To the Graduate Council:
I am submitting herewith a thesis written by Mahendra Duwal Shrestha entitled "Analysis and Simulation Of A Simple Evolutionary System." I have examined the final paper copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Computer Science.

Michael D. Vose, Major Professor

We have read this thesis
and recommend its acceptance:

Michael D. Vose

Hairong Qi

Judy D. Day

Accepted for the Council:

Dixie Thompson

Vice Provost and Dean of the Graduate School

To the Graduate Council:
I am submitting herewith a thesis written by Mahendra Duwal Shrestha entitled "Analysis and Simulation Of A Simple Evolutionary System." I have examined the final electronic copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Computer Science.

Michael D. Vose, Major Professor

We have read this thesis
and recommend its acceptance:

Michael D. Vose

Hairong Qi

Judy D. Day

Accepted for the Council:

Dixie Thompson

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

Analysis and Simulation Of A Simple Evolutionary System

A Thesis Presented for

The Master of Science

Degree

The University of Tennessee, Knoxville

Mahendra Duwal Shrestha

August 2016

dedication...

Acknowledgements

I would like to thank...

Some quotation...

Abstract

Abstract text goes here...

Contents

List of Tables

List of Figures

Chapter 1

Introduction

This thesis begins with notation that is used throughout this document.

1.1 Notation

Some standard mathematical notations as well as some non-standard mathematical notations are introduced here.

Angle brackets $\langle \cdots \rangle$ denote a tuple which is to be regarded as a column vector. The column vector of all 1s is denoted by $\mathbf{1}$. Transpose is indicated with superscript T. The standard vector norm is $\|x\| = \sqrt{x^T x}$. Modulus (or absolute value) is denoted by $|\cdot|$. When S is a set, $|S|$ denotes the cardinality of S.

The notation $O(f)$ denotes a function g (with similar domain and codomain as f), such that pointwise $g \leq cf$ for some constant c. The notation $\theta(f)$ is a function g such that pointwise $c_0 f \leq g \leq c_1 f$ for some constants c_0 , c_1. Curly brackets $\{\cdots\}$ are used as grouping symbols and to specify both sets and multisets. Square brackets $[\cdots]$ are, besides their standard use as specifying a closed interval of real numbers, used to denote an indicator function: if *expr* is an expression which may be true or false, then $[expr]$ denotes 1 if *expr* is true, and 0 otherwise; $[\cdots]$ is sometimes referred to as an *Iverson bracket*.

The supremum is the least upper bound, and is denoted by sup. The infimum is the greatest lower bound, and is denoted by inf.

The set of length ℓ binary strings is denoted by $\mathcal{R}$. It is a commutative ring under component-wise addition and multiplication modulo 2. If $x \in \mathcal{R}$, then it may be regarded as the vector $x = \langle x_0, x_1, \cdots, x_{\ell-1} \rangle$. Denote the additive identity of $\mathcal{R}$ by $\mathbf{0}$ and the multiplicative identity by $\mathbf{1}$. Let $\overline{g}$ abbreviate $\mathbf{1} + g$. Except when explicitly indicated otherwise, operations acting on elements of $\mathcal{R}$ are as defined in this paragraph. In particular, $g\overline{g} = \mathbf{0} = g + g$, $g^2 = g$, $g + \overline{g} = \mathbf{1}$ for all $g \in \mathcal{R}$.

1.2 Background

The genetic algorithm (GA) is inspired by nature, and seeks to evolve useful constructs. It is typically population based, and proceeds over a number of generations to evolve solutions to problems not yielding to other known methods. Basic elements of a GA are: selection according to fitness, crossover, and random mutation (see Mitchell (1999)). In the simplest case, populatoin members are fixed-length binary strings. The fitness function assigns a score (fitness) to the elements (chromosomes) of the current population.

Selection: select population members in the current population for reproduction; those with higher fitness are more likely to be selected to reproduce.

Crossover: with some probability (the crossover rate), choose a random point in two parents (population members selected for reproduction) and exchanges subsequences after that point to create two offspring.

Mutation: flip bits of an individual with some small probability, the mutation rate.

Figure 1.1 shows the procedural flow of a basic finite population genetic algorithm.

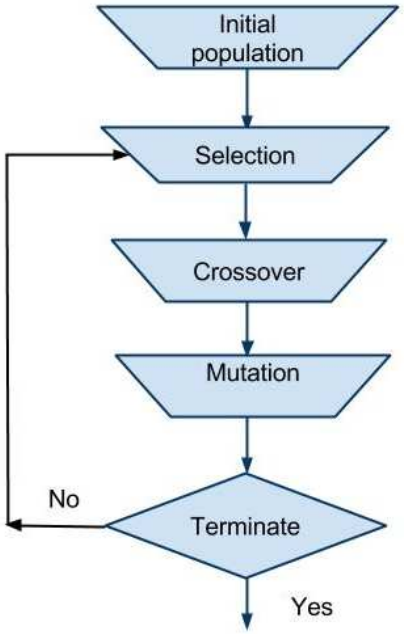


Figure 1.1: Finite GA

A simple Holland style genetic algorithm (see Holland (1992)): Each iteration of

1. Start with some initial population P containing r binary strings of length ℓ.
2. Choose (with replacement) parents u and v from the current population P (using any selection scheme).
   a. Cross u and v to produce children u' and v'.
   b. Mutate u' and v' with some probability to produce u'' and v''.
   c. Keep, with uniform probability, one of u'' and v'' for the next generation
3. If the next generation contains fewer than r members, repeat step 2.
4. Replace P by the new generation formed and go to step 2.

this process produces a generation. The process is repeated until the system stops to improve or some threshold is met.

The infinite population GA models a population as a probability vector $\boldsymbol{p}$ where component $\boldsymbol{p}_j$ can be interpreted as the proportion of string j in the population. If $\mathcal{G}$ is the function mapping infinite population $\boldsymbol{p}$ to the next generation, $\mathcal{G}(\boldsymbol{p})$ is a

probability vector such that

$$\mathcal{G}(\boldsymbol{p})_j \;=\; \text{the probability that string } j \text{ occurs in the next generation.}$$

The evolution of infinite population $\boldsymbol{p}$ is the sequence

$$\boldsymbol{p} \to \mathcal{G}(\boldsymbol{p}) \to \mathcal{G}(\mathcal{G}(\boldsymbol{p})) \to \cdots$$

Several people working in the 1950s and the 1960s – like Box (1957), Friedman (1959), Bledsoe (1961), Bremermann (1962), and Reed, Toombs and Baricelli (1967) – developed evolution-inspired algorithms, but little attention or theoretical analysis was given to them (see Mitchell (1999)). Genetic algorithms were popularized by Holland and his colleagues in the 1960s and the 1970s. Holland introduced a population-based algorithm with crossover and mutation, and promoted his schema theorem (see Holland (1992)) as a theoretical means to analyze on genetic algorithm dynamics. Holland's Schema theorem provides a lower bound for schema survival in next generation. * The schema theorem is an inequality however, and can not predict which strings are expected in the next generation. Bethke (see Bethke (1980)) gave equations computing the expected number of any string in the next generation. Goldberg (see Goldberg (1987)) used such equations to model the evolutionary trajectory of a two bit GA under crossover and proportional selection. Vose and Liepins (see Vose and Liepins (1991)) simplified and extended these equations by integrating mutation into the recombination of arbitrarily long binary strings. Although their model computes infinite population trajectories, given a *finite* population represented by vector $\boldsymbol{p}$ (component $\boldsymbol{p}_i$ is the proportion of string i in the *finite population*), the infinite population model computes the expected proportion

*A schema is a template that identifies a set of strings in the population with similarities at certain string positions; it is made up of 1s, 0s, and $*$s where $*$ is the 'don't care' symbol that matches either 0 or 1.

$\mathcal{G}(\boldsymbol{p})_i$ of string i in the next generation. This is perhaps the most direct connection between the infinite population model and a finite population GA.

Nix and Vose (see Nix and Vose (1992)) explored issues regarding the relationship between the finite population GA and the infinite population model. For a positive mutation rate, a finite population GA will form an ergodic Markov chain, visiting every state infinitely often in the long run. The short term trajectory followed by a finite population is related to the evolutionary path determined by the infinite population model. For large populations, the short term trajectory follows closely and with large probability, that path predicted by the infinite population model.

Vose compiled and extended previous work regarding the infinite population model in the book *Simple Genetic Algorithm: Foundations and Theory* (see Vose (1999)). In particular, he discussed how the Walsh transform can be applied to increase computational efficiency in calculating the infinite population model. There have been previous applications of the Walsh transform to GAs. Bethke first introduced the idea of using Walsh transforms to analyse GA fitness funcitons in terms of schemata (see Bethke (1980)). The idea was further developed in papers by Goldberg (see Goldberg (1989a), Goldberg (1989b)). However, such usage did not apply Walsh transforms to crossover, to mutation, or to any of their associated mathematical objects. In contrast, Vose and Liepins applied the Walsh transform directly to mutation and recombination, and proved that the twist M^* of the mixing matrix M is triangularized by the Walsh transform, and related eigenvalues of M^* to the stability of fixed points of $\mathcal{G}$ (see Vose and Liepins (1991)).[†] In a related paper, Koehler (see Koehler (1994)) gives a congruence transformation defined by a lower triangular matrix that diagonalizes the mixing matrix (for 1-point crossover and mutation given by a rate) and proved a conjecture of Vose and Liepins concerning eigenvalues of M^*. Koehler, Bhattacharyya and Vose (see Koehler et al. (1997)) applied the Fourier

[†]The mixing matrix M has rows and columns indexed by chromosomes; entry $M_{i,j}$ is the probability that mixing parents i and j (mixing is the combined effect of crossover and mutation) will produce a child having all bits zero. The twist (M^*) of the mixing matrix M is defined by $(M^*)_{i,j} = M_{i+j,i}$.

transform in generalizing results established for binary GAs to strings over an alphbet of cardinality c (in the binary case, the Fourier transform is the Walsh transform). From a computational perspective, a major contribution of Vose and Wright (see Vose and Wright (1998)) was demonstrating that the mixing matrix is sparse in the Walsh basis, and the computational efficiency of computing $\mathcal{G}(\boldsymbol{p})$ can thereby be improved from $O(8^\ell)$ to $O(3^\ell)$ where ℓ is the chromosome length. The cost of moving from standard coordinates to the Walsh basis need not be a bottleneck; the fast Walsh transform (see Shanks (1969)) does that in $O(\ell 2^\ell)$ time.

1.3 Random Heuristic Search

The work presented in this thesis is based on *Random Heuristic Search (RHS)*, a general search method, defined upon the central concept of state and transition between states (see Vose (1999)). The simple genetic algorithm is a particular type of RHS. An instance of *RHS* is an initial collection of elements P (referred to as the initial population) chosen from some search space Ω, together with a stochastic transition rule τ, which from P will produce another collection P^*; iterating τ produces a sequence of generations.

Let n be the cardinality of Ω, let $\mathbf{1}$ denote the column vector of all 1s. The *simplex* is the set of population descriptors:

$$\Lambda = \{x = \langle x_0, ..., x_{n-1}\rangle : \mathbf{1}^T x = 1, x_j \geq 0\}$$

Element $\boldsymbol{p} \in \Lambda$ corresponds to a population; p_j = the proportion in the population of the jth element of Ω. The cardinality of each population is a constant r, called the population size. Given r, a population descriptor $\boldsymbol{p}$ unambiguously determines a population.

Given current population vector $\boldsymbol{p}$, the next population vector $\tau(\boldsymbol{p})$ cannot be predicted with certainty because τ is stochastic; it results from r independent,

identically distributed random choices. Let $\mathcal{G} : \Lambda \to \Lambda$ be a function that maps current population vector $\boldsymbol{p}$ to a vector whose ith component is the probability that the ith element of Ω is chosen. Thus, $\mathcal{G}(\boldsymbol{p})$ specifies the distribution from which the aggregate of r choices forms the subsequent generation. The probability that population $\boldsymbol{q}$ is the next population vector given current population (vector) $\boldsymbol{p}$ is (see Vose (1999))

$$\begin{aligned} Q_{\boldsymbol{p},\boldsymbol{q}} &= r! \prod \frac{(\mathcal{G}(\boldsymbol{p})_j)^{r\boldsymbol{q}_j}}{(r\boldsymbol{q}_j)!} \\ &= \exp\{-r \sum \boldsymbol{q}_j \log \frac{\boldsymbol{q}_j}{\mathcal{G}(\boldsymbol{p})_j} - \sum (\log \sqrt{2\pi r \boldsymbol{q}_j} + \frac{1}{12r\boldsymbol{q}_j + \theta(r\boldsymbol{q}_j)}) \\ &\quad + O(\log r)\} \end{aligned} \tag{1.1}$$

where summation is restricted to indices for which $\boldsymbol{q}_j > 0$ and θ is a function such that $0 < \theta < 1$. Each random vector in the sequence $\boldsymbol{p}, \tau(\boldsymbol{p}), \tau^2(\boldsymbol{p}), ...$ depends only on the value of the preceding one, which is a special situation. The sequence forms a Markov chain with transition matrix Q. The conceptualization of RHS can be replaced by a Markov chain model which makes no reference to sampling Ω; from current population $\boldsymbol{p}$, produce $\boldsymbol{q}$ with probability $Q_{\boldsymbol{p},\boldsymbol{q}}$. The expected next generation $\mathcal{E}(\tau(\boldsymbol{p}))$ is $\mathcal{G}(\boldsymbol{p})$ (see Vose (1999)). The expression

$$\sum \boldsymbol{q}_j \log \frac{\boldsymbol{q}_j}{\mathcal{G}(\boldsymbol{p})!}$$

in (1.1) is the *discrepancy* of $\boldsymbol{q}$ with respect to $\mathcal{G}(\boldsymbol{p})$. It is a measure of how far $\boldsymbol{q}$ is from the expected next population $\mathcal{G}(\boldsymbol{p})$. Discrepancy is nonnegative and is zero only when $\boldsymbol{q}$ is the expected next population. Hence the first factor

$$\exp\{-r \sum \boldsymbol{q}_j \log \frac{\boldsymbol{q}_j}{\mathcal{G}(\boldsymbol{p})_j}\}$$

in (1.1) indicates the probability that $\boldsymbol{q}$ is the next generation decays exponentially, with constant r, as the discrepancy between $\boldsymbol{q}$ and $\mathcal{G}(\boldsymbol{p})$ increases. The expression

$$\sum(\log\sqrt{2\pi r\boldsymbol{q}_j} + \frac{1}{12r\boldsymbol{q}_j + \theta(r\boldsymbol{q}_j)})$$

measures the *dispersion* of the population vector $\boldsymbol{q}$ and the second factor in (1.1)

$$\exp\{-\sum(\log\sqrt{2\pi r\boldsymbol{q}_j} + \frac{1}{12r\boldsymbol{q}_j + \theta(r\boldsymbol{q}_j)})\}$$

indicates the probability that $\boldsymbol{q}$ is the next generation decays exponentially with increasing dispersion. As Vose stated in his book (see Vose (1999)):

> "The combined effect of the two influences of discrepancy and dispersion is that random heuristic search favors a less disperse population near the expected next generation. In particular, if the current population is near the expected next generation, then the first factor does not contribute a strong bias for change. When $\mathcal{G}(\boldsymbol{p})$ is nearly the initial population $\boldsymbol{p}$, the influence of discrepancy favors $\boldsymbol{p}$ as the next generation since the alternatives, being lattice points, are constrained to be some distance away from the expected next generation. This phenomenon is expressed quantitivelty by theorem 3.4. Moreover, the second factor may exert a stabilizing effect provided the current population has low dispersion compared to the alternatives."

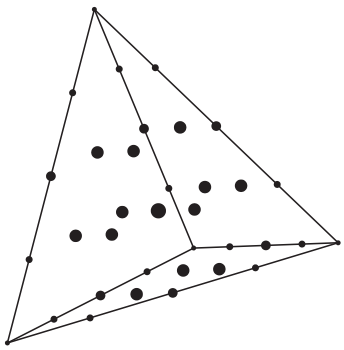


Figure 1.2: Population points

Figure 1.2 illustrates population points in a simplex for $\ell = 2, r = 4$. Finite populations are represented by dots, where smaller dots have lower dispersion and larger dots have higher dispersion. The diagram also illuminates that finite populations are constrained to occupy lattice points within Λ. As population size $r \to \infty$, the lattice points become dense in Λ, which corresponds to the fact that an infinite population can be (represented by) any point of Λ.

The variance of the next generation (with respect to the expected population) (see Vose (1999)) is

$$\mathcal{E}(\|\tau(\boldsymbol{p}) - \mathcal{G}(\boldsymbol{p})\|^2) = \frac{1 - \|\mathcal{G}(\boldsymbol{p})\|^2}{r} \tag{1.2}$$

It follows from Chebyshev's inequality (see Wikipedia (2016a)) that

$$P(\|\tau(\boldsymbol{p}) - \mathcal{G}(\boldsymbol{p})\| \geq \epsilon) \leq \frac{1 - \|\mathcal{G}(\boldsymbol{p})\|^2}{r\epsilon^2} \tag{1.3}$$

where P above denotes probability and $\epsilon > 0$ is arbitrary.

Let $f(r)$ be a function which grows arbitrarily slowly, such that

$$\lim_{r\to\infty} f(r) = \infty$$

and

$$\lim_{r\to\infty} f(r)/\sqrt{r} = 0.$$

If

$$\epsilon = f(r)/\sqrt{r} \tag{1.4}$$

then (1.3) becomes

$$\lim_{r\to\infty} P(\|\tau(\boldsymbol{p}) - \mathcal{G}(\boldsymbol{p})\| \geq \epsilon) \;\leq\; \lim_{r\to\infty} \frac{1 - \|\mathcal{G}(\boldsymbol{p}\|^2)}{f(r)^2} \;=\; 0$$

Therefore, $\tau(\boldsymbol{p})$ converges in probability to $\mathcal{G}(\boldsymbol{p})$ as the population size increases, and τ corresponds to $\mathcal{G}$ in the infinite population case. Moreover, 1.4 suggests that the expected distance between finite and infinite population in the next generation might decrease as $1/\sqrt{r}$.

In figure 1.2, finite population points can be only at certain points, but infinite population points can be anywhere in the simplex. Theorem 3.1 in 'The Simple Genetic Algorithm: Foundations and Theory' states (see Vose (1999)):

> *If $\boldsymbol{p}, \boldsymbol{q} \in \Lambda$ are arbitrary population vectors for population size r, and $\boldsymbol{\xi}$ denotes an arbitrary element of Λ, then*
>
> $$\inf_{\boldsymbol{p}\neq\boldsymbol{q}} \|\boldsymbol{p} - \boldsymbol{q}\| \;=\; \sqrt{2}/r \tag{1.5}$$
> $$\sup_{\boldsymbol{\xi}} \inf_{\boldsymbol{p}} \|\boldsymbol{\xi} - \boldsymbol{p}\| \;=\; O(1/\sqrt{r}) \tag{1.6}$$
>
> *where the constant (in the "big oh") is independent of the dimension n of Λ.*

From 1.6, the distance between an infinite population $\boldsymbol{\xi}$ and finite population p is $O(1/\sqrt{r})$. This suggests that the distance between $\tau(\boldsymbol{p})$ and $\mathcal{G}(\boldsymbol{p})$ might decrease as $1/\sqrt{r}$.

Let η be the random variable $\|\boldsymbol{q} - \mathcal{G}(\boldsymbol{p})\|$, and let $\phi(x) = x^2$. It follows from Jensen's Inequality (see Wikipedia (2016b)) that since ϕ is a convex function,

$$\phi(\mathcal{E}(\eta))) \leq \mathcal{E}(\phi(\eta))$$

Therefore,

$$\mathcal{E}(\|\boldsymbol{q} - \mathcal{G}(\boldsymbol{p})\|) \;=\; \mathcal{E}(\eta) \leq \sqrt{\mathcal{E}(\eta^2)} \;=\; \frac{\sqrt{1 - \|\mathcal{G}(\boldsymbol{p})\|^2}}{\sqrt{\boldsymbol{r}}} \tag{1.7}$$

This suggests that the distance between $\tau(\boldsymbol{p})$ and $\mathcal{G}(\boldsymbol{p})$ might decrease as $1/\sqrt{r}$.

1.4 Research Problems

Equations 1.4, 1.6, and 1.7 all suggest that the distance between $\tau\boldsymbol{p}$ and $\mathcal{G}(\boldsymbol{p})$ might decrease as $1/\sqrt{r}$. The first research question to consider is whether that rate of decrease is exhibited in practice. We investigate the rate of decrease with experiments in Chapter 2.

An instance of RHS is focused if $\mathcal{G}$ is continuously differentiable, and for every $\boldsymbol{p} \in \Lambda$ the sequence

$$\boldsymbol{p}, \mathcal{G}(\boldsymbol{p}), \mathcal{G}^2(\boldsymbol{p}), ...$$

converges. In this case, $\mathcal{G}$ is also called focused, and the path determined by following at each generation what τ is expected to produce will lead to some fixed point ω

$$\mathcal{G}(\omega) = \lim_{n \to \infty} \mathcal{G}^n(\boldsymbol{p}) = \omega.$$

When specialized to a simple GA (the details are explained in Chapter 2), it turns out that $\mathcal{G}$ is focused under certain conditions, but under other conditions the sequence $\boldsymbol{p}, \mathcal{G}(\boldsymbol{p}), \mathcal{G}^2(\boldsymbol{p}), ...$ converges to a periodic orbit which oscillates between fixed points of $\mathcal{G}^2$ (see Vose (1999)). If a finite population GA follows the infinite population GA closely, and if infinite populations oscillate under certain conditions, then finite populations might also show oscillating behavior. The second research question to consider is whether such finite population oscillation is exhibited in practice. We investigate finite population oscillation with experiments in Chapter 3.

The third research question concerns the robustness of finite population oscillation. Consider the lattice points in the simple Λ which represent finite populations (for

some fixed population size r) and let $\boldsymbol{P}_j$ denote the jth population represented by the jth lattice point. Let $\boldsymbol{\pi}^k$ be the probability vector having as jth component the probability that $\boldsymbol{P}_j$ is the kth generation. If $\boldsymbol{\pi}^0$ is the initial population distribution, the steady state distribution $\boldsymbol{\pi}$ is given by (see Wikipedia (2016c))

$$\boldsymbol{\pi} = \lim_{k\to\infty} \boldsymbol{\pi}^k = \lim_{k\to\infty} \boldsymbol{\pi}^0 Q^k \tag{1.8}$$

assuming the limit exists. The jth component $\boldsymbol{\pi}_j$ can be interpreted as the proporiton of time that a GA spends in population $\boldsymbol{P}_j$. If transition matrix Q is irreducible[‡] and aperiodic[§], then the Markov chian is ergodic, the steady state distribution $\boldsymbol{\pi}$ exists, and it has positive components (see Minc (1988)). The solution to equation 1.8 satisfies

$$\boldsymbol{\pi} = \boldsymbol{\pi} Q \tag{1.9}$$

where $\boldsymbol{\pi}$ is normalized so that its components sum to one. If GA were to oscillate between two populations $\boldsymbol{P}_i$ and $\boldsymbol{P}_j$, then $\boldsymbol{\pi}_i^k = 1$ (other components are 0) when k is odd, and $\boldsymbol{\pi}_j^k = 1$ (other components are 0) when k is even. Therefore, oscillation should not occur. In Chapter 4, we investigate oscillation behavior of finite populations when the Markov chain is ergodic. The third research question concerns whether finite population oscillation is exhibited in practice when the Markov chain is ergodic.

[‡]A Markov chain is said to be *irreducible* if it is possible to get to any state from any state.
[§]A Markov chain is *aperiodic* if it can return to state i at irregular times.

Chapter 2

Extending A Genetic Algorithm Model To The Diploid Case

This chapter describes a simple Markov model for evolution under the influence of crossing over and mutation; it is a non-overlapping, generational, infinite population model under the assumption of *complete panmixia* (random mating) and no selective pressure. This chapter shows how diploid evolution equations can be represented by haploid equations and can be specialized to Vose's infinite population model, which is a haploid model.

A basic syntactic model for haploid and diploid genomes is first considered. Then the mechanics of how next generation is obtained from current generation are defined abstractly in procedural terms, which serves to motivate the equations governing evolution. Next evolution equations are developed corresponding to the procedural description defining evolution for a population of diploid genomes. Observations concerning the form and symmetry of those equations directly lead to decoupling from the diploid case a haploid model sufficient to determine evolutionary trajectories for the diploid case. Mask based mutation and crossover operators are used to specialize haploid equations to Vose's infinite haploid population model. Analytical and computational simplification resulting from specialization to Vose's infinite population

model are explained and used in experimental simulation and study of convergence of finite population short-term behavior to behavior predicted by infinite population model.

2.1 Model

A haploid genome g is defined syntactically as a length ℓ binary string. A collection of h chromosomes may be modeled by partitioning g into h segments (of arbitrary lengths $\ell_1, \ldots, \ell_h$; thus $\ell = \ell_1 + \cdots + \ell_h$).

A diploid genome $\alpha = \langle \alpha_0, \alpha_1 \rangle$ is likewise defined syntactically as a pair of length ℓ binary strings. Although simple, that syntax is flexible and possesses significant modeling power by means of tailoring partitioning to application. We concentrate on the abstract level, considering the evolution of a non-overlapping, generational, infinite population model assuming panmixia and no selective pressure. We are not concerned with whether and how partitioning is defined as it is irrelevant to the development.

Following Hardy (see Hardy (1908)), the model q^n at generation n is a vector having for component q^n_α the prevalence of diploid α (the probability of selecting α at generation n, assuming unbiased selection).* Ordered diploid $\gamma = \langle \gamma_0, \gamma_1 \rangle$ is produced for generation $n+1$ according to following procedural description.

Assuming independent selection events:

- From parent α — selected with probability q^n_α — obtain gamete γ_0
- From parent β — selected with probability q^n_β — obtain gamete γ_1

Following Gieringer (see Geiringer (1944)), let the transmission function $t_\alpha(g)$ be the probability that gamete g is produced from parental genome α. It follows from the

*The representation here is the conceptual equivalent of Hardy's model.

above that the equation determining the next generation q^{n+1} is

$$q_\gamma^{n+1} \;=\; \sum_\alpha q_\alpha^n \, t_\alpha(\gamma_0) \sum_\beta q_\beta^n \, t_\beta(\gamma_1) \tag{2.1}$$

It should be appreciated that the Mendelian (see Mendel (1865)) laws of segregation[†] and independent assortment[‡] need not be respected by the transmission function.

The right hand side of (2.1) is invariant under interchange of the summation variables α and β, which is equivalent to interchanging γ_0 and γ_1. This symmetry reflects the fact that which haploid of γ is designated as γ_0 is arbitrary,

$$q_{\langle\gamma_0,\gamma_1\rangle}^{n+1} \;=\; q_{\langle\gamma_1,\gamma_0\rangle}^{n+1}$$

The model corresponding to (2.1) is low-level in the sense that it regards $\langle\gamma_0,\gamma_1\rangle$ and $\langle\gamma_1,\gamma_0\rangle$ as distinct when $\gamma_1 \neq \gamma_0$. A higher-level model based on sets is easily obtained,

$$q_{\{\gamma_0,\gamma_1\}} \;=\; \begin{cases} 2q_{\langle\gamma_0,\gamma_1\rangle} & \text{if } \gamma_0 \neq \gamma_1 \\ q_{\langle\gamma_0,\gamma_1\rangle} & \text{otherwise} \end{cases}$$

which is in agreement with Hardy (see Hardy (1908)).

2.2 Reduction

Evolution equation (2.1) may be reduced to the haploid case. Its right hand side is the product of two summations; denote the first by $p_{\gamma_0}^{n+1}$ and the second by $p_{\gamma_1}^{n+1}$ so that

$$q_{\langle\gamma_0,\gamma_1\rangle}^{n+1} \;=\; p_{\gamma_0}^{n+1} \, p_{\gamma_1}^{n+1} \tag{2.2}$$

[†]Alleles of a given locus segregate into separate gametes.
[‡]Alleles of one gene sort into gametes independently of the alleles of another gene.

where for any haploid γ_0,

$$p^{n+1}_{\gamma_0} \;=\; \sum_{\alpha} q^n_\alpha \, t_\alpha(\gamma_0) \tag{2.3}$$

It suffices to determine the evolution of the distributions p^n. Uncoupling p from q using (2.3), and equation (2.2) with superscript n — instantiate the n in (2.2) with $n-1$ — yields the evolution equation

$$\begin{aligned} p^{n+1}_{\gamma_0} \;&=\; \sum_{\alpha_0,\,\alpha_1} q^n_{\langle \alpha_0,\,\alpha_1\rangle} \, t_{\langle \alpha_0,\,\alpha_1\rangle}(\gamma_0) \\ &=\; \sum_{\alpha_0,\,\alpha_1} p^n_{\alpha_0}\, p^n_{\alpha_1} \, t_{\langle \alpha_0,\,\alpha_1\rangle}(\gamma_0) \end{aligned} \tag{2.4}$$

The p^n are in fact distributions; summing equation (2.2) with superscript n yields

$$1 \;=\; \sum_{\alpha} q^n_\alpha \;=\; \sum_{\alpha_0,\,\alpha_1} p^n_{\alpha_0}\, p^n_{\alpha_1} \;=\; \Big(\sum_{\alpha_0} p^n_{\alpha_0}\Big)^2$$

The weighted count of haploid g in generation n is

$$\sum_{\alpha_0,\,\alpha_1} q^n_{\langle \alpha_0,\alpha_1\rangle}([g=\alpha_0]+[g=\alpha_1]) \tag{2.5}$$

$$= \sum_{\alpha_0,\,\alpha_1} p^n_{\alpha_0}\, p^n_{\alpha_1}[g=\alpha_0] + \sum_{\alpha_0,\,\alpha_1} p^n_{\alpha_0}\, p^n_{\alpha_1}[g=\alpha_1] \tag{2.6}$$

$$= 2p^n_g \tag{2.7}$$

Hence the (normalized) prevalence of haploid g in generation n is the g th component of the distribution p^n. Moreover, (2.5) and (2.2) show (for $n>0$) invertibility of the map

$$\pi : \boldsymbol{q}^n \;\longmapsto\; \boldsymbol{p}^n$$

Evolution equation (2.4) in matrix form is

$$p'_g \;=\; p^T M_g \, p \tag{2.8}$$

where current state p (generation n) and next state p' (generation $n+1$) are column vectors, and the g th transmission matrix is

$$\Big(M_g\Big)_{u,v} \;=\; t_{\langle u,v\rangle}(g) \tag{2.9}$$

(vectors and matrices are indexed by haploids — length ℓ binary strings).

2.3 Specialization

This section summarizes from the development in Vose (see Vose (1999)). It specializes the haploid evolution equations in the previous section to a context where mask-based crossing over and mutation operators are used, leading to Vose's infinite population model for Genetic Algorithms. Whereas in previous sections *component* referred to a component of a distribution vector q^n or p^n, in this section a component is either a probability (when when speaking of a component of a distribution vector), or a bit (when speaking of a component of a haploid).

2.3.1 Mutation

Mutation simulates low probability errors in chromosome duplication. Mutation provides a mechanism to inject new strings into the next generation. The symbol $\boldsymbol{\mu}$ denotes mutation distribution describing the probability $\boldsymbol{\mu}_i$ with which $i \in \Omega$ is selected to be a mutation mask. The result of mutating g is $g+i$ with probability $\boldsymbol{\mu}_i$. Mutating g using mutation mask i alters the bits of g in those positions the mutation mask i is 1. If g should mutate to g' with probability ρ, let

$$\boldsymbol{\mu}_{g+g'} \;=\; \rho$$

Given distribution $\boldsymbol{\mu}$, mutation is the stochastic operator sending g to g' with probability $\boldsymbol{\mu}_{g+g'}$. Abusing notation, $\boldsymbol{\mu} \in [0, 0.5)$ is regarded as a *mutation rate* which

implicitly specifies distribution $\boldsymbol{\mu}$ according to rule (see Vose and Wright (1998))

$$\boldsymbol{\mu}_i = (\boldsymbol{\mu})^{\mathbf{1}^T i}(1-\boldsymbol{\mu})^{\ell-\mathbf{1}^T i}$$

2.3.2 Crossover

Crossover refers to crossing over (also termed recombination) between two chromosomes (strings in our case). Crossover like mutation also provides mechanism for injection of new strings into new generation population. Geiringer (see Geiringer (1944)) used crossover masks to generate offsprings from parent chromosomes in absence of mutation and selection. Let $\boldsymbol{\chi}_m$ be the probability distribution with which m is selected to be a crossover mask. Following Geiringer (see Geiringer (1944)), if crossing over u and v should produce u' and v' with probability ρ, let

$$\boldsymbol{\chi}_m \;=\; \rho$$

where m is 1 at components which u' inherits from u, and 0 at components inherited from v. It follows that

$$\begin{aligned} u' &= m\,u + \overline{m}\,v \\ v' &= m\,v + \overline{m}\,u \end{aligned}$$

Given distribution $\boldsymbol{\chi}$, crossover is the stochastic operator which sends u and v to u' and v' with probability $\boldsymbol{\chi}_m/2$.

Abusing notation, $\boldsymbol{\chi}$ can be considered as a *crossover rate* that specifies the distribution $\boldsymbol{\chi}$ given by rule (see Vose and Wright (1998))

$$\boldsymbol{\chi}_i = \begin{cases} \boldsymbol{\chi} c_i & \text{if } i > 0. \\ 1 - \boldsymbol{\chi} + \boldsymbol{\chi} c_0 & \text{if } i = 0. \end{cases}$$

where $c \in \Lambda$ is referred to as *crossover type*. Classical crossover types include *1-point crossover* and *uniform crossover*. For *1-point crossover*,

$$c_i = \begin{cases} 1/(\ell - 1) & \text{if } \exists k \in (0, \ell).i = 2^k - 1. \\ 0 & \text{otherwise.} \end{cases}$$

and for uniform crossover, $c_i = 2^{-\ell}$.

2.3.3 Mixing Matrix

The combined action of mutation and crossover is referred to as *mixing*. The *mixing matrix* M is the transmission matrix corresponding to the additive identity of $\mathcal{R}$ is

$$M \;=\; M_{\mathbf{0}}$$

Crossover and mutation are defined in a manner respecting arbitrary partioning and arbitrary linkage to preserve the ability to endow abstract syntax with specialized semantics. Groups of loci can mutate and crossover with arbitrarily specified probabilities as disscussed in above sections. For mutation distribution $\boldsymbol{\mu}$ and crossover distribution $\boldsymbol{\chi}$, then transmission function can be expressed as (see Vose and Wright (1998))

$$t_{\langle u,v \rangle}(g) \;=\; \sum_{i \in \mathcal{R}} \sum_{j \in \mathcal{R}} \sum_{k \in \mathcal{R}} \boldsymbol{\mu}_i \boldsymbol{\mu}_j \, \frac{\boldsymbol{\chi}_k + \boldsymbol{\chi}_{\overline{k}}}{2} \, [k(u+i) + \overline{k}(v+j) \;=\; g] \tag{2.10}$$

Here a child gamete g is produced via mutation and then crossover (which are operators that commute).

The mixing matrix M is a fundamental object, because (2.10) implies that evolution equation (2.8) can be expressed in the form

$$p'_g \;=\; (\sigma_g p)^T M \, (\sigma_g p) \tag{2.11}$$

where the permutation matrix σ_g is defined by component equations

$$(\sigma_g)_{u,v} \;=\; [u+v=g]$$

2.4 Walsh Transorm

If $n \in \mathcal{R}$ and $t \in \mathcal{R}$, and N is cardinality of $\mathcal{R}$, the Walsh matrix is defined by

$$W_{n,t} = N^{-1/2}(-1)^{n^T t}$$

where $N^{-1/2}$ is normalization factor.

The matrix is symmetric, i.e.,

$$W_{n,t} = W_{n,t}$$

and it has entries satisfying

$$W_{n,t+k} = N^{1/2} W_{n,t} W_{n,k}; \qquad k \in \mathcal{R}.$$

The practical importance of this symmetry is that the transform and inverse are the same mathematical operation, and *Walsh matrix* is its own inverse,

$$W = W^{-1}.$$

Given vector w and matrix A, let $\widehat{w}$ and $\widehat{A}$ denote the Walsh transform of w and A respectively. Then $\widehat{w} = Ww$ and $\widehat{A} = WAW$ (see Beauchamp (1975)).

2.4.1 Fast Walsh Transform

Computation of the discrete Walsh transform given by equation (**??**) might take n^2 operations (addition or subtraction) if implemented naively. An algorithm using

matrix factorization techniques is found to perform the transformation in $n \log_2 n$ operations. This algorithm is the Fast Walsh transform (FWT). Shanks (see Shanks (1969)) described FWT algorithm which is analogous to Cooley-Tukey algorithm (see Cooley and Tukey (1965)) for fast Fourier transformation. The algorithm for FWT can be translated into pseudocode as:

```
1: procedure FWT
2:     n = 2^d ← size of array X where d is positive integer
3:     for i = 0 to d − 1 do
4:         m = n/2^i
5:         z = m/2
6:         for j = 0 to 2^i − 1 do
7:             for k = 0 to z − 1 do
8:                 t1 = m × j + k
9:                 t2 = m × j + z + k
10:                a = X[t1]
11:                b = X[t2]
12:                X[t1] = a + b
13:                X[t2] = a − b
14:            end for
15:        end for
16:    end for
17:    return X
18: end procedure
```

Algorithm 1: FWT pseudocode

2.4.2 Walsh Transform Adaptation

We adapt Walsh transform methods which have already been established for Vose's haploid model (see Vose and Wright (1998)) for computing evolutionary trajectories. Adaptation of Walsh transformation efficiently models infinite diploid population evolution, making feasible comparisons between finite and infinte diploid population short-term evolutionary behavior. Evolution equation (2.11), specialized to Vose's

infinite population model without selection is

$$p'_g \;=\; (\sigma_g p)^T M\,(\sigma_g p)$$

where the permutation matrix σ_g is defined by component equations

$$(\sigma_g)_{u,v} \;=\; [u+v=g]$$

The Walsh matrix W is defined by the component equations

$$W_{u,v} \;=\; 2^{-\ell/2}(-1)^{u^T v}$$

where the subscripts u, v (which belong to $\mathcal{R}$) on the left hand side are interpreted on the right hand side as column vectors in $\mathbb{R}^\ell$. Columns of W form the orthonormal basis — the *Walsh basis* — which simultaneously diagonalizes the σ_g. Expressed in the Walsh basis (see Vose and Wright (1998)), the mixing matrix takes the form

$$\widehat{M}_{u,v} \;=\; 2^{\ell-1}\,[uv=\mathbf{0}]\,\widehat{\boldsymbol{\mu}}_u\widehat{\boldsymbol{\mu}}_v \sum_{k\in\overline{u+v}\mathcal{R}} \boldsymbol{\chi}_{k+u}+\boldsymbol{\chi}_{k+v} \tag{2.12}$$

and equation (2.11) takes the form

$$\widehat{p}'_g \;=\; 2^{\ell/2} \sum_{i\in g\mathcal{R}} \widehat{p}_i\,\widehat{p}_{i+g}\,\widehat{M}_{i,i+g} \tag{2.13}$$

where $g\mathcal{R}=\{gi\,|\,i\in\mathcal{R}\}$ (for any $g\in\mathcal{R}$).

The mapping from generation n to generation $n+1$, determined in natural coordinates by equation (2.8) in terms of the transmission function (2.9), and given in Walsh coordinates by equation (2.13) in terms of the mixing matrix (2.12), is Markovian; the next state p' depends only upon the current state p. Let $\mathcal{M}$ represent the mixing transformation,

$$p' \;=\; \mathcal{M}(p) \tag{2.14}$$

and let $\mathcal{M}^n(p)$ denote the n-fold composition of $\mathcal{M}$ with itself; thus generation $n+1$ is described by

$$p^{n+1} \;=\; \mathcal{M}^n(p^1)$$

where $p^1 \;=\; \pi(q^1)$. We have little to say about the matrix of the Markov chain corresponding to the mixing transformation $\mathcal{M}$, because it is uncountable; each state is a distribution vector p describing a population. However, that is not an obstacle to computing evolutionary trajectories; (2.14) can be computed in Walsh coordinates relatively efficiently via (2.12) and (2.13).

2.5 Distance

Let vector $\boldsymbol{f}$ represent a finite diploid population; component $\boldsymbol{f}_\alpha$ is the prevalence of diploid α. Let the support $S_{\boldsymbol{f}}$ of $\boldsymbol{f}$ be the set of diploids occurring in the population represented by $\boldsymbol{f}$,

$$S_{\boldsymbol{f}} \;=\; \{\alpha \,|\, \boldsymbol{f}_\alpha > 0\}$$

Let $\boldsymbol{q}$ similarly represent an infinite diploid population (see section 2.1). As points in $\mathbb{R}^{2^\ell \times 2^\ell}$, the Euclidean distance between $\boldsymbol{f}$ and $\boldsymbol{q}$ is

$$\|\boldsymbol{f} - \boldsymbol{q}\| \;=\; \sum_\alpha{}^{\frac{1}{2}} (\boldsymbol{f}_\alpha - \boldsymbol{q}_\alpha)^2$$

Whereas a naive computation of this distance involves $2^\ell \cdot 2^\ell$ terms, leveraging equation (2.2) can significantly reduce the number of terms involved. Note that

$$\|\boldsymbol{f} - \boldsymbol{q}\|^2 \;=\; \sum_{\alpha \notin S_{\boldsymbol{f}}} (\boldsymbol{f}_\alpha - \boldsymbol{q}_\alpha)^2 + \sum_{\alpha \in S_{\boldsymbol{f}}} (\boldsymbol{f}_\alpha - \boldsymbol{q}_\alpha)^2 \tag{2.15}$$

Using equation (2.2) — $\boldsymbol{q}_\alpha = \boldsymbol{p}_{\alpha_0} \boldsymbol{p}_{\alpha_1}$ (suppressing superscripts to streamline notation) — together with the fact that $\boldsymbol{f}_\alpha = 0$ in every term of the first sum above, the first

sum reduces to

$$\begin{aligned}\sum_{\langle\alpha_0,\alpha_1\rangle\notin S_{\boldsymbol{f}}}(\boldsymbol{p}_{\alpha_0}\boldsymbol{p}_{\alpha_1})^2 &= \sum_{\langle\alpha_0,\alpha_1\rangle}(\boldsymbol{p}_{\alpha_0})^2(\boldsymbol{p}_{\alpha_1})^2 - \sum_{\langle\alpha_0,\alpha_1\rangle\in S_{\boldsymbol{f}}}\left(\boldsymbol{p}_{\alpha_0}\boldsymbol{p}_{\alpha_1}\right)^2 \\ &= \sum_{g}^{2}(\boldsymbol{p}_g)^2 - \sum_{\alpha\in S_{\boldsymbol{f}}}(\boldsymbol{q}_\alpha)^2 \end{aligned} \tag{2.16}$$

It follows from (2.15) and (2.16) that

$$\begin{aligned}\|\boldsymbol{f}-\boldsymbol{q}\|^2 &= \sum_{g}^{2}(\boldsymbol{p}_g)^2 + \sum_{\alpha\in S_{\boldsymbol{f}}}(\boldsymbol{f}_\alpha-\boldsymbol{q}_\alpha)^2 - \sum_{\alpha\in S_{\boldsymbol{f}}}(\boldsymbol{q}_\alpha)^2 \\ &= \sum_{g}^{2}(\boldsymbol{p}_g)^2 + \sum_{\alpha\in S_{\boldsymbol{f}}}\boldsymbol{f}_\alpha(\boldsymbol{f}_\alpha-2\boldsymbol{q}_\alpha) \end{aligned} \tag{2.17}$$

which involves $2^\ell+|S_{\boldsymbol{f}}|$ terms, assuming that $S_{\boldsymbol{f}}$ is known as a byproduct of computing $\boldsymbol{f}$. Therefore, (2.17) computes distance between finite and infinite population efficiently.

2.6 Simplification

Computations in the haploid case are simplified by equations (2.12) and (2.13) which follow from specializing to Vose's infinite population model and computing in the Walsh basis. Time switching between the standard basis and the Walsh basis is negligible; the fast Walsh transform (in dimension n) has complexity $n\log n$ Shanks (1969).

Only one mixing matrix as opposed to 2^ℓ matrices is needed to compute the next generation; evolution equation (2.13) references the same matrix for every g, whereas evolution equation (2.8) depends upon a different matrix M_g for each choice of g. The matrix is computed by a single sum as opposed to a triple sum; compare equation (2.12) with equation (2.10). Also, the relevant quadratic form is computed with a single sum as opposed to a double sum; computing via (2.13) is linear time in the

size of $g\mathcal{R}$ (for each g) as opposed to the quadratic time computation (for each g) represented by equation (2.8).

From a computational standpoint, the best-case scenario is where recomputation of the matrices mentioned in the previous paragraph is obviated by sufficient memory. The reduction from 2^ℓ matrices to one matrix helps significantly in that regard. To demonstrate this advantage in concrete terms, consider genomes of length $\ell = 14$. Using 2^{14} matrices each of which contains $2^{14} \times 2^{14}$ entries of type `double` requires 32 terabytes, whereas the mixing matrix at 2 gigabytes fits easily within the memory of a laptop. Moreover, for a population size of $N \leq 2^{20}$, the distance computation described in the previous section reduces the number of terms involved by a factor of $2^{28}/(2^{14} + 2^N) > 252$.

2.7 Convergence

This section presents a cursory numerical investigation of the convergence of finite diploid population short-term behaviour to that of the infinite diploid population model as described in section 2 (the underlying haploid model for the infinite population case is described in section 2.1).

Equations (2.2), (2.12), (2.13), (2.17) were employed to efficiently compute the distance

$$d = \|\boldsymbol{f}^n - \boldsymbol{q}^n\|$$

where $\boldsymbol{f}^n$ and $\boldsymbol{q}^n$ represent finite and infinite diploid populations (respectively) at generation $n \in \{1, 2, 4, 8, 16, 32, 64, 128\}$, beginning from a random initial population ($\boldsymbol{f}^0 = \boldsymbol{q}^0$). Genome lengths $\ell \in \{4, 6, 8, 10, 12, 14\}$ and population sizes $N = 2^i$ for integer $0 \leq i \leq 20$ were considered. The crossover distribution $\boldsymbol{\chi}$ corresponds to independent assortment of bits, and the mutation distribution $\boldsymbol{\mu}$ corresponds to

independent bit mutation probability 0.001,

$$\boldsymbol{\chi}_m \;=\; 2^{-\ell}, \qquad \boldsymbol{\mu}_g \;=\; (0.001)^{\mathbf{1}^{\mathrm{T}}g}(0.999)^{\ell-\mathbf{1}^{\mathrm{T}}g}$$

(subscripts above on the left hand side of an equality are interpreted on the right hand side of the equality as column vectors in $\mathbb{R}^\ell$). The finite population case is computed using the itemized procedural definition given in section 2.1; the transmission function (2.10) corresponds to $\boldsymbol{\mu}$ and $\boldsymbol{\chi}$ above (bits mutate independently and are freely assorted).

The data, presented in six surface graphs in **Figure 2.1** and organized by genome length, shows a near linear dependence of $\log d$ on $\log N$. As expected, the graphs show smoothing with increasing genome length (the computation of d involves averaging over ℓ components), and also with increased population size (as explained in Vose (1999), the initial transient of a finite haploid population trajectory converges as $N \to \infty$ to the corresponding infinite population model).

Of particular interest is the linear trend exhibited above. The slope m and intercept b of the regression line

$$\log d = m \log N + b \tag{2.18}$$

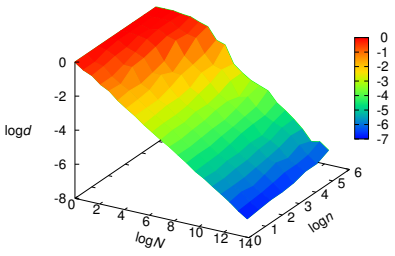
was computed using the data above; each was plotted against genome length ℓ and organized by generation n. The resulting graphs are displayed below.

Taking the exponential of the regression line (2.18) yields the estimate $d \approx N^m e^b$.

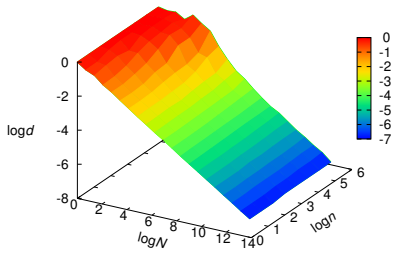
Slopes of the regression lines shown in **Figure 2.2** are approximately -0.5, indicating

$$d \;\approx\; k/\sqrt{N}. \tag{2.19}$$

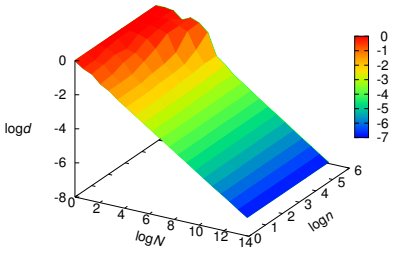
Equation 2.19 agrees with (1.3), (1.7) and theorem 3.1 from 'The Simple Genetic Algorithm: Foundations and Theory' (see Vose (1999)) which gives the bound for the expected rate of convergence for the single-step haploid case; the distance is inversely



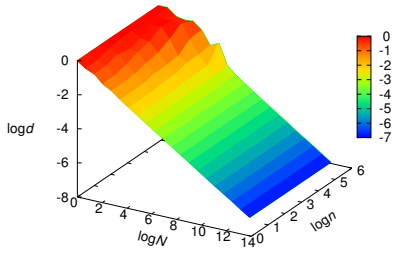
(a) $\ell = 4$.



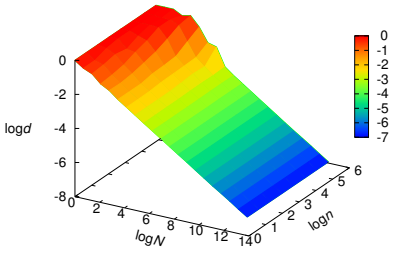
(b) $\ell = 6$.



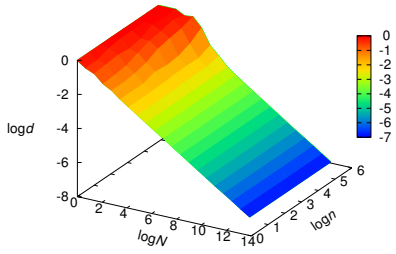
(c) $\ell = 8$.



(d) $\ell = 10$.

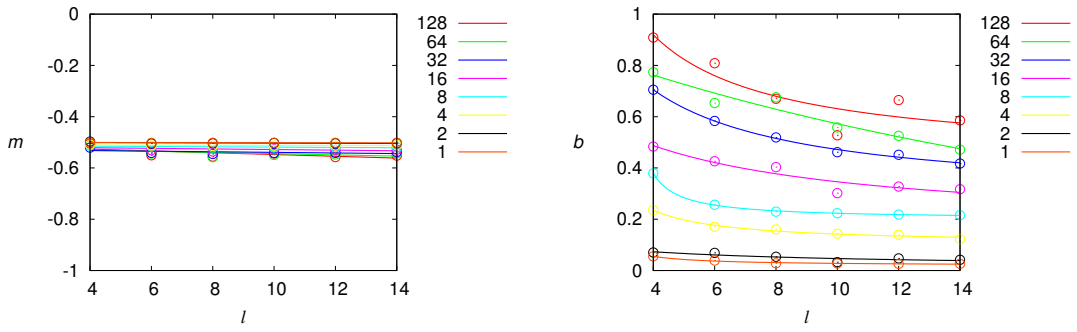


(e) $\ell = 12$.



(f) $\ell = 14$.

Figure 2.1: Convergence of finite population behaviour: d is distance between finite population $\boldsymbol{f}^n$ and infinite population $\boldsymbol{q}^n$ at generation n, population size N, for genome length ℓ (bits).



(a) Slope m, genome length ℓ.

(b) Intercept b, genome length ℓ.

Figure 2.2: Regression parameters: multi-plot of slope m and intercept b for generation $n \in \{1, 2, 4, 8, 16, 32, 64, 128\}$.

proportional to square root of population size. The consistent convergence rate across multiple generations shown in Figure (2.1) is somewhat surprising, simulation results above indicate it may persist to generation $n = 128$.

The intercept graphs in **Figure 2.2 b** show the constant of proportionality $k = e^b$ decreases monotonically with genome length ℓ, and increases monotonically with generation n. The increase in k for larger n seems to be a manifestation of the growing nonlinearity uniformly exhibited by the plots in **Figure 2.1** as n increases. It seems likely that the nonlinearity results from genetic drift experienced by finite populations (see Crow and Kimura (1970)).

2.8 Summary

We began with a description of simple diploid Markov model under mutation and crossover with no selective pressure. The model was reduced to the haploid case and specialized using mask based recombination operators to show Vose's infinite population model can be extended to the diploid case. Using computational benefits of this reduction, we showed via experiment and regression of the resulting data that distance between finite diploid population and infinite diploid population can indeed

decrease like $1/\sqrt{N}$ in practice. That rate of decrease is consistent with the single step-step convergence bounds predicted by Vose's infinite population model for the haploid case.

Chapter 3

Oscillation

This chapter investigates the qualitative similarity between finite population short-term behavior and infinite population evolutionary limits predicted by Vose. It uses computation to verify predicted infinite population limits and presents necessary and sufficient conditions for convergence to periodic orbits. We compute mutation distribution $\boldsymbol{\mu}$ and crossover distribution $\boldsymbol{\chi}$ to satisfy those conditions. And through experiments, we explore our second research question that whether finite populations oscillate following infinite populations behavior.

3.1 Limits

Vose states that under mild assumptions on mutations (explained later), infinite populations converge under repeated application of $\mathcal{M}$ in the absense of selective pressure. Vose mentions that periodic orbits are possible, but populations converge under repeated application of $\mathcal{M}^2$ and the limits $\boldsymbol{p}^* = \lim_{n\to\infty} \mathcal{M}^{2n}(\boldsymbol{p})$ and $\boldsymbol{q}^* = \lim_{n\to\infty} \mathcal{M}^{2n+1}(\boldsymbol{q})$ exist (see Vose (1999)).

Following Vose (see Vose (1999)), let $S_g = g\mathcal{R}/\{\mathbf{0}, g\}$, and let $|g|$ be the number of non zero bits in g. If $\widehat{\boldsymbol{p}}$ represents the current population in Walsh coordinates,

then the next generation $\widehat{\boldsymbol{p}}'_g$ (expressed in Walsh coordinates) is

$$\widehat{\boldsymbol{p}}'_g = \begin{cases} 2^{\ell/2} & \text{if } g = 0 \\ x_g \widehat{\boldsymbol{p}}_g + y_g(\widehat{\boldsymbol{p}}_g) & \text{otherwise} \end{cases}$$

where

$$x_g = 2\widehat{\mathcal{M}}_{g,0}, \qquad y_g(z) = 2^{\ell/2} \sum_{i \in S_g} z_i z_{i+g} \widehat{\mathcal{M}}_{i,i+g}.$$

Moreover,

$$\begin{aligned} |g| &= 1 \implies y_g = 0 \\ |g| &> 0 \implies |x_g| \leq 1 \\ |x_g| &= 1 \implies y_g = 0 \end{aligned}$$

With above notations, the limits can be expressed in Walsh basis by recursive equations (see Vose (1999))

$$\widehat{\boldsymbol{p}^*}_g = \begin{cases} (x_g y_g(\widehat{\boldsymbol{p}^*}) + y_g(\widehat{\boldsymbol{q}^*}))/(1 - x_g^2) & \text{if } |x_g| < 0 \\ \widehat{p}_g & \text{otherwise} \end{cases} \tag{3.1}$$

$$\widehat{\boldsymbol{q}^*}_g = \begin{cases} (x_g y_g(\widehat{\boldsymbol{q}^*}) + y_g(\widehat{\boldsymbol{p}^*}))/(1 - x_g^2) & \text{if } |x_g| < 0 \\ \widehat{\mathcal{M}(\boldsymbol{p})_g} & \text{otherwise} \end{cases} \tag{3.2}$$

If $x_g \neq -1$ for all g, then $\boldsymbol{p}^* = \boldsymbol{q}^* = \lim_{n\to\infty} \mathcal{M}(\boldsymbol{p})$ is the limit of mixing. In other cases, mixing converges to a periodic orbit oscillating between $\boldsymbol{p}^*$ and $\boldsymbol{q}^* = \mathcal{M}(\boldsymbol{p}^*)$.

Limits $\widehat{\boldsymbol{p}^*}_g$ and $\widehat{\boldsymbol{q}^*_g}$ can be computed considering gth components in order of increasing $|g|$. The necessary and sufficient condition for the sequence

$$\boldsymbol{p}, \mathcal{M}(\boldsymbol{p}), \mathcal{M}^2(\boldsymbol{p}), \cdots$$

to converge to a periodic orbit is that for some g

$$-1 = \sum_j (-1)^{g^T j} \boldsymbol{\mu}_j = -\sum_{k \in \bar{g}\mathcal{R}} \boldsymbol{\chi}_{k+g} + \boldsymbol{\chi}_k \tag{3.3}$$

3.2 Mutation and Crossover Distributions

The following describes the generation of mutation and crossover distributions that satisfy equation 3.3 for evolution to converge to a periodic orbit. Let $\boldsymbol{\mu}$ and $\boldsymbol{\chi}$ represent mutation and crossover distributions (respectively), and let $U01()$ return a random number between 0 and 1. For any $g \in \mathcal{R}$, $g \neq 0$, and for all $j \in \mathcal{R}$,

$$\boldsymbol{\mu}_j = \begin{cases} U01() & \text{if } (g^T j) \text{ is odd.} \\ 0 & \text{otherwise.} \end{cases}$$

Normalization yields $\boldsymbol{\mu}$ (the mutation distribution),

$$\boldsymbol{\mu}_j = \boldsymbol{\mu}_j / \sum_{j \in \mathcal{R}} \boldsymbol{\mu}_j.$$

Moreover, $\boldsymbol{\mu}$ satisfies condition 3.3.

Condition $k \in \bar{g}\mathcal{R}$ in equation 3.3 is

$$k = \bar{g}i \text{ for some } i \in \mathcal{R}$$

Multiplying through by $\bar{g}$ yields

$$\bar{g}k = \bar{g}\bar{g}i \;=\; \bar{g}i \;=\; k$$

The crossover distribution can therefore be generated as follows. For all $k \in \mathcal{R}$,

$$\begin{aligned} \boldsymbol{\chi}_k &= U01() \;\; if \;\; \bar{g}k \;=\; k \\ \boldsymbol{\chi}_{k+g} &= U01() \;\; if \;\; \bar{g}k \;=\; k \\ \boldsymbol{\chi}_k &= 0 \;\; otherwise. \end{aligned}$$

Normalization yields $\boldsymbol{\chi}$ (the crossover distribution),

$$\boldsymbol{\chi}_k = \boldsymbol{\chi}_k / \sum_{k \in \mathcal{R}} \boldsymbol{\chi}_k.$$

Moreover, $\boldsymbol{\chi}_k$ satisfies condition 3.3.

3.3 Initial Population

To investigate oscillation in infinite population and finite population behavior, it is desirable to have the same or corresponding initial populations.

For string length ℓ, the number of possible haploids is $x = 2^\ell$. Let array $\boldsymbol{t}$ represent a population size of N as follows: $\boldsymbol{t}_j$ is the jth population member (some element of $\{0, .., x-1\}$ where elements are base 2 length ℓ binary strings). Array $\boldsymbol{t}$ is generated from a random vector $\boldsymbol{u}$ of size x as follows.

$$\begin{aligned} \boldsymbol{u}_i &= U01(); \qquad i = 0, 1, .., x-1 \\ \boldsymbol{t}_j &= randp(\boldsymbol{u}); \qquad j = 0, .., N-1 \end{aligned}$$

where $U01()$ returns random number between 0 and 1, and $randp(\boldsymbol{u})$ returns random index i in array $\boldsymbol{u}$ with probability $\boldsymbol{u}_i$.

Let $\boldsymbol{c}_i$ be the count of haploid member i in population $\boldsymbol{t}$,

$$\boldsymbol{c}_i = \sum_{j=0}^{N-1} [\boldsymbol{t}_j = i]; \qquad i = 0, .., x-1$$

The infinite population vector $\boldsymbol{p}$ has ith component

$$\boldsymbol{p}_i = \frac{\boldsymbol{c}_i}{N}.$$

This randomly generated infinite haploid population vector $\boldsymbol{p}$ is used to obtain a diploid infinite population vector $\boldsymbol{q}$, and finite population vectors $\boldsymbol{s}$ and $\boldsymbol{f}$ as follows.

Infinite diploid population $\boldsymbol{q}$ is calculated corresponding to initial haploid population $\boldsymbol{p}$ as

$$\boldsymbol{q}_{i,j} = \boldsymbol{p}_i \boldsymbol{p}_j; \qquad (0 \leq i, j < x)$$

The finite haploid population members are the elements of array $\boldsymbol{t}$, the corresponding finite haploid population vector $\boldsymbol{s}$ is identical to $\boldsymbol{p}$. Let $\boldsymbol{v}$ be a finite diploid population member array of dimension two and of size N^2 whose diploid member $\boldsymbol{v}[i][j]$ at index $[i][j]$ is

$$\boldsymbol{v}[i][j] \;=\; \langle \boldsymbol{t}_i, \boldsymbol{t}_j \rangle \qquad 0 \leq i, j < N$$

The finite diploid population (proportion) vector $\boldsymbol{f}$ corresponding to finite diploid population member array $\boldsymbol{v}$ is identical to $\boldsymbol{q}$.

Thus, initial infinite haploid population vector $\boldsymbol{p}$ corresponds to initial infinite diploid population vector $\boldsymbol{q}$, and to initial finite haploid population vector $\boldsymbol{s}$ with population size N and population member array $\boldsymbol{t}$, and to initial finite diploid population vector $\boldsymbol{f}$ with population size N^2 and population member array $\boldsymbol{v}$.

3.4 Oscillation

Crossover distribution $\boldsymbol{\chi}$ and mutation distribution $\boldsymbol{\mu}$ satisfying condition (3.3) are considered to investigate oscillating behavior between predicted infinite population evolutionary limits.

Infinite haploid population evolutionary limits $\boldsymbol{p}_h^*$ and $\boldsymbol{q}_h^*$ were computed using equations (3.1) and (3.2). Infinite diploid population evolutionary limits $\boldsymbol{p}_d^*$ and $\boldsymbol{q}_d^*$ are obtained as follows

$$\begin{aligned}(\boldsymbol{p}_d^*)_{\langle\gamma_0,\gamma_1\rangle} &= (\boldsymbol{p}_h^*)_{\gamma_0}(\boldsymbol{p}_h^*)_{\gamma_1}\\ (\boldsymbol{q}_d^*)_{\langle\gamma_0,\gamma_1\rangle} &= (\boldsymbol{q}_h^*)_{\gamma_0}(\boldsymbol{q}_h^*)_{\gamma_1}\end{aligned}$$

where $\gamma = \langle\gamma_0, \gamma_1\rangle$ is a diploid genome.

For every genome length ℓ, the same initial population (calculated as described in (3.3)) was used for the infinite population and all sizes of finite populations conisdered. Genome lengths $\ell \in \{8, 10, 12, 14\}$ were used. Base population size of $N_0 = 64$ was used for the finite haploid case to compute initial population vector. The population sizes considered for plotting graphs were $N \in \{N_0^2, 10N_0^2, 20N_0^2\}$. To study oscillation in finite populations, the distances of $\boldsymbol{p}^n$ and $\boldsymbol{s}^n$ to haploid evolutionary limits $\boldsymbol{p}_h^*$ and $\boldsymbol{q}_h^*$ were plotted and the distances of $\boldsymbol{q}^n$ and $\boldsymbol{f}^n$ to diploid evolutionary limits $\boldsymbol{p}_d^*$ and $\boldsymbol{q}_d^*$ were plotted.

According to the results and conclusions from chapter 2, the expected distance d between finite population of size N and infinite population is

$$d \approx 1/\sqrt{N}$$

Table 3.1: Expected single step distance d for population size N

N	4096	40960	81920
d	0.0156	0.0049	0.0035

The distance between finite population and infinite population, for both haploid and diploid cases, were also plotted.

3.4.1 Haploid Population

Figures 3.1, 3.2, 3.3 and 3.4 showing oscillations in finite haploid populations, and distances between finite haploid populations and infinite populations arranged by genome length ℓ in ascending order. In each figure for unique genome length ℓ, sub-figures are arranged by population size (N). In each figure, first three rows of sub-figures on left column shows distance of finite population to limits and sub-figure in fourth row on left column shows distance of infinite population to limits. These sub-figures depicts oscillating behavior of both infinite and finite populations when the condition 3.3 is met. Finite population oscillation is sharper with increased population size. As population size increases, oscillation approaches the behavior exhibited by infinite population.

In each figure (3.1, 3.2, 3.3, and 3.4), first three rows of graphs on right side shows distance variation (difference in distance (d) and average distance (d_{avg})) where d is distance between haploid finite and infinite populations and d_{avg} is average value of d. In fourth row on right, a single graph for distances (d) of different finite population sizes ($N = 1N_0^2, 10N_0^2, 20N_0^2$) to infinite population are plotted. The resulting graphs show distance decreases as population size increases, which is consistent with results from section 2.1. The distance graphs smooth as population size increases. The graphs of $d - d_{avg}$ decreases in amplitude as population size increases. For fixed finite population size, as ℓ increases, distance graphs become smoother, and amplitude of oscillations decreases.

Distance data obtained from simulations for haploid populations are summarized in table 3.2. The last three columns tabulate average distance values between finite and infinite population for population sizes $N = 4096$, $N = 40960$ and $N = 81920$ respectively. Results from table 3.2 show average distance between finite and infinite population follows closely the expected single step distance given in table 3.1. The distance decreases as $1/\sqrt{N}$.

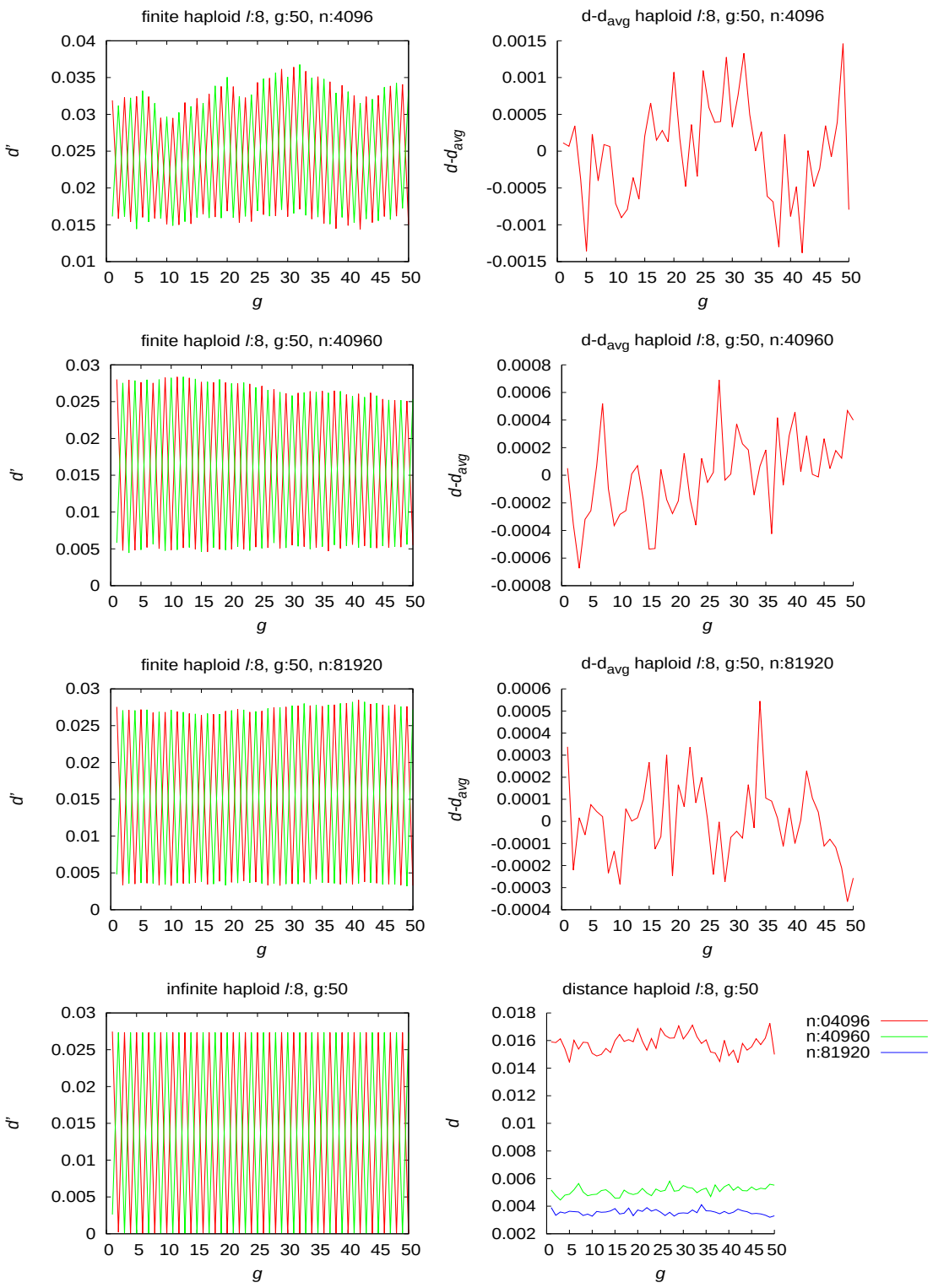


Figure 3.1: Infinite and finite haploid population oscillation behavior for genome length $\ell = 8$ (bits): In left column, d' is distance of finite population of size n or infinite population to limits for g generations. In right column, d is distance of finite population to infinite population for g generations and d_{avg} is average distances.

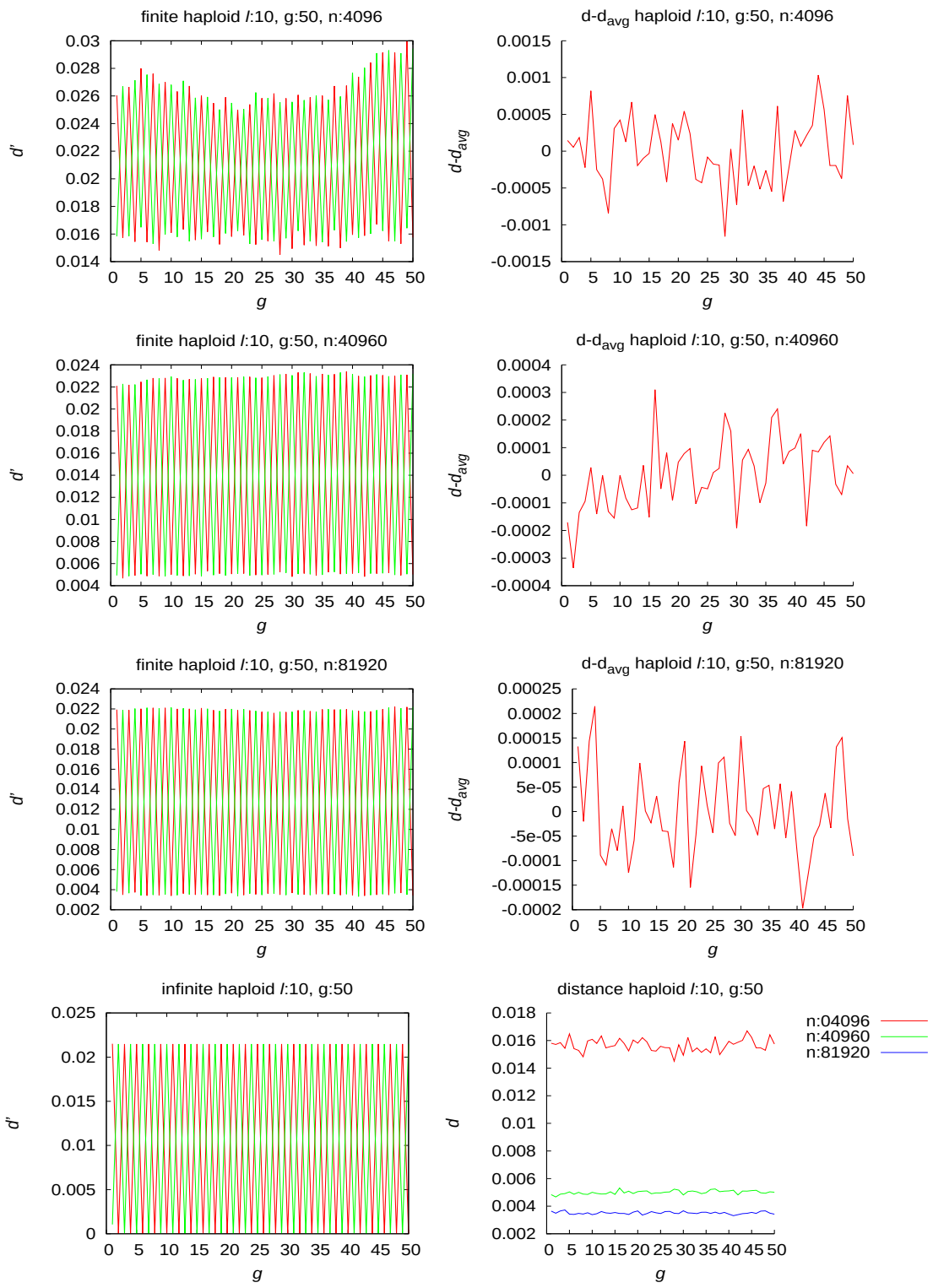


Figure 3.2: Infinite and finite haploid population oscillation behavior for genome length $\ell = 10$ (bits): In left column, d' is distance of finite population of size n or infinite population to limits for g generations. In right column, d is distance of finite population to infinite population for g generations and d_{avg} is average distances..

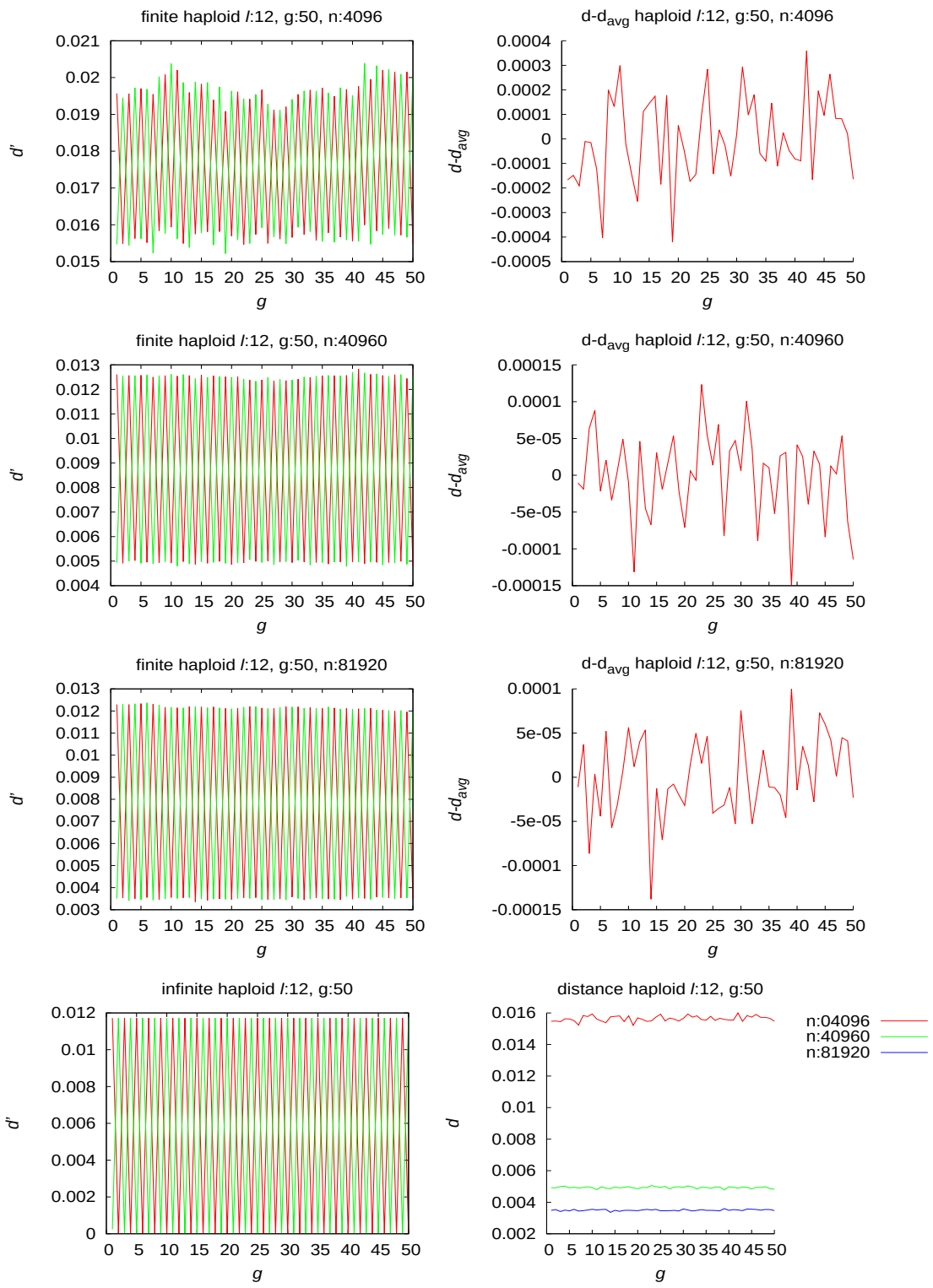


Figure 3.3: Infinite and finite haploid population oscillation behavior for genome length $\ell = 12$ (bits): In left column, d' is distance of finite population of size n or infinite population to limits for g generations. In right column, d is distance of finite population to infinite population for g generations and d_{avg} is average distances..

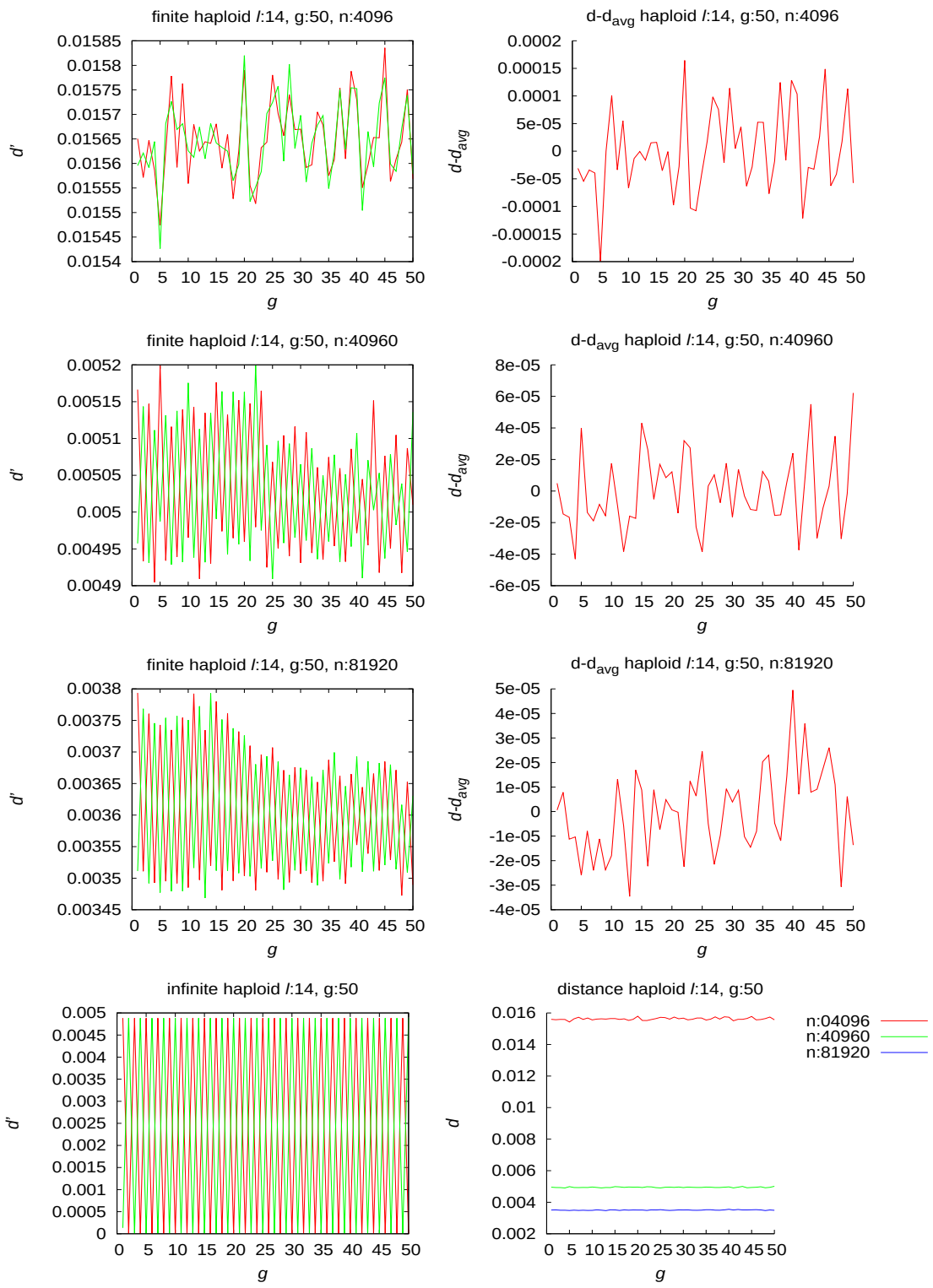


Figure 3.4: Infinite and finite haploid population oscillation behavior for genome length $\ell = 14$ (bits): In left column, d' is distance of finite population of size n or infinite population to limits for g generations. In right column, d is distance of finite population to infinite population for g generations and d_{avg} is average distances..

Table 3.2: Distance measured for haploid population: N is population size, ℓ is genome length and average distance between finite and infinite population is tabulated in the last three columns. $\{4096, 40960, 81920\}$

ℓ	$N = 4096$	$N = 40960$	$N = 81920$
8	0.0158	0.0051	0.0035
10	0.0157	0.0050	0.0035
12	0.0156	0.0049	0.0035
14	0.0156	0.0049	0.0035

3.4.2 Diploid Population

Figures 3.5, 3.6, 3.7 and 3.8 showing oscillations in finite diploid populations, and distances between finite haploid populations and infinite populations arranged by genome length ℓ in ascending order. In each figure for unique genome length ℓ, sub-figures are arranged by population size (N). In each figure, first three rows of sub-figures on left column shows distance of finite population to limits and sub-figure in fourth row on left column shows distance of infinite population to limits. These sub-figures depicts oscillating behavior of both infinite and finite populations when the condition 3.3 is met. Like in haploid population case, diploid finite population oscillation is also sharper with increased population size. As population size increases, oscillation approaches the behavior exhibited by infinite population.

In each figure (3.5, 3.6, 3.7, and 3.8), first three rows of graphs on right side shows distance variation (difference in distance (d) and average distance (d_{avg})) where d is distance between diploid finite and infinite populations and d_{avg} is average value of d. In fourth row on right, a single graph for distances (d) of different finite population sizes ($N = 1N_0^2, 10N_0^2, 20N_0^2$) to infinite population are plotted. The resulting graphs show distance decreases as population size increases, which is consistent with results from section 2.1. The distance graphs smooth as population size increases. The graphs of $d - d_{avg}$ decreases in amplitude as population size increases. For fixed finite population size, as ℓ increases, distance graphs become smoother, and amplitude of oscillations decreases.

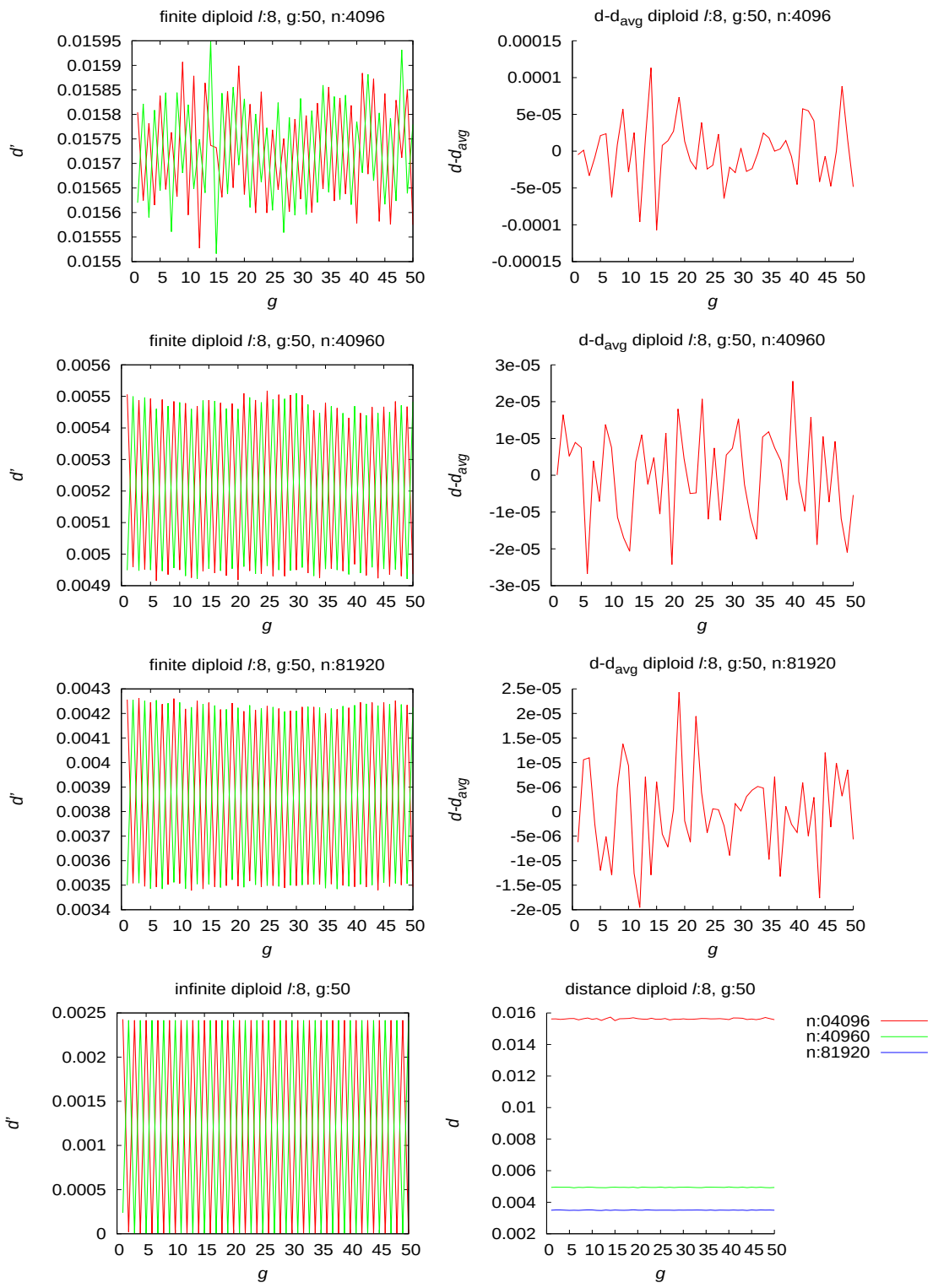


Figure 3.5: Infinite and finite diploid population oscillation behavior for genome length $\ell = 8$ (bits): In left column, d' is distance of finite population of size n or infinite population to limits for g generations. In right column, d is distance of finite population to infinite population for g generations and d_{avg} is average distance.

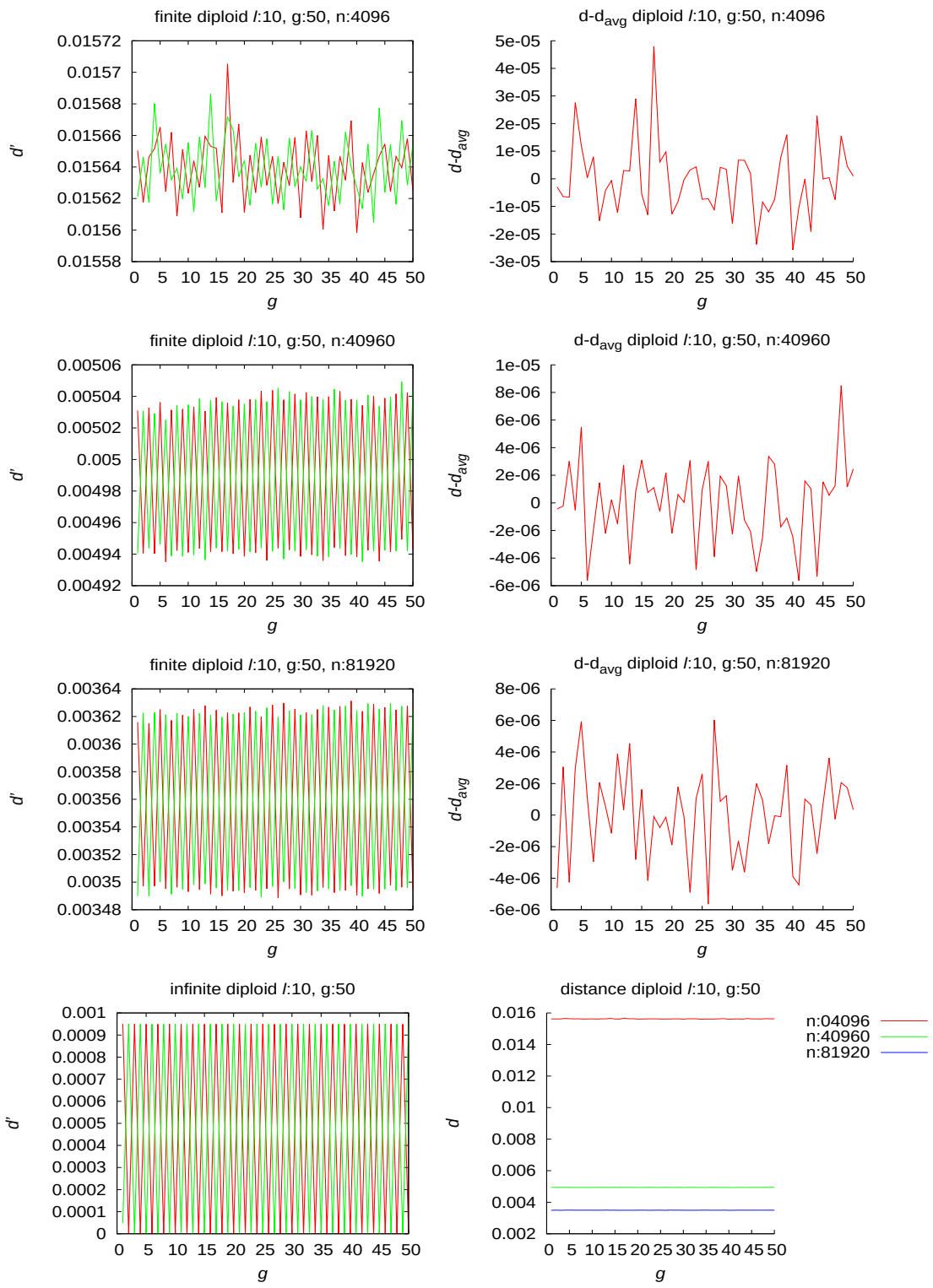


Figure 3.6: Infinite and finite diploid population oscillation behavior for genome length $\ell = 10$ (bits): In left column, d' is distance of finite population of size n or infinite population to limits for g generations. In right column, d is distance of finite population to infinite population for g generations and d_{avg} is average distance.

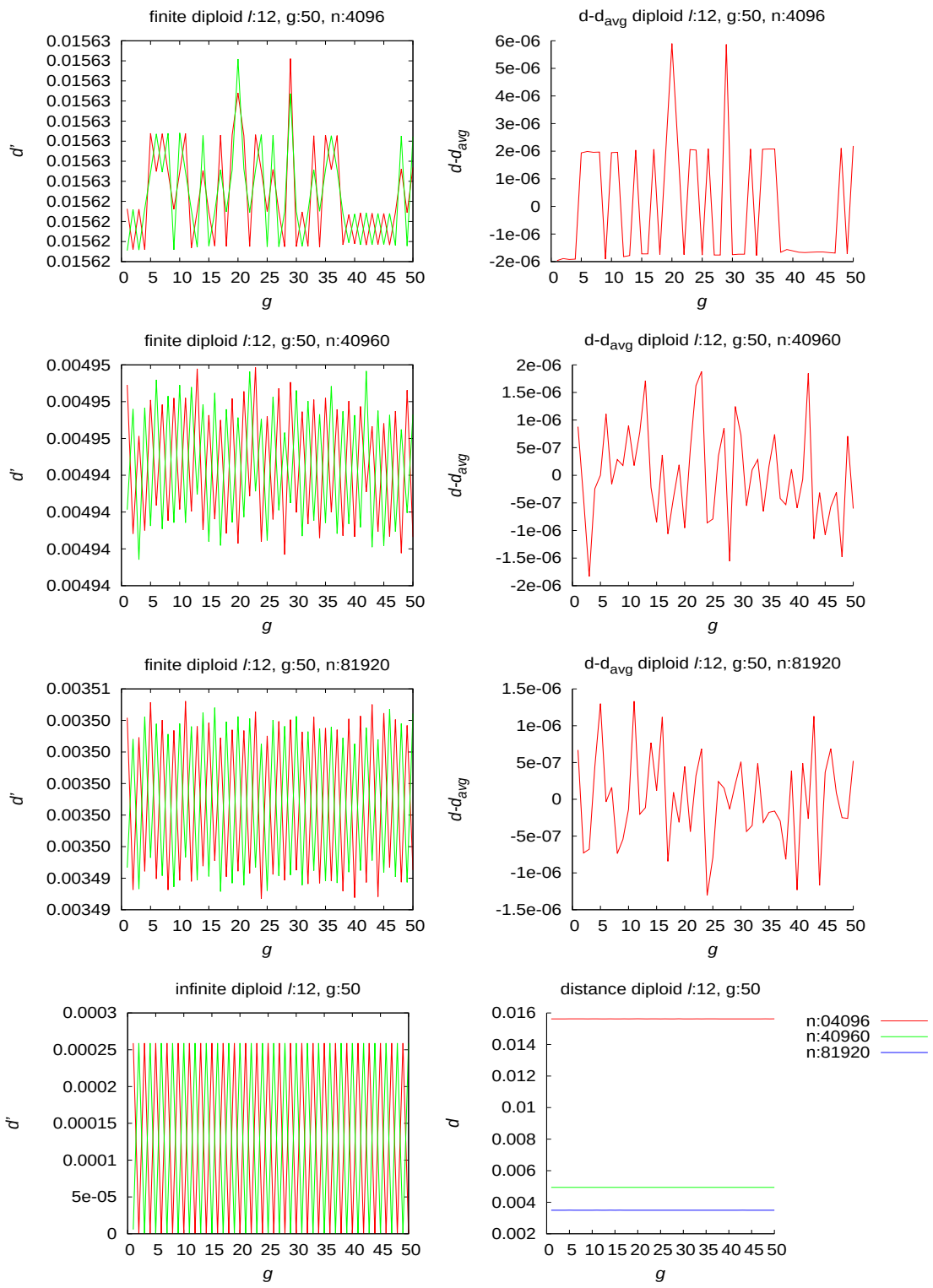


Figure 3.7: Infinite and finite diploid population oscillation behavior for genome length $\ell = 12$ (bits): In left column, d' is distance of finite population of size n or infinite population to limits for g generations. In right column, d is distance of finite population to infinite population for g generations and d_{avg} is average distance.

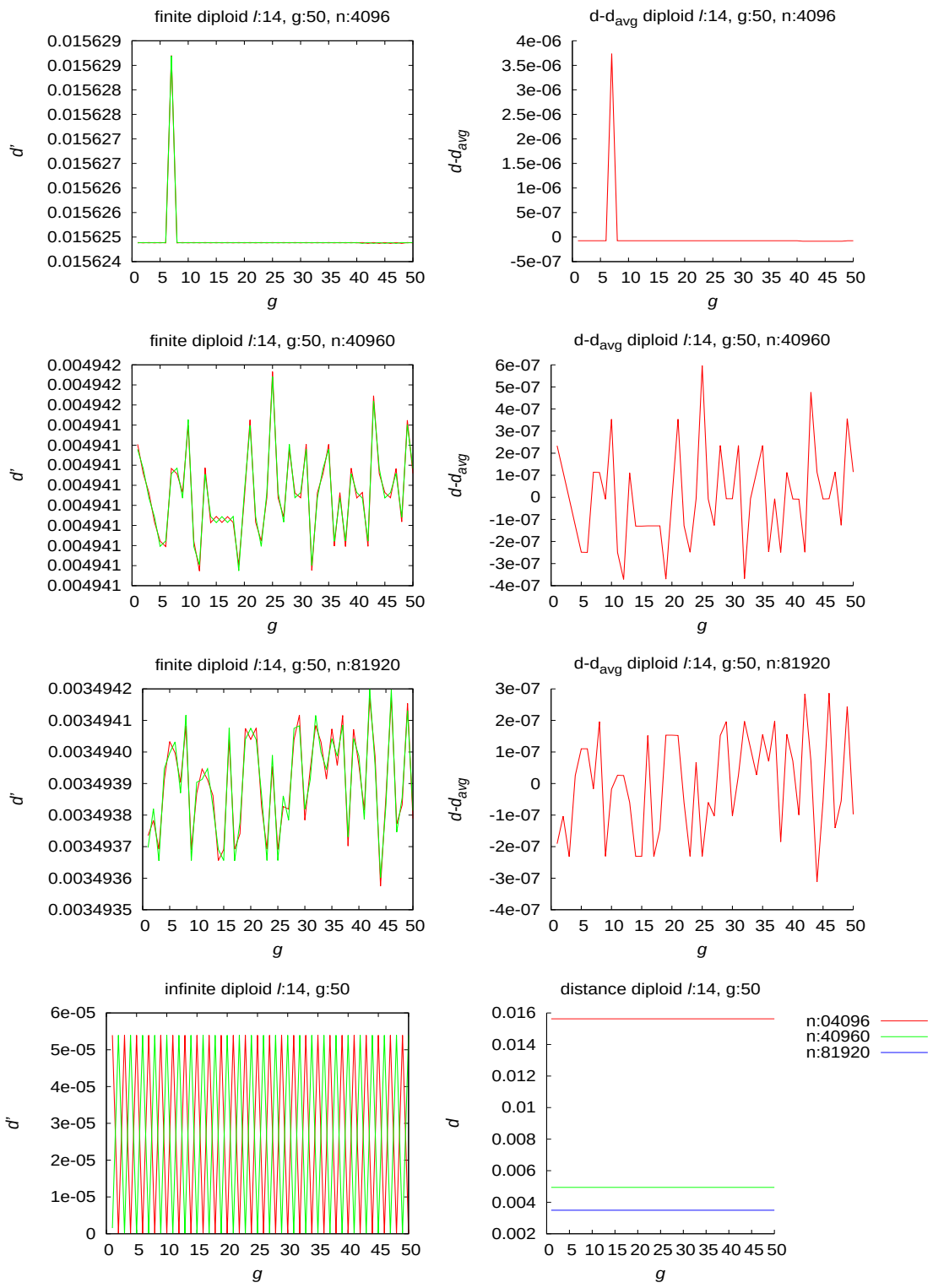


Figure 3.8: Infinite and finite diploid population oscillation behavior for genome length $\ell = 14$ (bits): In left column, d' is distance of finite population of size n or infinite population to limits for g generations. In right column, d is distance of finite population to infinite population for g generations and d_{avg} is average distance.

For same genome length ℓ and same size finite populations, graph showing distance between finite diploid population and infinite population is smoother than graph showing distance between finite haploid population and infinite population. Also amplitude of oscillations in diploid populations is smaller than in haploid populations.

Distance data obtained from simulations for diploid populations are summarized in table 3.3. The last three columns tabulate average distance values between finite and infinite population for population sizes $N = 4096$, $N = 40960$ and $N = 81920$ respectively. Results from table 3.3 show average distance between finite and infinite

Table 3.3: Distance measured for diploid population: N is population size, ℓ is genome length and average distance between finite and infinite population is tabulated in the last three columns. $\{4096, 40960, 81920\}$

ℓ	$N = 4096$	$N = 40960$	$N = 81920$
8	0.0156	0.0049	0.0035
10	0.0156	0.0049	0.0035
12	0.0156	0.0049	0.0035
14	0.0156	0.0049	0.0035

population follows closely the expected single step distance given in table 3.1. The distance decreases as $1/\sqrt{N}$.

3.5 Summary

In this chapter, we described limits predicted by Vose for infinite population, and conditions for convergence to periodic orbits. Mutation distributions and crossover distributions were computed to satisfy the conditions for infinite populations to converge to periodic orbits, and through experiment, we showed finite populations also oscillate, and converge to infinite population behavior as population size increases. The distance between finite population and infinite populations decreases as $1/\sqrt{N}$ which agrees with results from chapter 2.

Chapter 4

Violation

Chapter 5

Conclusion

This research shows how Vose's haploid model for Genetic Algorithms extends to the diploid case, improving the computation of infinite population evolutionary trajectories by significantly reducing the time and space used. Efficiency is achieved through decoupling haploid evolution from the evolution of infinite diploid populations and employing Walsh transform methods to compute the effects of mask-based crossover and mutation.

Simulations are thereby made feasible which otherwise would require excessive resources, as illustrated through computations exploring the convergence rate of finite population short-term behavior. Results agree with the expected rate of convergence for the single-step haploid case; distance is inversely proportional to square root of population size.

Evolutionary limits predicted by Vose for infinite population were explored and analysed. Simulations showed that when the necessary condition for oscillation in infinite populations is met, finite populations also show oscillating behavior, and approximately converge to infinite population evolutionary limits in the short term. When the condition is violated, i nfinite populations ceases to oscillate, but finite populations continue to oscillate if the violation is small.

In this research, we did not consider fitness factor for selection for simplicity of model. In future, we plan to extend our work accomodating fitness factor in our model and investigate convergence of short-term finite population behavior to infinite population limits.

Bibliography

Bibliography

Beauchamp, K. (1975). *Walsh functions and their applications*. Academic Press. 20

Bethke, A. D. (1980). *Genetic Algorithms As Function Optimizers*. PhD thesis, The University of Michigan. 4, 5

Cooley, J. W. and Tukey, J. W. (1965). An algorithm for the machine calculation of complex fourier series. *Mathematics of Computation*, 19(90):297–301. 21

Crow, J. and Kimura, M. (1970). *An introduction to population genetics theory*. New York, Evanston and London: Harper & Row. 28

Geiringer, H. (1944). On the probability theory of linkage in mendelian heredity. *Ann. Math. Stat.*, 15(1):25–27. 14, 18

Goldberg, D. E. (1987). Simple genetic algorithms and the minimal, deceptive problem. *Genetic algorithms and simulated annealing*, 74:74–88. 4

Goldberg, D. E. (1989a). Genetic algorithms and walsh functions: Part i, a gentle introduction. *Complex systems*, 3(2):129–152. 5

Goldberg, D. E. (1989b). Genetic algorithms and walsh functions-partii: Deception and its analysis. *Complex systems*, 3(153–171). 5

Hardy, G. H. (1908). Mendelian proportions in a mixed population. *Science*, 28(706):49–50. 14, 15

Holland, J. H. (1992). *Adaptation in natural and artificial systems*. Cambridge : MIT Press. 3, 4

Koehler, G. J. (1994). A proof of the vose-liepins conjecture. *Annals of Mathematics and Artificial Intelligence*, 10(4):409–422. 5

Koehler, G. J., Bhattacharyya, S., and Vose, M. D. (1997). General cardinality genetic algorithms. *Evol. Comput.*, 5(4):439–459. 5

Mendel, G. (1865). Versuche über plflanzenhybriden. *Verhandlungen des naturforschenden Vereines in Brünn*, IV:3–47. 15

Minc, H. (1988). *Nonnegative Matrices*. A Wiley-Interscience Publication. 12

Mitchell, M. (1999). *An Introduction to Genetic Algorithms*. The MIT Press. 2, 4

Nix, A. E. and Vose, M. D. (1992). Modeling genetic algorithms with markov chains. *Annals of Mathematics and Artificial Intelligence*, 5(1):79–88. 5

Shanks, J. L. (1969). Computation of the fast walsh-fourier transform. *IEEE Trans. Comput.*, 18(5):457–459. 6, 21, 24

Vose, M. and Liepins, G. E. (1991). Punctuated equilibria in genetic search. *Complex systems*, 5(1):31–44. 4, 5

Vose, M. D. (1999). *The simple genetic algorithm: foundations and theory*, volume 12. MIT press. 5, 6, 7, 8, 9, 10, 11, 17, 26, 30, 31

Vose, M. D. and Wright, A. H. (1998). The simple genetic algorithm and the walsh transform: Part i, theory. *Evol. Comput.*, 6(3):253–273. 6, 18, 19, 21, 22

Wikipedia, C. (2016a). Chebyshev's inequality. 9

Wikipedia, C. (2016b). Jensen's inequality. 10

Wikipedia, C. (2016c). Markov chain. 12

Appendix

Vita

Vita goes here...