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:

Committee Member 1

Committee Member 2

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:

Committee Member 1

Committee Member 2

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

1.1 Introduction

1.2 Literature

1.3 Random Heuristic Search

Vose **?** introduced abstract model, a generalized heuristic search method referred to as *Random Heuristic Search (RHS)* which is defined upon the central concept of state and transition between states. An instance of *RHS* can be thought of as an initial collection of elements P_0 chosen from some search space Ω , together with a stochastic transition rule τ , which from P_i will produce another collection P_{i+1}. In other words, τ will be iterated to produce a sequence of generations.

The beginning collection P_0 is referred to as the *initial population.* Let n be the cardinality of Ω and $\mathbf{1}$ denotes column vector of all 1s. The *simplex* is defined to be the set of population descriptors:

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

An element p of Λ corresponds to a population according to the rule:

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 p unambiguously determines a population.

Given the current population vector p, the next population vector $\tau(p)$ cannot be predicted with certainty because τ is stochastic and results from r independent, identically distributed random choices. Let $\mathcal{G} : \Lambda \to \Lambda$ be a function that given the current population vector p produces a new vector whose ith component is the probability that ith element of Ω is chosen. Thus, $\mathcal{G}(p)$ is the probability vector that specifies the distribution from which the aggregate of r choices forms the subsequent generation. Probability that population q given current population vector p can be computed as **?**

$$r! \prod \frac{(\mathcal{G}(p)_j)^{rq_j}}{(rq_j)!}$$

$$= exp\{-r \sum q_j \ln \frac{q_j}{\mathcal{G}(p)!} - \sum (\ln \sqrt{2\pi r q_j} + \frac{1}{12rq_j + \theta(rq_j)}) + O(\ln r)\}$$

where summation is restricted to indices for which $q_j > 0$.

Each random vector in the sequence $p, \tau(p), \tau^2(p), ...$ depends only on the value of the preceding one, which is a special situation, and such a sequence forms a Markov chain with transition matrix

$$Q_{p,q} = r! \prod \frac{(\mathcal{G}(p)_j)^{rq_j}}{(rq_j)!}$$

So the conceptualization of RHS can be replaced by Markov chain model abstraction which makes no reference to sampling Ω. That is from current population p, produce $q = \tau(p)$ with probability $Q_{p,q}$. With transition matrix defined for Markov chain model, Vose **?** says the expected next generation $E(\tau(p))$ is $\mathcal{G}(p)$ and the expression in transition matrix

$$\sum q_j \ln \frac{q_j}{\mathcal{G}(p)!}$$

gives the qualitative information regarding probable next generation which is the *discrepancy* of q with respect to $\mathcal{G}(p)$. It is a measure of how far q is from the expected next population $\mathcal{G}(p)$. Discrepancy is nonnegative and is zero only when q is the expected next population. Hence the factor

$$exp\{-r\sum q_j \ln \frac{q_j}{\mathcal{G}(p)!}\}$$

in the expression of transition matrix indicates the probability that q is the next generation decays exponentially, with constant r , as the discrepancy between q and the expected next population increases. The expression

$$\sum(\ln \sqrt{2\pi r q_j} + \frac{1}{12rq_j + \theta(rq_j)})$$

measures the *dispersion* of the population vector q and the factor

$$exp\{-\sum(\ln \sqrt{2\pi r q_j} + \frac{1}{12rq_j + \theta(rq_j)})\}$$

indicates the probability that q is the next generation decays exponentially with increasing dispersion.

Vose **?** calculated variance of next generation population with respect to expected population as

$$E(\|\tau(p) - \mathcal{G}(p)\|^2) = (1 - \|\mathcal{G}(p)\|^2)/r$$

and mentioned $\tau(p)$ converges in probability to $\mathcal{G}(p)$ as the population size increases. Therefore, τ corresponds to $\mathcal{G}$ in the infinite population case.

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 contributes to the elegance and simplicity of the abstract development and manifests diploid evolution equations can be represented by haploid equations.

A basic syntactic model for haploid and diploid genomes is considered in the beginning and commented on its expressive power. Then the mechanics of how the $(n+1)$th generation is obtained from the nth 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.

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 $\ell = \ell_1 + \cdots + \ell_h$). Partitioning may be extended to chromosomes so as to interpret each as a collection of genes. If continued to the granularity of pairs of bits, partitioning allows, for example, representing the four possibilities Adenine, Guanine, Cytosine, and Thymine.

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 **?**, 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 **?**, let the transmission function $t_\alpha(g)$ be the probability that gamete g is produced from parental genome α. It follows from the above that the equation determining the next generation q^{n+1} is

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

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

It should be appreciated that the Mendelian **?** 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,

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

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**?** (issues he considered and results he obtained relating to invariant distributions for a particular sort of transmission function are not here mentioned because they are irrelevant to the purpose of this section).

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 $p^{n+1}_{\gamma_0}$ and the second by $p^{n+1}_{\gamma_1}$ so that

$$q^{n+1}_{\langle\gamma_0,\gamma_1\rangle} \;=\; p^{n+1}_{\gamma_0}\, p^{n+1}_{\gamma_1} \tag{2.2}$$

where for any haploid γ_0,

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

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

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

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

Let $[\mathit{expression}]$ denote 1 if *expression* is true, and 0 otherwise.[§] 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

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

[§] $[\cdots]$ is sometimes referred to as an *Iverson bracket*.

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).

Chapter 3

Specialization

This chapter summarizes from the development in Vose **?**. 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).

The set of haploids (i.e., length ℓ binary strings) is a commutative ring $\mathcal{R}$ under component-wise addition and multiplication modulo 2. This algebraic structure is crucial to Vose's specialization and subsequent analysis of (2.8). Denote the additive identity by $\mathbf{0}$ and the multiplicative identity by $\mathbf{1}$, and 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.*

3.1 Mutation

Mutation simulates effects of error that happen with low probability during duplication of chromosome. Mutation provides mechanism to inject new strings into the

*In particular, $g\overline{g}=\mathbf{0}=g+g$, $g^2=g$, $g+\overline{g}=\mathbf{1}$ for all $g\in\mathcal{R}$.

next generation population which gives *RHS* ability to search beyond the confines of initial population.

Symbol μ is used to represent mutation distribution describing the probability μ_i with which $i \in \Omega$ is selected to be a mutation mask. $\mu : \Omega \to \Omega$ is nondeterministic mutation function where the result $\mu(x)$ of applying mutation function on x is $x \oplus i$ with probability μ_i of distribution μ where i is *mutation mask*. Mutating x using mutation mask i alters the bits of x in those positions the mutation mask i is 1. $\mu \in [0, 0.5)$ is regarded as a *mutation rate* which implicitly specifies distribution μ according to rule **?**

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

If g should mutate to g' with probability ρ, let

$$\mu_{g+g'} \;=\; \rho$$

Given distribution μ, mutation is the stochastic operator sending g to g' with probability $\mu_{g+g'}$.

Mutation considered is *independent* for all j and k which means **?**

$$\mu_j = \sum_{k \otimes i = 0} \mu_{i \oplus j} \sum_{k \overline{\otimes} i = 0} \mu_{i \otimes j}$$

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. Masked based crossover is used in this document. Geiringer **?** used crossover mask with probability (distribution) associated with the mask to generate offsprings from parent chromosomes in absence of mutation and selection. Let χ_m be probability distribution with which m is selected to be a crossover mask. Following Geiringer **?**, if crossing over u and v

should produce u' and v' with probability ρ, let

$$\chi_m \;=\; \rho$$

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 χ, crossover is the stochastic operator which sends u and v to u' and v' with probability $\chi_m/2$ for each u' and v'.

χ can be considered as a *crossover rate* that specifies the distribution χ given by rule **?**

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

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, $c_i = 2^{-\ell}$.

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 μ and crossover distribution χ, whether or not μ is independent if mutation is performed before crossover, then transmission function can be expressed as **?**

$$t_{\langle u,v\rangle}(g) \;=\; \sum_{i\in\mathcal{R}}\sum_{j\in\mathcal{R}}\sum_{k\in\mathcal{R}} \mu_i\mu_j \,\frac{\chi_k + \chi_{\overline{k}}}{2}\,[k(u+i)+\overline{k}(v+j) \;=\; g] \tag{3.1}$$

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 (3.1) implies that evolution equation (2.8) can be expressed in the form

$$p'_g \;=\; (\sigma_g p)^T M\,(\sigma_g p) \tag{3.2}$$

where the permutation matrix σ_g is defined by component equations

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

3.4 Walsh Transorm

A time series, f(t), in terms of a series of Walsh funcitons W(n,t) **?**, viz.

$$f(t) = a_0 W_{0,t} + \sum_{n=1}^{N-1} a_n W_{n,t}$$

where n is an ordering number, N is number of terms used in Walsh series to express time series and

$$\frac{a_0}{2} = \frac{1}{T} \int_0^T f(t) W_{n,t} dt$$

$$a_n = \frac{1}{T} \int_0^T f(t) W_{n,t} dt$$

Finite discrete Walsh transform pair on N sampling points, x_t, can be expressed as **?**

$$X_n = \frac{1}{N} \sum_{t=0}^{N-1} x_t W_{n,t} \tag{3.3}$$

$$n = 0, 1, 2...N-1$$

and

$$x_t = \sum_{n=0}^{N-1} X_n W_{n,t}$$

$$t = 0, 1, 2...N-1$$

The Walsh function series $W_{n,t}$ can be obtained using Walsh matrix also known as Hadamard matrix of order N. Walsh matrix or Hadamard matrix is a square matrix of order N whose coefficients comprise only +1 and -1 and where its rows (and columns) are orthogonal to one another. The Walsh matrix is defined by

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

where $N^{-1/2}$ is normalization factor and $n \cdot t$ is bitwise dot product of binary representation of number n and t.

The matrix is symmetric, i.e.,

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

and it has entries satisfying

$$W_{n,t\oplus k} = N^{1/2} W_{n,t} W_{n,k}$$

The practical importance of this symmetry is that the transform and inverse represent same mathematical operation, hence simplifying the derivation and application of the transform. With the normalized form, *Walsh matrix* is its own inverse, i.e.,

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

In the matrix form, 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 $\widehat{A} = WAW$. If w is a row vector, then w in its Walsh basis $\widehat{w}$ represents wW.

3.5 Walsh Transform Adaptation

The Walsh transform has spectacular ability to unravel the intricacies of mixing. And that is why we adapt Walsh transform methods for computing evolutionary trajectories, which have already been established for Vose's haploid model **?**. Adaptation of Walsh transformation efficiently models infinite diploid population evolution. This adaptation of Walsh transormation helps in making feasible comparisons between finite and infinte diploid population short-term evolutionary behavior. Recalling evolution equation (3.2), without selection, specialized to Vose's

infinite population model expressed in mixing matrix's term,

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

where the permutation matrix σ_g is defined by component equations

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

In our model, the Walsh matrix W is defined by 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.

A change of basis which simultaneously diagonalizes the σ_g unravels the evolution equation (3.2). Expressed in the Walsh basis (see **?**), the mixing matrix takes the form

$$\widehat{M}_{u,v} \;=\; 2^{\,\ell-1}\,[uv=\mathbf{0}]\,\widehat{\mu}_u\widehat{\mu}_v \sum_{k\in\overline{u+v}\mathcal{R}} \chi_{k+u}+\chi_{k+v} \tag{3.4}$$

and equation (3.2) takes the form

$$\widehat{p}'_g \;=\; 2^{\,\ell/2} \sum_{i\in g\mathcal{R}} \widehat{p}_i\,\widehat{p}_{i+g}\,\widehat{M}_{i,i+g} \tag{3.5}$$

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 (3.5) in terms of the mixing matrix (3.4), is Markovian; the next state p' depends only upon the current state p. Let $\mathcal{M}$ represent the mixing

transformation,

$$p' \;=\; \mathcal{M}(p) \tag{3.6}$$

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; (3.6) can be computed in Walsh coordinates relatively efficiently via (3.4) and (3.5).

3.6 Fast Walsh Transform

However, computation of discrete Walsh transform given by equation (3.3) takes N^2 operations (addition or subtraction). An algorithm using matrix factorization techniques is found to perform transformation in $N \log_2 N$ operations. This algorithm in fast Walsh transform (FWT). Shanks **?** described FWT algorithm which is analogous to Cooley-Tukey **?** algorithm for fast Fourier transformation. Shanks assumed walsh function to be periodic with period N, where N is an integral power of 2. So a complete orthogonal set will have N function $W_{m,n}$ where $m = 0, 1, 2, .., N-1$ and $n = 0, 1, 2, .., N-1$. The first two discrete walsh functions are defined as

$$W_{0,n} = 1 \text{ for } n = 0, 1, 2, .., N-1 \tag{3.7}$$

$$W_{1,n} = \begin{cases} 1 & \text{for } n = 0, 1, 2, .., (N/2) - 1. \\ -1 & \text{for } n = N/2, (N/2)+1, .., N-1. \end{cases} \tag{3.8}$$

Remainder of set can be generated by using multiplicative iterative equation (**??**):

$$W_{m,n} = W_{[m/2],2n} \cdot W_{m-2[m/2],n} \tag{3.9}$$

where $[m/2]$ indicates the integer part of $m/2$.

The discrete Walsh functions as defined here are symmetric with respect to the argument (m, n). That is,

$$W_{m,n} = W_{n,m}.$$

For real array of length N, Walsh transform can be defined as

$$F(m) = \sum_{n=0}^{N-1} f(n) W_{m,n}, \text{where } m = 0, 1, 2, .., N-1 \tag{3.10}$$

Similarly, for inverse transform is

$$f(n) = \frac{1}{N} \sum_{m=0}^{N-1} F(m) W_{m,n}, \text{where } n = 0, 1, 2, .., N-1 \tag{3.11}$$

Since walsh functions $W_{m,n}$ have values either 1 or -1, computation of (3.10) and (3.11) does not require multiplication. Shanks **?** derived using (3.9) an algorithm analogous to Cooley-Tukey algorithm that will require $N \log_2 N$ summations to compute complete Walsh transform.
For $N = 8$ (which can be extended to general case), indices in (3.10) can be replaced with a set which can only have values 0 and 1. That is,

$$m = 4j_2 + 2j_1 + j_0, j_2, j_1, j_0 = 0 \text{ or } 1 \tag{3.12}$$

$$n = 4k_2 + 2k_1 + k_0, k_2, k_1, k_0 = 0 \text{ or } 1 \tag{3.13}$$

Using these new notations, $W_{m,n}$ becomes $W(j_2, j_1, j_0; k_2, k_1, k_0)$. So (3.10) becomes

$$F(j_2, j_1, j_0) = \sum_{k0=0}^{1} \sum_{k1=0}^{1} \sum_{k2=0}^{1} f(k2, k1, k0) \cdot W(j_2, j_1, j_0; k_2, k_1, k_0) \tag{3.14}$$

Here, j_2, j_1, j_0 is binary representation of m. So, dividing m by 2 equivalents to shifting binary representation of m by 1 to the right and dropping fractional bit. That is, if

$$m \leftrightarrow j_2 j_1 j_0$$

then,

$$[m/2] \leftrightarrow 0 j_2 j_1.$$

Similarly, if

$$n \leftrightarrow k_2 k_1 k_0$$

then,

$$2n \leftrightarrow k_2 k_1 k_0 0.$$

An 8-length Walsh function is periodic with period 8. Thus any index (such as $2n$) can be evaluated modulo 8. This is equivalent to deleting any bits above the third bit and we have

$$2n(modulo 8) \leftrightarrow k_1 K_0 0$$

Using these indices, (3.9 becomes

$$W(j_2, j_1, j_0; k_2, k_1, k_0) = W(0, j_2, j_1; k_1, k_0, 0) \cdot W(0, 0, j_0; k_2, k_1, k_0). \tag{3.15}$$

j_0 can only be 0 or 1, so $W(0, 0, j_0; k_2, k_1, k_0)$ represents either $W_{0,n} or W_{1,n}$. The function $W_{0,n}$ is 1 for all n. And function $W_{1,n}$ is 1 if $0 < n < 4$ and $W_{1,n}$ is -1.0 if $4 < n < 7$. Thus, from (3.13), $W_{1,n}$ is +1.0 if $k_2 = 0$ and $W_{1,n}$ is -1.0 if $k_2 = 1$. Therefore,

$$W(0, 0, j_0; k_2, k_1, k_0) = (-1)^{j_0 k_2}. \tag{3.16}$$

(3.15) can be used to factor $W(0, j_2, j_1; k_1, k_0, 0)$ to get

$$W(0, j_2, j_1; k_1, k_0, 0) = W(0, 0, j_2; k_0, 0, 0) \cdot W(0, 0, j_1; k_1, k_0, 0). \tag{3.17}$$

Using (3.16) and (3.17) in (3.15), we get completely factored expression for the 8-length Walsh function

$$W(j_2, j_1, j_0; k_2, k_1, k_0) = (-1)^{j_2 k_0} (-1)_{j_1 k_1} (-1)^{j_0 k_2}. \tag{3.18}$$

Using (3.18), (3.14) becomes

$$F(j_2, j_1, j_0; k_2, k_1, k_0) = \sum_{k_0=0}^{1} (-1)^{j_2 k_0} \sum_{k_1=0}^{1} (-1)^{j_1 k_1} \cdot \sum_{k_2=0}^{1} (-1)^{j_0 k_2} f(k_2, k_1, k_0). \tag{3.19}$$

We can define

$$A_1(j_0, k_1, k_0) = \sum_{k_2=0}^{1} (-1)^{j_0 k_2} f(k_2, k_1, k_0) \tag{3.20}$$

Array A_1 is first set of intermediate calculations to compute $F(m)$. $A_2(j_0, j_1, k_0)$ can be computed from A_1 and final result $A_3(j_0, j_1, j_2)$ from A_2 for 8-length Walsh transform. However, the order of the values will be in the bit reversed form. It can be generalized for the case $N = 2^p$ by defining intermediate Walsh transform arrays as

$$A_l(j_O, j_1, ..., j_{l-1}, k_{p-l-1}, ..., k_0) = \sum_{k_{p-l}=0}^{1} A_{l-1}(j_0, j_1, ..., j_{l-2}, k_{p-l}, k_{p-l-1}, ..., k_0) \cdot (-1)^{j_{l-1} k_{p-l}} \tag{3.21}$$

where $l = 1, 2, ..., p$ and

$$A_0(k_{p-1}, k_{p-2}, ..., k_0) = f(k_{p-1}, k_{p-2}, ..., k_0). \tag{3.22}$$

The general equation for the $M = 2^P$-length discrete Walsh function is

$$W(j_{p-1}, j_{p-2}, ..., j_0; k_{p-1}, k_{p-2}, ..., k_0) = \prod_{i=0}^{p-1} (-1)^{j_{p-1-i} k_i}.$$

Chapter 4

Experimental Simulations and Measurements

This chapter describes how distance between finite diploid population and infinite population is calculated. Then it discusses simplifications in computations made by our evolutionary equations and in distance computation. And it goes on to discuss results from simulations and convergence of finite diploid population short-term behavior to evolutionary behavior predicted by infinite population model.

4.1 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

$$\|\boldsymbol{f}-\boldsymbol{q}\|^2 \;=\; \sum_{\alpha\notin S_{\boldsymbol{f}}}(\boldsymbol{f}_\alpha-\boldsymbol{q}_\alpha)^2+\sum_{\alpha\in S_{\boldsymbol{f}}}(\boldsymbol{f}_\alpha-\boldsymbol{q}_\alpha)^2 \tag{4.1}$$

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

$$\begin{aligned}\sum_{\langle\alpha_0,\alpha_1\rangle\notin S_{\boldsymbol{f}}}(\boldsymbol{p}_{\alpha_0}\boldsymbol{p}_{\alpha_1})^2 \;&=\; \sum_{\langle\alpha_0,\alpha_1\rangle}(\boldsymbol{p}_{\alpha_0})^2(\boldsymbol{p}_{\alpha_1})^2-\sum_{\langle\alpha_0,\alpha_1\rangle\in S_{\boldsymbol{f}}}\left(\boldsymbol{p}_{\alpha_0}\boldsymbol{p}_{\alpha_1}\right)^2\\ &=\; \sum_{g}^{2}(\boldsymbol{p}_g)^2-\sum_{\alpha\in S_{\boldsymbol{f}}}(\boldsymbol{q}_\alpha)^2\end{aligned} \tag{4.2}$$

It follows from (4.1) and (4.2) that

$$\begin{aligned}\|\boldsymbol{f}-\boldsymbol{q}\|^2 \;&=\; \sum_{g}^{2}(\boldsymbol{p}_g)^2+\sum_{\alpha\in S_{\boldsymbol{f}}}(\boldsymbol{f}_\alpha-\boldsymbol{q}_\alpha)^2-\sum_{\alpha\in S_{\boldsymbol{f}}}(\boldsymbol{q}_\alpha)^2\\ &=\; \sum_{g}^{2}(\boldsymbol{p}_g)^2+\sum_{\alpha\in S_{\boldsymbol{f}}}\boldsymbol{f}_\alpha(\boldsymbol{f}_\alpha-2\boldsymbol{q}_\alpha)\end{aligned} \tag{4.3}$$

which involves $2^\ell+|S_{\boldsymbol{f}}|$ terms, assuming that $S_{\boldsymbol{f}}$ is known as a byproduct of computing $\boldsymbol{f}$.

(4.3) computes distance between finite and infinite population efficiently.

4.2 Simplification

The haploid case simplified by equations (3.4) and (3.5) are the consequence of 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 $n \log n$ **?**.

Only one mixing matrix as opposed to 2^ℓ matrices is needed to compute the next generation; evolution equation (3.5) 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 (3.4) with equation (3.1). Also, the relevant quadratic form is computed with a single sum as opposed to a double sum; computing via (3.5) 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 $2^{28}/(2^{14} + 2^N) > 252$.

Chapter 5

Evolutionary Limits

Bibliography

Appendix

Appendix A

Summary of Equations

A.1 Cartesian

some equations here

A.2 Cylindrical

some equations also here

Vita

Vita goes here...