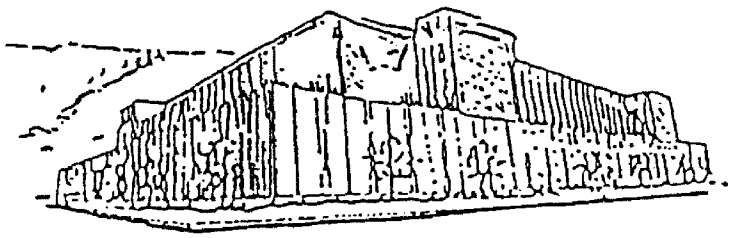
1996

Applying diploidy and dominance to artificial genetic search

Garrett L. Bidwell
The University of Montana



Maureen and Mike
MANSFIELD LIBRARY

The University of **MONTANA**

Applying Diploidy and Dominance to Artificial Genetic Search

by

Garrett L. Bidwell

B.S., Duke University, 1989

presented in partial fulfillment of the requirements

for the degree of

Master of Science

in Computer Science

The University of Montana

July 22, 1996

Approved by:

Chairperson

Dean, Graduate School

8-13-96

Date

Garrett L. Bidwell, M.S., July 22, 1996 Computer Science

Applying Diploidy and Dominance to Artificial Genetic Search (81 pp.)

Director: Alden H. Wright

Genetic Algorithms (GAs) are search and optimization procedures based on the mechanics of natural selection. They encode the parameters of a problem in a single-stranded or haploid binary string. However, most haploid organisms in the biological world are simple lifeforms such as bacteria. More complex lifeforms such as plants, animals, and humans rely on a diploid chromosome, which contains homologous chromosome pairs at each locus. When chromosome pairs contain different values at the same location, a dominance operator usually resolves the conflict.

The primary motivation for incorporating diploidy and dominance into GAs is to increase population diversity and thus avoid premature convergence to a suboptimal solution. In a multimodal fitness landscape, this added diversity may enable a GA to avoid convergence to local optima. In the case of non-stationary function optimization problems, the objective is to use a diploid GA to adapt more readily to changing requirements and thus exhibit improved performance over that of the haploid GA. This paper will show analytically and empirically that a diploid GA is capable of maintaining greater population diversity than the haploid GA, and that it is better able to avoid complete convergence than the haploid GA. In addition, empirical tests are performed to demonstrate the effectiveness of a diploid GA in multimodal and non-stationary environments.

Table of Contents

List of Figures

Chapter 1

Introduction

1. Background

Genetic Algorithms (GAs) are search and optimization procedures based on the mechanics of natural selection and genetics. Working with an encoding of a problem's parameter set, GAs search from a random initial population of points. Using fitness-biased selection, the best individuals (or solutions to a problem) are chosen to pass all or some of their genetic information on to a new generation. Stochastic operators analogous to biological crossover and mutation are then used to create offspring from the selected individuals. The resulting offspring become part of a new generation, which, once a specified maximum population size is reached, replaces the previous generation. As simulated evolution proceeds, the average fitness of the population is likely to increase from one generation to the next as better solutions to the problem are discovered. The entire procedure, (selection, crossover, and mutation), continues until some stopping criterion is met. The cannonical GA is described in Goldberg [6] and Mitchell [16], and the infinite population model—an idealized mathematical model used to study the properties of the cannonical GA—is described in Vose [19] as well as Vose [20].

2. Motivation

According to Hunter [9], in order to be effective, search techniques such as GAs require two types of activity: exploration and exploitation. In exploration, the algorithm should traverse different regions of search-space, looking for promising areas. In exploitation, a known good region should be examined to find its best point. A purely random search is good at exploration, but it does not perform exploitation. A purely hillclimbing technique, on the other hand, is good at exploitation, but does little exploration. The two types of activity are contradictory, and a search algorithm must find a good tradeoff between them. In practice, GAs are typically much more effective at exploitation than they are at exploration. Granted, they start with a random population, which means that many points in search-space are initially explored. However, as selection takes effect, the genes of a few relatively highly fit (but possibly suboptimal) individuals may rapidly come to dominate the population. Once the population loses its diversity and begins to converge, it is extremely difficult to re-enter the exploration mode. Crossover of almost identical chromosomes produces little in the way of new genetic material. Thus, new and innovative solutions are no longer being sought out to any great extent. Only mutation remains to explore new search-space, and this performs an unsatisfactorily slow random search.

This situation has become known as the problem of *premature convergence*. As examples, consider each of the following scenarios: In the optimization of a multimodal function, the population may converge to a local, suboptimal point without ever locating the global optimum. In the optimization of a non-stationary function, (i.e. one which varies over time), the population may sufficiently converge so that alleles are lost at many loci. When the objective function changes, it is unlikely that the algorithm will be able to introduce alleles necessary to achieve the new optimum.

In terms of a GA, this can be expressed as a particular bit of a binary string becoming essentially fixed. However, it is precisely in these examples and other complicated domains that GAs have the versatility to be applied and the potential to outperform other specialized search techniques such as hillclimbing and gradient methods.

Attempts to combat premature convergence have centered around modifying the selection operator by remapping raw fitness values. As listed in Beasley [1], they include fitness scaling (or compression), fitness windowing, and fitness ranking. While each of these techniques may avoid convergence to a local maximum, they may also incur unwanted side effects, the most common of which is over-compression. In over-compression, the presence of just one "super-fit" individual can cause a flattening-out of the fitness function where the rest of the population is densely clustered about a single value once the fitness scale is compressed. With a finite population, if the fitness function is too flat, an accumulation of stochastic errors termed *genetic drift* may dictate the trajectory of the population. The rate of genetic drift provides a lower-bound on the rate at which a finite population GA can converge to a correct solution. As a result, the fitness function must contain a gradient that supersedes genetic drift. Researchers have found that overcompression not only leads to slower performance, but, if it occurs to an extent that genetic drift is allowed to dominate, may actually lead the population away from a maximum. Unfortunately, the degree of over-compression may be dictated by a single, extreme individual, either the fittest or the worst. Thus, unless the remapped fitness values are evenly distributed, these techniques will break down.

3. Objective

The purpose of this paper is to propose the study of a novel method for maintaining population diversity and thus avoiding premature convergence in finite population GAs. In the case of stationary optimization problems, it is important that this be done without adversely affecting the algorithm's overall performance. In the case of non-stationary optimization functions, the proposed method should not only increase diversity but also exhibit improved performance over that of the cannonical GA.

An explanation of the terminology used herein is warranted:

- A given string is commonly referred to as an individual's *chromosome.*
- A position in a string is called a *locus.*
- The entity at a locus is called a *gene.*
- The possible values of each gene are called *alleles.*
- The complete collection of chromosomes is termed an individual's *genome.*
- The particular set of genes contained in a genome is called a *genotype.*
- The external manifestation or behavior pattern specified by a genotype is called a *phenotype.*
- A *dominant* allele is expressed in the phenotype when paired with some other allele.
- A *recessive* allele is NOT expressed in the phenotype when paired with a dominant allele.

Most GAs are based on a single-stranded haploid chromosome. In this simple model, a single-stranded string contains all of the problem-related information in a binary encoding. However, most of the haploid organisms in the natural world tend to be rather uncomplicated lifeforms. Most organisms rely on a diploid chromosome, which consists of one or more pairs of homologous chromosomes, each containing information for the same functions. When chromosome pairs have different values

(or alleles) at the same locus, dominance usually resolves the conflict by allowing the dominant allele to take precedence over the recessive allele. Although this seems redundant, there are distinct advantages to a diploid scheme.

One of the advantages of diploidy is that it allows a wider diversity of alleles to be kept in the population over time. Currently harmful, but potentially useful genetic information can be maintained in a recessive position, shielded by the dominance operator. In addition, when the dominance operator is allowed to evolve, it has been hypothesized that this scheme can be used to infuse a form of "long-term distributed memory" into the GA by permitting old solutions to be carried along, (but not expressed), and rapidly reinstated if it becomes desirable in the context of the current environment to do so. Biological studies such as Fisher's [3] have indicated that dominance evolves in diploid and polyploid plant and animal species, giving them the ability to adapt more readily to changing environments. The intriguing implication is that a dominance shift can produce a rapid change in an organism's phenotype not possible through simple mutation. Applied to GAs, this could provide a mechanism for enhancing exploration or, in the case of a non-stationary problem, reintroducing once useful alleles that have again become useful.

4. Previous Work

Surprisingly, there have been only a small number of studies applying diploidy and dominance to GAs. In 1971, Hollstien [8] introduced a triallelic diploid scheme with an evolving dominance map to represent diploidy and dominance in artificial genetic search. His simulations maintained better population diversity (as measured by population variance) than a haploid scheme, but he used a test bed consisting entirely of stationary functions and found no overall improvements in performance.

In 1987, Goldberg and Smith [5] compared the performance characteristics of Hollstien's triallelic scheme with those of a fixed (1-dominates-0) dominance map and a simple haploid scheme. More importantly, they applied an oscillating 0-1 knapsack (non-stationary function optimization) problem to each of these schemes. However, they were interested only in improving performance, and they did not record population diversity statistics in their simulations. Their experimental results showed that both diploid schemes were better able to satisfy the changing requirements of a non-stationary environment than was the haploid scheme. Furthermore, the evolving dominance map was better able to respond to changing optima than was the fixed one. Because they used an oscillating constraint function that reverted back to previous states, Goldberg and Smith claimed to have induced a form of long-term distributed memory into the GA with very little computational overhead. In other words, the redundant memory of diploidy allowed old solutions to be stored as recessive alleles and recovered again when the dominance operator shifted.

A more recent paper by Ng and Wong [18] examines and repeats the experiments of Goldberg and Smith, bringing into question some of the conclusions from the 1987 paper and introducing a different diploid scheme along with a unique dominance change mechanism. They conduct experiments which demonstrate that their novel diploid scheme is able to achieve greater diversity than both a haploid scheme and the triallelic scheme used by Hollstien, Goldberg, and Smith. In tests that apply the oscillating 0-1 knapsack function, their results indicate that if the mutation rate is kept sufficiently low, ($\mu < 0.05$), their scheme also outperforms the others when responding to changes in the functional constraints. They point out that by changing the oscillation frequency, the population size, and the mutation rate, the haploid scheme is actually able to outperform the triallelic diploid scheme when given the proper parameters. This is a caution to anyone using a finite population GA to sup-

port conclusions—results may represent only an isolated case generated by a specific range of parameters.

The disparity between the results of Ng and Wong and those of Goldberg and Smith may stem from the fact that their analyses are based in population genetics and schema theory respectively. Whereas Ng and Wong use an infinite population viability model to compute allele recursions, Goldberg and Smith compute a recursion for the proportion of recessive alleles based on a schema growth equation. While it is debatable whether an infinite population model is superior to the schema theorem for the purposes of analysis, it is true that the theory of population genetics generally assumes an infinite population.

Despite differing viewpoints, both of the aforementioned studies agree that the idea of applying diploidy and dominance to genetic search appears to hold promise. Moreover, we should remember that these concepts have their origin in the biological realm, and there are numerous related studies, as well as a large body of analytical work concerning the mathematics of genetics. In the following chapters, the advantages of diploidy over haploidy are presented in a more formal, mathematical context. Several models based on those used by population geneticists are analyzed both from a theoretical standpoint and for their worth in application to GAs. Finally, empirical tests are used as a supplement to support and visualize the results of the analysis.

Chapter 2

Why Diploidy?

1. A Diploid Viability Model

To see how diploidy differs from haploidy, it is useful to compare their respective viability models.

For the diploid case, population geneticists such as Hartl and Clark [7] have presented a simple viability selection model that conveniently explores selection-based behavior of a population despite the many complexities introduced by fitness. The model makes the following assumptions:

1. a diploid organism
2. non-overlapping generations
3. infinite population size
4. viability selection only
5. random mating
6. no mutation

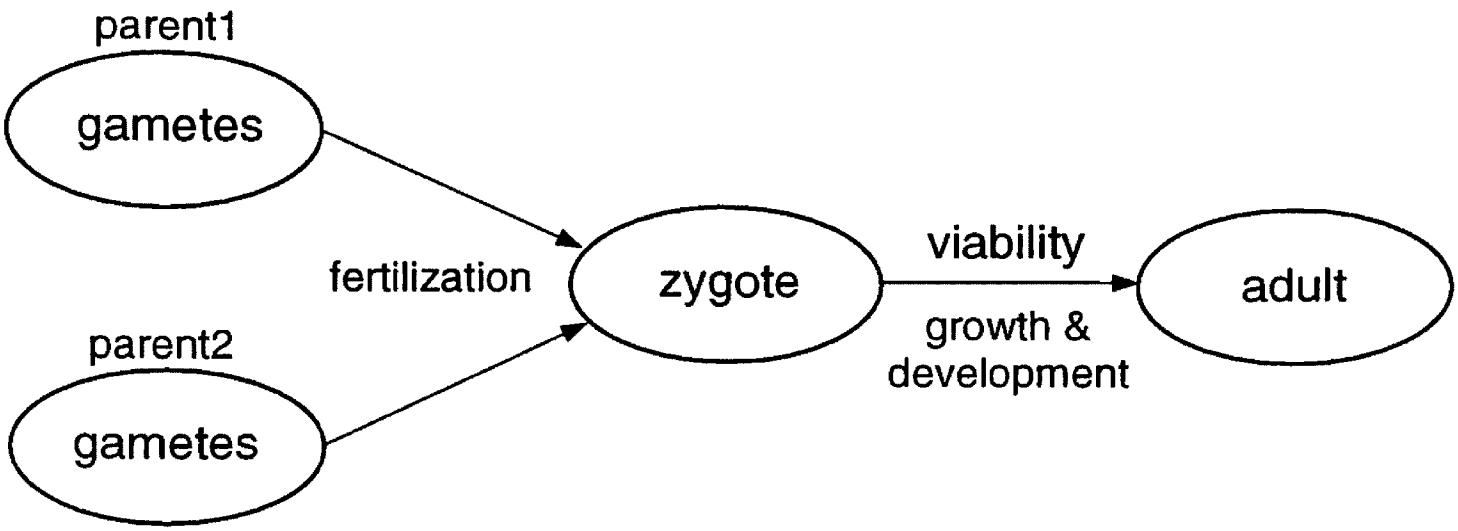


Figure 2.1: A generalized diploid lifecycle

It is informative to describe the steps of the model in terms of the stages in the lifecycle of a diploid organism. We begin with the gamete phase, a biological example of which is sperm or egg. This is a haploid phase, because the gametes each contain only half of the genetic information of a diploid individual. The remaining phases are all diploid phases, and they are much more conspicuous and are of greater duration than the gamete phase. Upon fertilization, we reach the zygote phase. The organism then undergoes growth and development to reach the adult phase. It is during the transition from zygote to adult that proportional selection acts, based on the differential viabilities of the genotypes. The stages of the diploid lifecycle are summarized pictorially in Figure 2.1.

Perhaps the simplest example is the one-locus, two-allele viability selection model. Let 0 and 1 denote the alleles. Let x denote the frequency of 0, $1 - x$ the frequency of 1. The random mating assumption gives x^2 as the frequency of the zygote 00,

$2x(1-x)$ as the frequency of 01, and $(1-x)^2$ as the frequency of 11. Note that a genotype of 10 is equivalent to 01. Let the relative fitnesses (or viabilities) of 00, 01, and 11 be f_{00}, f_{01}, and f_{11} respectively, so that the zygotes survive in the ratio f_{00}:f_{01}:f_{11}. The resulting ratio of 00:01:11 among adults is

$$f_{00}x^2 : 2f_{01}x(1-x) : f_{11}(1-x)^2.$$

The sum of these terms represents the *average fitness* of the population and is denoted by

$$\bar{f} = f_{00}x^2 + 2f_{01}x(1-x) + f_{11}(1-x)^2.$$

To obtain the gametic frequencies for the next generation, each of the terms in the above ratio must be normalized so that the frequencies sum to 1. This is accomplished by dividing by the average fitness. Thus, the frequency x' of the gamete 0 in the next generation is given by

$$x' = \frac{f_{00}x^2 + f_{01}x(1-x)}{\bar{f}} \tag{2.1}$$

Note that the coefficient 2 associated with 01 frequencies until this point has been lost, because 01 heterozygotes produce half 0 and half 1 gametes due to Mendelian segregation.

Another useful relation is the change in allele frequency in one generation, $\Delta x = x' - x$ or $\Delta x = \frac{f_{00}x^2 + f_{01}x(1-x)}{\bar{f}} - x$. With some algebraic manipulation, this can be expressed in a more convenient form:

$$\Delta x = \frac{x(1-x)[x(f_{00} - f_{01}) + (1-x)(f_{01} - f_{11})]}{\bar{f}} \tag{2.2}$$

There are four cases to consider, based on the assignment of the fitnesses.

case 1: $f_{00} > f_{01} > f_{11}$

Examining equation 2.2 above, it is evident that Δx is positive, since $f_{00} - f_{01} > 0$,

$f_{01} - f_{11} > 0$, and the allele frequencies and $\bar{f}$ must always be nonnegative. This implies that $x \rightarrow 1$.

case 2: $f_{00} < f_{01} < f_{11}$

This case is analagous to case 1, except that $f_{00} - f_{01} < 0$, $f_{01} - f_{11} < 0$ and Δx is now negative, implying that $x \rightarrow 0$. Cases 1 and 2 are said to exhibit *directional selection*, since at equilibrium $\hat{x} = 1$ and $\hat{x} = 0$ respectively. These fixed points are of little interest, however, since in each case one of the alleles has been completely eliminated.

case 3: $f_{00} < f_{01} > f_{11}$

When the heterozygote fitness is superior to that of both of the homozygote fitnesses, we have a condition known as *overdominance.* Here, there is a third equilibrium in addition to $\hat{x} = 1$ and $\hat{x} = 0$, because $x(f_{00} - f_{01}) + (1 - x)(f_{01} - f_{11})$ can equal 0 for some value of x. Because this third equilibrium point is of some interest, the overdominant case is given more thorough treatment below.

case 4: $f_{00} > f_{01} < f_{11}$

When the heterozygote fitness is inferior to that of both of the homozygote fitnesses, we have a condition known as *underdominance.* Again, there is a third equilibrium point, and the equation for $\hat{x}$ is identical to that derived below for the overdominant case. However, the resulting equilibrium for this case is unstable, so that even if the value of x is close to $\hat{x}$, it diverges away from the polymorphic equilibrium point to a value of either 0 or 1. Furthermore, the trajectories of the allele frequencies, and hence their final values, are dependent upon their initial values.

Based on examination of the above cases, case 3 seems worthy of further treatment. It is well known that for the overdominant case, there exists a polymorphic equilibrium and that this point is globally stable. This means that regardless of the initial allele frquencies, the system will always converge to the equilibrium point. The

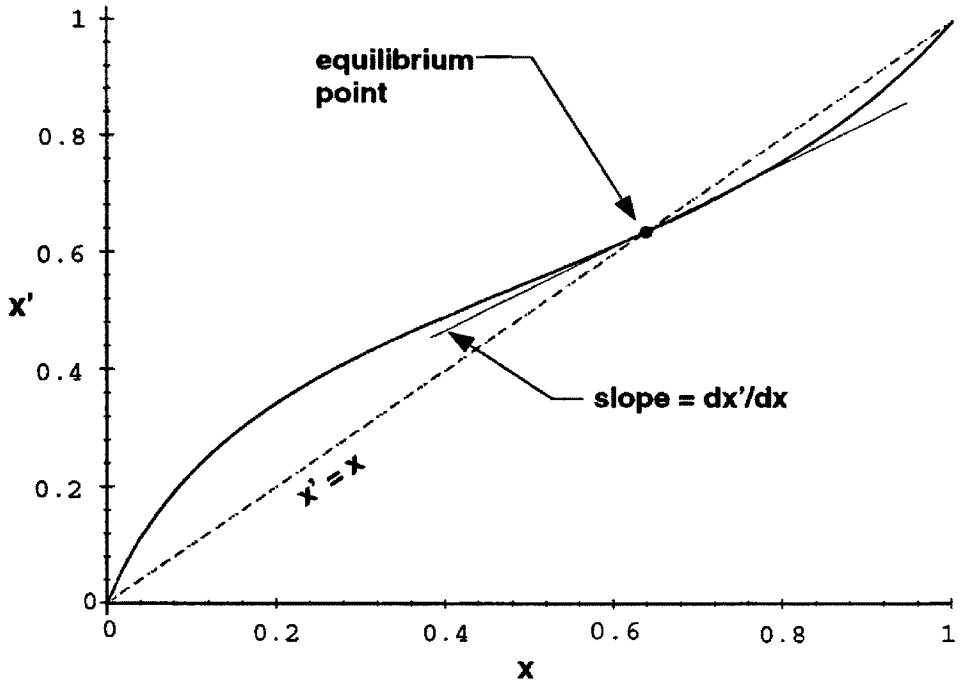


Figure 2.2: A plot of x' vs. x for the diploid model

equilibrium point is "polymorphic", because there is some non-zero fraction of each allele (i.e. the fixed point lies within the interval $(0,1)$). Figure 2.2 is a plot of x' versus x for fitness values of $f_{00} = 0.6$, $f_{01} = 1$, and $f_{11} = 0.3$.

For a formal proof, the reader is referred to Nagylaki [17]. However, local asymptotic stability of a fixed point can be determined based on the condition $\frac{dx'}{dx}|_{x'=x} < 1 \Rightarrow$ an asymptotically stable fixed point. The fixed point itself can be derived in terms of the fitnesses by setting $x' = x$ in equation 1 and solving for x to get

$$\hat{x} = \frac{f_{11} - f_{01}}{f_{00} - 2f_{01} + f_{11}}.$$

Computing the derivative of equation 2.1 with respect to x and evaluating it at the fixed point gives

$$\left.\frac{dx'}{dx}\right|_{x'=\hat{x}} = \frac{f_{00}f_{01} - 2f_{00}f_{11} + f_{01}f_{11}}{f_{01}^2 - f_{00}f_{11}}$$

Without loss of biological generality, it is convenient to let $f_{00} = 1 - r, f_{01} = 1$, and $f_{11} = 1 - s$ with $0 < r, s \leq 1$. This gives

$$\left.\frac{dx'}{dx}\right|_{x'=\hat{x}} = \frac{r + s - 2rs}{r + s - rs} < 1$$

since $2rs > rs$.

2. A Haploid Viability Model

To contrast this with the haploid case, a single step of the single-locus Simple Genetic Algorithm as described in Vose [19] with zero mutation (and no crossover) is outlined. Let the initial population vector be $\mathbf{x} = [x\ (1-x)]^T$, and the fitness vector be $[f_0\ f_1]^T$. Begin by performing a proportional selection step according to the fitness function defined in [19]. This yields

$$\mathcal{F}(\mathbf{x}) = \begin{bmatrix} \frac{f_0 x}{f_0 x + f_1(1-x)} \\ \frac{f_1(1-x)}{f_0 x + f_1(1-x)} \end{bmatrix}$$

Next, this vector is subjected to the recombination function $\mathcal{M}$. This gives

$$\mathbf{x}' = \mathcal{M}(\mathcal{F}(\mathbf{x})) = \begin{bmatrix} \frac{f_0^2 x^2 + f_0 f_1 x(1-x)}{[f_0 x + f_1(1-x)]^2} \\ \frac{f_1^2(1-x)^2 + f_0 f_1 x(1-x)}{[f_0 x + f_1(1-x)]^2} \end{bmatrix}$$

The next generation frequency x' of 0 is

$$x' = \frac{f_0 x[f_0 x + f_1(1-x)]}{\bar{f}} = \frac{f_0 x}{f_0 x + f_1(1-x)}, \tag{2.3}$$

where $\bar{f} = [f_0 x + f_1(1-x)]^2$.

It is not hard to see that the recurrence in equation 2.3 can only have fixed points at 0 and 1. A plot of x' versus x for the haploid model is shown is Figure 2.3 using fitness values of $f_0 = 0.8$ and $f_1 = 0.2$.

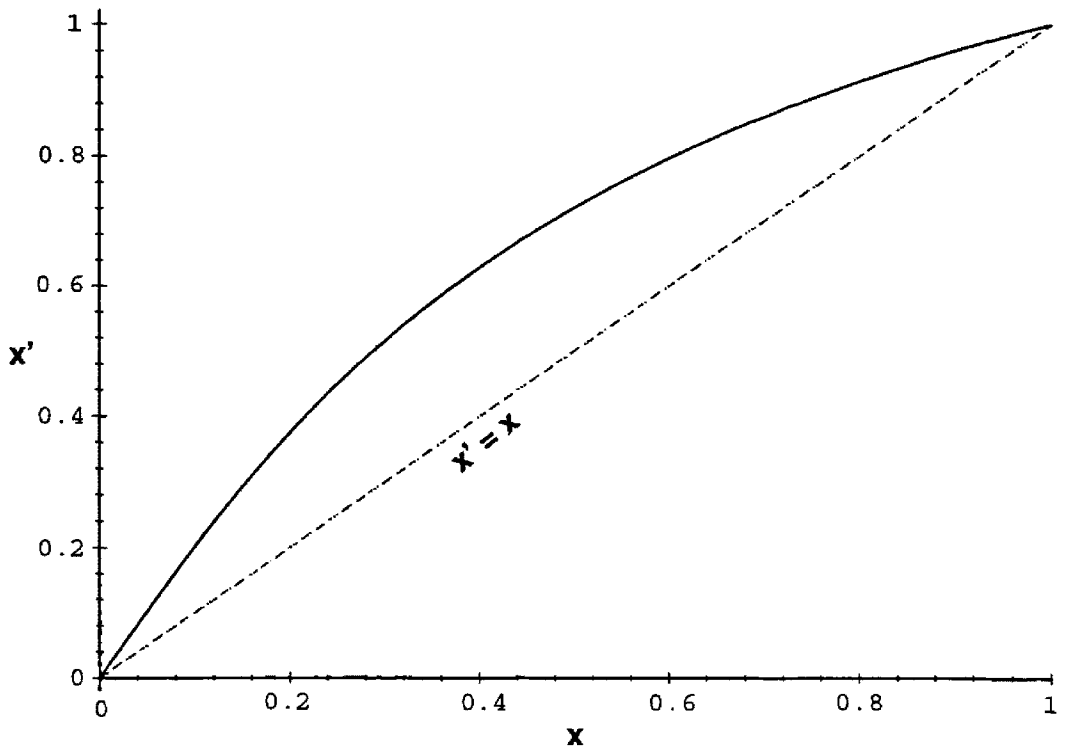


Figure 2.3: A plot of x' vs. x for the haploid model

3. Conclusions

Mathematically, the recurrence for the diploid model (equation 2.1) is the ratio of degree 2 polynomials, whereas the recurrence for the haploid model (equation 2.3) is the ratio of linear polynomials, which gives the diploid model inherently greater complexity. Biologically, the overdominant polymorphism of the diploid model is one of the basic mechanisms for maintaining genetic diversity in a population, and it has no analogue in the haploid model.

Chapter 3

The Dominance Operator

1. The Function of Dominance

To illustrate how the dominance operator works, consider a diploid chromosomal structure where different letters represent different alleles:

$$AAaa$$

$$aAAa$$

Here, there are two alleles, or two possible values that a gene may take on at a given locus, namely A or a. By convention, an uppercase letter is used to denote a dominant allele, while a lowercase letter denotes a recessive allele. In nature, if a given locus contains a gene for say, eye color, then the A allele might represent brown eyes, while the a allele might represent blue eyes. Although nature sometimes allows hybrids or intermediate forms, we will not allow that possibility. We make the restriction that the phenotype cannot have both brown and blue eyes. Hence, there is a pair of genes describing a given function, and the potential exists for conflict. The dominance operator resolves this conflict by allowing one allele (the dominant allele) to take precedence over the other allele (the recessive allele) at that locus. When there are more than two alleles, more than one allele may play a dominant

role, depending on the allele with which it is paired, and situations may arise when an allele is dominant when paired with one allele, but recessive when paired with another. The action of the dominance operator can, at least in part, be defined in terms of observable phenomena. An allele is dominant if it is expressed (i.e. it is apparent in the phenotype) when paired with an identical allele—the homozygous case where $AA \rightarrow A$ —or with a different allele—the heterozygous case where $Aa \rightarrow A$ or $aA \rightarrow A$. An allele is recessive if it is expressed only when paired with an identical allele—the homozygous case where $aa \rightarrow a$. Thus, the chromosome pairs above may be rewritten as:

$$\begin{array}{r} AAaa \\ aAAa \\ \hline AAAa \end{array}$$

This can also be expressed in terms of the following dominance map:

	A	a
A	A	A
a	A	a

In an abstract sense, dominance is a function that maps from genotypes to phenotypes. More importantly, as Goldberg [6] notes, it serves as a form of genotype reduction. This means that the dominance operator can be used in the context of GAs as a means of mapping a diploid chromosome to a haploid chromosome, which in turn can be subjected to a haploid fitness function. In this manner, a diploid GA can be constructed with minimal computational overhead. The fitness function does not have to be completely redefined for a diploid chromosome.

We return to the discussion of diploidy from Chapter 2. Diploidy facilitates population diversity by allowing heterozygotes (individuals with one dominant and one recessive allele at a locus) to exist and reproduce. Heterozygotes not only protect recessive alleles from extinction, they propagate them. It seems reasonable to

follow a biological parallel here and introduce a "heterozygote advantage", i.e. to endow heterozygotes with a higher fitness than homozygotes. In so doing, there is a greater probability of avoiding rapid convergence to a single genotype. This could be simulated by assigning a "fitness bonus" to heterozygotes. As the diploid genome is mapped to a haploid genome, the number of heterozygous loci is recorded. The resulting haploid genome is subjected to the fitness function, just as is done in the cannonical GA. However, for each heterozygous locus, we add the value of the fitness bonus to the value obtained from the haploid fitness function. The size of the bonus is important–a bonus that is too small will not permit overdominance, and a bonus that is too large will actually speed convergence to a population of heterozygotes. In addition, problems may arise with large strings if the resulting fitnesses are not evenly distributed. Continuing with our example, let s represent the fitness bonus and f the haploid fitness function. Then,

$$\begin{array}{c} AAaa \\ aAAa \\ \hline f(AAAa) + 2s \end{array}$$

In seeking guidelines for assigning fitnesses, Chapter 4 examines viability models for multiple allele polymorphisms.

2. Dominance Maps

One of the earliest schemes for incorporating diploidy and dominance in artificial genetic search is due to Hollstien [8]. He began with a two-locus, evolving dominance map. At one locus, 0 and 1 are the allowable alleles. For each of these loci, there is an associated locus, reserved for a modifier gene, at which M and m are the allowable alleles. The 0 alleles are dominant when there is at least one M allele present at the homologous modifier locus. Hollstien assumed that the numerical and modifier loci are

adjacent on the chromosomes and that they are never separated by crossover. Thus, the combinations of alleles—$0M, 0m, 1M$, and $1m$—may be treated as four alleles at a single locus. The sixteen possible genotypes produce the following dominance map:

	$0M$	$0m$	$1M$	$1m$
$0M$	0	0	0	0
$0m$	0	0	0	1
$1M$	0	0	1	1
$1m$	0	1	1	1

Note that there is a greater number of genotypes that produce 0 alleles than those that produce 1 alleles. This requires that measures be taken to counteract this bias, such as giving 0 alleles a slightly higher probability of occurrence, both in the initial population and through mutation.

Hollstien recognized that three alleles are sufficient to achieve the effects of dominance interaction and to provide the capability of dominance shifts through selection. His triallelic scheme used $0, 1$, and 2 as the possible alleles at a locus. As indicated in the dominance map that follows, 0 alleles dominate 1 alleles, (which are always recessive), and 2 alleles, (which play the role of a "dominant 1"), dominate 0 alleles.

	0	1	2
0	0	0	1
1	0	1	1
2	1	1	1

When it is advantageous to have 1 dominate 0, selection can replace 1 alleles by 2 alleles to effect a dominance shift at each locus. Again, note that there is a bias (this time towards 1 alleles) if all three alleles are evenly distributed in the population.

Ng and Wong [18] use a two-allele, two-locus scheme that attempts to remove the bias inherent in Hollstien's. There is no evidence to that they fully succeed in this,

however, because they resolve dominance contention arbitrarily. Curiously, they also prohibit certain heterozygote genotypes by promoting recessive alleles to dominant alleles. More interesting than their dominance map is perhaps their approach to dominance shifts. They use a dominance change mechanism that takes effect on a rapid (i.e. a single generation) rather than an evolutionary time scale. They use the following criteria: if an individual's fitness decreases by more than 20% over a single generation, then a dominance change occurs for that individual wherein dominant alleles are demoted to recessives and recessive alleles are promoted to dominants. In light of this, it is little wonder that their scheme outperformed that of Goldberg and Smith on test problems that involved a rapid change in fitness over a single generation. Although it is doubtful that it has any precedence in nature, their method is consistent with one of the attractive features of GAs—it achieves global performance through local action.

The analysis in Chapter 4 is rooted in and experiments with a dominance map based on Hollstien's original two-locus evolving dominance map. The simplifying assumption is made that it can be treated as a single-locus model with four alleles as justified above. A different symbology is used, and the map is symmetric with respect to ones and zeros in order to eliminate the need to counteract a bias. Dominance contention is resolved to maintain this symmetry. The table below depicts the four-allele dominance map where 0 and 1 are dominant alleles and o and i are the corresponding recessive alleles.

	0	o	1	i
0	0	0	0	0
o	0	0	1	1
1	0	1	1	1
i	0	1	1	1

Chapter 4

Four Alleles at a Single Locus

1. A multiple allele viability model

Chapter 2 presented a viability model for two alleles at a single locus. However, the dominance map from the previous chapter utilizes four alleles —a dominant and recessive 0, and a dominant and recessive 1—at a single locus. Fortunately, biologists have considered the case where an autosomal gene may have more than two alleles segregating the population. (In fact, this occurs quite commonly in nature.) Hartl and Clark [7] describe a generalized model with viability selection operating on a gene with k alleles. The model is reproduced here for the special case where $k = 4$. Let the frequencies of alleles 0, o, 1, and i be p_1, p_2, p_3, and p_4 respectively. The allele frequencies must still sum to 1, i.e.

$$\sum_{i=1}^{4} p_i = 1$$

Arranging the alleles along the rows and columns of a Punnet square gives the possible genotypes and their respective frequencies when random mating is assumed:

	0	o	1	i
0	00 p_1^2	0o p_1p_2	01 p_1p_3	0i p_1p_4
o	o0 p_2p_1	oo p_2^2	o1 p_2p_3	oi p_2p_4
1	10 p_3p_1	1o p_3p_2	11 p_3^2	1i p_3p_4
i	i0 p_4p_1	io p_4p_2	i1 p_4p_3	ii p_4^2

Assuming that there is no distinction between the genotype composed of alleles A_iA_j and that composed of A_jA_i, then there are ten distinct genotypes in the table above. (Hereafter, the convention will be to list 0 and o before 1 and i, and dominant alleles before recessives.) Each heterozygote genotype thus has two entries in the table, so that its corresponding frequency will have a coefficient of 2.

The next step is to assign fitnesses to each genotype. This is most easily depicted as a 4×4 *fitness matrix*

$$W = \begin{bmatrix} w_{11} & w_{12} & w_{13} & w_{14} \\ w_{21} & w_{22} & w_{23} & w_{24} \\ w_{31} & w_{32} & w_{33} & w_{34} \\ w_{41} & w_{42} & w_{43} & w_{44} \end{bmatrix} \tag{4.1}$$

where each entry w_{ij} corresponds to the genotype composed of alleles A_i and A_j. Note that this matrix can be simplified into an upper (or lower) triangular matrix, since $w_{ij} = w_{ji}$. In deriving the recursion for the allele frequencies in the next generation, it is helpful to set up a table that summarizes the information presented thus far:

genotype	frequency	fitness	gametes produced			
			0	o	1	i
00	p_1^2	w_{11}	1	0	0	0
0o	$2p_1p_2$	w_{12}	$\frac{1}{2}$	$\frac{1}{2}$	0	0
01	$2p_1p_3$	w_{13}	$\frac{1}{2}$	0	$\frac{1}{2}$	0
0i	$2p_1p_4$	w_{14}	$\frac{1}{2}$	0	0	$\frac{1}{2}$
oo	p_2^2	w_{22}	0	1	0	0
o1	$2p_2p_3$	w_{23}	0	$\frac{1}{2}$	$\frac{1}{2}$	0
oi	$2p_2p_4$	w_{24}	0	$\frac{1}{2}$	0	$\frac{1}{2}$
11	p_3^2	w_{33}	0	0	1	0
1i	$2p_3p_4$	w_{34}	0	0	$\frac{1}{2}$	$\frac{1}{2}$
ii	p_4^2	w_{44}	0	0	0	1

In general, p_i' is derived by computing *frequency* $\times$ *fitness* $\times$ *gametes produced* for each row and summing these products for the appropriate column. For p_1', this gives $w_{11}p_1^2 + w_{12}p_1p_2 + w_{13}p_1p_3 + w_{14}p_1p_4$. This can be generalized as $p_i \sum_j w_{ij}p_j$. The summation is commonly referred to as the *marginal fitness* of an allele, and is denoted as $w_i = \sum_j w_{ij}p_j$. As before, the allele frequency must be normalized to 1 by dividing by the sum of all the allele frequencies (or the average fitness of the population), which is once again labelled $\bar{w}$. For multiple alleles, this is expressed as $\bar{w} = \sum_i \sum_j w_{ij}p_ip_j$. Thus, the general expression for the allele frequencies in the next generation is

$$p_i' = \frac{p_i w_i}{\bar{w}} \tag{4.2}$$

At equilibrium, this becomes $\hat{p}_i = \frac{\hat{p}_i \hat{w}_i}{\hat{\bar{w}}}$. Since we desire a polymorphic equilibrium such that each allele is present in some fraction—this is commonly referred to as a *complete* polymorphism —we introduce the stipulation that $0 < \hat{p}_i < 1 \ \forall i$. Now it is possible to divide by $\hat{p}_i$ and rearrange to get

$$\hat{w}_i = \hat{\bar{w}} \text{ for } i = 1, 2, 3, 4$$

which means that all of the marginal fitnesses are equal when the population comes to equilibrium. This can be rewritten as $\hat{w}_i - \hat{w}_1 = 0$ for $i = 2, 3, 4$. By adding the condition that $\sum_i p_i = 1$, we have a system of four linear equations in four unknowns, which can readily be solved. Nagylaki [17] provides an elegant method for computing the allele frequencies at equilibrium based on techniques in linear algebra. It is quite amenable to implementation in a mathematical software package or a programming language. Nagylaki makes use of the following identity, which can be found in Lancaster and Tismenetsky [14]:

$$adj(W)W = det(W)I$$

where $adj(W)$ denotes the adjoint of the fitness matrix W, which is defined to be the transposed matrix of cofactors of W, $det(W)$ is the determinant of W, and I is the identity matrix. The equilibrium equation $\hat{w}_i = \hat{\bar{w}} = \sum_j w_{ij}\hat{p}_j = w_i$ is expressed in vector form as

$$W\hat{\mathbf{p}} = \hat{\bar{w}}\mathbf{1}$$

where $\mathbf{1}$ is the 4×1 column vector of ones. Multiplying this by $adj(W)$ and using the identity yields

$$det(W)\hat{\mathbf{p}} = \hat{\bar{w}}(adj(W))\mathbf{1} \tag{4.3}$$

In order to break this into its components, denote the ith component of the vector $adj(W)\mathbf{1}$ as V_i. We now have

$$det(W)\hat{p}_i = \hat{\bar{w}}V_i$$

If $det(W) \neq 0, V_i \neq 0\ \forall i$, and all the V_i have the same sign, then there exists a unique internal equilibrium

$$\hat{p}_i = \frac{\hat{\bar{w}}V_i}{det(W)} = \frac{V_i}{\sum_j V_j} \tag{4.4}$$

where the substitution $det(W) = \hat{\bar{w}} \sum_j V_j$ has been made by post-multiplying both sides of equation 4.3 by $\mathbf{1}^{\mathbf{T}}$, the 1×4 row vector of ones, and recalling that $\mathbf{1}^{\mathbf{T}}\hat{\mathbf{p}} = \mathbf{1}$. Note that this also serves to normalize the equilibrium allele frequencies.

In summary, not only does equation 4.4 give a terse form for the allele frequencies at equilibrium solely in terms of the fitness matrix, but it also provides criteria for their admissibility, (i.e. $0 < \hat{p}_i < 1 \ \forall i$), in terms of the values in the fitness matrix, namely

1. $det(W) \neq 0$
2. $V_i \neq 0 \ \forall i$
3. $sgn(V_i) = sgn(V_j) \ \forall i, j$

Although equation 4.4 gives admissibility criteria based on the fitness matrix as a whole, it does *not* provide heuristics for assigning individual fitness values *a priori*. The question remains, how does one assign these fitnesses in order to guarantee a complete polymorphism that is biologically admissible? For the case of two alleles at a single locus, overdominance was a sufficient condition for a globally stable polymorphic equilibrium. One possible means of extending this condition to the four-allele case would be to assign the fitnesses such that

$$w_{ii} < w_{ij} > w_{jj}$$

Here, each heterozygote is more fit than the homozygote for either of its constituent alleles. Unfortunately, this condition is neither necessary nor sufficient to guarantee a complete polymorphism when extended to more than two alleles. Lewontin, et. al. [15] were able to derive conditions *necessary* for a complete polymorphism based on a triallelic model, but these cannot be readily applied to a four-allele model. Most distressing, however, are the results of their experiments that examined the

probability that multiple allele polymorphisms could be maintained by random choice of viabilities:

1. Only a very small portion of the parameter space admits a stable polymorphism
2. As the number of alleles is increased, the probability of a stable polymorphism decreases dramatically
3. For genes with seven alleles, even if all heterozygotes have higher fitnesses than the respective homozygotes, only 0.1% of the randomly chosen viabilities admit a stable polymorphism

Based on this information, a purely random search was ruled out. An attempt was made at a "narrowed" random search that reduced the parameter space by extending Lewontin's triallelic conditions and adding a small "fitness bonus" to heterozygotes such as 0o and a "double bonus" to double heterozygotes such as 0i. Specifically, viabilities had to satisfy the condition

$$w_{ij} > (w_{ii} + w_{jj})/2$$

The search procedure—implemented in Maple—uncovered two possible fitness matrices that yielded complete polymorphisms with allele frequencies within the admissible range at equilibrium. One was extremely sensitive to and highly dependent on the value of the fitness bonus, while the other was not.

At this point, the stability of the equilibrium point has to be addressed. If the complete polymorphic equilibrium exists and is biologically admissible, then a result of Kingman [13] provides a method for determining its stability. For a gene with k alleles, if the fitness matrix W has j positive eigenvalues, then at most $k-j+1$ alleles will exist with positive frequencies at equilibrium. Stated slightly differently, a unique admissible solution to equation 4.4 will be globally stable *if and only if* W has exactly

one positive eigenvalue and at least one negative eigenvalue. In such a case, the system moves, for any initial frequency point for which each p_i is positive, to this equilibrium. If the equilibrium is inadmissible or unstable, then the system evolves in such a way that one or more alleles becomes eliminated and the complete polymorphism is lost. Perhaps the simplest example to demonstrate Kingman's theory is the $k \times k$ fitness matrix where all heterozygotes have fitness 1 and all homozygotes have fitness $1 - s$, where $0 < s < 1$, i.e.

$$W = \begin{bmatrix} 1-s & 1 & 1 & \dots & 1 \\ 1 & 1-s & 1 & \dots & 1 \\ 1 & 1 & 1-s & \dots & 1 \\ \vdots & \vdots & \vdots & & \\ 1 & 1 & 1 & \dots & 1-s \end{bmatrix}$$

The adjoint of W is

$$adj(W) = \begin{bmatrix} (k-1)s^{k-2} - s^{k-1} & -s^{k-2} & -s^{k-2} & \dots & -s^{k-2} \\ -s^{k-2} & (k-1)s^{k-2} - s^{k-1} & -s^{k-2} & \dots & -s^{k-2} \\ -s^{k-2} & -s^{k-2} & (k-1)s^{k-2} - s^{k-1} & \dots & -s^{k-2} \\ \vdots & \vdots & \vdots & & \\ -s^{k-2} & -s^{k-2} & -s^{k-2} & \dots & (k-1)s^{k-2} - s^{k-1} \end{bmatrix}$$

Summing the elements along any row, we have $V_i = -s^{k-1}\ \forall i$. Since there are k rows, $\sum_j V_j = -ks^{k-1}$. Substituting these values into equation 4.4 gives

$$\hat{p}_i = \frac{-s^{k-1}}{-ks^{k-1}} = \frac{1}{k}\ \forall i$$

With the assumption that $k > 1$, the equilibrium is clearly admissible. The eigenvalues of W are $(k - s), -s, -s, \dots, -s$, and thus the stability conditions are met. However, this configuration assigns a fitness penalty to homozygotes, which may incur unwanted side effects such as negative or zero fitness values after repeated applications of the recursion.

Kingman's criteria provides a convenient alternative to the method used for the two allele model, which in this case would involve computing the partial differential of the allele frequency recursion—equation 4.2—for each allele and evaluating it at the equilibrium point determined by equation 4.4. It should also be noted that it is quite logical for the stability of the equilibrium to be dependent solely upon the fitness matrix and completely independent of the allele frequencies.

2. Mapping haploid fitnesses to a diploid fitness matrix

Recall that the objective is to map haploid fitnesses to a diploid fitness matrix, while introducing a heterozygote advantage by a assigning a small fitness bonus to the heterozygote genotypes. The aforementioned search revealed a fitness matrix that accomplished this objective and yielded an equilibrium that was both admissible and stable according to Kingman's criteria. The matrix is based on a pair of haploid fitnesses, f_0 and f_1, and a fitness bonus s. Using the fitness matrix W of equation 4.1, we substitute actual fitness values for each entry w_{ij} as follows:

$$W = \begin{bmatrix} f_0 & f_0 + s & max + s & max + s \\ f_0 + s & f_0 & max + s & max + s \\ max + s & max + s & f_1 & f_1 + s \\ max + s & max + s & f_1 + s & f_1 \end{bmatrix} \tag{4.5}$$

where $max = maximum(f_0, f_1)$ and $s < f_i$ for $i = 0, 1$. Here, the homozygote genotypes appear along the diagonal and have the smallest fitness values. The "single" heterozygotes are next in fitness ranking, and the "double" heterozygotes have the highest possible fitness values.

Proposition: The fitness matrix in equation 4.5 yields an admissible and globally stable complete polymorphic equilibrium for all possible values of f_0, f_1, and s subject to the constraint $s < f_i$ for $i = 0, 1$.

Empirical testing—a short computer program that chose random values for the parameters f_0, f_1, and s with the objective of finding a set of parameters for which the equilibrium point was either inadmissible or unstable—failed to find a counterexample. An analytical proof thus seems warranted.

Proof: We start with the fitness matrix W from equation 4.5. A case-by-case analysis based on the value of max is required.

Case 1: $max = f_0$

The initial assumptions are that $f_0 > f_1 > s > 0$. For the admissibility of the equilibrium point, we need to show that $0 < \hat{p}_i < 1 \ \forall i$. Using Nagylaki's method as described earlier, the allele frequencies at equilibrium are

$$\hat{p}_1 = \hat{p}_2 = \frac{s}{4(f_0 - f_1 + s)}$$

$$\hat{p}_3 = \hat{p}_4 = \frac{2f_0 - 2f_1 + s}{4(f_0 - f_1 + s)}$$

Clearly, since $f_0 > f_1 > s > 0$, the numerator and denominator of each equation is positive. This implies that $\hat{p}_i > 0$. We also have that $s < 4(f_0 - f_1) + 4s$ and that $2(f_0 - f_1) + s < 4(f_0 - f_1) + 4s$, which implies that $\hat{p}_i < 1$. Thus, $0 < \hat{p}_i < 1 \ \forall i$ and the equilibrium point is admissible.

It can be shown that the eigenvalues of W in equation 4.4 are $-s, -s, f_0 + f_1 + s + \sqrt{5f_0^2 + f_1^2 - 2f_0f_1 + 8f_0s + 4s^2}$, and $f_0 + f_1 + s - \sqrt{5f_0^2 + f_1^2 - 2f_0f_1 + 8f_0s + 4s^2}$. We need to show that exactly one of these eigenvalues is positive and the others are all negative. (This is actually stricter than Kingman's criteria, which requires that *at least* one of the eigenvalues be negative, because we are only interested in a *complete* polymorphic equilibrium.)

1. Since $s > 0$, $-s < 0$.

2. $f_0 + f_1 + s + \sqrt{5f_0^2 + f_1^2 - 2f_0f_1 + 8f_0s + 4s^2}$

$> f_0 + f_1 + s + \sqrt{5f_0^2 + f_1^2 - 2f_0f_1}$, since $f_0 > 0$ and $s > 0$.

$> f_0 + f_1 + s + \sqrt{f_0^2 + f_1^2 - 2f_0f_1}$

$= f_0 + f_1 + s + \sqrt{(f_0 - f_1)^2}$

$= f_0 + f_1 + s + f_0 - f_1$

$= 2f_0 + s$

> 0, since $f_0 > 0$ and $s > 0$.

3. $f_0 + f_1 + s - \sqrt{5f_0^2 + f_1^2 - 2f_0f_1 + 8f_0s + 4s^2}$

$< f_0 + f_1 + s - \sqrt{5f_0^2 + f_1^2 - 2f_0^2 + 8f_0s + 4s^2}$, since $f_0 > f_1$.

$= f_0 + f_1 + s - \sqrt{3f_0^2 + f_1^2 + (4f_0s + 4f_0s) + 4s^2}$

$< f_0 + f_1 + s - \sqrt{(f_0^2 + 2f_0^2) + f_1^2 + 4f_0s + 4f_1s + 4s^2}$, since $f_1 < f_0$.

$< f_0 + f_1 + s - \sqrt{f_0^2 + 2f_0f_1 + f_1^2 + 4f_0s + 4f_1s + 4s^2}$, since $f_1 < f_0$.

$< f_0 + f_1 + s - \sqrt{f_0^2 + 2f_0f_1 + f_1^2 + 2f_0s + 2f_1s + s^2}$

$= f_0 + f_1 + s - \sqrt{(f_0 + f_1 + s)^2}$

$= f_0 + f_1 + s - (f_0 + f_1 + s)$

$= 0$

Case 2: $max = f_1$

Observing the allele frequencies at equilibrium,

$$\hat{p_1} = \hat{p_2} = \frac{s}{4(f_1 - f_0 + s)}$$

$$\hat{p_3} = \hat{p_4} = \frac{2f_1 - 2f_0 + s}{4(f_1 - f_0 + s)}$$

and the eigenvalues of the fitness matrix, $-s, -s, f_1+f_0+s+\sqrt{5f_1^2 + f_0^2 - 2f_1f_0 + 8f_1s + 4s^2}$, and $f_1 + f_0 + s - \sqrt{5f_1^2 + f_0^2 - 2f_1f_0 + 8f_1s + 4s^2}$, we see that this case is symmetric to case 1.

Therefore, we have three negative eigenvalues and one positive eigenvalue, which satisfies Kingman's criteria for stability. □

3. Remarks

The results of the previous section prompt a number of questions:

1. How does one map the haploid gene frequencies x_0 and x_1 to the diploid allele frequencies p_1, p_2, p_3, and p_4?

2. If we start with the initial conditions that $p_1 = p_2$ and $p_3 = p_4$, will this system maintain these equalities?

3. If so, can this model with four alleles be equated to a simpler model that uses only two alleles?

4. Is this model consistent with the dominance map in Chapter 3?

5. Can this model be extended to multiple loci?

We first address the problem of mapping the haploid gene frequencies, x_0 and x_1 to the four diploid allele frequencies, p_1, p_2, p_3, and p_4. Referring to the summary table in the first section of this chapter and the dominance map in Chapter 3, we see that genotypes 00, 0o, 01, 0i, and oo map to 0 and genotypes o1, oi, 11, 1i, and ii map to 1. Summing the frequencies of each of these genotypes and setting this equal to either x_0 or x_1 as appropriate yields

$$x_0 = p_1^2 + 2p_1p_2 + 2p_1p_3 + 2p_1p_4 + p_2^2$$

$$x_1 = 2p_2p_3 + 2p_2p_4 + p_3^2 + 2p_3p_4 + p_4^2$$

Note that

$$x_0 + x_1 = (p_1 + p_2 + p_3 + p_4)^2 = 1^2 = 1$$

We now proceed to analyze the four allele system and attempt to simplify it.

Proposition: If the system from the previous section is initialized with $p_1 = p_2$ and $p_3 = p_4$, then the iterates of the system will maintain these equalities.

Proof: Let the initial allele frequencies be such that $p_1 = p_2$ and $p_3 = p_4$. We also have that $p_1 + p_2 + p_3 + p_4 = 1$. This can now be expressed as $2p_1 + 2p_3 = 1$, so that we can solve for each of the allele frequencies in terms of p_1. That is, $p_2 = p_1$ and $p_3 = p_4 = \frac{1}{2} - p_1$. We substitute these values into the allele frequecy update equations that result when equation 4.2 is expanded for each allele. The w_{ij}s are expressed in terms of the fitnesses in the matrix of equation 4.5.

$$p_1' = \frac{w_{11}p_1^2 + w_{12}p_1p_2 + w_{13}p_1p_3 + w_{14}p_1p_4}{\bar{w}} = \frac{(w_{11} + w_{12})p_1^2 + (w_{13} + w_{14})p_1(\frac{1}{2} - p_1)}{\bar{w}}$$

$$= \frac{(2f_0 - 2max - s)p_1^2 + (max + s)p_1}{\bar{w}}$$

$$p_2' = \frac{w_{21}p_2p_1 + w_{22}p_2^2 + w_{23}p_2p_3 + w_{24}p_2p_4}{\bar{w}} = \frac{(w_{21} + w_{22})p_1^2 + (w_{23} + w_{24})p_1(\frac{1}{2} - p_1)}{\bar{w}}$$

$$= \frac{(2f_0 - 2max - s)p_1^2 + (max + s)p_1}{\bar{w}} = p_1'$$

$$p_3' = \frac{w_{31}p_3p_1 + w_{32}p_3p_2 + w_{33}p_3^2 + w_{34}p_3p_4}{\bar{w}} = \frac{(w_{31} + w_{32})p_1(\frac{1}{2} - p_1) + (w_{33} + w_{34})(\frac{1}{2} - p_1)^2}{\bar{w}}$$

$$= \frac{(2f_1 - 2max - s)p_1^2 + (max - 2f_1)p_1 + \frac{1}{4}(2f_1 + s)}{\bar{w}}$$

$$p_4' = \frac{w_{41}p_4p_1 + w_{42}p_4p_2 + w_{43}p_4p_3 + w_{44}p_4^2}{\bar{w}} = \frac{(w_{41} + w_{42})p_1(\frac{1}{2} - p_1) + (w_{43} + w_{44})(\frac{1}{2} - p_1)^2}{\bar{w}}$$

$$= \frac{(2f_1 - 2max - s)p_1^2 + (max - 2f_1)p_1 + \frac{1}{4}(2f_1 + s)}{\bar{w}} = p_3'$$

It follows that if $p_1 = p_2$ and $p_3 = p_4$, then $p_1' = p_2'$ and $p_3' = p_4'$. □

An important result of this proof is that it is possible to express all four allele frequencies in terms of one allele frequency. This allows the expression of x_0 and x_1 in terms of a single allele, which in turn permits a simple mapping from x_0 and x_1 to p_1, p_2, p_3, and p_4, namely $p_1 = p_2 = \frac{1}{2}x_0$ and $p_3 = p_4 = \frac{1}{2}x_1$.

We now proceed to equate the system of four alleles to one with two alleles, q_1 and q_2, by setting $q_1 = 2p_1$ and $q_2 = 2p_3$. This gives

$$q_1' = 2p_1' = \frac{(f_0 - max - \frac{1}{2}s)4p_1^2 + (max + s)2p_1}{\bar{w}} = \frac{(f_0 - max - \frac{1}{2}s)q_1^2 + (max + s)q_1}{\bar{w}}$$

$$q_2' = 2p_3' = \frac{(w_{31} + w_{32})(\frac{1}{2} - p_3)p_3 + (w_{33} + w_{34})p_3^2}{\bar{w}} = \frac{(f_1 - max - \frac{1}{2}s)4p_3^2 + (max + s)2p_3}{\bar{w}}$$

$$= \frac{(f_1 - max - \frac{1}{2}s)q_2^2 + (max + s)q_2}{\bar{w}}$$

It can readily be verified that this two allele system is derived from the following 2×2 fitness matrix:

$$W = \begin{bmatrix} f_0 + \frac{1}{2}s & max + s \\ max + s & f_1 + \frac{1}{2}s \end{bmatrix} \tag{4.6}$$

Furthermore, empirical tests showed that the two allele model represented by the fitness matrix in equation 4.6 and the four allele model represented by the fitness matrix in equation 4.5 exhibited identical behavior.

Tests of particular interest were those that compared the rate of convergence for a haploid model with that of the diploid model. For the haploid case, the implementation details followed those of Vose [19] for the infinite population model. The diploid case was implemented based on the four allele model described above. A plot of the 0-bit convergence over a time scale of 250 generations is shown in Figure 4, where the initial values of x_0 and x_1 are 0.01 and 0.99 respectively, and the fitness values are $f_0 = 1.00$ and $f_1 = 0.90$. The three curves representing the diploid model correspond

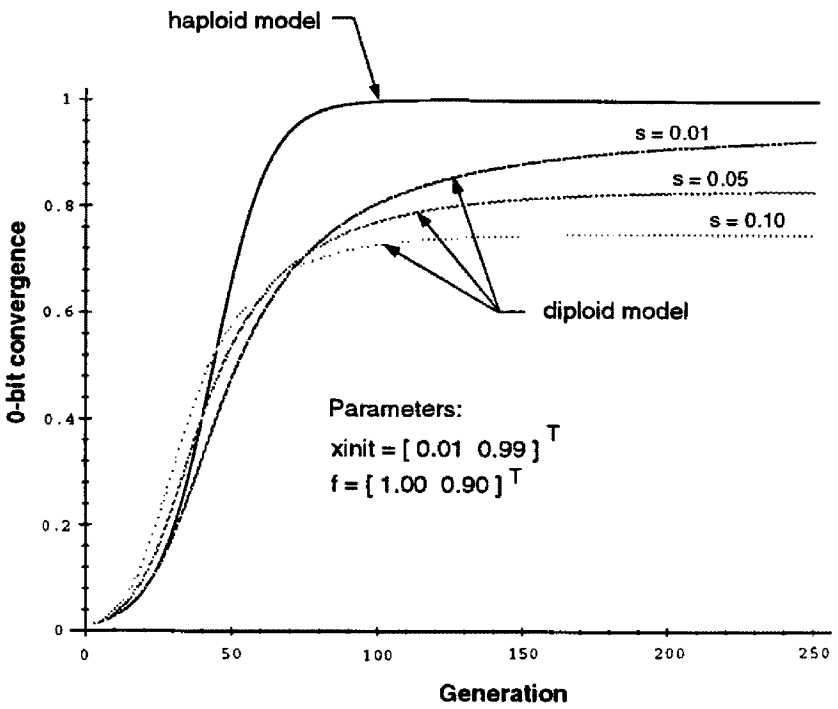


Figure 4.1: A comparison of convergence rates

to three different values of s, the heterozygote fitness bonus. Observe that s can be used to control the rate and the asymptotic value of convergence, and that for any value of s greater than 0, the diploid curve lies below the haploid curve. The case for 1-bit convergence is symmetric, and the curves of Figure 4.1 can be duplicated by interchanging the fitnesses and the initial values of x_0 and x_1. From the figure, it is evident that the diploid model is capable of both slowing the rate of convergence (to a homozygote genotype) and avoiding complete convergence (by forming a stable polymorphism).

There arises a problem, however, when we attempt to reconcile the four-allele fitness matrix with the dominance map from Chapter 3. These two entities are superimposed in the table below. Let w_{ij} and d_{ij} denote the i,jth entries of the fitness

matrix W and the dominance map, respectively. Comparing the fitness matrix with the dominance map for consistency, we should see that the subscript on the fitness of w_{ij} corresponds to the value in d_{ij}.

	0	o	1	i
0	f_0 0	$f_0 + s$ 0	$max + s$ 0	$max + s$ 0
o	$f_0 + s$ 0	f_0 0	$max + s$ 1	$max + s$ 1
1	$max + s$ 0	$max + s$ 1	f_1 1	$f_1 + s$ 1
i	$max + s$ 0	$max + s$ 1	$f_1 + s$ 1	f_1 1

In the case of the eight $max + s$ entries in W, max must evaluate to either f_0 or f_1. It is clear, though, that some of the corresponding entries of the dominance map contain a 0, while others contain a 1. For example, $w_{13} = w_{23} = max + s$, but $d_{13} = 0$ and $d_{23} = 1$. The dominance map could be altered so that $d_{13} = d_{14} = d_{23} = d_{24} = d_{31} = d_{32} = d_{41} = d_{42}$, but this would create an imbalance heavily favoring either 0 or 1. Furthermore, it is not possible to know in advance whether max will evaluate to f_0 or f_1. This problem is inherent when using max in the fitness matrix, and there is no fixed-value dominance map that can be used consistently with it.

Consistency between the fitness matrix and the dominance map becomes an important issue when the single-locus diploid model is extended to mulitple loci, i.e. bit strings of arbitrary length. With the current model, the entries of the dominance map that correspond to the $max + s$ entries of the fitness matrix may be resolved consistently only after max has been evaluated. As defined previously, the computation of max relies on having knowledge of f_0 and f_1 *at a particular locus.* This does

not present a problem in a single-locus or 1-bit GA, but for arbitrary string lengths and fitness functions, this information is not available, since in traditional GA*s* the fitnesses typically correspond to entire strings, not to particular locations within a string.

4. Conclusions

Although the four allele model presented in this chapter is not extendible to multiple loci and therefore not applicable to GA*s* in general, it does provide some insight into the assignment of fitnesses in order to achieve overdominance, the convergence characteristics of diploid models relative to the haploid GA, and the number of alleles required to effect the desired behavior.

Specifically, two alleles are sufficient to bring about the desired improvement in the convergence characteristics. The assignment of the heterozygote fitnesses is critical to achieving overdominance. To take a two-allele example, let w_{11}, w_{12}, w_{21}, and w_{22} be the fitnesses of genotypes $00, 01, 10$, and 11 respectively. Note that $w_{12} = w_{21}$. Choose $w_{11} = f_0, w_{12} = w_{21} = f_0 + s$, and $w_{11} = f_1$. This will allow overdominance if $f_0 > f_1$, but if $f_1 > f_0 + s$, we have directional selection with $w_{22} > w_{12} = w_{21} > w_{11}$ and a globally stable polymorphic equilibrium is not possible. We could assign the fitness bonus so that $s > |f_0 - f_1|$, but this requires knowledge of f_0 and f_1 at a single locus, and it may result in inordinately large values for s. Because of symmetry, $w_{12} = w_{21} = f_1 + s$ suffers from the same problems. We might try some combination of f_0 and f_1, e.g. $w_{12} = w_{21} = 0.5f_0 + 0.5f_1 + s$. The entries in the dominance map for 01 and 10 could be assigned based on the outcome of a "coin flip" for each entry. In other words, with probability 0.5 we assign a value of 0 to an entry, and with probability 0.5 we assign a value of 1. However, this looks beyond a more

fundamental problem—it is still possible that, for instance, $f_0 > 0.5f_0 + 0.5f_1 + s$ if $f_0 > f_1 + 2s$, so we cannot achieve overdominance for arbitrary values of f_0 and f_1. Thus, using *max* in the fitness matrix creates irresolvable conflicts in the dominance map. Using some combination of f_0 and f_1 can be resolved in the dominance map with non-deterministic entries, but up to this point, no combination has been presented that can guarantee overdominance for all possible values of f_0 and f_1.

Chapter 5

A Scheme With Varying Heterozygote Fitness

1. Explanation

Building on the results of Chapter 4 and the concept of a dominance map with non-deterministic entries, we present a scheme that uses two alleles, 0 and 1, and a heterozygote fitness that varies over time. Instead of a fixed combination of f_0 and f_1, we use the allele frequencies to determine the relative contributions of f_0 and f_1 to the fitness of the heterozygote genotypes. In the fitness matrix below, p is the frequency of allele 0, and q is the frequency of allele 1. Note that since $p + q = 1$, we have $q = 1 - p$. Thus, once p has been assigned, q is fixed. As before, s represents a small additive fitness bonus.

$$W = \begin{bmatrix} f_0 & f_0p + f_1(1-p) + s \\ f_0p + f_1(1-p) + s & f_1 \end{bmatrix} \tag{5.1}$$

Recall that w_{ij} refers to the i,jth entry of W and that w_{11}, w_{12}, w_{21}, and w_{22} are the fitnesses of zygotes $00, 01, 10$, and 11 respectively, where $w_{12} = w_{21}$. The corresponding dominance map would look like the following:

	0	1
0	0	0 : p 1 : 1-p
1	0 : p 1 : 1-p	1

Here, the genotypes 01 and 10 map to 0 with probability p and 1 with probability $1-p$, where p is defined as the frequency of allele 0 at the locus under consideration in the current generation.

Using the matrix of equation 5.1, we can derive an equation for the allele frequencies in the next generation in the same manner that equation 2.1 was derived. This gives

$$p' = \frac{f_0 p^2 + [f_0 p + f_1(1-p) + s]p(1-p)}{f_0 p^2 + 2(f_0 p + f_1(1-p) + s)p(1-p) + f_1(1-p)^2} \tag{5.2}$$

For arbitrary initial values of f_0, f_1, and p, it is quite possible that the system of equation 5.2 will initially exhibit directional selection. For example, take the case where $p = q = 0.5$ and $f_0 > f_1 + 2s$. This gives $w_{12} = w_{21} = 0.5f_0 + 0.5f_1 + s < 0.5f_0 + 0.5(f_0 - 2s) + s = f_0 = w_{11}$ and $w_{12} = w_{21} = 0.5f_0 + 0.5f_1 + s > 0.5(f_1 + 2s) + 0.5f_1 + s = f_1 + 2s > f_1 = w_{22}$. Recall from Chapter 2 that this situation represents directional selection, where the allele frequencies will approach a limit based on the differential fitnesses. That is, for $w_{11} > w_{12} = w_{21} > w_{22}, p \rightarrow 1$ and $q \rightarrow 0$. However, although the system will begin to converge toward $p = 1$, it will become overdominant before it actually reaches $p = 1$ and eliminates all 1 alleles. To see why this is true, let $w_{11} = f_0 > w_{22} = f_1$ and let $p = 1 - \epsilon$. Suppose that for sufficiently small ϵ, it is the case that $w_{12} = w_{21} < w_{11}$. Substituting $p = 1 - \epsilon$ into the matrix of equation 5.1,

$$w_{12} = w_{21} = (1-\epsilon)f_0 + \epsilon f_1 + s$$

$$= f_0 + (f_1 - f_0)\epsilon + s$$
$$> f_0 = w_{11} \text{ if } \epsilon < \frac{s}{f_0 - f_1} > 0$$

This contradicts the assumption that $w_{12} = w_{21} < w_{11}$ for sufficiently small ϵ. We see that for $\epsilon < \frac{s}{f_0 - f_1}$, $w_{12} = w_{21} > w_{11} > w_{22}$, and we conclude that the system of equation 5.2 is overdominant. A similar argument can be made for initial values where $w_{22} = f_1 > w_{11} = f_0$.

2. Analysis

In a sense, we have added a feedback mechanism which adjusts the heterozygote fitness until it produces overdominance, regardless of the initial allele frequencies and the fitnesses. This comes at some expense, though, as equation 5.2 is a ratio of degree 3 polynomials, and the analysis becomes significantly more difficult. Moreover, because the entries of the fitness matrix are no longer all constant, we cannot apply the methods of Nagylaki and Kingman to solve for the fixed point and determine its stability. Consequently, the methods used in Chapter 2 for two alleles will be used again here. By setting $p' = p$ in equation 5.2 and solving for p, we can derive the fixed point in terms of the fitnesses. This procedure yields four solutions:

$$\hat{p}_1 = 0$$

$$\hat{p}_2 = 1$$

$$\hat{p}_3 = \frac{f_0 - f_1 - s + \sqrt{f_0^2 - 2f_0f_1 + f_1^2 + s^2}}{2(f_0 - f_1)}$$

$$\hat{p}_4 = \frac{f_0 - f_1 - s - \sqrt{f_0^2 - 2f_0f_1 + f_1^2 + s^2}}{2(f_0 - f_1)}$$

At this point, we need to show that exactly one of these solutions gives a fixed point in the open interval (0,1). Clearly, $\hat{p}_1 = 0$ and $\hat{p}_2 = 1$ do not lie within (0,1), so they

can be eliminated. We wish to show that $\hat{p}_4$ can also be eliminated.

To show: $\hat{p}_4 \leq 0$ or $\hat{p}_4 \geq 1$

Case 1: $f_0 > f_1$

$$\begin{aligned}
\hat{p}_4 &= \frac{f_0 - f_1 - s - \sqrt{f_0^2 - 2f_0f_1 + f_1^2 + s^2}}{2(f_0 - f_1)} \\
&= \frac{f_0 - f_1 - s - \sqrt{(f_0 - f_1)^2 + s^2}}{2(f_0 - f_1)} \\
&< \frac{f_0 - f_1 - s - \sqrt{(f_0 - f_1)^2}}{2(f_0 - f_1)} \quad \text{since} \quad s^2 > 0 \\
&= \frac{f_0 - f_1 - s - (f_0 - f_1)}{2(f_0 - f_1)} \\
&= \frac{-s}{2(f_0 - f_1)} \\
&< 0 \quad \text{since} \quad f_0 > f_1 \quad \text{and} \quad s > 0
\end{aligned}$$

Case 2: $f_1 > f_0$

$$\begin{aligned}
\hat{p}_4 &= \frac{f_0 - f_1 - s - \sqrt{f_0^2 - 2f_0f_1 + f_1^2 + s^2}}{2(f_0 - f_1)} \\
&> \frac{f_0 - f_1 - s - \sqrt{f_1^2 - 2f_0f_1 + f_0^2 + 2(f_1 - f_0)s + s^2}}{2(f_0 - f_1)} \quad \text{since} \quad 2(f_1 - f_0)s > 0 \\
&= \frac{f_0 - f_1 - s - \sqrt{(f_1 - f_0 + s)^2}}{2(f_0 - f_1)} \\
&= \frac{(f_0 - f_1) - s - (f_1 - f_0 + s)}{2(f_0 - f_1)} \\
&= \frac{2(f_0 - f_1) - 2s}{2(f_0 - f_1)} \\
&= 1 + \frac{s}{f_1 - f_0} \\
&> 1 \quad \text{since} f_1 > f_0 \quad \text{and} s > 0
\end{aligned}$$

Therefore, $\hat{p}_4 < 0$ for $f_0 > f_1$ and $\hat{p}_4 > 1$ for $f_1 > f_0$, so $\hat{p}_4$ is not a biologically valid equilibrium point.

It remains to be shown that $\hat{p}_3$ falls within (0,1) and is biologically valid.

To show: $0 < \hat{p}_3 < 1$

Case 1: $f_0 > f_1$

We begin by showing that $\hat{p}_3 > 0$.

$$\begin{aligned}
\hat{p}_3 &= \frac{f_0 - f_1 - s + \sqrt{f_0^2 - 2f_0f_1 + f_1^2 + s^2}}{2(f_0 - f_1)} \\
&= \frac{f_0 - f_1 - s + \sqrt{(f_0 - f_1)^2 + s^2}}{2(f_0 - f_1)} \\
&> \frac{f_0 - f_1 - s + \sqrt{s^2}}{2(f_0 - f_1)} \quad \text{since} \quad (f_0 - f_1)^2 > 0 \\
&= \frac{f_0 - f_1 - s + s}{2(f_0 - f_1)} \\
&= \frac{f_0 - f_1}{2(f_0 - f_1)} \\
&= 1/2 \\
&> 0
\end{aligned}$$

Now, we show that $\hat{p}_3 < 1$.

$$\begin{aligned}
\hat{p}_3 &= \frac{f_0 - f_1 - s + \sqrt{f_0^2 - 2f_0f_1 + f_1^2 + s^2}}{2(f_0 - f_1)} \\
&< \frac{f_0 - f_1 - s + \sqrt{f_0^2 - 2f_0f_1 + f_1^2 + 2(f_0 - f_1)s + s^2}}{2(f_0 - f_1)} \quad \text{since} \quad 2(f_0 - f_1)s > 0 \\
&= \frac{f_0 - f_1 - s + \sqrt{(f_0 - f_1 + s)^2}}{2(f_0 - f_1)} \\
&= \frac{f_0 - f_1 - s + (f_0 - f_1 + s)}{2(f_0 - f_1)} \\
&= \frac{2(f_0 - f_1)}{2(f_0 - f_1)} \\
&= 1
\end{aligned}$$

Case 2: $f_1 > f_0$

The steps from the first part of Case 1 may be duplicated to show that $\hat{p}_3 > 0$ for

$f_1 > f_0$.

The following demonstrates that $\hat{p_1} < 1$.

$$\begin{aligned}
\hat{p_3} &= \frac{f_0 - f_1 - s + \sqrt{f_0^2 - 2f_0f_1 + f_1^2 + s^2}}{2(f_0 - f_1)} \\
&= \frac{f_1 - f_0 + s - \sqrt{f_0^2 - 2f_0f_1 + f_1^2 + s^2}}{2(f_1 - f_0)} \\
&< \frac{f_1 - f_0 + s - \sqrt{f_0^2 - 2f_0f_1 + f_1^2 + 2(f_0 - f_1)s + s^2}}{2(f_1 - f_0)} \quad since\ 2(f_0 - f_1)s < 0 \\
&= \frac{f_1 - f_0 + s - \sqrt{(f_0 - f_1 + s)^2}}{2(f_1 - f_0)} \\
&= \frac{f_1 - f_0 + s - (f_0 - f_1 + s)}{2(f_1 - f_0)} \\
&= \frac{2(f_1 - f_0)}{2(f_1 - f_0)} \\
&= 1
\end{aligned}$$

Therefore, $0 < \hat{p_3} < 1$ for both $f_0 > f_1$ and $f_1 > f_0$, so $\hat{p_3}$ is a biologically valid equilibrium point within (0,1). For notational convenience, we let $\hat{p} = \hat{p_3}$ so that we have

$$\hat{p} = \frac{f_0 - f_1 - s + \sqrt{f_0^2 - 2f_0f_1 + f_1^2 + s^2}}{2(f_0 - f_1)}$$

Without loss of generality, we can assume that f_0 and f_1 differ by a multiplicative factor, say $2f$, so that $f_0 = 1 + f$ and $f_1 = 1 - f$ and the above equilibrium can be rewritten as

$$\hat{p} = \frac{2f - s + \sqrt{4f^2 + s^2}}{4f} \tag{5.3}$$

We proceed to determine the stability of the equilibrium point. As in Chapter 2, we take the first derivative of the allele recursion (equation 5.2) and evaluate it at the equilibrium point. For local stability, this must yield a value less than 1.

To show: $\frac{dp'}{dp}|_{p=\hat{p}} < 1$

Method: Since the quantity is a quotient, we show that the numerator is less than

the denominator by matching common terms.

$$\begin{aligned}\left.\frac{dp'}{dp}\right|_{p=\hat{p}} &= 16f^2(16f_0^2 + 8f^2\sqrt{4f^2+s^2} - 4f^2s\sqrt{4f^2+s^2} + 16f^4 + 2(4f^2+s^2)^{3/2} \\ &\quad +2s^4 + 12f^2s^2 + 2s^2\sqrt{4f^2+s^2} - 16f^2s - 4s^3 - s^3\sqrt{4f^2+s^2} \\ &\quad -s(4f^2+s^2)^{3/2}) \;/\; (16f^2 + 12f^2\sqrt{4f^2+s^2} - (4f^2+s^2)^{3/2} + s^2\sqrt{4f^2+s^2})^2\end{aligned}$$

Expanding the denominator and eliminating common terms, we need to show that

$$\begin{aligned}&16f^2\sqrt{4f^2+s^2} - 8f^2s^2 - 4(4f^2+s^2)^{3/2} + 4f^2s\sqrt{4f^2+s^2} + s(4f^2+s^2)^{3/2} \\ &+s^3\sqrt{4f^2+s^2} - 2s^4 + 16f^2s + 4s^3 \\ >\ & 0\end{aligned}$$

We will need to assume that $s < f$ in order to manipulate the inequality further. Expressing $(4f^2+s^2)^{3/2}$ as $(4f^2+s^2)\sqrt{4f^2+s^2}$,

$$\begin{aligned}&16f^2\sqrt{4f^2+s^2} - 8f^2s^2 - 4(4f^2+s^2)\sqrt{4f^2+s^2} + 4f^2s\sqrt{4f^2+s^2} \\ &+s(4f^2+s^2)\sqrt{4f^2+s^2} + s^3\sqrt{4f^2+s^2} - 2s^4 + 16f^2s + 4s^3 \\ =\ & 16f^2\sqrt{4f^2+s^2} - 8f^2s^2 - 16f^2\sqrt{4f^2+s^2} - 4s^2\sqrt{4f^2+s^2} \\ &+8f^2s\sqrt{4f^2+s^2} + 2s^3\sqrt{4f^2+s^2} - 2s^4 + 16f^2s + 4s^3 \\ >\ & -8f^2s^2 - 4s^2\sqrt{4f^2+s^2} + 8f^2s(2f) + 2s^3(2f) - 2s^4 + 16f^2s + 4s^3 \\ & \text{since } \sqrt{4f^2+s^2} > 2f \\ =\ & -8f^2s^2 - 4s^2\sqrt{4f^2+s^2} + 16f^3s + 4fs^3 - 2s^4 + 16f^2s + 4s^3 \\ >\ & -8f^2s^2 - 4s^2\sqrt{4f^2+s^2} + 16f^2s^2 + 4s^4 - 2s^4 + 16f^2s + 4s^3 \\ & \text{since } f > s \\ =\ & -4s^2\sqrt{4f^2+s^2} + 8f^2s^2 + 2s^4 + 16f^2s + 4s^3 \\ >\ & -4s^2(2f+s) + 8f^2s^2 + 2s^4 + 16f^2s + 4s^3 \text{ since } \sqrt{4f^2+s^2} < 2f+s \\ =\ & -8fs^2 - 4s^3 + 8f^2s^2 + 2s^4 + 16f^2s + 4s^3\end{aligned}$$

$$
\begin{aligned}
&> -8fs^2 + 8f^2s^2 + 2s^4 + 16fs^2 \quad \text{since} \quad f > s \\
&= 8f^2s^2 + 2s^4 + 8fs^2 \\
&> 0 \quad \text{since} \quad f > 0
\end{aligned}
$$

Since the numerator is less than the denominator, we have shown that

$$\left.\frac{dp'}{dp}\right|_{p=\hat{p}} < 1$$

provided that $s < f = \frac{1}{2}|f_0 - f_1|$.

In order to prove that the internal equilibrium point is *globally* stable, we need to show that the system defined by equation 5.2 satisfies two additional criteria:

1. $\frac{dp'}{dp} > 0$ for $0 < p < 1$
2. $\Delta p > 0$ for $0 < p < \hat{p}$

   $\Delta p < 0$ for $\hat{p} < p < 1$

To show: $\frac{dp'}{dp} > 0$ for $0 < p < 1$

$$
\begin{aligned}
\frac{dp'}{dp} = {} & (f_0^2p^4 - 2f_0f_1p^4 + f_1^2p^4 + 4f_0f_1p^3 - 4f_1^2p^3 + f_0sp^2 + 6f_1^2p^2 - 6f_0f_1p^2 \\
& + f_1sp^2 + 4f_0f_1p - 4f_1^2p - 2f_1sp + f_1s + f_1^2) \; / \\
& (2f_0p^3 - 2f_1p^3 - 3f_0p^2 + 3f_1p^2 + 2sp^2 - 2sp - f_1)^2 \qquad (5.4)
\end{aligned}
$$

Clearly, the quantity in the denominator is greater than 0. We proceed to evaluate the numerator. Grouping the terms of the numerator,

$$(f_0-f_1)^2p^4+4(f_0f_1-f_1^2)p^3+(f_0+f_1)sp^2-6(f_0f_1-f_1^2)p^2+4(f_0f_1-f_1^2)p-2f_1sp+f_1s+f_1^2$$

Case 1: $f_0 > f_1 > s$

$$
\begin{aligned}
&(f_0 - f_1)^2p^4 + 4(f_0f_1 - f_1^2)p^3 + (f_0 + f_1)sp^2 - 6(f_0f_1 - f_1^2)p^2 + 4(f_0f_1 - f_1^2)p \\
&-2f_1sp + f_1s + f_1^2
\end{aligned}
$$

$$= (f_0 - f_1)^2p^4 + (f_0f_1 - f_1^2)(4p^3 - 6p^2 + 4p) + (f_0 + f_1)sp^2 - 2f_1sp + f_1s + f_1^2$$

$$> (f_0 - f_1)^2p^4 + (f_0f_1 - f_1^2)(4p^3 - 6p^2 + 4p) + (f_0 + f_1)sp^2 - 2f_1sp + f_1sp + f_1^2$$

since $p < 1$

$$> (f_0 - f_1)^2p^4 + (f_0f_1 - f_1^2)(4p^3 - 6p^2 + 4p) + (f_0 + f_1)sp^2 - 2f_1sp + f_1sp + f_1s$$

since $f_1 > s$

$$> (f_0 - f_1)^2p^4 + (f_0f_1 - f_1^2)(4p^3 - 6p^2 + 4p) + (f_0 + f_1)sp^2 - 2f_1sp + f_1sp + f_1sp$$

since $p < 1$

$$= (f_0 - f_1)^2p^4 + (f_0f_1 - f_1^2)(4p^3 - 6p^2 + 4p) + (f_0 + f_1)sp^2$$

$$> (f_0f_1 - f_1^2)(4p^3 - 6p^2 + 4p) + (f_0 + f_1)sp^2 \text{ since } (f_0 - f_1)^2p^4 > 0$$

$$> (f_0f_1 - f_1^2)(4p^3 - 6p^2 + 4p) \text{ since } (f_0 + f_1)sp^2 > 0$$

$$> 0 \text{ since } f_0f_1 - f_1^2 > 0 \text{ and } 4p^3 - 6p^2 + 4p > 0$$

We can elaborate further on the latter quantity by stating that $4p^3 - 6p^2 + 4p = p(4p^2 - 6p + 4)$. Our original assumption is that $p > 0$. It can easily be verified that $y = 4x^2 - 6x + 4$ is parabolic with a global minimum at $x = 0.75$, which corresponds to $y = 1.75 > 0$.

Case 2: $f_1 > f_0 > s$

$$(f_0 - f_1)^2p^4 + 4(f_0f_1 - f_1^2)p^3 + (f_0 + f_1)sp^2 - 6(f_0f_1 - f_1^2)p^2 + 4(f_0f_1 - f_1^2)p$$
$$-2f_1sp + f_1s + f_1^2$$

$$= f_0^2p^4 - f_0f_1p^4 + (f_1^2 - f_0f_1)(p^4 - 4p^3 + 6p^2 - 4p) + (f_0 + f_1)sp^2 - 2f_1sp + f_1s + f_1^2$$

$$> f_0^2p^4 - f_0f_1 + (f_1^2 - f_0f_1)(p^4 - 4p^3 + 6p^2 - 4p) + (f_0 + f_1)sp^2 - 2f_1sp + f_1s + f_1^2$$

since $p^4 < 1$

$$= f_0^2p^4 + (f_1^2 - f_0f_1)(p^4 - 4p^3 + 6p^2 - 4p + 1) + (f_0 + f_1)sp^2 - 2f_1sp + f_1s + f_1^2$$

$$> f_0^2p^4 + (f_1^2 - f_0f_1)(p^4 - 4p^3 + 6p^2 - 4p + 1) + (f_0 + f_1)sp^2 - 2f_1sp + f_1sp + f_1^2$$

since $p < 1$

$> f_0^2p^4 + (f_1^2 - f_0f_1)(p^4 - 4p^3 + 6p^2 - 4p + 1) + (f_0 + f_1)sp^2 - 2f_1sp + f_1sp + f_1s$

since $f_1 > s$

$> f_0^2p^4 + (f_1^2 - f_0f_1)(p^4 - 4p^3 + 6p^2 - 4p + 1) + (f_0 + f_1)sp^2 - 2f_1sp + f_1sp + f_1sp$

since $p < 1$

$= f_0^2p^4 + (f_1^2 - f_0f_1)(p^4 - 4p^3 + 6p^2 - 4p + 1) + (f_0 + f_1)sp^2$

$> (f_1^2 - f_0f_1)(p^4 - 4p^3 + 6p^2 - 4p + 1) + (f_0 + f_1)sp^2$ since $f_0^2p^4 > 0$

$> (f_1^2 - f_0f_1)(p^4 - 4p^3 + 6p^2 - 4p + 1)$ since $(f_0 + f_1)sp^2 > 0$

> 0 since $f_1^2 - f_0f_1 > 0$ and $p^4 - 4p^3 + 6p^2 - 4p + 1 = (p-1)^4 > 0$

Note that we have added the additional restriction that $s < min(f_0, f_1)$.

It remains to be shown that $\Delta p > 0$ for $0 < p < \hat{p}$ and $\Delta p < 0$ for $\hat{p} < p < 1$. In determining the equilibrium points, we showed that the curve of p' versus p intersects the line $p' = p$ at exactly three points in the closed interval [0,1], namely at $p = 0, p = \hat{p}$, and $p = 1$. The line $p' = p$, or the diagonal, represents the set of points where $\Delta p = 0$. Thus, points above this line will have $\Delta p > 0$, and points below it will have $\Delta p < 0$. We have shown that $\frac{dp'}{dp} > 0$ in the open interval (0,1). This implies that the curve of p' versus p is strictly increasing within (0,1). Evaluating equation 5.4 at $p = 0$ yields

$$\left.\frac{dp'}{dp}\right|_{p=0} = \frac{f_1 + s}{f_1} > 1$$

This implies that the curve of p' versus p lies above the diagonal for sufficiently small, nonnegative values of p. Since the curve does not cross the diagonal again with increasing values for p until $p = \hat{p}$, we claim that for $0 < p < \hat{p}$, $\Delta p > 0$. At $p = \hat{p}$, the curve passes through the diagonal with slope less than 1, as implied by the earlier result that $\frac{dp'}{dp}|_{p=\hat{p}} < 1$. Hence, the curve of p' versus p is below the diagonal when

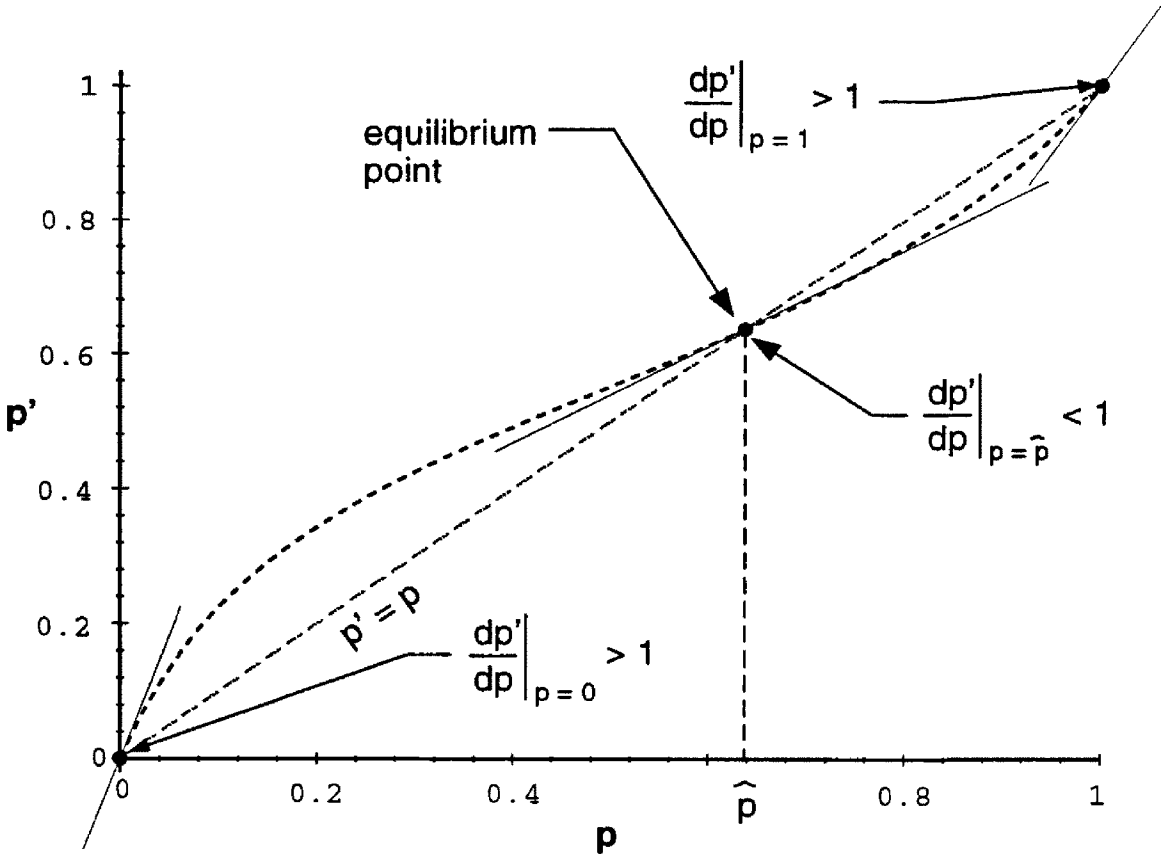


Figure 5.1: A geometric argument for global stability

$p = \hat{p} + \epsilon$ for sufficiently small ϵ. As the value of p increases, the curve does not intersect the diagonal again until $p = 1$, at which point it has slope

$$\left.\frac{dp'}{dp}\right|_{p=1} = \frac{f_0 + s}{f_0} > 1$$

Figure 5.1 depicts this argument graphically. Since the slope of p' versus p is positive in (0,1), we can place an additional bound that its curve does not extend above the line $p' = \hat{p}$ for $0 < p < \hat{p}$ or below this line for $\hat{p} < p < 1$. Thus, the curve of p' versus p must lie within the shaded region of Figure 5.2. For any curve within this region, the iterates of p will staircase into the equilibrium point as depicted in Figure 5.3.

An analytical argument can be made for $\Delta p > 0$ for $0 < p < \hat{p}$ and $\Delta p < 0$ for $\hat{p} < p < 1$. From calculus, (see [4]), the curve of a function f can be described by the formula

$$f(p) = f(a) + f'(a)(p - a) + \frac{1}{2}f''(c)(p - a)^2$$

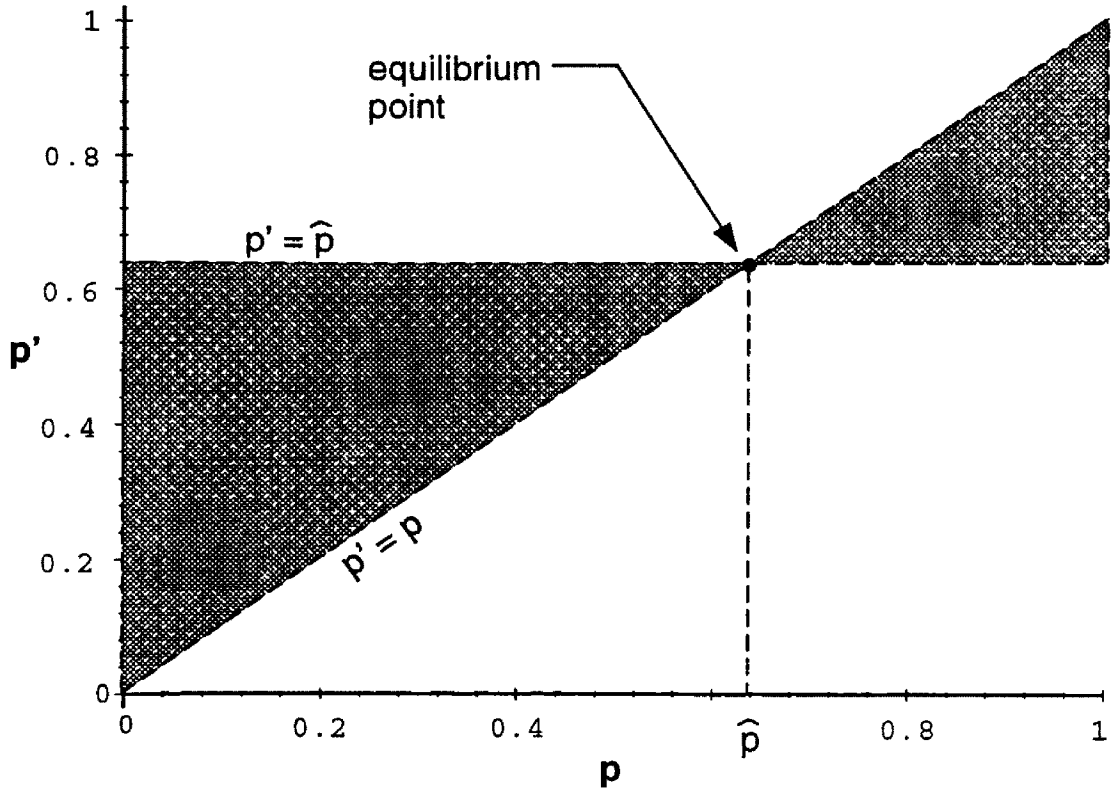


Figure 5.2: The curve for p' versus p must lie within the shaded region

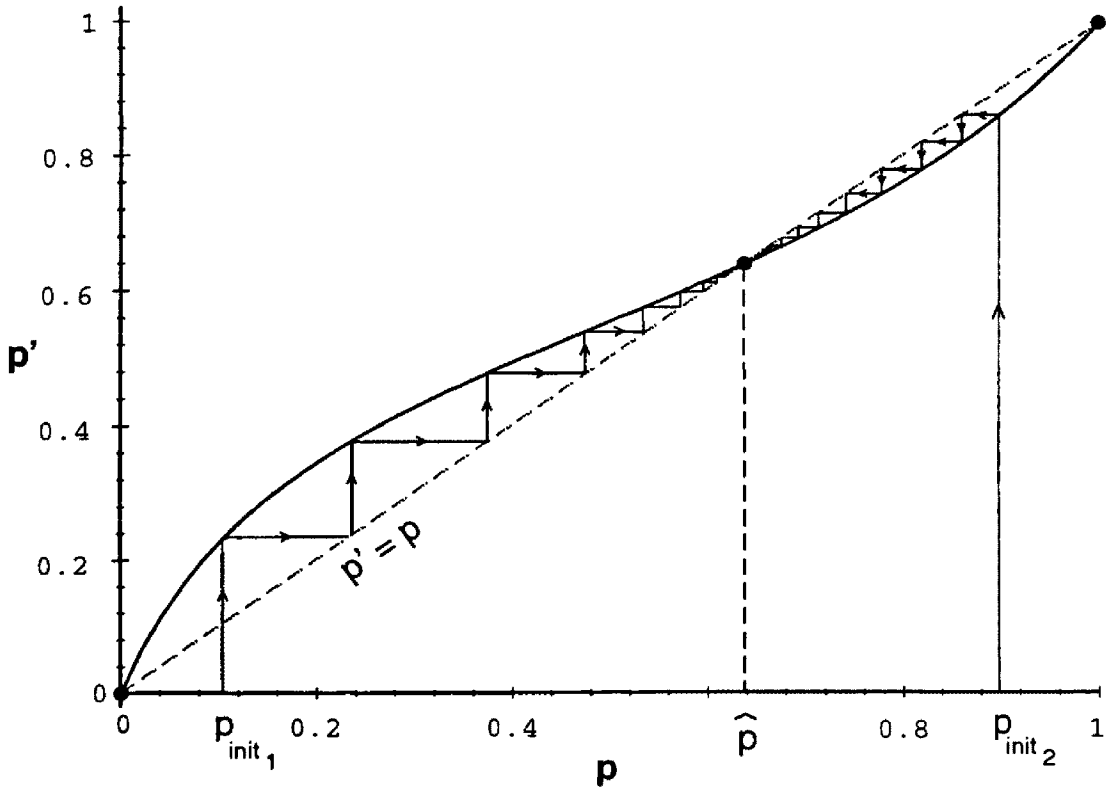


Figure 5.3: The iterates of p staircase into the equilibrium point

where a is a point in the neighborhood of p, the point under consideration, $f'(a)$ is the first derivative of the function f evaluated at a, and $f''(c)$ is the second derivative of f evaluated at a point c between a and p. The first two terms of this equation comprise a linear approximation of the curve, i.e.

$$f(p) \approx f(a) + f'(a)(p-a) = L(p)$$

The third term is an error term, the absolute value of which represents the distance from the line described by $L(p)$ and the curve of $f(p)$.

$$E(p) = \frac{1}{2}f''(c)(p-a)^2$$

The error term varies with the proximity of a to p. The closer the proximity, the smaller the error. The value of $f''(c)$ is bounded, i.e. there is some B for which $f''(c) \leq B$ for all c between a and p. Note that when a is close to p, $(p-a) > (p-a)^2$ and thus $L(p) > E(p)$. Let $f(p) = \Delta p = p' - p$. Then $f'(p) = \frac{df(p)}{dp} = \frac{dp'}{dp} - 1$. At $a = 0, \Delta p = 0$, so $f(0) = 0$. $f'(0) = \frac{dp'}{dp}|_{p=0} - 1 > 0$, since $\frac{dp'}{dp}|_{p=0} > 1$. Thus, $L(p) > 0$, and the linear approximation for Δp lies above the x-axis. The actual curve of Δp lies either above or below $L(p)$, depending on the sign of the error term. The distance from the x-axis is either $L(p) + |E(p)|$ or $L(p) - |E(p)|$, respectively. Clearly, $L(p) + |E(p)| > 0$. The case when the curve of Δp is below $L(p)$ is shown in Figure 5.4. We wish to show that for sufficiently small values of p, $L(p) > E(p)$. In other words, for sufficiently small p,

$$f(0) + f'(0)(p-0) = f'(0)(p) > \frac{1}{2}f''(c)(p-0)^2 = \frac{1}{2}f''(c)p^2$$

Since $p > 0$, both sides of the equation can be divided by p to give $f'(0) > \frac{1}{2}f''(c)p$ or

$$p < \frac{2f'(0)}{f''(c)}$$

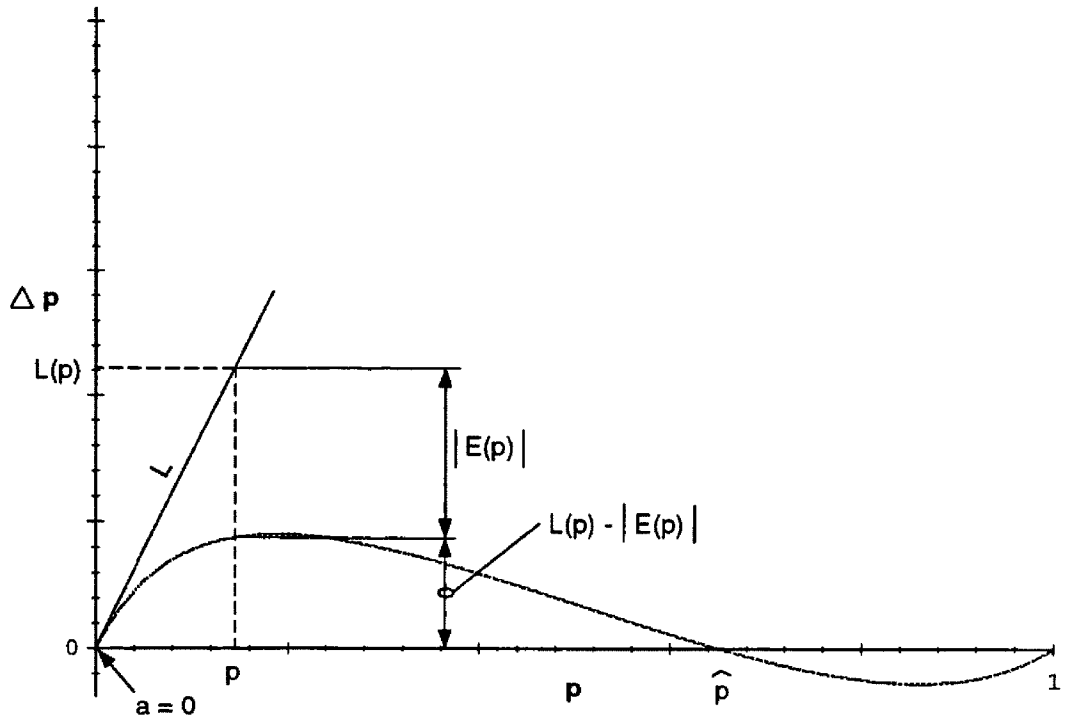


Figure 5.4: The curve for Δp and its linear approximation

which holds for sufficiently small p, since $f''(c)$ is uniformly bounded by B in a neighborhood. Hence, $\Delta p > 0$ for $0 < p < \hat{p}$. Similarly, it can be shown that for $\hat{p} < p < 1, \Delta p < 0$ by taking $a = \hat{p}$ and examining values of p sufficiently close to, but greater than $\hat{p}$. We conclude that the equilibrium point of equation 5.3 is globally stable. □

Finally, Figure 5.5 is a plot of the 0 allele frequency for the system of equation 5.2 superimposed with a plot of the 0-bit frequency for the haploid model over a time scale of 500 generations. The initial values of x_0 and p_0 are both 0.01, and the initial values of x_1 and p_1 are 0.99. The fitness values are $f_0 = 1.00$ and $f_1 = 0.90$. The two curves representing the diploid model correspond to two different values of s, the heterozygote fitness bonus. The case for 1 allele convergence is symmetric, and the curves of Figure 5.4 can be duplicated by interchanging the fitnesses and the initial values of p_0 with p_1 and x_0 with x_1. Once again, it is evident that the

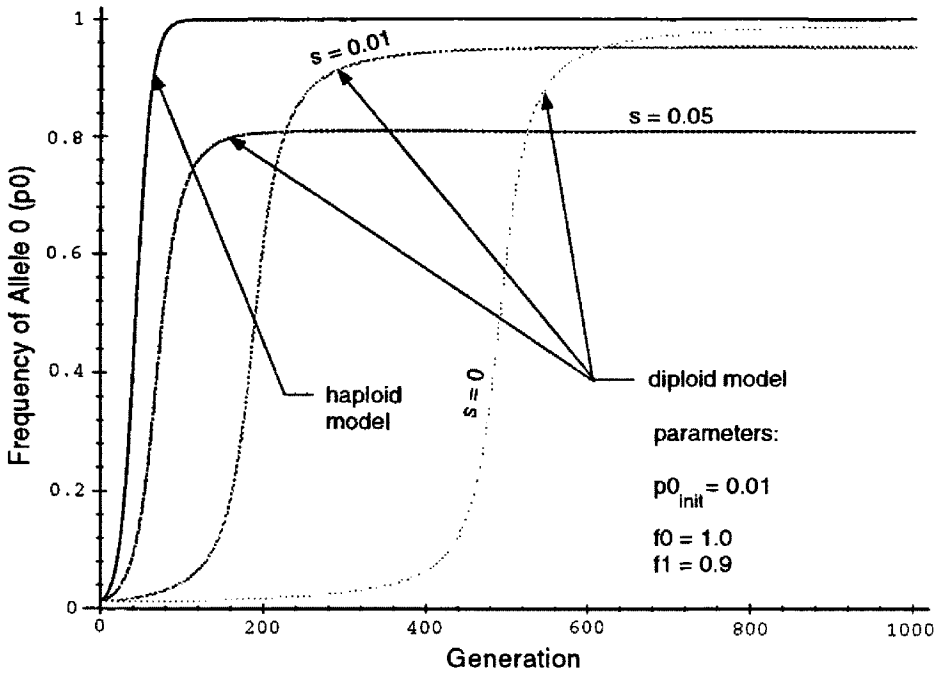


Figure 5.5: Convergence characteristics: haploid vs. diploid models

diploid model is capable of both slowing the rate of convergence (to a homozygote genotype) and avoiding complete convergence (by forming a stable polymorphism). More importantly, the scheme of this chapter can be extended to multiple loci.

3. Extending the Model

Geneticists such as Hartl and Clark [7] have taken the next logical step by analyzing a two-locus, two-allele viability model. The primary difference between this model and single-locus models is the addition of recombination between pairs of genes linked on the same chromosome. This is simulated in GAs with the crossover operator. An allele recursion can be derived, but the two-locus selection problem has not been solved for the general case. In other words, for an arbitrary fitness matrix, there

is no formula for the equilibria and their stability. Several papers have examined a special case of the two-locus selection problem referred to as the *additive* model, where the simplifying assumption is made that the fitness of a given genotype is the sum of the fitness effects at each locus. Most notable among these are perhaps a series of papers by Karlin and Liberman [10], [11], and [12] that analyze the two-locus additive fitness model and extend it to an arbitrary number of loci. The complexity of the analysis demands a level of mathematics and a system of notation that are quite beyond the scope of this paper. In [12], Karlin and Liberman develop a global convergence criterion and then apply it to establish that the polymorphic equilibrium of a general multilocus additive viability model is globally stable provided:

1. Each of the loci is diallelic.

2. Each of the loci is overdominant.

3. The multilocus recombination rate is positive.

Clearly, item 1 is satisfied with the system outlined in this chapter. We have shown that the single-locus case is capable of attaining overdominance, thus meeting the requirement of item 2. With a positive crossover rate, item 3 can be satisfied. However, it is not clear whether a multi-locus extension of the system in this chapter can be equated to the additive model. As stated earlier, fitnesses in a traditional GA usually correspond to entire strings, not to particular locations within a string. For an arbitrary fitness function, there is no way to derive quantitatively the fitness of a given string from the sum of the fitnesses of its component bits, since an individual bit typically has no fitness associated with it. The issue of whether a globally stable polymorphic equilibrium exists for a multilocus diploid GA will have to be resolved by empirical methods.

Chapter 6

Empirical Test Results

1. Implementing a Diploid GA

The procedure for implementing a diploid GA that conforms to the scheme discussed in chapter 5 is very similar to the procedure for the haploid GA as presented in Mitchell [16]. The main differences lie in the need to compute allele frequencies during each generation, the computation of fitnesses, and the application of the crossover operator. For the diploid GA, we perform the following steps:

1. Randomly generate an initial population of n diploid individuals, where each individual consists of two l-bit binary strings.
2. Compute the allele frequencies in the total population for each locus.
3. Evaluate the fitness of each individual.
4. Generate a new population of n diploid individuals by repeatedly performing:
   (a) selection—select two parents based on fitness
   (b) gametogenesis—generate a pair of gametes from each parent, performing crossover with probability *pcross* and bit-wise mutation with probability *pmut*.

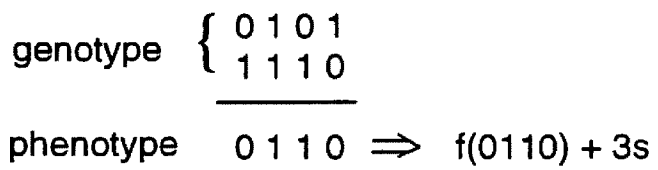


Figure 6.1: Computing the fitness of a diploid genome

(c) fertilization—randomly combine one set of gametes from each parent to create a (diploid) zygote or child.

5. Goto step 2.

The two l-bit binary strings are aligned so that a locus of the diploid chromosome refers to the same position in each string. The allele frequency at a given locus is computed by counting the number of 0 alleles at that locus for each individual in the population, then dividing by two times the population size. This is done for all l loci. These frequencies are then used to assist in the resolution of the heterozygote entries of the dominance map. To compute the fitness of an individual, its diploid genotype must first be mapped to a haploid phenotype. The genotype is examined on a locus-by-locus basis. 00 maps to 0 and 11 maps to 1. 01 and 10 map to 0 with probability p, where p is the frequency of allele 0 at that locus. Thus, 01 and 10 map to 1 with probability $1 - p$. The number of heterozygote loci is recorded and stored in a *bonuscount* variable. The haploid fitness function f is applied to the phenotype and a bonus equivalent to $bonusvalue \times bonuscount$ is added to the resulting fitness. This is depicted for a string of length 4 in Figure 6.1. The method of selection is stochastic sampling with replacement, or "roulette wheel" selection.

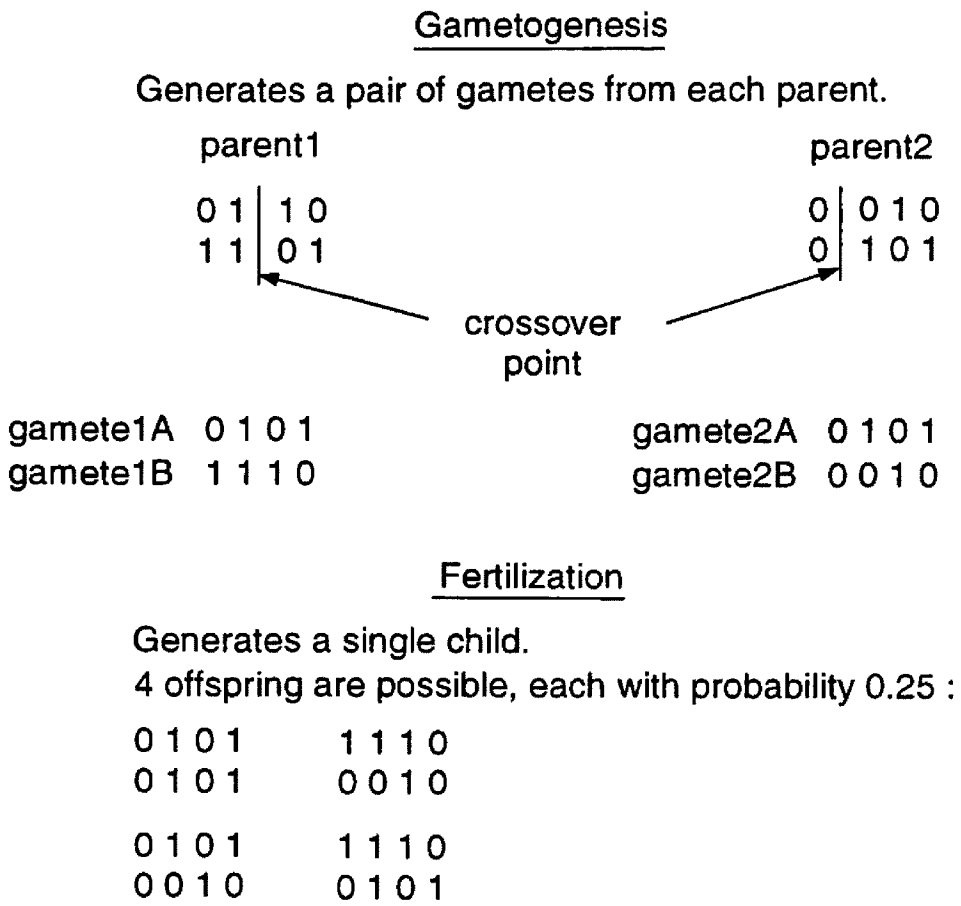


Figure 6.2: Diploid gametogenesis and fertilization

Crossover in the diploid GA occurs at a different stage of the lifecycle than in the haploid GA. Since each diploid parent consists of two strings, recombination of genetic material can occur within a single parent. One-point crossover is performed. Before a parent can donate a pair of gametes to the fertilization process, the mutation operator is able to act upon each bit of the gametes with a small probability. Fertilization consists of randomly choosing one gamete from each parent and combining them to form a new diploid individual. Gametogenesis and fertilization are shown in Figure 6.2 for a 4-bit example where $bonusvalue = s$ and $bonuscount = 3$. In the figure, one-point crossover is performed between locus 2 and locus 3 for parent1, and between locus 1 and locus 2 for parent2. For the sake of clarity, no mutation is performed in

this example.

The first tests attempted to duplicate the results attained in Figure 5.5 for allele convergence. Chapter 5 presented an idealized model of a GA with no mutation and an infinite population. We saw that the infinite population haploid model converged rapidly and completely, while the diploid model exhibited a slower rate of convergence and retained both types of alleles. The diploid implementation described above should be able to achieve similar results for large size populations. Figure 6.3 shows a comparison of the convergence rates for the haploid GA and the diploid GA. With the same initial parameters, $pcross = 0$ and $pmut = 0$, and a population size of $10,000$, the results appear to agree quite closely with the models. The small perturbations or lack of "smoothness" in the curves are due to stochastic errors. Again, we see that we can alter the rate of convergence and percentage of alleles remaining at equilibrium by varying the value of the heterozygote fitness bonus, s.

2. Measuring Diversity

A pairwise Hamming distance function is used to measure the diversity of the haploid and diploid GAs. The function works as follows: Each individual's binary representation is compared locus-by-locus with that of every other individual in the population. In order to correlate diploid results with haploid results, the Hamming distances will be computed from each individual's *phenotype* in the diploid case. Each time the allele values differ at a given locus, the Hamming distance is incremented by one. If there are n individuals in the population, each consisting of a string of length l, then a total of $ln(n-1)/2$ bitwise comparisons are required. The Hamming distance is then normalized over the population size and the string length, so that diversity results can be compared among differing population sizes and string lengths. Because the

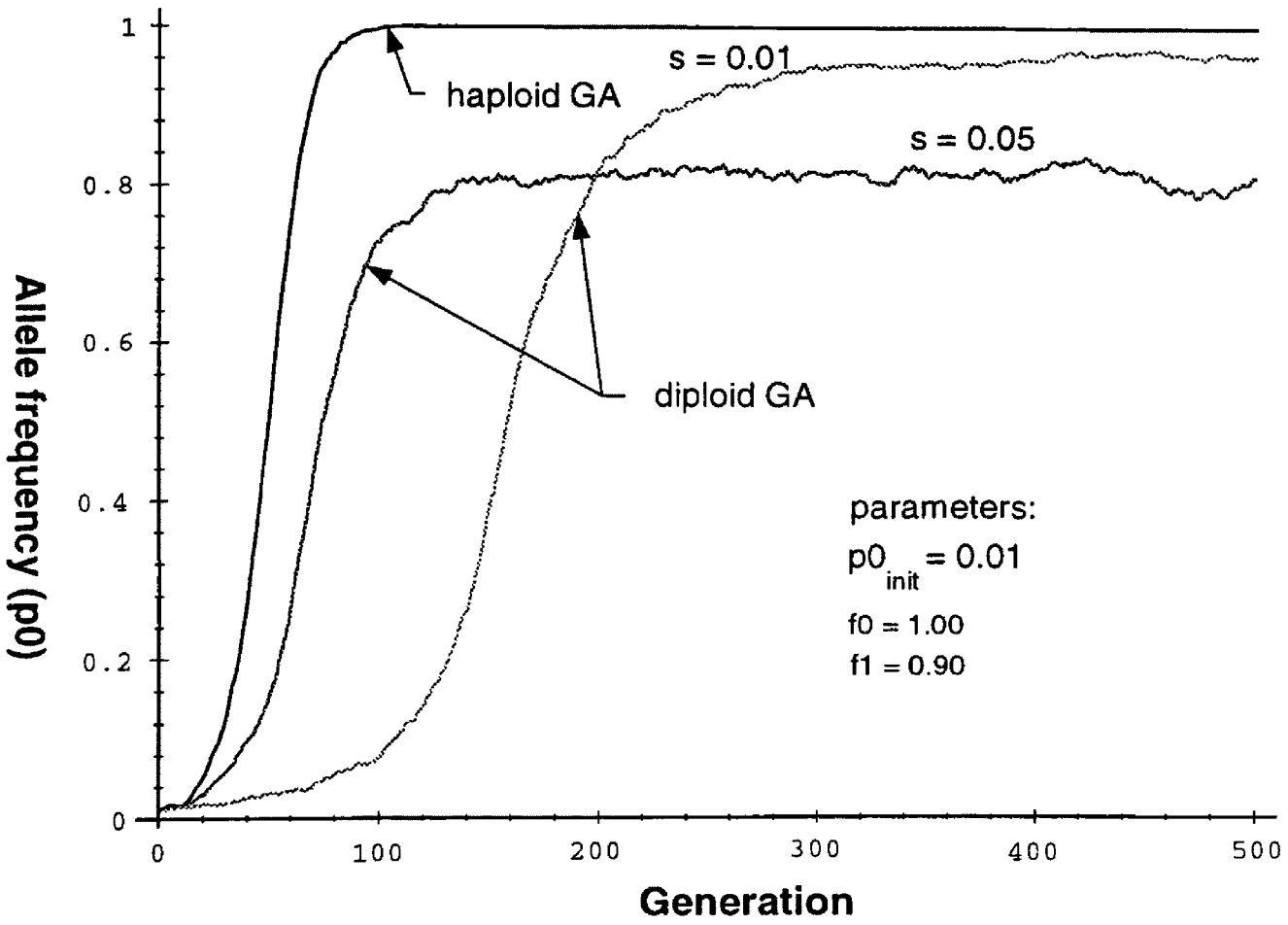


Figure 6.3: Convergence characteristics: haploid vs. diploid GA

degree of possible diversity decreases with increasing population size for small string lengths, we normalize the Hamming distance values only when $2^l \gg n$. Another important measure in the case of the diploid GA is the number of heterozyous loci in the population. By recording this statistic, we can determine whether overdominance is being maintained in the diploid population.

In the absence of selective pressure, changes in allele frequency can result from chance alone, a phenomenon biologists refer to as *random genetic drift*. Left to the influence of random genetic drift, the allele frequencies in a haploid or diploid population will wander about, but will eventually converge as alleles are either lost or become fixed. The rate of convergence is dependent upon population size, initial allele frequencies, and other factors. The reader is referred to Hartl and Clark [7] for an overview of random genetic drift, including studies, models, and a list of further references. We would like to show that a diploid GA will converge at a slower rate than a haploid GA under these conditions and that the heterozygote fitness bonus can affect the rate of convergence.

Random genetic drift can be simulated in a GA by using a flat fitness function that gives every individual in the population equal probability of being selected to parent an offspring. In addition, the mutation rate is set to zero. We assume that the randomly generated initial population provides an even distribution of allele frequencies. The heterozygote fitness bonus is computed as a small percentage of the average fitness of the population in the previous generation, e.g. 0.01 or 0.05. This ensures that the bonus is relatively small with respect to the fitness of a given individual during a given generation. The bonus is set to zero when the fitnesses of the first generation are evaluated. Figures 6.4 and 6.5 compare the pairwise Hamming distance values for the haploid GA with those for the diploid GA with various values of s, the heterozygote fitness bonus. Figure 6.4 was generated with a population size

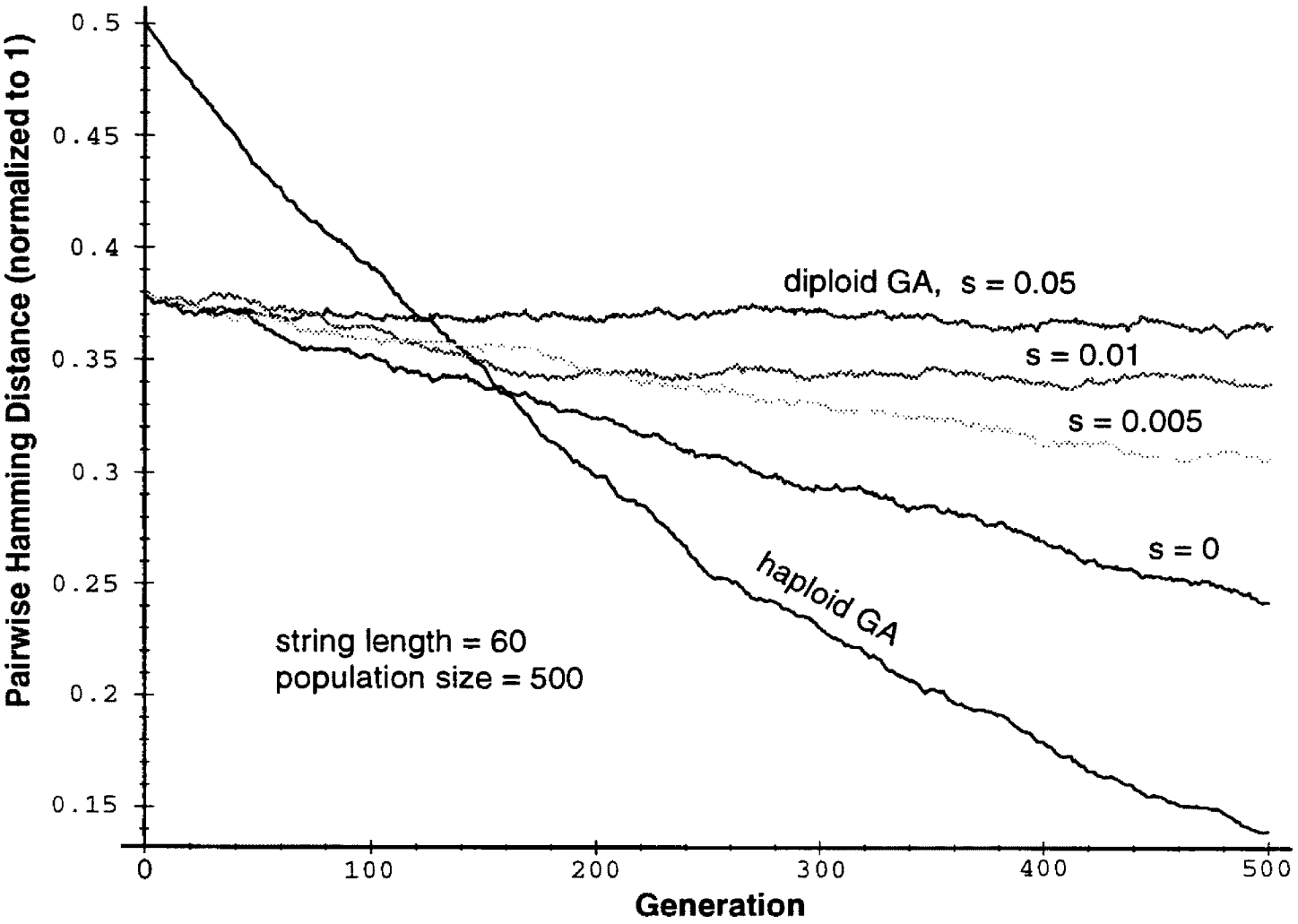


Figure 6.4: Pairwise Hamming distance values for $n = 500$ and $l = 60$

of 500 and a string length of 60, while Figure 6.5 was generated with a population size of 100 and a string length of 60. Each figure represents results averaged over 10 runs. Both GAs use one-point crossover with a rate of $pcross = 0.5$. We see that the diploid GA does indeed converge at a slower rate than the haploid GA, even without the benefit of the heterozygote fitness bonus. Moreover, increasing the bonus decreases the rate of convergence. As expected, Hamming distance values are smaller and convergence rates are faster for the smaller sized population.

Figures 6.6 and 6.7 show the percentage of heterozygous loci for successive gen-

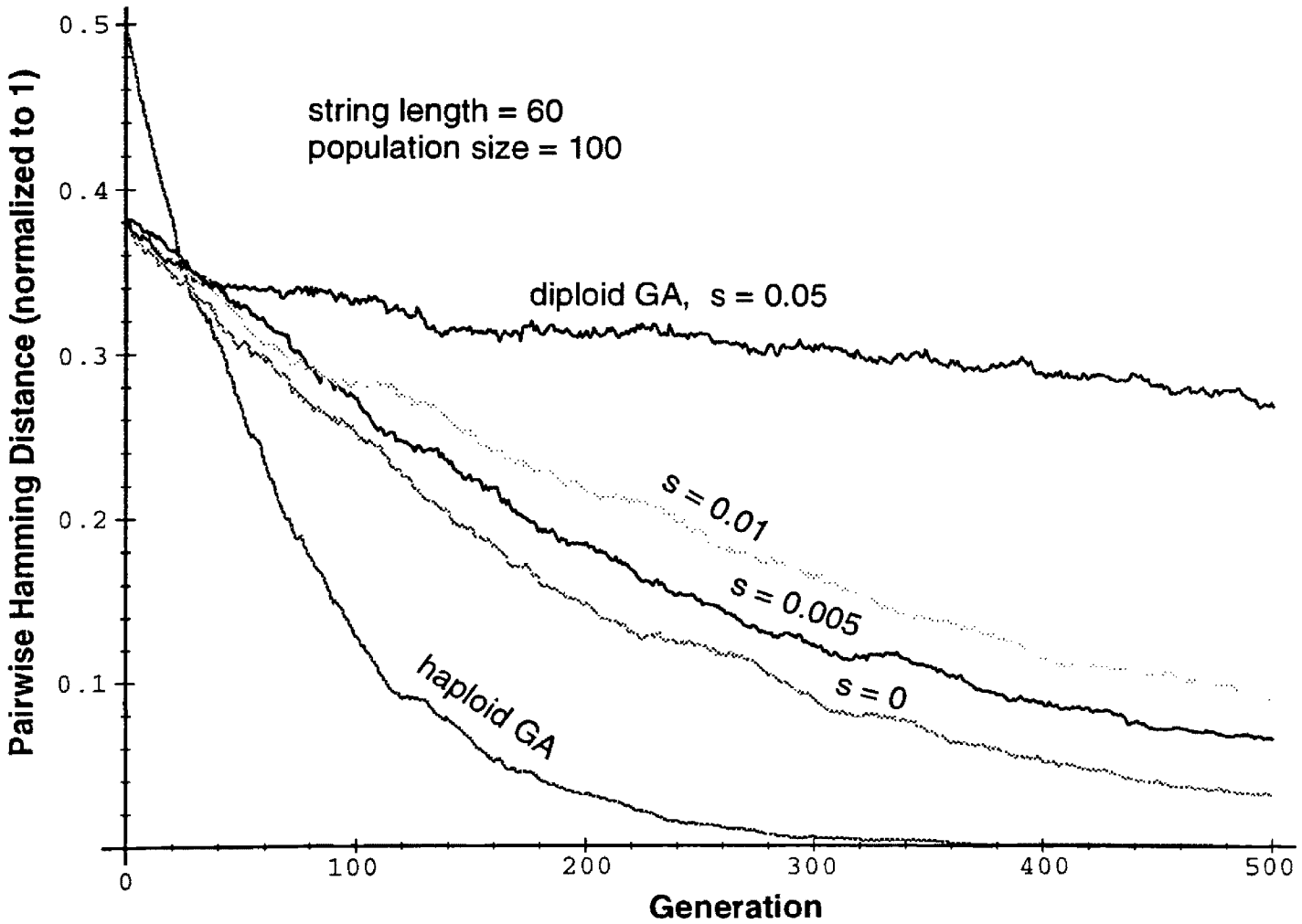


Figure 6.5: Pairwise Hamming distance values for $n = 100$ and $l = 60$

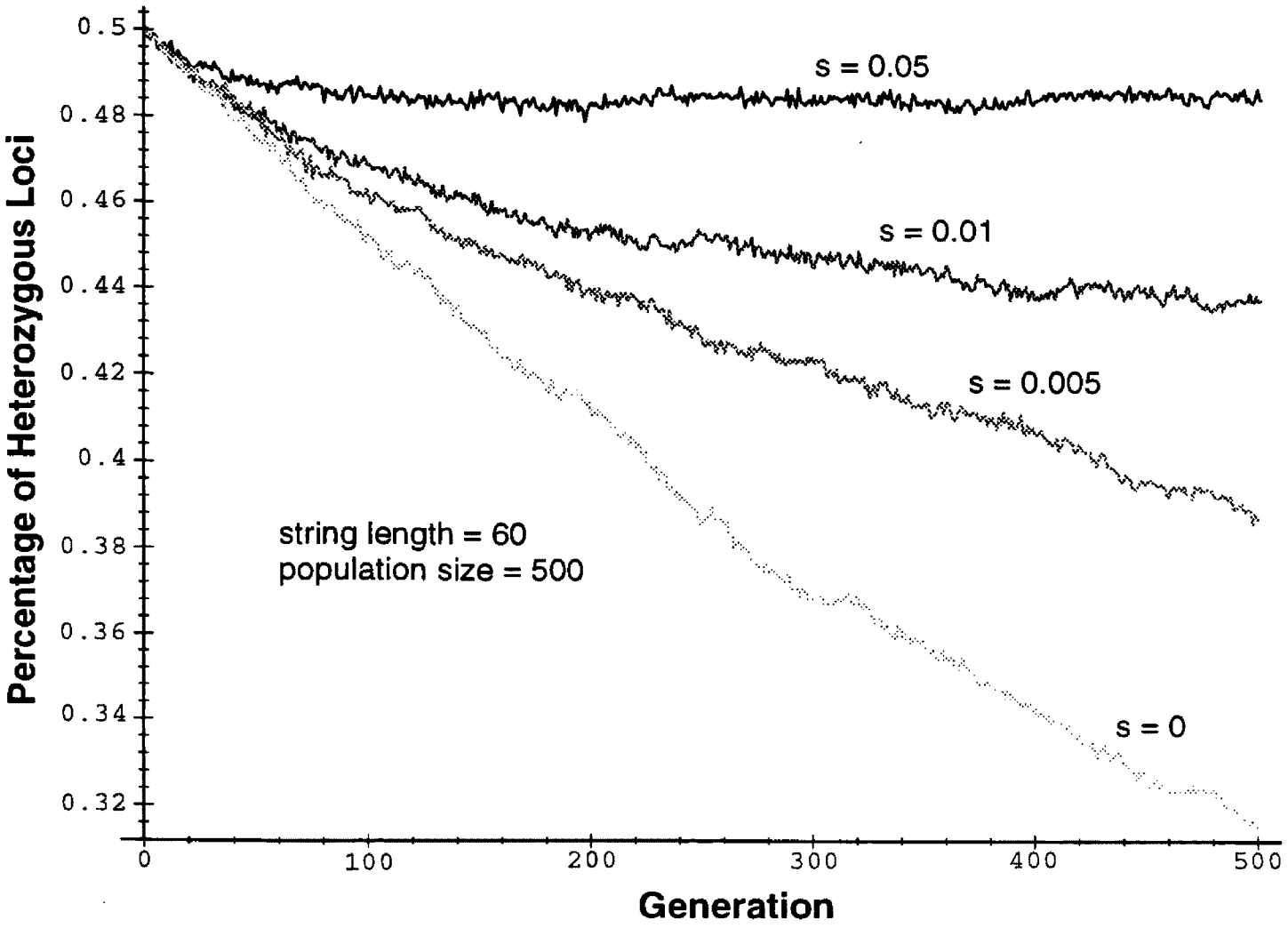


Figure 6.6: Fraction of heterozygous loci for $n = 500$ and $l = 60$

erations and correspond to Figures 6.4 and 6.5, respectively.

3. The Oscillating 0-1 Knapsack Problem

The goal of the 0-1 knapack problem is to maximize the total value of a subset of objects selected from a set of N possible objects that may be placed in a knapsack, subject to a weight constraint. Letting v_i be the value of the ith object and w_i be the weight of the ith object, the problem may be expressed mathematically as

$$\max \sum_{i=1}^{N} v_i x_i$$

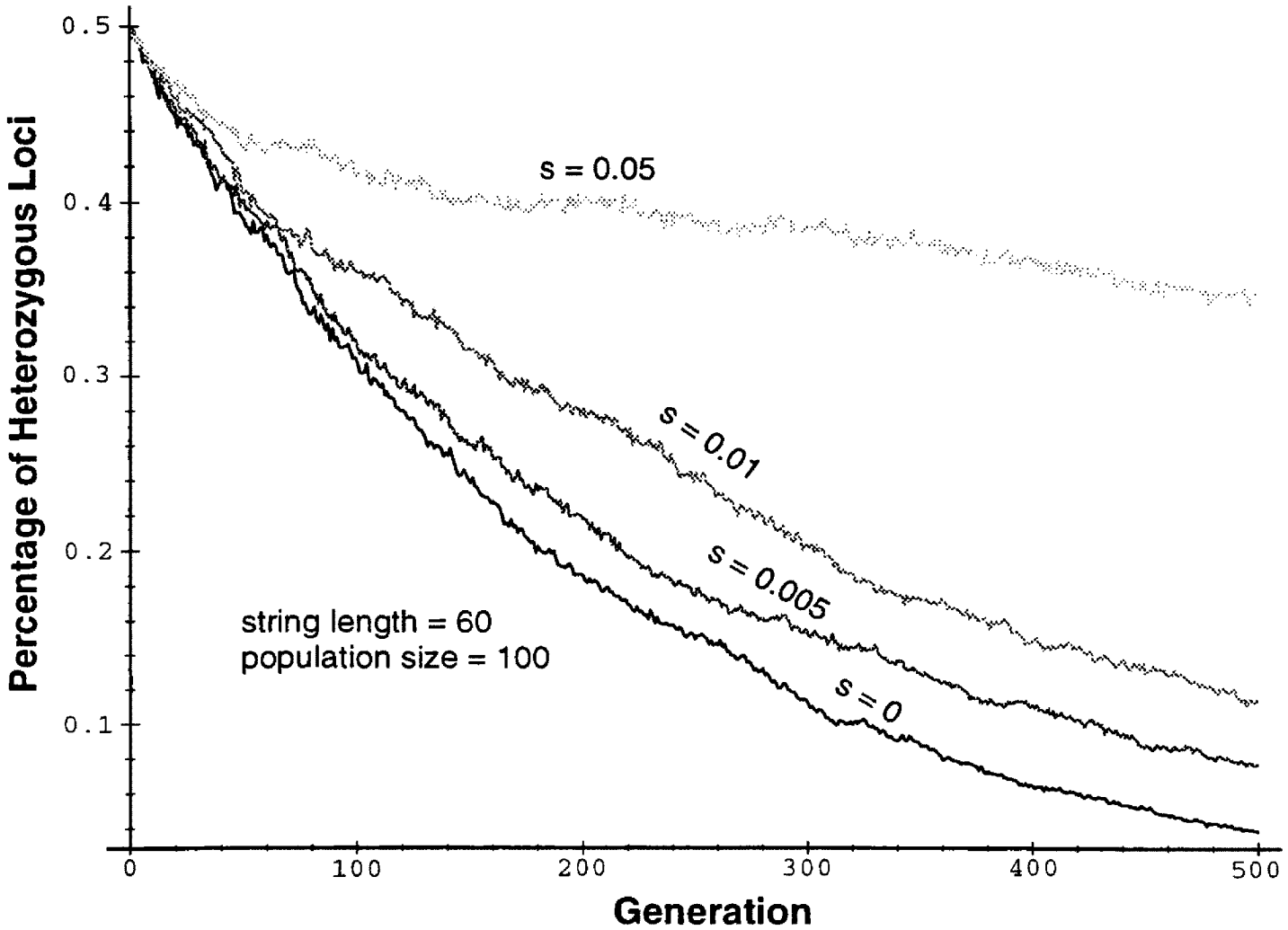


Figure 6.7: Fraction of heterozygous loci for $n = 100$ and $l = 60$

subject to the weight constraint

$$\sum_{i=1}^{N} w_i x_i \leq W$$

where $x_i \in \{0, 1\}$ denotes whether the ith object is in or out of the knapsack, and W is the maximum permissible weight. As Goldberg and Smith note in [5], the problem is presented to the GA blindly. That is, the algorithm has no knowledge of the structure or parameters of the problem, since they are represented externally as part of the fitness function. In addition, nonstationarity is introduced by varying the weight constraint as a step function between two values—82% and 50% of the total object weights—every 50 generations. The weight constraint is handled as follows: a knapsack weight that exceeds the maximum permissible weight results in a fitness penalty which is deducted from the total value. Specifically, the penalty function applied to overweight knapsacks is

$$\text{penalty} = 20 \times (\sum_{i=1}^{N} w_i x_i - W)^2$$

Negative fitness values that result from applying the penalty function are set to zero. The table below depicts the parameters in the 17-object knapsack problem used by both Goldberg and Smith [5] and Ng and Wong [18].

Object number i	Object value v_i	Object weight w_i
0	2	12
1	3	5
2	9	20
3	2	1
4	4	5
5	4	3
6	2	10
7	7	6
8	8	8
9	10	7
10	3	4
11	6	12
12	5	3
13	5	3
14	7	20
15	8	1
16	6	20
totals	91	122

This results in weight constraints of $W_{82\%} = 100$ and $W_{50\%} = 61$. The optimal strings for each case are as follows:

W	string	value	weight
100	01111101111111111	87	100
61	01011101111111011	71	57

Unfortunately, it is very difficult to correlate the results of the two papers, because they disagree on the selection and crossover strategies. While Goldberg and Smith use stochastic remainder selection with replacement and two-point crossover with

pcross = 0.75, Ng and Wong use linear ranking selection and uniform crossover with *pcross* = 0.5. Our own tests indicated that the selection method can have a significant impact on the results obtained. For example, stochastic sampling with replacement resulted in slower convergence (and thus better recovery from changes in the weight constraint) than did stochastic remainder selection with replacement when used in the haploid GA runs. Because Goldberg and Smith provide sufficient information to repeat their experiments, their GA parameters and implementation were chosen for the tests used in this chapter. The weight constraint was switched every 50 generations, and test runs were performed with *pmut* = 0.001 and *pmut* = 0.01. Figure 6.8 plots the average and maximum fitnesses over 500 generations with *pmut* = 0.001 for the haploid GA, triallelic diploid GA(as per Hollstien, Goldberg, and Smith), and diallelic diploid GA (as presented in this chapter and modelled in the previous chapter). In the diallelic diploid GA, the heterozygote fitness bonus is computed as 0.01 of the average fitness of the previous generation. Once again, the bonus is used only in the selection process and is not included in the fitness results. Each plot represents average and maximum generational fitnesses averaged over 10 runs. Figure 6.9 presents the results averaged over 10 runs with *pmut* = 0.01. Clearly, when the weight constraint is switched to the lower value, the diploid GAs are able to reach a good solution before the next weight constraint change, while the haploid GA with *pmut* = 0.001 converges sufficiently so that all strings have zero fitness after application of the penalty function. With *pmut* = 0.001 and an oscillation period of 100 generations, none of the GAs are able to achieve the optimal fitnesses of 87 and 71. Although both the triallelic and diallelic diploid GAs have similar fitness values for the 82% constraint, the diallelic scheme exhibits a slight performance advantage for the 50% constraint when *pmut* = 0.001 and a decidedly greater advantage for this constraint when *pmut* = 0.01. When *pmut* = 0.01, the

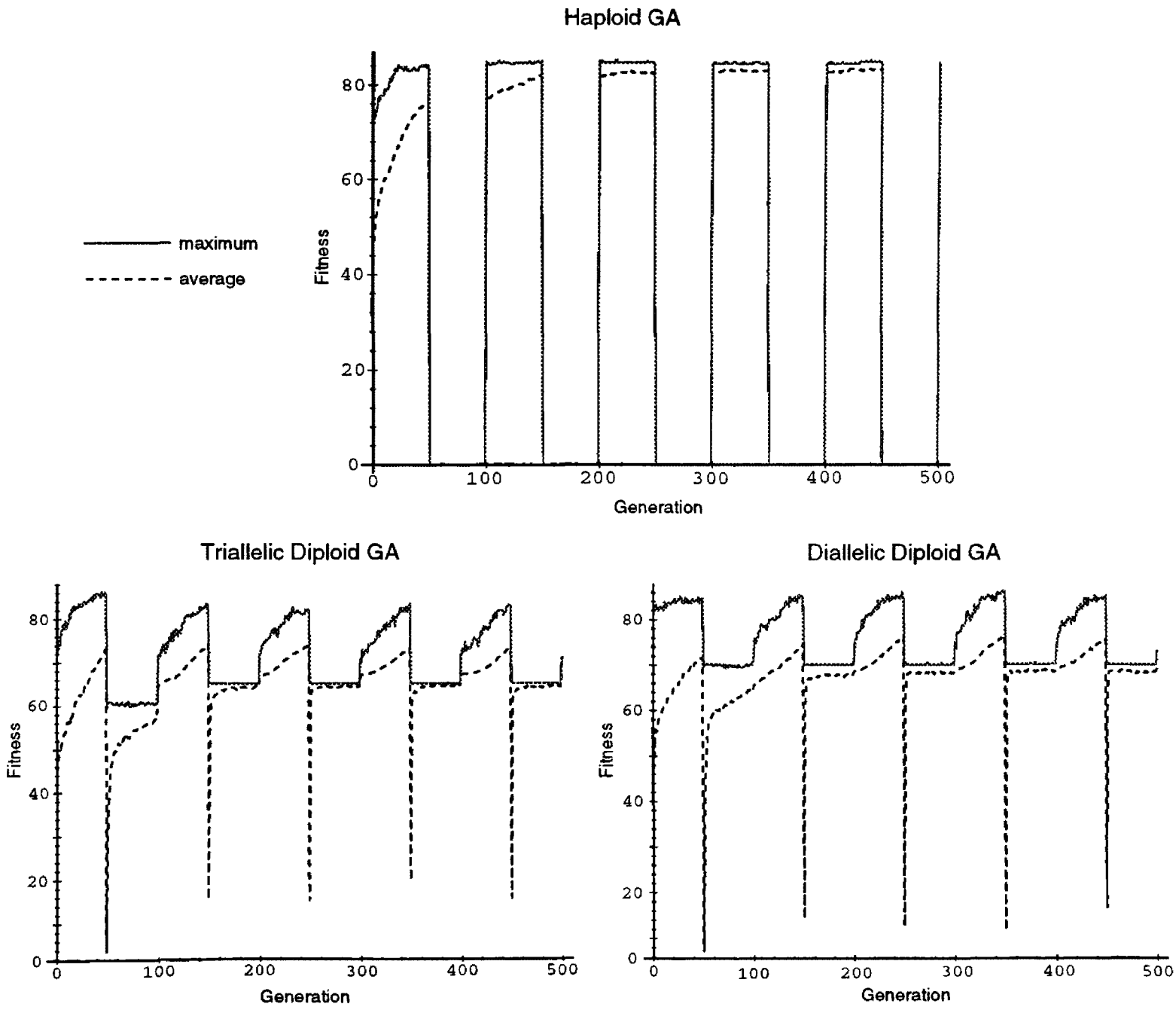


Figure 6.8: 0-1 oscillating knapsack results, $pmut = 0.001$

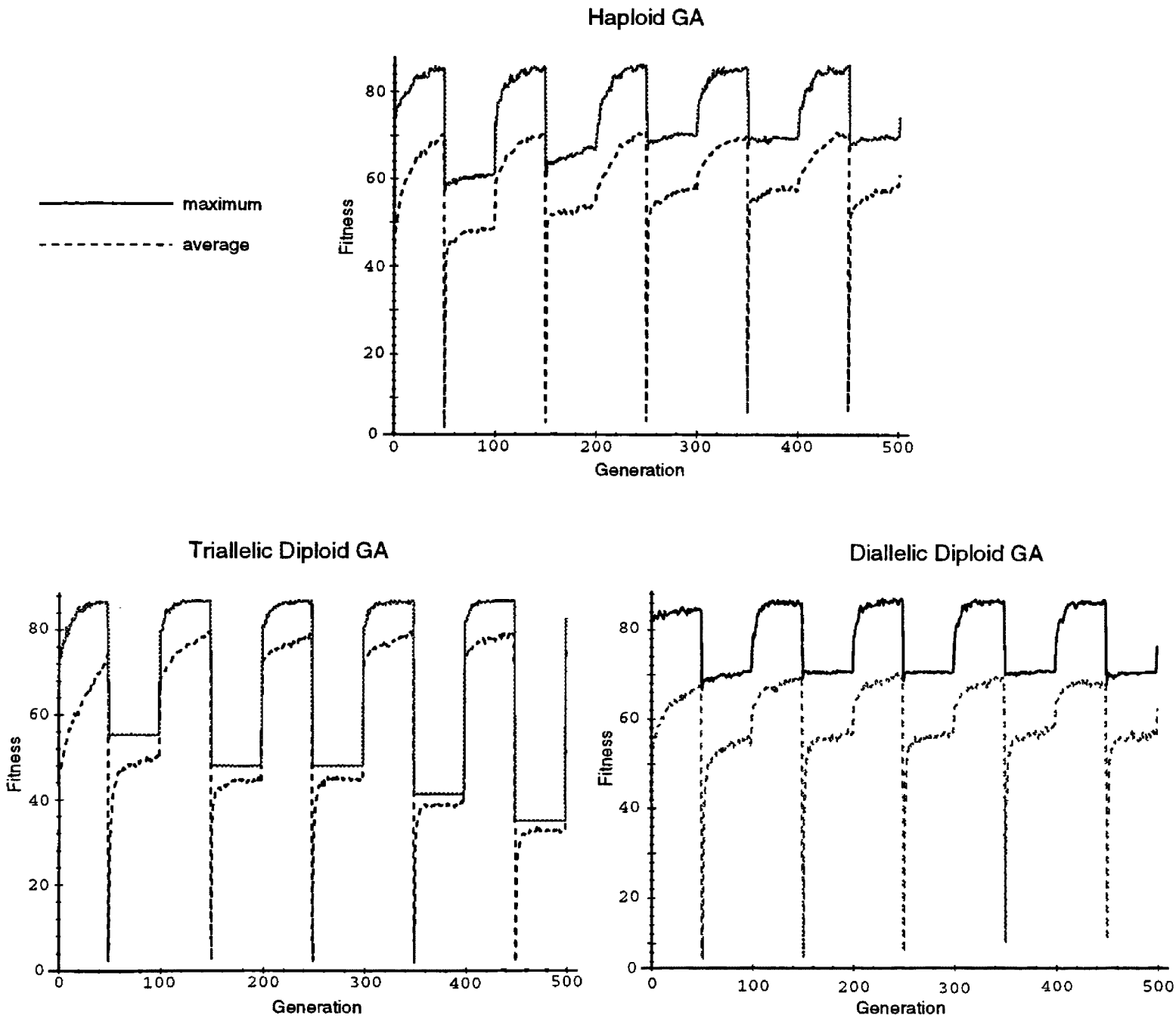


Figure 6.9: 0-1 oscillating knapsack results, $pmut = 0.01$

triallelic scheme not only reaches lower fitness values for the 50% constraint, but also degrades with successive oscillations. The higher mutation rate gives the haploid GA better performance for both weight constraints, but it falls short of the diallelic diploid GA, which finds the optimum for both weight constraints when $pmut = 0.01$.

4. Multimodal Function Optimization

A fundamental hypothesis that attempts to explain how GAs work is the *building block hypothesis* [6]. The hypothesis states that strings which include substrings that are contained in the globally optimal string (or building blocks) will increase in frequency. Fitter strings are thus constructed from the most fit partial solutions of past samplings. To test this hypothesis, GA researchers such as Goldberg have devised fitness functions specifically designed to deceive a GA. A deceptive fitness function is one in which the average fitness of substrings which are *not* contained in the global optimum is higher than the average fitness of those which are. We present a 3-bit deceptive problem based on the minimal deceptive problem of Goldberg [6]. We assign fitnesses to each of the possible 3-bit substrings as follows:

string	fitness
000	3
001	2
010	2
011	1
100	2
101	1
110	1
111	4

Here, 111 is the optimal substring, but all other substrings have fitnesses that produce a gradient away from 111 toward a local optimum at 000. We concatenate 10 of these 3-bit substrings together to form a string of length of 30. A 30-bit string's fitness is evaluated 3 bits at a time (using the fitness values in the above table) and is the sum of 10 of these fitness values. Thus, the globally optimal string consists of all 1s, and there are $2^{10} - 1$ local optima designed to entrap a rapidly converging GA on a suboptimal peak.

We apply this fitness function to the haploid and diploid GAs, measuring average and maximum fitness and pairwise Hamming distance. The mutation rate is varied in the haploid GA, while mutation is set to zero and the value of the heterozygote fitness bonus is varied in the diploid GA. In order to ensure that fitness comparisons are made fairly, the fitness bonus is incorporated only during the selection process, but is not included in an individual's contribution to the average fitness of the population, which is used in the fitness plots. Again, results are averaged over 10 runs, and a crossover rate of $pcross = 0.5$ is used with one-point crossover. Examining figures 6.10 and 6.11, we see that for population sizes of 500 and 100 respectively, the diploid GA performs better under any fitness bonus selection scheme than does the haploid GA. We also note that the haploid GA never reaches the global optimum in its best-of-generation fitness results (not plotted). The corresponding diversity results are reported in terms of the pairwise Hamming distance in figures 6.12 and 6.13 for population sizes of 500 and 100 respectively.

We see that for the smaller population, the diploid GA requires a higher heterozygote fitness bonus to achieve the same degree of diversity as it did with the larger population. Although a relatively high mutation rate of $pmut = 0.01$ enables the haploid GA to maintain the greatest diversity in the smaller population, the corresponding fitnesses indicate that its performance suffers greatly as a side effect of a

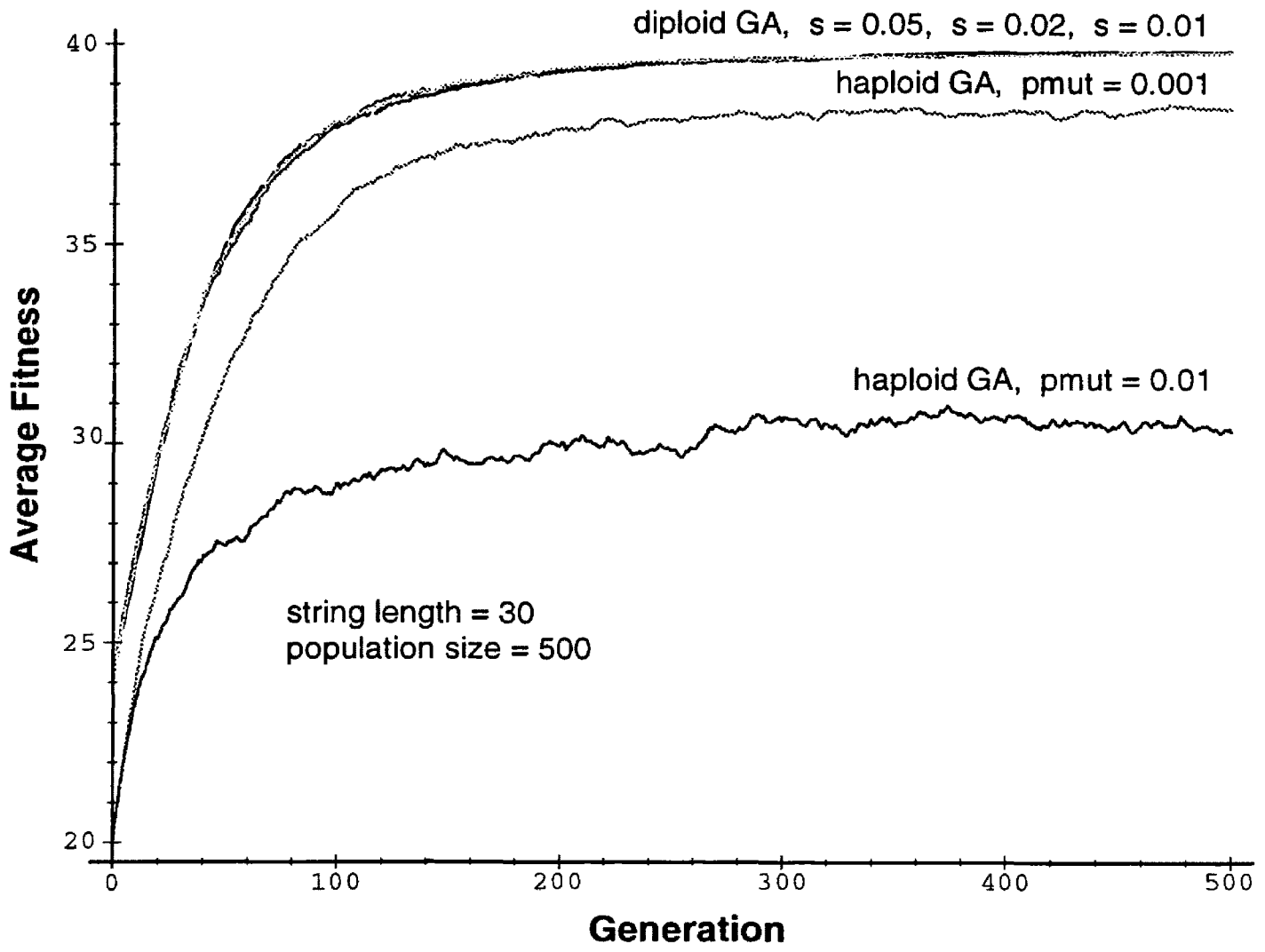


Figure 6.10: Deceptive problem fitness results, $n = 500$ and $l = 30$

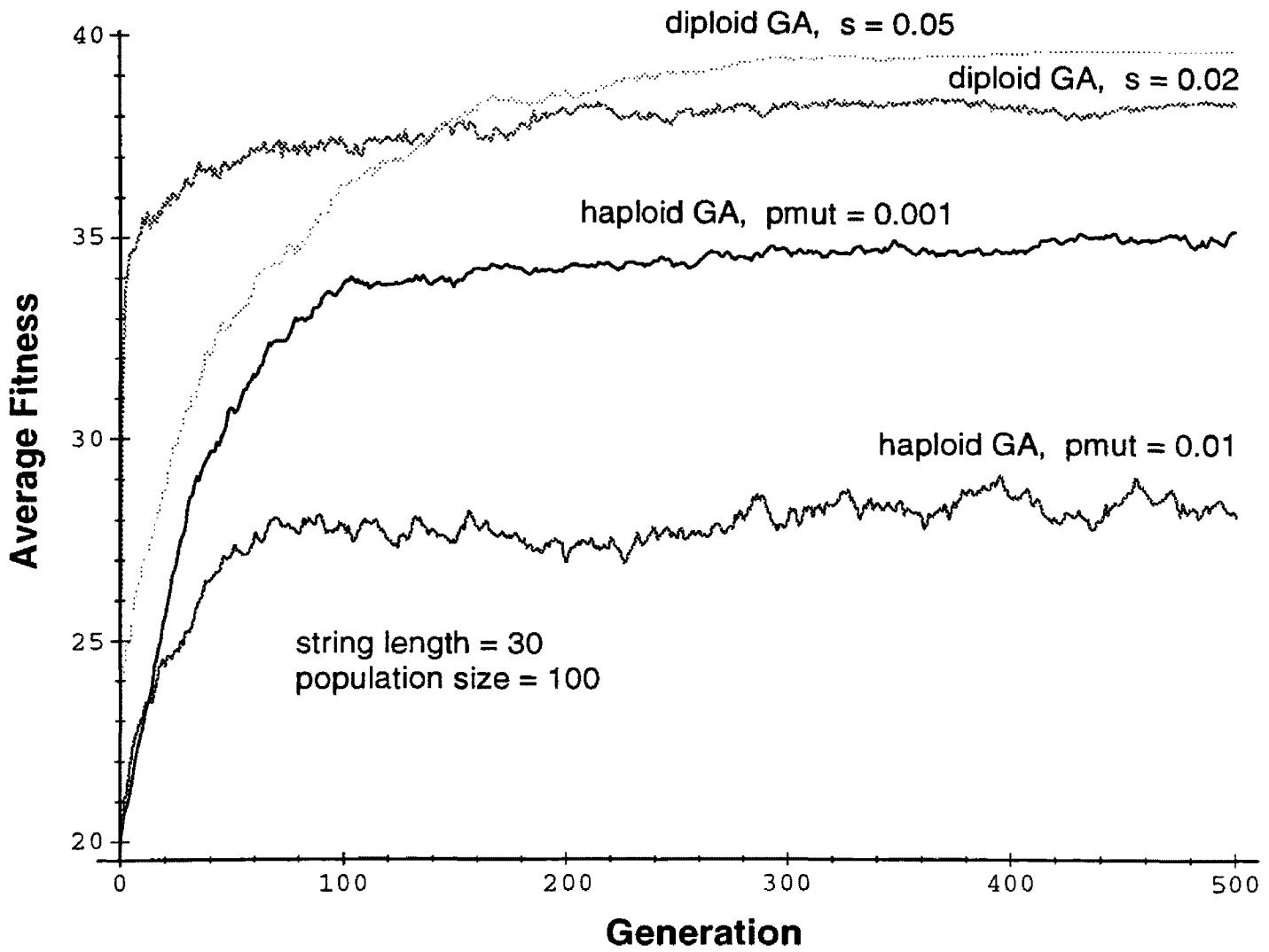


Figure 6.11: Deceptive problem fitness results, $n = 100$ and $l = 30$

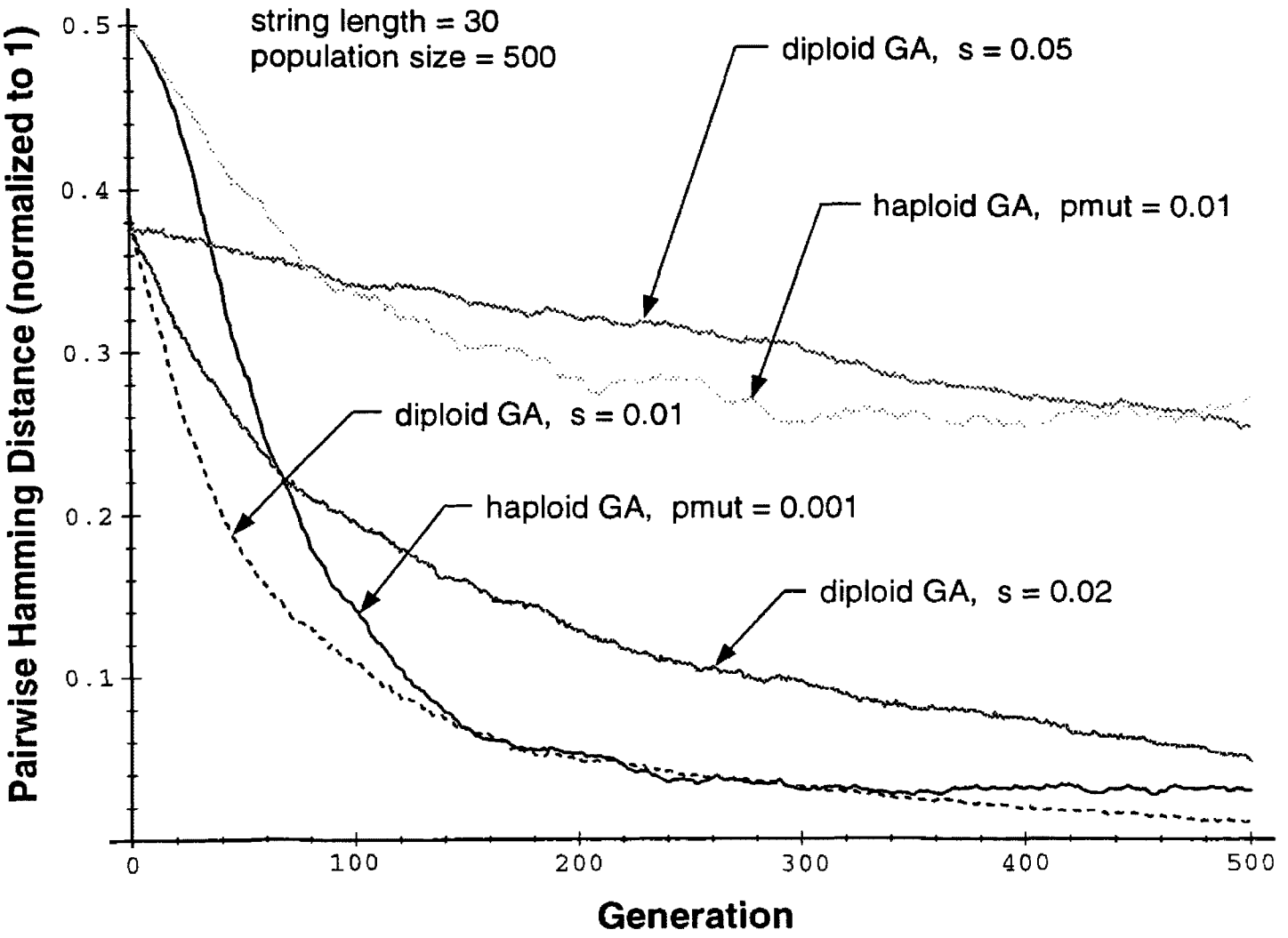


Figure 6.12: Deceptive problem diversity results, $n = 500$ and $l = 30$

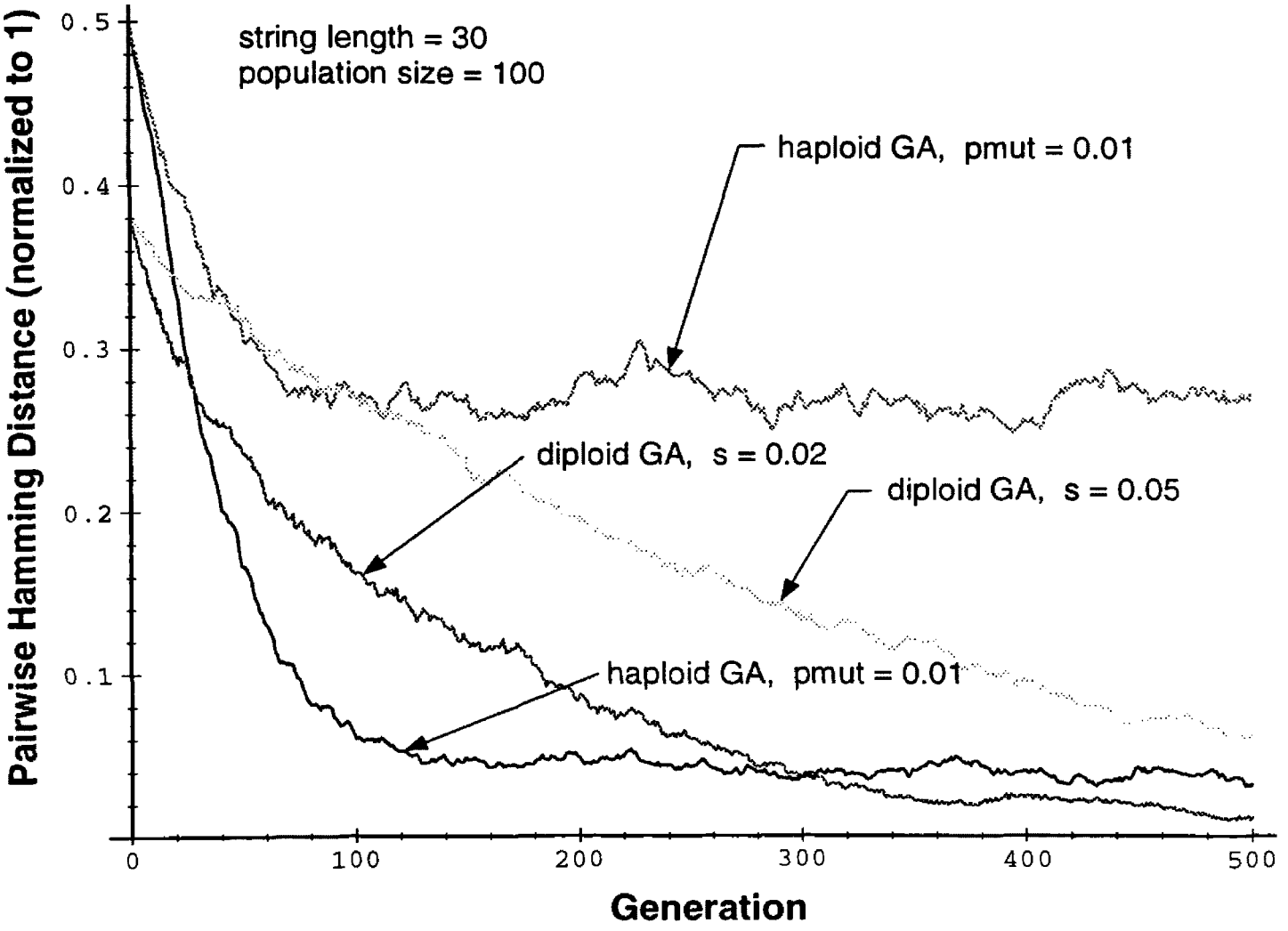


Figure 6.13: Deceptive problem diversity results, $n = 100$ and $l = 30$

high rate of mutation. It appears that for this deceptive problem, the diploid GA is able to give both better performance and increased diversity when given a sufficient heterozygote fitness bonus.

5. A Runtime Study

With large population sizes and long bit strings, conventional GAs may require a significant amount of time to run in order to reach a desired stopping criterion. Certainly, the diploid GA introduces additional computational overhead when evaluating the fitness of an individual. Allele frequencies at each locus must be computed and stored for each generation. The individual's genotype must be mapped to a phenotype, and the number of heterozygous loci must be determined before an individual can be assigned a fitness. In measuring runtime performance, we are most interested in determining whether the diploid GA gets linearly or exponentially worse with increasing string lengths and population sizes. We take the difference of the diploid minus the haploid runtime for various string-length $\times$ population-size products. "Runtime" is defined as the user-mode time as measured by the Unix **time** facility. All programs are written in C, compiled with the IBM **xlc** compiler, and run under AIX 4.2 on an RS-6000/250 workstation. The deceptive fitness function of the previous section is used in both the haploid and diploid GAs. The crossover rate for both GAs is $pcross = 0.5$. While the haploid GA is given a mutation rate of $pmut = 0.001$, the diploid GA is given $pmut = 0$ and a heterozygote fitness bonus of $s = 0.01 \times$ avg. fitness. The following (string length, population size) pairs were used:

string length l	population size n
30	100
60	100
90	100
30	500
45	500
60	500
75	500
90	500

The results are shown in Figure 6.14, and they appear to indicate a linear rather than an exponential relationship.

6. Conclusions

When selection and mutation are eliminated, the diploid GA is able to slow the rate of convergence associated with random genetic drift. By modifying the fitness bonus for heterozygotes, we can control the rate of allele loss and the percentage of heterozygous loci in the population. With a multimodal fitness function, the diploid GA gives both greater diversity and improved performance over that of the haploid GA. Moreover, it does so without the need for mutation. When applied to the oscillating 0-1 knapsack problem, the diploid GA presented herein outperforms both the haploid GA and the triallelic diploid GA of Goldberg and Smith in adjusting to periodic, large changes in fitness and recalling previous problem solutions. Although the runtime differential between the diploid and haploid GAs increases with increasing string length and population size, it does so at a linear, rather than an exponential rate.

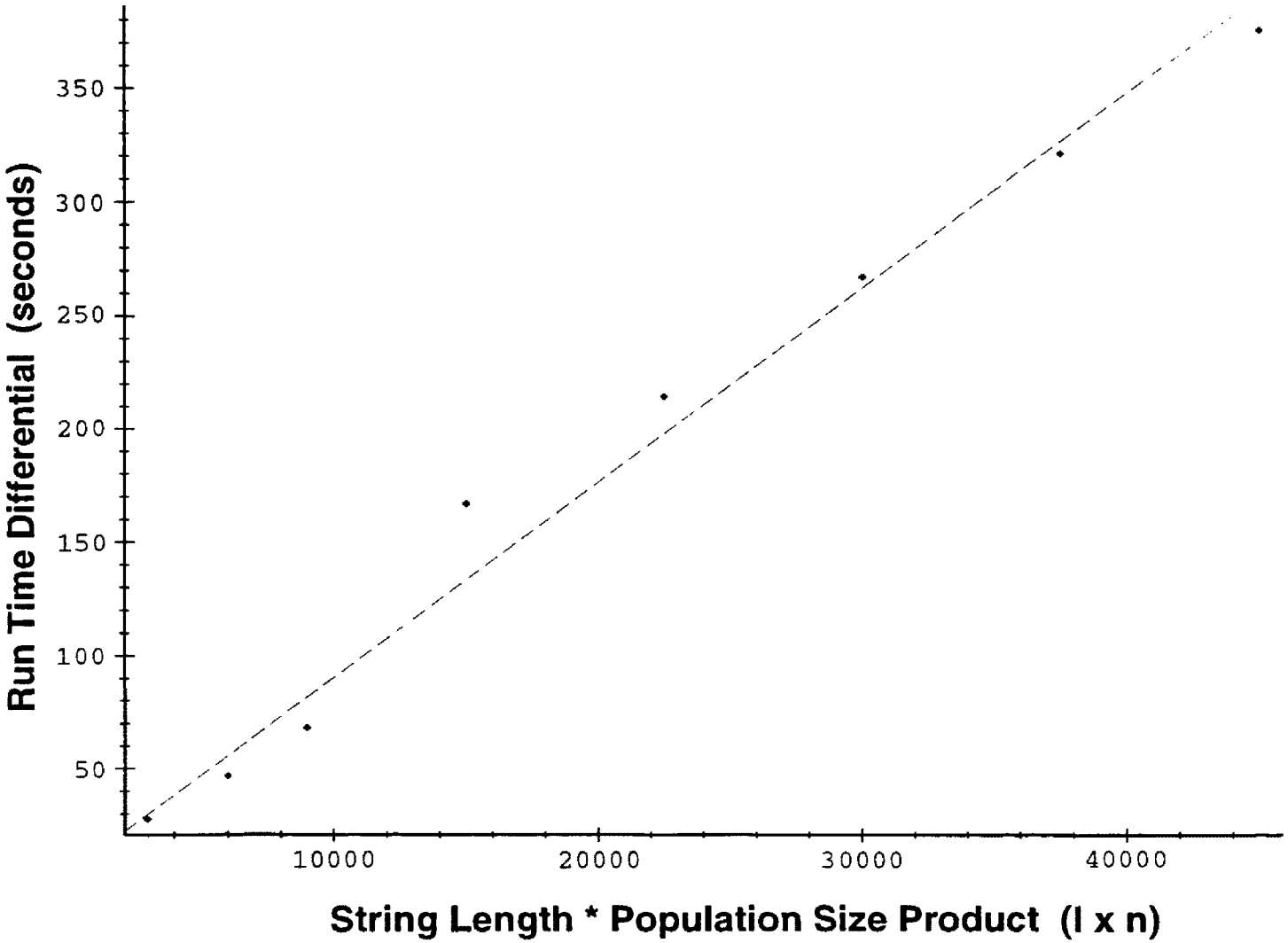


Figure 6.14: Runtime Differential, diploid - haploid

Chapter 7

Conclusions

From the preceding chapters, we arrive at the following conclusions:

1. The diploid model has an allele recursion equation which is inherently more complex than the corresponding equation for the haploid model. This fact, along with biological observations, suggests that diploid populations are capable of exhibiting more complex behavior than haploid ones.

2. Two alleles are sufficient to provide overdominance and thus globally stable polymorphisms in diploid populations, given the proper assignment of fitnesses.

3. Adapting a diploid genome to a haploid fitness function requires variable heterozygote fitnesses in order to guarantee overdominance for arbitrary haploid fitness values.

4. A diploid model with variable heterozygote fitnesses can be realized as a practical GA that exhibits the properties of overdominance and globally stable polymorphisms.

5. The diploid GA is able to introduce and maintain greater population diversity to prevent (or at least mitigate) the problem of premature convergence.

6. On a highly multimodal, deceptive fitness function, the diploid GA maintained greater population diversity and achieved better fitness results than the haploid GA. While the haploid GA converged to local optima for all runs, the diploid GA found the global optimum for all runs.

7. While the mutation operator gives the haploid GA a means to introduce diversity into the population, it is an undirected method that may have unwanted side effects. High mutation rates are usually deleterious to GA performance.

8. The heterozygote fitness bonus of the diploid GA appears to provide and maintain population diversity without large negative effects on performance.

9. The diploid GA presented herein outperforms both the haploid GA and the triallelic diploid GA of Goldberg and Smith in tests with an oscillating 0-1 knapsack problem.

We have achieved the objectives of introducing greater population diversity, preventing (or in some cases mitigating) the problem of premature convergence, and improving GA performance in complex problem domains such as multimodal and nonstationary fitness landscapes. Based on the wealth of theory available in the field of population genetics and the fact that GAs already borrow heavily from some of this theory, there appears to be great potential in using biological analogues to further GA research.

Bibliography

[1] Beasley, David, Bull, David R., and Martin, Ralph R. "An Overview of Genetic Algorithms: Part 1, Fundamentals", *University Computing*, 15(2) pp. 58-69, 1993.

[2] Beasley, David, Bull, David R., and Martin, Ralph R. "An Overview of Genetic Algorithms: Part 2, Research Topics", *University Computing*, 15(4) pp. 170-181, 1993.

[3] Fisher, R.A. *The Genetical Theory of Natural Selection*, second edition. New York: Dover Press, 1958.

[4] Gillett, P. *Calculus and Analytic Geometry*, second edition. Lexington, MA: D.C. Heath and Company, 1984.

[5] Goldberg, David E. and Smith, Robert E. " Nonstationary Function Optimization Using Genetic Algorithms with Dominance and Diploidy", *Proceedings of the Second International Congress on Genetic Algorithms*, pp. 59-68, July, 1987.

[6] Golberg, David E. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading, MA: Addison-Wesley, 1989

[7] Hartl, D. L. & Clark, A. G. *Principles of Population Genetics*, second edition. Sinauer Associates, Inc., 1989.

[8] Hollstien, R.B. "Artificial Genetic Adaptation in Computer Control Systems", *Dissertation Abstracts International*, 32(3), 1510B, 1971.

[9] Hunter, A. "Introduction to Genetic Algorithms", *Sugal V2.0 User Manual*, Section 2, University of Sunderland, England, 1995.

[10] Karlin, S. and Liberman, U. "The Two-Locus Multi-Allele Additive Viability Model", *Journal of Mathematical Biology, 5*, 201-211, 1978.

[11] Karlin, S. and Liberman, U. "Representation of Nonepistatic Selection Models and Analysis of Multilocus Hardy-Weinberg Equilibrium Configurations", *Journal of Mathematical Biology, 7*, 353-374, 1979.

[12] Karlin, S. and Liberman, U. "Global Convergence Properties in Multilocus Viability Selection Models: The Additive Model and the Hardy-Weinberg Law", *Journal of Mathematical Biology, 29*, 161-176, 1990.

[13] Kingman, J.F.C. "A Mathematical Problem in Population Genetics", *Proc. Cambridge Phil. Soc., 57*, 574-582, 1961.

[14] Lancaster, P. and Tismenetsky, M. *The Theory of Matrices*, second edition. Academic Press, 1985.

[15] Lewontin, R.C., Ginzburg, L.R., and Tuljapurkar, S.D. "Heterosis as an explanation for large amounts of genic polymorphism", *Genetics, 88*, 149-170, 1978.

[16] Mitchell, M. *An Introduction to Genetic Algorithms.* MIT Press, 1996.

[17] Nagylaki, T. *Introduction to Theoretical Population Genetics.* Springer-Verlag, 1992.

[18] Ng, Khim Peow and Wong, Kok Cheong "A New Diploid Scheme and Dominance Change Mechanism for Nonstationary Function Optimization", *Proceedings of the Sixth International Congress on Genetic Algorithms*, pp. 159-167, July, 1995.

[19] Vose, M. D. *The Simple Genetic Algorithm: foundations and theory.* MIT Press, (to appear).

[20] Vose, M. D. "Formalizing Genetic Algorithms", *Proc. IEEE wksp. on G.A.s, N.N.s, & S.A. applied to problems in Signal & Image Processing*, Glasgow, U.K., May 1990.