Morgan Kaufmann Publishers Inc., 2004.

[16] I. Wegener, *The Complexity of Boolean Functions.* Wiley-Teubner, 1987.

[17] J. Holland, *Adapdation in Natural and Artificial Systems.* Ann Arbor, Michigan: University of Michigan Press, 1975.

[18] A. Turing, "Intelligent Machinery," in *Machine Intelligence*, B. Meltzer and D. Michie, Eds. Edinburgh: Edinburgh University Press, 1948, vol. 5.

[19] ——, "Computing Machinery and Intelligence," *Mind*, vol. 59, no. 236, pp. 433–460, 1950.

[20] A. S. Fraser, "Simulation of genetic systems by automatic digital computers. i. introduction," *Aust. J. Biol. Sci*, vol. 10, pp. 484–491, 1957.

[21] ——, "Simulation of genetic systems," *Journal Theoretical Biology*, vol. 2, pp. 329–346, 1962.

[22] A. Fraser and D. Burnell, *Computer M:odels in Genetics.* McGraw-Hill New York,, 1970.

[23] D. Fogel, "Foundations of evolutionary computation," in *Modeling and Simulation for Military Applications*, W. Schum, Ed. Society of Photo Optical, 2006, pp. 1–13.

[24] H. Bremermann, "The evolution of intelligence, the nervous system as a model of its environment," *Tech. Report, No. 1, Contract No. 477(17), Dept. Mathematics, Univ. Washington, Seattle*, 1958.

[25] H. J. Bremermann, "Optimization through evolution and recombination," in *Self Organizing Systems*, M. C. Y. et al, Ed. Washington, DC: Spartan Books, 1962, pp. 93–106.

[26] H. Mühlenbein, "How genetic algorithms really work: Mutation and hillclimbing," in *PPSN*, R. Männer and B. Manderick, Eds. Elsevier, 1992, pp. 15–26.

[27] R. W. Anderson and M. Conrad, "Hans J. Bremermann: a pioneer in mathematical biology." *Biosystems*, vol. 34, no. 1, pp. 1–10, 1995.

[28] D. B. Fogel and R. W. Anderson, "Revisiting Bremermann's genetic algorithm: I. simultaneous mutation of all parameters," in *Proceedings of the Congress on Evolutionary Computation (CEC) 2000.* IEEE Press, 2000, pp. 1204–1209.

[29] G. J. Friedman, "Digital simulation of an evolutionary process," *General Systems Yearbook*, vol. 4, pp. 171–184, 1959.

[30] G. Box, “Evolutionary operation: a method of increasing industrial productivity,” *Applied Statistics*, vol. 6, pp. 81–101, 1957.

[31] L. J. Fogel, *Biotechnology: Concepts and Applications.* Englewood Cliffs, NJ: Prentice–Hall, 1963.

[32] I. Rechenberg, *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution.* Stuttgart, Germany: Frommann-Holzboog, 1973.

[33] H.-P. Schwefel, “Adaptive Mechanismen in der biologischen Evolution und ihr Einfluss auf die Evolutionsgeschwindigkeit,” Technical University of Berlin, Technical Report of the Working Group of Bionics and Evolution Techniques at the Institute for Measurement and Control Technology Re 215/3, July 1974.

[34] D. B. Fogel, *Evolutionary Computation: The Fossil Record.* Wiley-IEEE Press, 1998.

[35] J. Holland, *Adaptation in Natural and Artificial Systems.* University of Michigan Press, 1975.

[36] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller, “Equation of state calculations for fast computing machines,” *Journal of Chemical Physics*, vol. 21, pp. 1087–1092, 1953.

[37] G. A. Croes, “A method for solving traveling-salesman problems,” *Operations Research*, vol. 6, no. 6, pp. 791–812, 1958.

[38] C. Reeves, *Modern Heuristic Techniques for Combinatorial Problems.* Oxford: Blackwell Scientific Press, 1993.

[39] K. A. DeJong, “An analysis of the behavior of a class of genetic adaptive systems,” Ph.D. dissertation, University of Michigan, Ann Arbor, MI, 1975.

[40] C. R. Reeves and J. E. Rowe, *Genetic Algorithms: Principles and Perspectives: A Guide to GA Theory.* Norwell, MA: Kluwer Academic Publishers, 2002.

[41] J. J. Grefenstette, Ed., *Proceedings of the 1st International Conference on Genetic Algorithms, Pittsburgh, PA, USA, July 1985.* Lawrence Erlbaum Associates, 1985.

[42] H.-P. Schwefel and R. Männer, Eds., *Parallel Problem Solving from Nature, 1st Workshop, PPSN I, Dortmund, FRG, October 1-3, 1990, Proceedings*, ser. Lecture Notes in Computer Science, vol. 496. Springer, 1991.

[43] W. Banzhaf, J. M. Daida, A. E. Eiben, M. H. Garzon, V. Honavar, M. J. Jakiela, and R. E. Smith, Eds., *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 1999), 13-17 July 1999, Orlando, Florida, USA.* Morgan Kaufmann, 1999.

[44] *Proceedings of the 1999 Congress on Evolutionary Computation, CEC 99, 6 - 9 July 1999, Washington, DC.* IEEE, 1999.

[45] D. E. Goldberg, "Optimal initial population size for binary-coded genetic algorithms." *Technical Report, University of Alabamam Tuscaloosa*, vol. TCGA TR No. 85001, 1991.

[46] ——, "Sizing populations for serial and parallel genetic algorithms," in *ICGA*, J. D. Schaffer, Ed. Morgan Kaufmann, 1989, pp. 70–79.

[47] D. E. Goldberg, K. Deb, and J. H. Clark, "Genetic algorithms, noise, and the sizing of populations," *Complex Systems*, vol. 6, pp. 333–362, 1991.

[48] G. Harik, E. Cantú-Paz, D. E. Goldberg, and B. L. Miller, "The gambler's ruin problem, genetic algorithms, and the sizing of populations," pp. 7–12, 1997.

[49] J. Grefenstette, "Optimization of control parameters for genetic algorithms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 16, no. 1, pp. 122–128, 1986.

[50] J. T. Alander, "On optimal population size of genetic algorithms," in *CompEuro '92: Proceedings of the 1992 IEEE Computer Systems and Software Engineering Conference.* New York: IEEE, 1992, pp. 65–70.

[51] D. Cvetkovic and H. Mühlenbein, "The optimal population size for uniform crossover and truncation selection," 1994.

[52] Y. Gao, "Population size and sampling complexity in genetic algorithms," in *GECCO 2003: Proceedings of the Bird of a Feather Workshops, Genetic and Evolutionary Computation Conference*, A. M. Barry, Ed. Chigaco: AAAI, 11 July 2003, pp. 178–181.

[53] M. Mitchell, *An Introduction to Genetic Algorithms.* Cambridge, MA: MIT Press, 1996.

[54] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning.* New York: Addison-Wesley, 1989.

[55] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[56] H. Mühlenbein and H.-M. Voigt, "Gene pool recombination in genetic algorithms," in *Metaheuristics: Theory and Applications.* Kluwer Academic Publishers, 1995, pp. 53–62.

[57] J. Jägersküpper and T. Storch, "When the plus strategy outperforms the comma strategy and when not," in *FOCI.* IEEE, 2007, pp. 25–32.

[58] R. Pearce and P. Cowley, "Use of fuzzy logic to describe constraints derived from engineering judgment in genetic algorithms," *IEEE Transactions on Industrial Electronics*, vol. 43, no. 5, pp. 535–540, 1996.

[59] H. A. N. L. Elmihoub, T. and A. Battersby, "Hybrid genetic algorithms - a review," *Engineering Letters*, vol. 13, pp. 124–137, 2006, iSSN: 1816-093X.

[60] R. Kicinger, T. Arciszewski, and K. A. De Jong, "Evolutionary computation and structural design: A survey of the state of the art," *Computers and Structures*, vol. 83, no. 23-24, pp. 1943–1978, 2005.

[61] C. R. Stephens and J. Cervantes, "Just what are building blocks?" in *FOGA*, 2007, pp. 15–34.

[62] A. E. Eiben, R. Hinterding, and Z. Michalewicz, "Parameter control in evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 3, pp. 124–141, 1999.

[63] J. Smith and T. Fogarty, "Operator and parameter adaptation in genetic algorithms," *Soft Computing*, vol. 1, no. 2, pp. 81–87, 1997.

[64] J. N. Richter, "Fuzzy evolutionary cellular automata," Master's thesis, Utah State University, 2003.

[65] R. Craighurst and W. N. Martin, "Enhancing ga performance through crossover prohibitions based on ancestry," in *ICGA*, L. J. Eshelman, Ed. Morgan Kaufmann, 1995, pp. 130–135.

[66] L. D. Whitley, "Cellular genetic algorithms," in *Proceedings of the 5th International Conference on Genetic Algorithms.* San Francisco, CA: Morgan Kaufmann Publishers Inc., 1993, p. 658.

[67] S. W. Mahfoud, "Crowding and preselection revisited," in *PPSN*, R. Männer and B. Manderick, Eds. Elsevier, 1992, pp. 27–36.

[68] S. Luke, *Essentials of Metaheuristics*, 2009, available at http://cs.gmu.edu/∼sean/book/metaheuristics/.

[69] t. f. e. Wikipedia, "Metaheuristic," http://en.wikipedia.org/wiki/Metaheuristic, downloaded April, 2010.

[70] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*, 2nd ed. Cambridge, MA: MIT Press, 1992.

[71] G. J. E. Rawlins, Ed., *Proceedings of the First Workshop on Foundations of Genetic Algorithms. Bloomington Campus, Indiana, USA, July 15-18 1990.* Morgan Kaufmann, 1991.

[72] H.-G. Beyer, K. De Jong, D. Fogel, and I. Wegener, Eds., *Theory of Evolutionary Algorithms, , Feb 13 - 18, 2000.* Wadern, Germany: Schloss Dagstuhl, 2000.

[73] S. J. Gould and R. C. Lewontin, "The spandrels of San Marco and the panglossian paradigm: A critique of the adaptationist programme," *Proceedings of the Royal Society of London. Series B, Biological Sciences*, vol. 205, no. 1161, pp. 581–598, 1979.

[74] C. Darwin and A. Wallace, "On the tendency of species to form varieties; and on the perpetuation of varieties and species by natural means of selection," *Journal of the Proceedings of The Linnean Society*, vol. 3, June 1858.

[75] C. Darwin, *On the Origin of Species by Means of Natural Selection.* London: Murray, 1859.

[76] E. Mayr, *The Growth of Biological Thought : Diversity, Evolution, and Inheritance.* Belknap Press, Cambridge, Mass, 1982.

[77] P. J. Bowler, *Evolution, the history of an idea*, 3rd ed. Berkeley: University of California Press, 2003.

[78] J. Brown, *Darwin's Origin of Species: A Biography.* New York: Atlantic Monthly Press, 2007.

[79] J. Felsenstein, "The evolutionary advantage of recombination," *Genetics*, vol. 78, no. 2, pp. 737–756, 1974.

[80] t. f. e. Wikipedia, "Modern evolutionary synthesis," http://en.wikipedia.org/wiki/Modern_evolutionary_synthesis, downloaded April, 2009.

[81] ——, "History of evolutionary thought," http://en.wikipedia.org/wiki/History_of_evolutionary_thought, downloaded April, 2009.

[82] ——, "Mutationism," http://en.wikipedia.org/wiki/Mutationism, downloaded April, 2009.

[83] ——, "On the origin of species," http://en.wikipedia.org/wiki/On_the_Origin_of_Species, downloaded April, 2009.

[84] ——, "Neutral theory of molecular evolution," http://en.wikipedia.org/wiki/Neutral_theory_of_molecular_evolution, downloaded April, 2009.

[85] J. B. Lamarck, *Philosophie Zoologique.* Paris: Chez Dentu, 1809.

[86] S. J. Gould, *The Structure of Evolutionary Theory.* Harvard: Belknap Press, 2002.

[87] R. Chambers, *Vestiges of the Natural History of Creation and Other Evolutionary Writings.* University of Chicago Press, 1844,1994. [Online]. Available: http://books.google.com/books?id=fkrGtdA9tDQC\&printsec=front_cover

[88] A. O. Lovejoy, *The Great Chain of Being: a Study of the History of an Idea.* Harvard University Press, Cambridge, Mass. :, 1936.

[89] W. B. Provine, *The Origins of Theoretical Population Genetics.* Chicago: University of Chicago Press, 1971.

[90] W. Bateson, *Materials for the Study of Variation.* London: Macmillan, 1894.

[91] E. J. Steele, R. A. Lindley, and R. V. Blanden, *Lamarck's Signature : How Retrogenes are Changing Darwin's Natural Selection Paradigm.* St. Leonards, N.S.W. Australia: Allen & Unwin, 1998.

[92] A. Weismann, J. A. Thomson, and M. R. Thomson, *The evolution theory.* E. Arnold, London,, 1904.

[93] M. G. Bulmer, *Francis Galton: pioneer of heredity and biometry.* John Hopkins University Press, 2003.

[94] G. Mendel, "Experiments in plant hybridization," 1865, ch. 1, pp. 8–17, this is a translation by the Royal Horticultural Society of London of the original paper, entitled "Vesuche über Pflanzen-Hybriden", published in the *Verb. naturf. Ver. in Brunn, Abandlungen,* iv. 1865, which appeared in 1866.

[95] O. Harman, "Darlington and the 'invention' of the chromosome," *Endeavour*, vol. 27, no. 22, pp. 69–74, 2003.

[96] N. Roll-Hansen, "The crucial experiment of wilhelm johannsen," *Biology and Philosophy*, vol. 4, no. 3, 1989. [Online]. Available: http://www.springerlink.com/content/rl441372173j8490/fulltext.pdf

[97] A. Stoltzfus, "Mutationism and the dual causation of evolutionary change," *Evolution & Development*, vol. 8, no. 3, pp. 304–317, June 2006.

[98] V. L. Kellogg, *Darwinism today.* Bell, London, 1907.

[99] T. H. Morgan, *Evolution and Adaptation.* New York: Macmillan, 1903.

[100] J. Tabery, "The 'evolutionary synthesis' of george udny yule," *Journal of the History of Biology*, vol. 37, no. 1, pp. 73–101, 2004.

[101] E. ÁKERBERG, "Nilsson-ehle and the development of plant breeding at svalöf during the period 1900-1915," *Hereditas*, vol. 105, no. 1, pp. 73–101, 1986.

[102] T. H. Morgan, *The mechanism of Mendelian heredity,.* Holt, New York, 1915, http://www.biodiversitylibrary.org/bibliography/6001. [Online]. Available: http://www.biodiversitylibrary.org/item/27295

[103] E. M. East, "A mendelian interpretation of variation that is apparently continuous," *The American Naturalist*, vol. 44, pp. 65–82, 1910.

[104] J. F. Crow, "Herbert spencer jennings (1868-1947)," *Genetics*, vol. 33, no. 1, pp. 1–4, 1948.

[105] T. H. Morgan, *A Critique of the Theory of Evolution.* Princeton NJ: Princeton University Press, 1916.

[106] M. Lederman, "Genes on chromosomes: The conversion of thomas hunt morgan," *Journal of the History of Biology*, vol. 22, no. 1, pp. 163–176, 1989.

[107] G. H. Hardy, "Mendelian proportions in a mixed population," *Science*, vol. 28, no. 706, pp. 49–50, 1908.

[108] W. Weinberg, "Uber den Nachweis der Vererbung beim Menschen," *Jahreshefte Verein f. vaterl. Naturk, in Wurttemberg*, vol. 64, pp. 368–82, 1908.

[109] E. Mayr, *Evolution and the Diversity of Life : Selected Essays.* Belknap Press of Harvard University press, Cambridge, Mass., 1976.

[110] R. Fisher, "The correlation between relatives on the supposition of mendelian inheritance," *Transactions of the Royal Society of Edinburgh*, vol. 52, pp. 399–433, 1918.

[111] J. F. Crow, "Fisher's contributions to genetics and evolution." *Theoretical Population Biology*, vol. 38, no. 3, pp. 263–75, 1990.

[112] R. A. Fisher, *The Genetical Theory of Natural Selection.* Oxford: Clarendon Press, 1930.

[113] J. F. Crow, "Sewall Wright (1889-1988)," *Genetics*, vol. 119, no. 1, pp. 1–4, 1988.

[114] S. Wright, "Fisher's theory of dominance," *The American Naturalist*, vol. 63, pp. 274–279, 1929.

[115] J. B. S. Haldane, *The causes of evolution.* Longmans, Green and co., London, 1932.

[116] J. F. Crow, "Centennial: J. b. s. haldane 1892-1964," *Genetics*, vol. 130, no. 1, pp. 1–6, 1992.

[117] N. V. Timofeeff-Ressovsky, "Mutations and geographical variation," in *The New systematics*, J. Huxley, Ed. Oxford University Press, 1940, pp. 73–136.

[118] T. Dobzhansky, *Genetics and the origin of species*, 3rd ed. New York: Columbia University Press, 1937,1964.

[119] J. Huxley, *Evolution: The Modern Synthesis.* New York: Harper & Brothers, 1942.

[120] G. G. Simpson, *Tempo and Mode in Evolution.* New York: Columbia University Press, 1944.

[121] E. Mayr and W. B. Provine, *The Evolutionary synthesis : perspectives on the unification of biology / edited by Ernst Mayr and William B. Provine.* Harvard University Press, Cambridge, Mass. :, 1980.

[122] M. B. Creighton H., "A correlation of cytological and genetical crossing-over in zea mays," *Proceedings of the National Academy of Sciences USA*, vol. 17, no. 8, pp. 492–497, 1931.

[123] C. D. Darlington, *Recent advances in cytology.* Churchill, London, 1932.

[124] ——, *The Evolution of Genetic Systems*, 1939.

[125] O. T. Avery, C. M. MacLeod, and M. McCarty, "Studies on the chemical nature of the substance inducing transformation of pneumococcal types. Induction of transformation by a desoxyribonucleic acid fraction isolated from pneumococcus type III," *Journal of Experimental Medicine*, vol. 79, pp. 137–159, 1944.

[126] J. D. Watson and F. H. Crick, "Molecular structure of nucleic acids; a structure for deoxyribose nucleic acid." *Nature*, vol. 171, no. 4356, pp. 737–738, 1953.

[127] M. Kimura, "Evolutionary rate at the molecular level." *Nature*, vol. 217, pp. 624–626, 1968.

[128] ——, "Genetic variability maintained in a finite population due to mutational production of neutral and nearly neutral isoalleles." *Genetics Research*, vol. 11, no. 3, pp. 247–269, 1968.

[129] W. Provine, "Rise of the null selection hypothesis," in *Genes and Ecology in History*, A. Cain and W. Provine, Eds. Oxford: Blackwell, 1991, pp. 15–23.

[130] J. L. King and T. H. Jukes, "Non-darwinian evolution," *Science*, vol. 164, no. 3881, pp. 788–798, 1969.

[131] M. Kimura, *The neutral theory of molecular evolution.* Cambridge, London: Cambridge University Press, 1983.

[132] T. Ohta, "Slightly deleterious mutant substitutions in evolution," *Nature*, vol. 246, pp. 96–98, 1973.

[133] ——, "The nearly neutral theory of molecular evolution," *Annual Review of Ecology and Systematics*, vol. 23, no. 1, pp. 263–286, 1992.

[134] S. Karlin, "Motoo Kimura and evolutionary theory," *Theoretical Population Biology*, vol. 49, pp. 119–121, 1996.

[135] B. Clarke, "Darwinian evolution of proteins," *Science (New York, N.Y.)*, vol. 168, no. 934, 1970.

[136] R. C. Richmond, "Non-Darwinian evolution: A critique," *Nature*, vol. 225, no. 5237, p. 1025–1028, 1970.

[137] M. Kreitman, "The neutral theory is dead. long live the neutral theory," *Bioessays*, vol. 18, no. 8, pp. 678–683, 1996.

[138] T. Ohta, "The current significance and standing of neutral and nearly neutral theories," *Bioessays*, vol. 18(8), pp. 673–677, 1996.

[139] G. Dover, "There's more to life than selection and neutrality," *Bioessays*, vol. 19, no. 1, pp. 91–92, 1997.

[140] L. Duret, "Neutral theory: The null hypothesis of molecular evolution," *Nature Education*, vol. 1, no. 1, 2008.

[141] M. Nei, "Selectionism and neutralism in molecular evolution," *Molecular Biology and Evolution*, vol. 22, no. 12, pp. 2318–2342, December 2005.

[142] ——, "The new mutation theory of phenotypic evolution," *Proceedings of the National Academy of Sciences*, vol. 104, no. 30, pp. 12 235–12 242, July 2007.

[143] R. Michod and B. Levin, *The Evolution of Sex: An Examination of Current Ideas.* Sunderland, MA: Sinauer Associates, 1988.

[144] A. Burt, "Resolving the paradox of sex and recombination," *Evolution*, vol. 54, no. 2, pp. 337–351, 2000.

[145] T. H. Morgan, *Heredity and Sex.* New York: Columbia University Press, 1913.

[146] H. J. Muller, “Some genetic aspects of sex.” *The American Naturalist*, vol. 66, no. 703, pp. 118–138, 1932.

[147] J. F. Crow, “H. J. Muller and the ”competition hoax”,” *Genetics*, vol. 173, no. 2, pp. 511–514, 2006.

[148] H. J. Muller, “Evolution by mutation.” *Bulletin of the American Mathematical Society*, vol. 64, no. 4, pp. 137–160, 1958.

[149] ——, “The relation of recombination to mutational advance.” *Mutation Research*, vol. 1, no. 1, pp. 2–9, 1964.

[150] J. F. Crow, “Hermann Joseph Muller, evolutionist,” *Nature Reviews Genetics*, vol. 6, no. 12, pp. 941–945, 2005.

[151] S. Wright, “The roles of mutation, inbreeding, crossbreeding and selection in evolution,” in *Proceedings of the Sixth Annual Congress of Genetics 1.*, D. F. Jones, Ed., 1932, pp. 356–366.

[152] G. G. Simpson, *Major Features of Evolution.* New York: Columbia University Press, 1953.

[153] E. Mayr, *What Evolution Is.* New York: Basic Books, 2001.

[154] K. M. Crow, J. F., “Evolution in sexual and asexual populations,” *The American Naturalist*, vol. 99, pp. 439–450, 1965.

[155] J. Maynard Smith, “Evolution in sexual and asexual populations,” *The American Naturalist*, vol. 102, pp. 469–473, 1968.

[156] K. M. Crow, J. F., “Evolution in sexual and asexual populations: A reply,” *The American Naturalist*, vol. 103, pp. 89–91, 1969.

[157] J. Maynard Smith, “What use is sex?” *Journal Theoretical Biology*, vol. 30, pp. 319–335, 1971.

[158] W. G. Hill and A. Robertson, “The effect of linkage on limits to artificial selection.” *Genetical Research*, vol. 8, no. 3, pp. 269–294, 1966.

[159] J. Maynard Smith and J. Haigh, “The hitch-hiking effect of a favourable gene,” *Genetical Research*, vol. 23, no. 1, pp. 23–35, 1974.

[160] N. H. Barton, “Genetic hitchhiking,” *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 355, no. 1403, pp. 1553–1562, Nov 2000.

[161] S. Otto and T. Lenormand, “Resolving the paradox of sex and recombination,” *Nature Reviews Genetics*, vol. 3, no. 4, pp. 252–262, 2002.

[162] N. Eldredge and S. J. Gould, “Punctuated equilibria: An alternative to phyletic gradualism,” in *Models in Paleobiology*, T. J. M. Schopf, Ed. San Francisco: Freeman, Cooper and Co., 1972, pp. 82–115.

[163] K. Sterelny, *Dawkins vs. Gould: Survival of the Fittest.* London: Icon Books, 2001.

[164] D. C. Dennett, *Darwin's Dangerous Idea: Evolution and the Meanings of Life.* New York: Simon & Schuster, 1995.

[165] S. J. Gould, “Darwinian fundamentalism,” *The New York Review of Books*, vol. 44, no. 12, June 1997.

[166] R. Goldscmidt, *The Material Basis of Evolution.* New Haven, CT: Yale University Press, 1940.

[167] S. Gould, “The return of hopeful monsters,” *Natural History*, vol. 86, pp. 22–30, 1977.

[168] S. J. Gould, “Tempo and mode in the macroevolutionary reconstruction on darwinism,” *PNAS*, vol. 91, no. 15, pp. 6764–6771, 1994.

[169] D. L. Hartl and A. G. Clark, *Principles of population genetics*, 4th ed. Sinauer Associates, Sunderland, Mass. :, 2006.

[170] W. J. Ewens, *Mathematical Population Genetics*, 2nd ed. New York: Springer, 2004.

[171] D. M. Walsh, “The trials of life: Natural selection and random drift,” *Philosophy of Science*, vol. 69, no. 3, pp. 452–473, 2002.

[172] M. Matthen and A. Ariew, “Two ways of thinking about fitness and natural selection,” *Journal of Philosophy*, vol. 119, no. 2, pp. 55–83, 2002.

[173] C. L. Stephens, “Selection, drift, and the “forces”of evolution,” *Philosophy of Science*, vol. 71, no. 4, 2004.

[174] R. G. B. Reid, *Evolutionary Theory : the Unfinished Synthesis.* London: Croom Helm, 1985.

[175] S. Løvtrup, *Darwinism: The Refutation of a Myth.* London: Croom Helm, 1987.

[176] J. A. Shapiro, “A third way,” *Boston Review*, vol. 22, no. 1, pp. 32–33, 1997.

[177] K. Mochizuki, N. A. Fine, T. Fujisawa, and M. A. Gorovsky, “Analysis of a piwi-related gene implicates small RNAs in genome rearrangement in tetrahymena,” *Cell*, vol. 110, no. 6, pp. 689–699, 2002.

[178] E. Jablonka and M. J. Lamb, *Evolution in Four Dimensions: Genetic, Epigenetic, Behavioral, and Symbolic Variation in the History of Life.* Cambridge, MA: MIT Press, 2006.

[179] R. Bürger, *The Mathematical Theory of Selection, Recombination and Mutation.* New York: J. Wiley and Sons, 2000.

[180] J. H. Gillespie, *Population Genetics: A Concise Guide.* Baltimore, MD: Johns Hopkins Press, 1998.

[181] M. Brin and G. Stuck, *Introduction to Dynamical Systems.* Cambridge: Cambridge University Press, 2002.

[182] S. Strogatz, *Nonlinear Dynamics and Chaos.* Reading, Massachusetts: Addison–Wesley Publishing Company, 1994.

[183] L. Altenberg, "The schema theorem and Price's theorem," in *FOGA*, L. D. Whitley and M. D. Vose, Eds. Morgan Kaufmann, 1994, pp. 23–49.

[184] R. C. Lewontin, "The interaction of selection and linkage. I. general considerations; heterotic models," *Genetics*, vol. 49, no. 1, pp. 49–67, 1964.

[185] M. Slatkin, "Selection and polygenic characters," *PNAS*, vol. 66, no. 1, pp. 87–93, May 1970.

[186] M. D. Vose, "Generalizing the notion of schema in genetic algorithms," *Artif. Intell.*, vol. 50, no. 3, pp. 385–396, 1991.

[187] M. D. Vose and G. E. Liepins, "Punctuated equilibria in genetic search," *Complex Systems*, vol. 5, no. 1, pp. 31–44, 1991.

[188] C. R. Stephens and H. Waelbroeck, "Effective degrees of freedom in genetic algorithms and the block hypothesis," in *Proceedings of the Seventh International Conference on Genetic Algorithms*, T. Bäck, Ed. San Mateo: Morgan Kaufman, 1997, pp. 34–40.

[189] ——, "Schemata evolution and building blocks," *Evolutionary Computation*, vol. 7, no. 2, pp. 109–124, 1999.

[190] G. R. Price, "Fisher's 'fundamental theorem' made clear," *Annals of Human Genetics*, vol. 36, no. 2, pp. 129–140, 1972.

[191] J. F. Crow, "Perspective: Here's to fisher, additive genetic variance, and the fundamental theorem of natural selection." *Evolution*, vol. 56, no. 7, pp. 1313–1316, 2002.

[192] W. J. Ewens, “A generalized fundamental theorem of natural selection,” *Genetics*, vol. 63, pp. 531–537, 1969.

[193] ——, “Mean fitness increases when fitnesses are additive,” *Nature*, vol. 221, p. 1076, 1969.

[194] ——, “An interpretation and proof of the fundamental theorem of natural selection,” *Theoretical Population Biology*, vol. 36, pp. 167–180, 1989.

[195] M. Kimura, “On the change of population fitness by natural selection,” *Heredity*, vol. 12, pp. 145–167, 1959.

[196] H. Geiringer, “On the probability of linkage in Mendelian heredity,” *Annals of Mathematical Statistics*, vol. 15, pp. 25–57, 1944.

[197] L. B. Booker, “Recombination distributions for genetic algorithms,” in *FOGA*, L. D. Whitley, Ed. Morgan Kaufmann, 1992, pp. 29–44.

[198] E. Akin, “Cycling in simple genetic systems,” *Journal of Mathematical Biology*, vol. 13, pp. 305–324, 1982.

[199] A. Hastings, “Stable cycling in discrete-time genetic models,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 78, no. 11, pp. 7224–7225, 1981.

[200] A. H. Wright and G. L. Bidwell, “A search for counterexamples to two conjectures on the simple genetic algorithm,” in *Foundations of Genetic Algorithms (FOGA-4)*, R. K. Belew and M. D. Vose, Eds. Morgan Kaufmann, 1997, pp. 73–84.

[201] A. H. Wright and A. Agapie, “Cycling and chaotic behavior in genetic algorithms,” in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2001)*, A. W. W. B. L. H.-M. V. M. G. S. S. M. D. S. P. M. H. G. E. B. Lee Spector, Erik D. Goodman, Ed. Morgan Kaufmann, 2001, pp. 718–724.

[202] U. Chakraborty and H. Muhlenbein, “Linkage equilibrium and genetic algorithms,” in *Proceedings IEEE International Conf. on Evolutionary Computation (ICEC’97).* IEEE, 1997, pp. 25–29.

[203] S. C. R. W. A. Poli, R. and J. Rowe, “A schema-theory-based extension of Geiringer’s theorem for linear gp and variable-length gas under homologous crossover.” in *Foundations of genetic algorithms (FOGA-7)*, J. E. Rowe, K. DeJong, and R. Poli, Eds. San Mateo: Morgan Kaufmann, 2003.

[204] B. Mitavskiy and J. E. Rowe, “A schema-based version of Geiringer’s theorem for nonlinear genetic programming with homologous crossover,” in *FOGA*, ser. Lecture Notes in Computer Science, A. H. Wright, M. D. Vose, K. A. D. Jong, and L. M. Schmitt, Eds., vol. 3469. Springer, 2005, pp. 156–175.

[205] ——, “An extension of Geiringer’s theorem for a wide class of evolutionary search algorithms,” *Evolutionary Computation*, vol. 14, no. 1, pp. 87–118, 2006.

[206] A. Robertson, “A mathematical model of the culling process in dairy cattle,” *Animal Production*, vol. 8, pp. 95–108, 1966.

[207] ——, “The spectrum of genetic variation,” in *Population Biology and Evolution*, R. C. Lewontin, Ed., 1968.

[208] B. Walsh and M. Lynch, *Evolution and Selection of Quantitative Traits.* http://nitro.biosci.arizona.edu/zbook/NewVolume_2/newvol2.html, 2009.

[209] G. R. Price, “Selection and covariance.” *Nature*, vol. 277, pp. 520–521, 1970.

[210] S. H. Rice, *Evolutionary Theory: Mathematical and Conceptual Foundations.* Sunderland, MA: Sinauer Associates, 1930.

[211] G. P. Wagner and L. Altenberg, “Complex adaptations and the evolution of evolvability,” *Evolution*, vol. 50, no. 3, pp. 967–976, 1996.

[212] L. Altenberg, “Open problems in the spectral analysis of evolutionary dynamics,” in *Frontiers of Evolutionary Computation (Genetic Algorithms And Evolutionary Computation Series)*, A. Menon, Ed. Boston, MA: Kluwer Academic Publishers, 2004, vol. 11, ch. 4, pp. 73–99.

[213] ——, “Modularity in evolution: Some low-level questions,” in *Modularity: Understanding the Development and Evolution of Complex Natural Systems*, D. Rasskin-Gutman and W. Callebaut, Eds. Cambridge, MA: MIT Press, 2004.

[214] D. E. Goldberg and K. Deb, “A comparative analysis of selection schemes used in genetic algorithms,” in *FOGA*, G. J. E. Rawlins, Ed. Morgan Kaufmann, 1990, pp. 69–93.

[215] M. Kimura and T. Ohta, “The average number of generations until fixation of a mutant gene in a finite population.” *Genetics*, vol. 61, pp. 763–771, 1969.

[216] D. Thierens and D. E. Goldberg, “Mixing in genetic algorithms,” in *ICGA*, S. Forrest, Ed. Morgan Kaufmann, 1993, pp. 38–47.

[217] ——, “Elitist recombination: An integrated selection recombination ga,” in *International Conference on Evolutionary Computation*, 1994, pp. 508–512.

[218] ——, "Convergence models of genetic algorithm selection schemes," in *PPSN*, ser. Lecture Notes in Computer Science, Y. Davidor, H.-P. Schwefel, and R. Männer, Eds., vol. 866. Springer, 1994, pp. 119–129.

[219] T. Bäck, "Selective pressure in evolutionary algorithms: A characterization of selection mechanisms," in *International Conference on Evolutionary Computation*, 1994, pp. 57–62.

[220] G. Rudolph, "Convergence of non-elitist strategies," in *International Conference on Evolutionary Computation*, 1994, pp. 63–66.

[221] H. Mühlenbein and D. Schlierkamp-Voosen, "Analysis of selection, mutation and recombination in genetic algorithms," in *Evolution and Biocomputation*, ser. Lecture Notes in Computer Science, W. Banzhaf and F. H. Eeckman, Eds., vol. 899. Springer, 1995, pp. 142–168.

[222] T. Blickle and L. Thiele, "A comparison of selection schemes used in evolutionary algorithms," *Evolutionary Computation*, vol. 4, no. 4, pp. 361–394, 1996.

[223] ——, "A mathematical analysis of tournament selection," in *ICGA*, L. J. Eshelman, Ed. Morgan Kaufmann, 1995, pp. 9–16.

[224] S. W. Mahfoud, "Genetic drift in sharing methods," in *International Conference on Evolutionary Computation*, 1994, pp. 67–72.

[225] H. Asoh and H. Mühlenbein, "On the mean convergence time of evolutionary algorithms without selection and mutation," in *PPSN*, ser. Lecture Notes in Computer Science, Y. Davidor, H.-P. Schwefel, and R. Männer, Eds., vol. 866. Springer, 1994, pp. 88–97.

[226] J. Hesser and R. Männer, "Towards an optimal mutation probability for genetic algorithms," in *PPSN*, ser. Lecture Notes in Computer Science, H.-P. Schwefel and R. Männer, Eds., vol. 496. Springer, 1990, pp. 23–32.

[227] T. Bäck, "Optimal mutation rates in genetic search," in *ICGA*, S. Forrest, Ed. Morgan Kaufmann, 1993, pp. 2–8.

[228] D. M. Tate and A. E. Smith, "Expected allele coverage and the role of mutation in genetic algorithms," in *ICGA*, S. Forrest, Ed. Morgan Kaufmann, 1993, pp. 31–37.

[229] J. F. Crow and M. Kimura, *Introduction to Population Genetics Theory.* New York: Harper and Row, 1972.

[230] F. Rothlauf, *Representations for Genetic and Evolutionary Algorithms.* Physica-Verlag, 2002.

[231] J. E. Rowe, M. D. Vose, and A. H. Wright, “Representation invariant genetic operators,” *Evolutionary Computation*, Accepted for publication.

[232] D. E. Goldberg, *Genetic Algorithms in Search, Optimization & Machine Learning.* Reading, MA: Addison Wesley, 1989.

[233] C. Thornton, “Why GAs are hard to use,” *Complexity International*, vol. 4, 1997.

[234] H. Mühlenbein, “Evolution in time and space-the parallel genetic algorithm,” in *FOGA*, G. J. E. Rawlins, Ed. Morgan Kaufmann, 1990, pp. 316–337.

[235] G. Rudolph, *Convergence Properties of Evolutionary Algorithms.* Hamburg: Dr. Kovac, 1997.

[236] M. D. Vose, “A closer look at mutation in genetic algorithms,” *Annals of Mathematics and Artificial Intelligence*, vol. 10, pp. 423–434, 1994.

[237] t. f. e. Wikipedia, “Price equation,” http://en.wikipedia.org/wiki/Price_equation, downloaded May, 2009.

[238] N. J. Radcliffe, “Forma analysis and random respectful recombination,” in *ICGA*, R. K. Belew and L. B. Booker, Eds. Morgan Kaufmann, 1991, pp. 222–229.

[239] ——, “Equivalence class analysis of genetic algorithms,” *Complex Systems*, vol. 5, pp. 183–205, 1991.

[240] R. A. Watson, *Compositional Evolution: The Impact of Sex, Symbiosis, and Modularity on the Gradualist Framework of Evolution (Vienna Series in Theoretical Biology).* The MIT Press, 2006.

[241] R. A. Watson and T. Jansen, “A building-block royal road where crossover is provably essential,” in *GECCO*, H. Lipson, Ed. ACM, 2007, pp. 1452–1459.

[242] R. Poli, “Recursive conditional schema theorem, convergence and population sizing in genetic algorithms,” in *Foundations of Genetic Algorithms (FOGA-6).* Morgan Kaufman, 2000, pp. 143–163.

[243] ——, “Schema theorems without expectations,” in *In: Proceedings of the Genetic and Evolutionary Computation Conference (Banzhaf, et al. Eds).* Morgan Kaufmann, 1999, pp. 80–6.

[244] T. Jones, “Crossover, macromutation and population-based search,” in *ICGA*, L. J. Eshelman, Ed. Morgan Kaufmann, 1995, pp. 73–80.

[245] C. R. Stephens, H. Waelbroeck, and R. Aguirre, "Schemata as building blocks: does size matter," in *Foundations of Genetic Algorithms (FOGA-5).* San Mateo: Morgan Kaufmann, 1997, pp. 117–133.

[246] C. Chryssomalakos and C. R. Stephens, "What basis for genetic dynamics?" 2004, intituto de Ciencias Nuclearas UNAM.

[247] A. Lozano, V. Mireles, D. Monsivais, C. R. Stephens, S. A. Alcalá, and F. Cervantes, "Building blocks and search," in *MICAI*, ser. Lecture Notes in Computer Science, A. H. Aguirre, R. M. Borja, and C. A. R. García, Eds., vol. 5845. Springer, 2009, pp. 704–715.

[248] C. L. Bridges and D. E. Goldberg, "An analysis of reproduction and crossover in a binary-coded genetic algorithm," in *Proceedings of the Second International Conference on Genetic Algorithms*, J. Grefenstette, Ed. Hillsdale, N. J.: Lawrence Erlbaum Associates, 1987, pp. 9–13.

[249] A. E. Eiben, E. H. L. Aarts, and K. M. van Hee, "Global convergence of genetic algorithms: A markov chain analysis," in *PPSN*, ser. Lecture Notes in Computer Science, H.-P. Schwefel and R. Männer, Eds., vol. 496. Springer, 1990, pp. 4–12.

[250] A. E. Nix and M. D. Vose, "Modeling genetic algorithms with markov chains." *Ann. Math. Artif. Intell.*, vol. 5, no. 1, pp. 77–88, 1992.

[251] T. E. Davis and J. C. Principe, "A markov chain framework for the simple genetic algorithm," *Evolutionary Compututation*, vol. 1, no. 3, pp. 269–288, 1993.

[252] M. D. Vose and G. E. Liepins, "Schema disruption," in *ICGA*, R. K. Belew and L. B. Booker, Eds. Morgan Kaufmann, 1991, pp. 237–243.

[253] M. D. Vose, "Modeling simple genetic algorithms," in *FOGA*, L. D. Whitley, Ed. Morgan Kaufmann, 1992, pp. 63–73.

[254] ——, "Modeling simple genetic algorithms," *Evolutionary Computation*, vol. 3, no. 4, pp. 453–472, 1996.

[255] ——, "Formalizing genetic algorithms," in *Proceedings of Genetic Algorithms, Neural Nets, and Simulated Annealing Applied to Problems in Signal and Image Processing.* Glasgow, UK: IEEE Press, 1990.

[256] ——, *The Simple Genetic Algorithm: Foundations and Theory.* Cambridge, MA: MIT Press, 1998.

[257] J. E. Rowe, "The dynamical systems model of the simple genetic algorithm," in *Theoretical Aspects of Evolutionary Computing*, L. Kallel, B. Naudts, and A. Rogers, Eds. Berlin: Springer, 2001, pp. 31–57.

[258] C. H. van Kemenade, J. N. Kok, H. L. Poutré, and D. Thierens, "Transmission function models of infinite population genetic algorithms," Centrum voor Wiskunde en Informatica, Amsterdam, The Netherlands, Tech. Rep. SEN-R9838.

[259] C. R. Stephens, R. Poli, A. H. Wright, and J. E. Rowe, "Exact results from a coarse grained formulation of the dynamics of variable-length genetic algorithms," in *GECCO*, W. B. Langdon, E. Cantú-Paz, K. E. Mathias, R. Roy, D. Davis, R. Poli, K. Balakrishnan, V. Honavar, G. Rudolph, J. Wegener, L. Bull, M. A. Potter, A. C. Schultz, J. F. Miller, E. K. Burke, and N. Jonoska, Eds. Morgan Kaufmann, 2002, pp. 578–585.

[260] A. A. Contreras, J. E. Rowe, and C. R. Stephens, "Coarse-graining in genetic algorithms: Some issues and examples," in *GECCO*, ser. Lecture Notes in Computer Science, E. Cantú-Paz, J. A. Foster, K. Deb, L. Davis, R. Roy, U.-M. O'Reilly, H.-G. Beyer, R. K. Standish, G. Kendall, S. W. Wilson, M. Harman, J. Wegener, D. Dasgupta, M. A. Potter, A. C. Schultz, K. A. Dowsland, N. Jonoska, and J. F. Miller, Eds., vol. 2723. Springer, 2003, pp. 874–885.

[261] C. R. Stephens and R. Poli, "Coarse-grained dynamics for generalized recombination," *IEEE Trans. Evolutionary Computation*, vol. 11, no. 4, pp. 541–557, 2007.

[262] R. Poli and C. R. Stephens, "Theoretical analysis of generalised recombination," in *Congress on Evolutionary Computation*. IEEE, 2005, pp. 411–418.

[263] C. R. Stephens and R. Poli, "Coarse graining in an evolutionary algorithm with recombination, duplication and inversion," in *Congress on Evolutionary Computation*. IEEE, 2005, pp. 1683–1690.

[264] J. Shapiro, A. Prügel-Bennett, and M. Rattray, "A statistical mechanical formulation of the dynamics of genetic algorithms," in *Evolutionary Computing, AISB Workshop*, ser. Lecture Notes in Computer Science, T. C. Fogarty, Ed., vol. 865. Springer, 1994, pp. 17–27.

[265] A. Prügel-Bennett and J. L. Shapiro, "An analysis of genetic algorithms using statistical mechanics." *Physical Review Letters*, vol. 72, no. 9, pp. 1305 – 1309, 1994.

[266] A. Rogers and A. Prügel-Bennett, "Modelling the dynamics of a steady state genetic algorithm," in *FOGA*, W. Banzhaf and C. R. Reeves, Eds. Morgan Kaufmann, 1998, pp. 57–68.

[267] A. Prügel-Bennett, "Modeling genetic algorithm dyanamics," in *Theoretical Aspects of Evolutionary Computation*, L. Kallel, B. Naudts, and A. Rogers, Eds. New York: Springer Verlag, 2001, pp. 59–85.

[268] J. L. Shapiro, “Statistical mechanics theory of genetic algorithms,” in *Theoretical Aspects of Evolutionary Computation*, L. Kallel, B. Naudts, and A. Rogers, Eds. New York: Springer Verlag, 2001, pp. 87–108.

[269] E. van Nimwegen, “The statistical dynamics of epochal evolution,” Ph.D. dissertation, Utrecht University, Utrecht, The Netherlands, 1999.

[270] P. S. Oliveto, J. He, and X. Yao, “Time complexity of evolutionary algorithms for combinatorial optimization: A decade of results,” *Int. Journal of Automation and Computing*, vol. 04, no. 3, pp. 281–293, 2007.

[271] S. Droste, T. Jansen, and I. Wegener, “On the optimization of unimodal functions with the (1 + 1) evolutionary algorithm,” in *PPSN*, ser. Lecture Notes in Computer Science, A. E. Eiben, T. Bäck, M. Schoenauer, and H.-P. Schwefel, Eds., vol. 1498. Springer, 1998, pp. 13–22.

[272] ——, “A rigorous complexity analysis of the (1 + 1) evolutionary algorithm for separable functions with boolean inputs,” *Evolutionary Computation*, vol. 6, no. 2, pp. 185–196, 1998.

[273] J. Nesetril, Ed., *Algorithms - ESA ’99, 7th Annual European Symposium, Prague, Czech Republic, July 16-18, 1999, Proceedings*, ser. Lecture Notes in Computer Science, vol. 1643. Springer, 1999.

[274] T. Jansen and I. Wegener, “On the utility of populations in evolutionary algorithms,” in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO) 2001.* Morgan Kaufmann, 2001, pp. 1034–1041.

[275] R. Motwani and P. Raghavan, *Randomized Algorithms.* New York: Cambridge University Press, 1995.

[276] J. He and X. Yao, “Drift analysis and average time complexity of evolutionary algorithms,” *Artif. Intell.*, vol. 127, no. 1, pp. 57–85, 2001.

[277] C. Witt, “Runtime analysis of the (mu + 1) ea on simple pseudo-boolean functions,” *Evolutionary Computation*, vol. 14, no. 1, pp. 65–86, 2006.

[278] D. H. Wolpert and W. G. Macready, “No free lunch theorems for optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, April 1997.

[279] D. H. Wolpert, “The lack of a priori distinctions between learning algorithms,” *Neural Comput.*, vol. 8, no. 7, pp. 1341–1390, 1996.

[280] D. H. Wolpert and W. G. Macready, “Coevolutionary free lunches,” *IEEE Trans. Evolutionary Computation*, vol. 9, no. 6, pp. 721–735, 2005.

[281] S. G. Ficici, “Solution concepts in coevolutionary algorithms,” Ph.D. dissertation, Brandeis University, Waltham, MA, 2004.

[282] A. Bucci and J. B. Pollack, “A mathematical framework for the study of coevolution,” in *Foundations of Genetic Algorithms FOGA-7).* Morgan Kaufmann, 2003, pp. 221–235.

[283] R. P. Wiegand, “An analysis of cooperative coevolutionary algorithms,” Ph.D. dissertation, George Mason University, Fairfax, VA, USA, 2004.

[284] S. Droste, T. Jansen, and I. Wegener, “Perhaps not a free lunch but at least a free appetizer,” in *GECCO*, W. Banzhaf, J. M. Daida, A. E. Eiben, M. H. Garzon, V. Honavar, M. J. Jakiela, and R. E. Smith, Eds. Morgan Kaufmann, 1999, pp. 833–839.

[285] D. Whitley and J. E. Rowe, “Focused no free lunch theorems,” in *GECCO*, C. Ryan and M. Keijzer, Eds. ACM, 2008, pp. 811–818.

[286] J. E. Rowe, M. D. Vose, and A. H. Wright, “Reinterpreting no free lunch,” *Evolutionary Computation*, vol. 17, no. 1, pp. 117–129, 2009.

[287] D. B. Fogel and J. Atmar, “Comparing genetic operators with gaussian mutations in simulated evolutionary processes using linear systems,” *Biological Cybernetics*, vol. 63, pp. 111–114, 1990.

[288] L. J. Eshelman and J. D. Schaffer, “Crossover’s niche,” in *ICGA*, S. Forrest, Ed. Morgan Kaufmann, 1993, pp. 9–14.

[289] M. Mitchell, S. Forrest, and J. H. Holland, “The royal road for genetic algorithms: Fitness landscapes and GA performance,” in *Proceedings of the First European Conference on Artificial Life. Toward a Practice of Autonomous Systems*, F. J. Varela and P. Bourgine, Eds. Cambridge, MA: MIT Press, 1992, pp. 243–254.

[290] T. Jones, “A description of holland’s royal road function,” *Evolutionary Computation*, vol. 2, no. 4, pp. 409–415, 1994.

[291] S. Forrest and M. Mitchell, “Relative building-block fitness and the building-block hypothesis,” in *Foundations of genetic algorithms 2*, L. D. Whitley, Ed. San Mateo: Morgan Kaufmann, 1993, pp. 109–126.

[292] ——, “What makes a problem hard for a Genetic Algorithm? Some anomalous results and their explanation,” *Machine Learning*, vol. 13, pp. 285–319, 1993.

[293] M. Mitchell, J. H. Holland, and S. Forrest, “When will a genetic algorithm outperform hill climbing.” in *NIPS*, J. D. Cowan, G. Tesauro, and J. Alspector, Eds. Morgan Kaufmann, 1993, pp. 51–58.

[294] A. Prügel-Bennett, "When a genetic algorithm outperforms hill-climbing," *Theor. Comput. Sci.*, vol. 320, no. 1, pp. 135–153, 2004.

[295] T. Jansen and I. Wegener, "Real royal road functions–where crossover provably is essential," *Discrete Applied Mathematics*, vol. 149, no. 1-3, pp. 111–125, 2005.

[296] K. Deb and D. E. Goldberg, "Analyzing deception in trap functions." in *FOGA*, L. D. Whitley, Ed. Morgan Kaufmann, 1992, pp. 93–108.

[297] R. A. Watson, "Analysis of recombinative algorithms on a non-separable building-block problem," in *Foundations of Genetic Algorithms (FOGA-6)*, W. N. Martin and W. M. Spears, Eds. Morgan Kaufmann, 2001, pp. 69–89.

[298] R. A. Watson and J. B. Pollack, "Hierarchically consistent test problems for genetic algorithms," in *Proceedings of the Congress on Evolutionary Computation*, P. J. Angeline, Z. Michalewicz, M. Schoenauer, X. Yao, and A. Zalzala, Eds., vol. 2. IEEE Press, 1999, pp. 1406–1413.

[299] M. Dietzfelbinger, B. Naudts, C. V. Hoyweghen, and I. Wegener, "The analysis of a recombinative hill-climber on h-iff," *IEEE Trans. Evolutionary Computation*, vol. 7, no. 5, pp. 417–423, 2003.

[300] T. Jansen and I. Wegener, "The analysis of evolutionary algorithms - a proof that crossover really can help," *Algorithmica*, vol. 34, no. 1, pp. 47–66, 2002.

[301] T. Storch and I. Wegener, "Real royal road functions for constant population size," *Theor. Comput. Sci.*, vol. 320, no. 1, pp. 123–134, 2004.

[302] S. Fischer and I. Wegener, "The one-dimensional ising model: Mutation versus recombination," *Theor. Comput. Sci.*, vol. 344, no. 2-3, pp. 208–225, 2005.

[303] D. Sudholt, "Crossover is provably essential for the ising model on trees," in *GECCO '05: Proceedings of the 2005 conference on Genetic and evolutionary computation*. New York: ACM, 2005, pp. 1161–1167.

[304] J. Horn, D. E. Goldberg, and K. Deb, "Long path problems." in *PPSN*, ser. Lecture Notes in Computer Science, Y. Davidor, H.-P. Schwefel, and R. Männer, Eds., vol. 866. Springer, 1994, pp. 149–158.

[305] J. Garnier and L. Kallel, "Statistical distribution of the convergence time of evolutionary algorithms for long-path problems." *IEEE Trans. Evolutionary Computation*, vol. 4, no. 1, pp. 16–30, 2000.

[306] D. A. Rosenblueth and C. R. Stephens, "An analysis of recombination in some simple landscapes," in *MICAI*, ser. Lecture Notes in Computer Science, A. H. Aguirre, R. M. Borja, and C. A. R. García, Eds., vol. 5845. Springer, 2009, pp. 716–727.

[307] C. Stephens, "The renormalization group and the dynamics of genetic systems," *Acta Physica Slovaka*, vol. 52, no. 6, pp. 515–524, 2002.

[308] M. D. Vose and A. H. Wright, "The simple genetic algorithm and the walsh transform: Part ii, the inverse," *Evol. Comput.*, vol. 6, no. 3, pp. 275–289, 1998.

[309] K. M. Burjorjee, "Two remarkable computational competencies of the simple genetic algorithm," *CoRR*, vol. abs/0810.3357, 2008.

[310] R. Poli, A. H. Wright, N. McPhee, and W. Langdon, "Emergent behaviour, population-based search and low-pass filtering," in *Proceedings of the Congress on Evolutionary Computation (CEC) 2006.* IEEE Press, 2006, pp. 88–95.

[311] L. Davis, *Handbook of Genetic Algorithms.* New York: Van Nostrand Reinhold, 1991.

[312] L. J. Eshelman and J. D. Schaffer, "Preventing premature convergence in genetic algorithms by preventing incest," in *ICGA*, R. K. Belew and L. B. Booker, Eds. Morgan Kaufmann, 1991, pp. 115–122.

[313] S. Arnone, M. Dell'Orto, and A. Tettamanzi, "Toward a fuzzy government of genetic populations," in *ICTAI*, 1994, pp. 585–591.

[314] H. Y. Xu and G. Vukovich, "A fuzzy genetic algorithm with effective search and optimization," in *Proceedings of the 1993 International Joint Conference on Neural Networks*, 1993, pp. 2967–2970.

[315] M. A. Lee and H. Takagi, "A framework for studying the effects of dynamic crossover, mutation, and population sizing in genetic algorithms," in *IEEE/Nagoya-University World Wisepersons Workshop*, ser. Lecture Notes in Computer Science, T. Furuhashi, Ed., vol. 1011. Springer, 1994, pp. 111–126.

[316] S. W. Mahfoud, "Niching methods for genetic algorithms," Ph.D. dissertation, University of Illinois at Urbana-Champaign, Urbana, IL, May 1995, illiGAL Report 95001.

[317] J. D. Schaffer, M. Mani, L. J. Eshelman, and K. E. Mathias, "The effect of incest prevention on genetic drift," in *Foundations of Genetic Algorithms (FOGA-5)*, W. Banzhaf and C. R. Reeves, Eds., 1998, pp. 235–244.

[318] T. Friedrich, N. Hebbinghaus, and F. Neumann, "Rigorous analyses of simple diversity mechanisms," in *GECCO*, H. Lipson, Ed. ACM, 2007, pp. 1219–1225.

[319] T. Friedrich, P. S. Oliveto, D. Sudholt, and C. Witt, "Theoretical analysis of diversity mechanisms for global exploration," in *GECCO*, C. Ryan and M. Keijzer, Eds. ACM, 2008, pp. 945–952.

[320] T. Friedrich, N. Hebbinghaus, and F. Neumann, "Comparison of simple diversity mechanisms on plateau functions," *Theor. Comput. Sci.*, vol. 410, no. 26, pp. 2455–2462, 2009.

[321] X. Yao, "An overview of evolutionary computation," *Chinese Journal of Advanced Software Research*, vol. 3, pp. 12–29, 1996.

[322] G. Rudolph, "Finite markov chain results in evolutionary computation: A tour d'horizon," *Fundam. Inform.*, vol. 35, no. 1-4, pp. 67–89, 1998.

[323] A. E. Eiben and G. Rudolph, "Theory of evolutionary algorithms: A bird's eye view," *Theor. Comput. Sci.*, vol. 229, no. 1, pp. 3–9, 1999.

[324] H.-G. Beyer, H.-P. Schwefel, and I. Wegener, "How to analyse evolutionary algorithms," *Theoretical Computer Science*, vol. 287, no. 1, pp. 101–130, 2002.

[325] J. E. Rowe, "The dynamical systems model of the simple genetic algorithm," in *Theoretical Aspects of Evolutionary Computation*, L. Kallel, B. Naudts, and A. Rogers, Eds. New York: Springer Verlag, 2001, pp. 51–57.

[326] T. E. Davis, "Toward an extrapolation of the simulated annealing convergence theory onto the simple genetic algorithm," Ph.D. dissertation, University of Florida, Gainesville, FL, USA, 1991.

[327] W. E. Hart and R. K. Belew, "Optimising an arbitrary function is hard for the genetic algorithm," in *ICGA*, R. K. Belew and L. B. Booker, Eds. Morgan Kaufmann, 1991, pp. 190–195.

[328] H.-G. Beyer, "Toward a theory of evolution strategies: Some asymptotical results from the (1, +lambda)-theory," *Evolutionary Computation*, vol. 1, no. 2, pp. 165–188, 1993.

[329] ——, "Toward a theory of evolution strategies: The (mu, lambda)-theory," *Evolutionary Computation*, vol. 2, no. 4, pp. 381–407, 1994.

[330] ——, "Towards a theory of evolution strategies: Results for $(1+, \lambda)$-strategies on (nearly) arbitrary fitness functions," in *PPSN*, ser. Lecture Notes in Computer Science, Y. Davidor, H.-P. Schwefel, and R. Männer, Eds., vol. 866. Springer, 1994, pp. 58–67.

[331] G. Rudolph, "Convergence of evolutionary algorithms in general search spaces," in *International Conference on Evolutionary Computation*, 1996, pp. 50–54.

[332] ——, “Asymptotical convergence rates of simple evolutionary algorithms under factorizing mutation distributions,” in *Artificial Evolution*, ser. Lecture Notes in Computer Science, J.-K. Hao, E. Lutton, E. M. A. Ronald, M. Schoenauer, and D. Snyers, Eds., vol. 1363. Springer, 1997, pp. 223–236.

[333] I. Wegener, “On the design and analysis of evolutionary algorithms,” in *Proceedings of theWorkshop on Algorithm Engineering as a New Paradigm*, 2000, pp. 36–47. [Online]. Available: citeseer.nj.nec.com/wegener00design.html

[334] ——, *Methods for the Analysis of Evolutionary Algorithms on Pseudo-Boolean Functions.* Kluwer, 2002, ch. 14, pp. 349–369.

[335] M. O. Rabin, “Probabilistic algorithms,” 1976, pp. 21–39.

[336] R. M. Karp, “An introduction to randomized algorithms,” *Discrete Appl. Math.*, vol. 34, no. 1-3, pp. 165–201, 1991.

[337] H.-G. Beyer, *The Theory of Evolution Strategies.* Heidelberg: Springer-Verlag, 2001.

[338] S. Droste, T. Jansen, and I. Wegener, “A natural and simple function which is hard for all evolutionary algorithms,” in *”26th Annual Conference of the IEEE Industrial Electronics Society, IECON*, vol. 4. Nagoya: IEEE, 22-28 Oct. 2000, pp. 2704 – 2709.

[339] T. Jansen and I. Wegener, “On the choice of the mutation probability for the (1+1) ea.” in *PPSN*, ser. Lecture Notes in Computer Science, M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J. J. M. Guervós, and H.-P. Schwefel, Eds., vol. 1917. Springer, 2000, pp. 89–98.

[340] S. Droste, T. Jansen, and I. Wegener, “Dynamic parameter control in simple evolutionary algorithms,” in *Foundations of Genetic Algorithms (FOGA-6)*, W. N. Martin and W. M. Spears, Eds. San Francisco CA: Morgan Kaufmann, 2001, pp. 275–294. [Online]. Available: citeseer.ist.psu.edu/droste00dynamic.html

[341] ——, “On the analysis of the (1+1) evolutionary algorithm,” *Theor. Comput. Sci.*, vol. 276, no. 1-2, pp. 51–81, 2002.

[342] T. Jansen and I. Wegener, “A comparison of simulated annealing with a simple evolutionary algorithm on pseudo-boolean functions of unitation,” *Theor. Comput. Sci.*, vol. 386, no. 1-2, pp. 73–93, 2007.

[343] ——, “Evolutionary algorithms - how to cope with plateaus of constant fitness and when to reject strings of the same fitness,” *IEEE Trans. Evolutionary Computation*, vol. 5, no. 6, pp. 589–599, 2001.

[344] I. Wegener and C. Witt, "On the analysis of a simple evolutionary algorithm on quadratic pseudo-boolean functions," *J. Discrete Algorithms*, vol. 3, no. 1, pp. 61–78, 2005.

[345] T. Jansen, K. A. D. Jong, and I. Wegener, "On the choice of the offspring population size in evolutionary algorithms," *Evolutionary Computation*, vol. 13, no. 4, pp. 413–440, 2005.

[346] C. Witt, "Population size vs. runtime of a simple ea," in *Evolutionary Computation, 2003. CEC '03. The 2003 Congress on*, vol. 3, Dec. 2003, pp. 1996–2003 Vol.3.

[347] ——, "Population size versus runtime of a simple evolutionary algorithm," *Theor. Comput. Sci.*, vol. 403, no. 1, pp. 104–120, 2008.

[348] J. He and X. Yao, "From an individual to a population: an analysis of the first hitting time of population-based evolutionary algorithms," *IEEE Trans. Evolutionary Computation*, vol. 6, no. 5, pp. 495–511, 2002.

[349] T. Jansen and I. Wegener, "On the analysis of evolutionary algorithms - a proof that crossover really can help," in *ESA*, ser. Lecture Notes in Computer Science, J. Nesetril, Ed., vol. 1643. Springer, 1999, pp. 184–193.

[350] ——, "Real royal road functions - where crossover provably is essential," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, L. Spector, E. D. Goodman, A. Wu, W. Langdon, H.-M. Voigt, M. Gen, S. Sen, M. Dorigo, S. Pezeshk, M. H. Garzon, and E. Burke, Eds. San Francisco, CA: Morgan Kaufmann, 7-11 July 2001, pp. 375–382.

[351] T. Storch and I. Wegener, "Real royal road functions for constant population size," in *GECCO*, ser. Lecture Notes in Computer Science, E. Cantú-Paz, J. A. Foster, K. Deb, L. Davis, R. Roy, U.-M. O'Reilly, H.-G. Beyer, R. K. Standish, G. Kendall, S. W. Wilson, M. Harman, J. Wegener, D. Dasgupta, M. A. Potter, A. C. Schultz, K. A. Dowsland, N. Jonoska, and J. F. Miller, Eds., vol. 2724. Springer, 2003, pp. 1406–1417.

[352] ——, "Real royal road functions for constant population size," *Theor. Comput. Sci.*, vol. 320, no. 1, pp. 123–134, 2004.

[353] J. C. Culberson, "Crossover versus mutation: Fueling the debate: Tga versus giga," in *ICGA*, S. Forrest, Ed. Morgan Kaufmann, 1993, p. 632.

[354] B. Doerr, T. Jansen, and C. Klein, "Comparing global and local mutations on bit strings," in *GECCO*, C. Ryan and M. Keijzer, Eds. ACM, 2008, pp. 929–936.

[355] P. P.S.Oliveto, P. Lehre, and F. Neumann, “Theoretical analysis of rank-based mutation - combining exploration and exploitation,” in *Proceedings of the IEEE Congress on Evolutionary Computation 2009 (CEC’09).* IEEE Press, 2009, pp. 1455–1462.

[356] F. Neumann, P. S. Oliveto, and C. Witt, “Theoretical analysis of fitness-proportional selection: landscapes and efficiency,” in *GECCO*, F. Rothlauf, Ed. ACM, 2009, pp. 835–842.

[357] P. S. Oliveto and C. Witt, “Simplified drift analysis for proving lower bounds in evolutionary computation,” in *PPSN*, ser. Lecture Notes in Computer Science, G. Rudolph, T. Jansen, S. M. Lucas, C. Poloni, and N. Beume, Eds., vol. 5199. Springer, 2008, pp. 82–91.

[358] F. Neumann, “Expected runtimes of evolutionary algorithms for the eulerian cycle problem,” in *Proceedings of the IEEE Congress on Evolutionary Computation 2004 (CEC’04).* IEEE Press, 2004, pp. 904–910.

[359] B. Doerr, N. Hebbinghaus, and F. Neumann, “Speeding up evolutionary algorithms through asymmetric mutation operators,” *Evolutionary Computation*, vol. 15, no. 4, pp. 401–410, 2007.

[360] “Analysis of a simple evolutionary algorithm for the multiobjective shortest path problem,” in *FOGA*, C. Horoba, Ed. New York: ACM Press, 2009.

[361] B. Doerr, E. Happ, and C. Klein, “Crossover can provably be useful in evolutionary computation,” in *GECCO*, C. Ryan and M. Keijzer, Eds. ACM, 2008, pp. 539–546.

[362] E. van Nimwegen, J. P. Crutchfield, and M. Mitchell, “Finite populations induce metastability in evolutionary search,” *Physics Letters A*, vol. 229, pp. 144–150, 1997.

[363] M. D. Vose, *The Simple Genetic Algorithm: Foundations and Theory.* Cambridge, MA: MIT Press, 1999.

[364] J. Dean and S. Ghemawat, “Mapreduce: Simplified data processing on large clusters,” in *OSDI*, 2004, pp. 137–150.

[365] S. Brin and L. Page, “The anatomy of a large-scale hypertextual web search engine,” *Computer Networks*, vol. 30, no. 1-7, pp. 107–117, 1998.

[366] J. E. Rowe, M. D. Vose, and A. H. Wright, “Coarse graining selection and mutation,” in *FOGA*, ser. Lecture Notes in Computer Science, A. H. Wright, M. D. Vose, K. A. D. Jong, and L. M. Schmitt, Eds., vol. 3469. Springer, 2005, pp. 176–191.

[367] ——, “Differentiable coarse graining,” *Theor. Comput. Sci.*, vol. 361, no. 1, pp. 111–129, 2006.

[368] K. M. Burjorjee, “Sufficient conditions for coarse-graining evolutionary dynamics,” in *FOGA*, 2007, pp. 35–53.

[369] B. Mitavskiy, J. E. Rowe, A. H. Wright, and L. M. Schmitt, “Quotients of markov chains and asymptotic properties of the stationary distribution of the markov chain associated to an evolutionary algorithm,” *Genetic Programming and Evolvable Machines*, vol. 9, no. 2, pp. 109–123, 2008.

[370] A. Menon, *Frontiers of Evolutionary Computation.* Norwell, MA: Kluwer Academic Publishers, 2004.

[371] W. Feller, *An Introduction to Probability Theory and Its Applications, Volume I*, 3rd ed. New York: John Wiley and Sons, Inc., 1968.

[372] G. Bernardi, “The neoselectionist theory of genome evolution,” *PNAS*, vol. 104, no. 20, pp. 8385–8390, May 2007.

[373] J. N. Richter, A. H. Wright, and J. Paxton, “Ignoble trails - where crossover is provably harmful,” in *PPSN*, ser. Lecture Notes in Computer Science, G. Rudolph, T. Jansen, S. M. Lucas, C. Poloni, and N. Beume, Eds., vol. 5199. Springer, 2008, pp. 92–101.

[374] M. Mitzenmacher and E. Upfal, *Probability and Computing: Randomized Algorithms and Probabilistic Analysis.* New York, NY, USA: Cambridge University Press, 2005.

[375] J. E. Rowe, “Population fixed-points for functions of unitation,” in *Foundations of genetic algorithms (FOGA-5)*, W. Banzhaf and C. Reeves, Eds. San Mateo: Morgan Kaufmann, 1999, pp. 60–84.

[376] A. H. Wright, J. E. Rowe, and J. R. Neil, “Analysis of the simple genetic algorithm on the single-peak and double-peak landscapes,” in *Proceedings of the Congress on Evolutionary Computation (CEC) 2002.* IEEE Press, 2002, pp. 214–219. [Online]. Available: http://www.cs.umt.edu/u/wright/

[377] G. Ochoa and I. Harvey, “Recombination and error thresholds in finite populations,” in *Foundations of Genetic Algorithms (FOGA-5).* San Mateo: Morgan Kaufmann, 1997, pp. 245–264.

[378] M. Eigen, J. McCaskill, and P. Schuster, “Molecular quasi-species,” *J. Phys. Chem.*, vol. 92, pp. 6881–6891, 1988.

[379] R. Dawkins, *The Blind Watchmaker.* New York: W. W. Norton & Company, Inc., September 1986.

[380] H. M. Taylor and S. Karlin, *An Introduction to Stochastic Modeling*, 3rd ed. San Diego: Academic Press, 1998.

[381] H. Suzuki and Y. Iwasa, “Crossover accelerates evolution in gas with a babel-like fitness landscape: Mathematical analyses,” *Evolutionary Computation*, vol. 7, no. 3, pp. 275–310, 1999.

[382] M. C. Boerlijst, S. Bonhoeffer, and M. A. Nowak, “Viral quasi-species and recombination,” *Proc. Royal Society London B*, vol. 263, pp. 1577–1584, 1996.

[383] A. H. Wright, J. E. Rowe, R. Poli, and C. R. Stephens, “Bistability in a gene pool GA with mutation,” in *Foundations of Genetic Algorithms (FOGA-7)*, J. E. Rowe, K. DeJong, and R. Poli, Eds. San Mateo: Morgan Kaufmann, 2003.

[384] ——, “A fixed point analysis of a gene pool GA with mutation,” in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO).* San Francisco, CA.: Morgan Kaufmann Publishers, 2002.

[385] H. Mühlenbein, “The equation for the response to selection and its use for prediction,” *Evolutionary Computation*, vol. 5, no. 3, pp. 303–346, 1997.

[386] S. Baluja and R. Caruana, “Removing the genetics from the standard genetic algorithm,” in *The Int. Conf. on Machine Learning 1995*, A. Prieditis and S. Russel, Eds. San Mateo, CA: Morgan Kaufmann, 1995, pp. 38–46. [Online]. Available: citeseer.nj.nec.com/baluja95removing.html

[387] A. H. Wright and G. Cripe, “Bistability of the needle function in the presence of truncation selection,” in *GECCO 2004: Proceedings of the Genetic and Evolutionary Computation Conference.* Springer Verlag, 2004. [Online]. Available: http://www.cs.umt.edu/u/wright/

[388] M. D. Vose and A. H. Wright, “Stability of vertex fixed points and applications.” in *FOGA*, L. D. Whitley and M. D. Vose, Eds. Morgan Kaufmann, 1995, pp. 103–113.

[389] A. H. Wright and J. N. Richter, “Strong recombination, weak selection, and mutation,” in *GECCO*, M. Cattolico, Ed. ACM, 2006, pp. 1369–1376.

[390] B. Wallace, *Topics in Population Genetics.* New York: W. W. Norton, 1968.

[391] H. Voigt and H. Mühlenbein, “Gene pool recombination and the utilization of covariances for the breeder genetic algorithm,” in *Proc. of the 2nd IEEE International Conference on Evolutionary Computation*, Z. Michalewicz, Ed. New York: IEEE Press, 1995, pp. 172– 177.

[392] T. Gedeon, C. Hayes, and R. Swanson, "Genericity of the fixed point set for the infinite population genetic algorithm," in *FOGA*, 2007, pp. 97–109.

[393] R. Seydel, *Practical Bifurcation and Stability Analysis.* Berlin: Springer, 1994.

[394] R. Hinterding, Z. Michalewicz, and A. Eiben, "Adaptation in evolutionary computation: A survey," in *IEEECEP: Proceedings of The IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence*, 1997, pp. 65–69. [Online]. Available: citeseer.ist.psu.edu/article/hinterding97adaptation.html

[395] T. Bäck and M. Schütz, "Intelligent mutation rate control in canonical genetic algorithms," in *ISMIS*, ser. Lecture Notes in Computer Science, Z. W. Ras and M. Michalewicz, Eds., vol. 1079. Springer, 1996, pp. 158–167.

[396] D. Thierens, "Adaptive mutation rate control schemes in genetic algorithms," in *Proceedings of the 2002 IEEE Congress on Evolutionary Computation CEC2002*, D. B. Fogel, M. A. El-Sharkawi, X. Yao, G. Greenwood, H. Iba, P. Marrow, and M. Shackleton, Eds. Piscataway, NJ: IEEE Press, 2002, pp. 980–985.

[397] Y. Shi, R. Eberhart, and C. Y., "Implementation of evolutionary fuzzy systems," *IEEE Transactions on Fuzzy Systems*, vol. 7, no. 2, pp. 109–119, 1999.

[398] T. Ross, *Fuzzy Logic with Engineering Applications.* New York: McGraw-Hill, 1993.

[399] A. H. Wright, J. E. Rowe, and J. R. Neil, "Analysis of the simple genetic algorithm on the single-peak and double-peak landscapes," in *Proceedings of the 2002 Congress on Evolutionary Computation CEC2002*, D. B. Fogel, M. A. El-Sharkawi, X. Yao, G. Greenwood, H. Iba, P. Marrow, and M. Shackleton, Eds. Piscataway, NJ: IEEE Press, 2002, pp. 214–219.

[400] J. N. Richter, J. Paxton, and A. H. Wright, "EA models and population fixed-points versus mutation rates for functions of unitation," in *GECCO*, H.-G. Beyer and U.-M. O'Reilly, Eds. ACM, 2005, pp. 1233–1240.

[401] t. f. e. Wikipedia, "Chernoff bound," http://en.wikipedia.org/wiki/Chernoff_bound, downloaded February, 2010.

[402] T. Hagerup and C. Rüb, "A guided tour of chernoff bounds," *Information Processing Letters*, vol. 33, no. 6, pp. 305–308, 1990.

[403] J. Canny, "Cs174 lecture 10 chernoff bounds," http://www.eecs.berkeley.edu/~jfc/cs174/lecs/lec10/lec10.pdf, downloaded February, 2010.

[404] J. N. Richter, "Mathematica code for phd research," http://github.com/nealrichter/dissertation_code/tree/master/mathematica/.

[405] A. W. J. Neal Richter, "Maple code for phd research," http://github.com/nealrichter/dissertation_code/tree/master/maple/.

[406] J. N. Richter, "Code for iasted 2005 experiments," http://github.com/nealrichter/dissertation_code/tree/master/iasted_experiments/, dec 6, 2005.

[407] R. K. Belew and L. B. Booker, Eds., *Proceedings of the 4th International Conference on Genetic Algorithms, San Diego, CA, USA, July 1991.* Morgan Kaufmann, 1991.

[408] Y. Davidor, H.-P. Schwefel, and R. Männer, Eds., *Parallel Problem Solving from Nature - PPSN III, International Conference on Evolutionary Computation. The Third Conference on Parallel Problem Solving from Nature, Jerusalem, Israel, October 9-14, 1994, Proceedings*, ser. Lecture Notes in Computer Science, vol. 866. Springer, 1994.

[409] *Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2005, 2-4 September 2005, Edinburgh, UK.* IEEE, 2005.

[410] C. Ryan and M. Keijzer, Eds., *Genetic and Evolutionary Computation Conference, GECCO 2008, Proceedings, Atlanta, GA, USA, July 12-16, 2008.* ACM, 2008.

[411] R. Männer and B. Manderick, Eds., *Parallel Problem Solving from Nature 2, PPSN-II, Brussels, Belgium, September 28-30, 1992.* Elsevier, 1992.

[412] L. J. Eshelman, Ed., *Proceedings of the 6th International Conference on Genetic Algorithms, Pittsburgh, PA, USA, July 15-19, 1995.* Morgan Kaufmann, 1995.

[413] S. Forrest, Ed., *Proceedings of the 5th International Conference on Genetic Algorithms, Urbana-Champaign, IL, USA, June 1993.* Morgan Kaufmann, 1993.

[414] L. D. Whitley, Ed., *Proceedings of the Second Workshop on Foundations of Genetic Algorithms. Vail, Colorado, USA, July 26-29 1992.* Morgan Kaufmann, 1993.

[415] H. Lipson, Ed., *Genetic and Evolutionary Computation Conference, GECCO 2007, Proceedings, London, England, UK, July 7-11, 2007.* ACM, 2007.

[416] L. D. Whitley and M. D. Vose, Eds., *Proceedings of the Third Workshop on Foundations of Genetic Algorithms. Estes Park, Colorado, USA, July 31 - August 2 1994.* Morgan Kaufmann, 1995.

[417] A. H. Wright, M. D. Vose, K. A. D. Jong, and L. M. Schmitt, Eds., *Foundations of Genetic Algorithms (FOGA-8)*, ser. Lecture Notes in Computer Science, vol. 3469. Springer, 2005.

[418] A. H. Aguirre, R. M. Borja, and C. A. R. García, Eds., *MICAI 2009: Advances in Artificial Intelligence, 8th Mexican International Conference on Artificial Intelligence, Guanajuato, México, November 9-13, 2009. Proceedings*, ser. Lecture Notes in Computer Science, vol. 5845. Springer, 2009.

[419] G. Rudolph, T. Jansen, S. M. Lucas, C. Poloni, and N. Beume, Eds., *Parallel Problem Solving from Nature - PPSN X, 10th International Conference Dortmund, Germany, September 13-17, 2008, Proceedings*, ser. Lecture Notes in Computer Science, vol. 5199. Springer, 2008.

APPENDICES

APPENDIX A

MATHEMATICAL NOTES

A.1 Chernoff Bounds Examples

The Chernoff bound enables the calculation of bounds on the tail distributions created by the sum of independent random variables and is named for Herman Chernoff[401, 402]. It is commonly used to analyze the performance of randomized algorithms [275].

Assume there exists randomized algorithm A, and that the algorithm is being run against a computational problem for which it finds the correct (or optimal) answer with probability p, and an incorrect (or sum-optimal) answer with probability $1 - p$. Let algorithm A be run n times with its answer recorded. It is useful to know the probability that the answer given by a majority of the n rules is correct. This constitutes a Bernoulli trial and can be modeled with a trial of n coin flips where the coin lands heads with probability p. Note that each trial must be independent of all other trials.

The below example is taken from John Canny's UC Berkeley lecture on Chernoff bounds [403].

Example Question: What is the probability of tossing a fair coin less than $m < n/2$ heads in n tosses?

This question can be answered with the the lower tail Chernoff bound.

Let X_i be a random variable taking binary values where $Pr[X_i = 1] = p$ and $Pr[X_i = 0] = (1 - p)$. The bound is defined as:

$$Pr[X < (1 - \delta)\mu] < exp(-\mu\delta^2/2) \tag{A.1}$$

For the question above, let X be the sum of heads tossed and let $\mu = n/2$. Pick $\delta = (1 - 2m/n)$. The Chernoff bound gives the below closed form upper bound as an answer to the question:

$$Pr[X < m] < exp(-(n/4)(1 - 2m/n)^2) \tag{A.2}$$

For the specific question of tossing ten or fewer heads in 100 trials, this results in a probability less than $exp(-(100/4)(1 - 20/100)^2) = exp(-16) = 1.12 \times 10^{-7}$.

Note that one must pick $\delta \leq 1$ as negative X values are not of interest. Next this method is applied to a randomized algorithm.

Example EA Question: What is the expected number of generations until a $(1+1)EA$ jumps a k-bit unitation gap from state x_i to state x_j where $0 < f(x_i) < f(x_j)$ and all points between the two states have zero fitness?

Note that $k < n$ and assume one is interested only in the probability of flipping some string with m ones to any string with $m + k$ ones. Assume also that the mutation probability of flipping one bit is $1/n$ and every bit is evaluated for mutation independently. The expected waiting time to jump this gap is $O(n^k)$ steps.

The bound to jump this gap in $O(n^k \ln n)$ steps is calculated via Chernoff as follows:

1. Let the number of steps be $m = en^k \ln n$.
2. Let the probability of jumping this gap in one step be $p = 1/n^k$.
3. $\mu = m\,p = \frac{en^k \ln n}{n^k} = e \ln n$
4. Pick $\delta = \left(\frac{2}{e}\right)^{1/2}$

5. Calculate the lower tail bound

$$\begin{aligned} e^{-\left(\mu\delta^2/2\right)} &= e^{-\left(\frac{e\ln n(2/e)}{2}\right)} \\ &= e^{-\ln n} \\ &= \frac{1}{n} \end{aligned}$$

6. Thus the probability to jump this gap in $O(n^k \ln n)$ steps is $1 - 1/n$.

The bound to jump this gap in $O(n^{k+1})$ steps is calculated via Chernoff as follows:

1. Let the number of steps be $m = en^{k+1}$.
2. Let the probability of jumping this gap in one step be $p = 1/n^k$.
3. $\mu = m\,p = \frac{en^{k+1}}{n^k} = en$
4. Pick $\delta = \left(\frac{2}{e}\right)^{1/2}$
5. Calculate the lower tail bound

$$\begin{aligned} e^{-\left(\mu\delta^2/2\right)} &= e^{-\left(\frac{en(2/e)}{2}\right)} \\ &= e^{-n} \\ &= \frac{1}{e^n} \end{aligned}$$

6. Thus the probability to jump this gap in $O(n^{k+1})$ steps is $1 - 1/e^n$, which is bounded by $1 - 1/2^n$.

APPENDIX B

PUBLICATIONS AND PERMISSIONS

Publications contained in this work are as follows. In most cases this work extends these publications.

- J. Neal Richter, Alden Wright, John Paxton. "Exploration of Population Fixed Points Versus Mutation Rates for Functions of Unitation", Workshop on Evolutionary Computation Theory, GECCO 2004, Seattle, WA
- J. Neal Richter, John Paxton, Alden Wright. "EA Models of Population Fixed Points Versus Mutations Rates for Functions of Unitation", GECCO 2005, Washington, D.C.
- J. Neal Richter, John Paxton. "Adaptive Evolutionary Algorithms on Unitation, Royal Road and Longpath Functions." IASTED Computation Intelligence Conference 2005, Calgary, Alberta
- Alden Wright, J. Neal Richter "Strong Recombination, Weak Selection, and Mutation" GECCO 2006, Seattle, WA
- J. Neal Richter, Alden Wright and John Paxton. "Ignoble Trails - where crossover is provably harmful", PPSN 2008, Dortmund, Germany

Permissions to reuse copyrighted work were obtained for the following:

- Figures 3.1 and 3.2 from Morgan [105] by Princeton University Press,
- Figure 3.3 from Darlington [123] by his estate and the Taylor and Francis Group,
- Figure 3.4 from Wright [151] by the Genetics Society of America.

APPENDIX C

SOURCE CODE

C.1 Mathematica Code

All code for Chapter 7 and 8 is available at [404].

C.2 Maple Code

Some code for Chapter 10 is available at [405].

C.3 GALib Code

All code for Chapter 11 is available at [406].

Document Build: 0.99.255

Date: Mon Apr 19 10:57:51 MDT 2010