

Efficient Simulation Of A Simple Evolutionary System

Mahendra Duwal Shrestha

The University Of Tennessee

March 17, 2017

Outline

Part-I: Efficient Computations

- Background

- Computations

Part-II: Applications

- Question 1: Convergence of finite population

- Question 2: Finite population oscillation

- Question 3: Finite population oscillation under mutation-violation

- Question 4: Finite population oscillation under crossover-violation

- Conclusion

Part-I: Efficient Computations

Terms

Population P : a collection of length ℓ binary strings

Population vector: \mathbf{p}_j proportion of string j in the population

If $P = 00, 01, 01, 10, 11, 11$, then $\mathbf{p}_3 = 2/6 = 1/3$

\mathcal{R} denotes the set of binary strings of length ℓ , $|\mathcal{R}| = Z = 2^\ell$

Addition and multiplication of elements in \mathcal{R} are bitwise operations modulo 2

$$x = 1101, y = 1010$$

$$x + y = 1101 + 1010 = 0111$$

$$xy = 1101 \cdot 1010 = 1000$$

$$\bar{x} = 0010$$

Crossover & Mutation

Crossover : Choose parents u and v , exchange bits using crossover mask m :

$$u' = um + v\bar{m}, v' = u\bar{m} + vm$$

$$u = \mathbf{1100}, v = 1101, m = 1100$$

$$\{\mathbf{1100}, 1101\} \rightarrow \{\mathbf{1100} + 0001, 0000 + 1100\} \rightarrow \{\mathbf{1101}, \mathbf{1100}\}$$

χ_m = probability of using crossover mask m

Mutation: Flip bits using mutation mask:

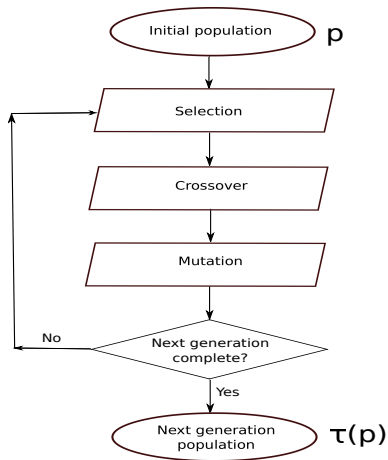
$$x \rightarrow x + m$$

$$x = 1100, m = 0001$$

$$1100 \rightarrow 1100 + 0001 \rightarrow 1101$$

μ_m = probability of using mutation mask m

Finite Population GA



Randomly select parents u and v

Crossover u and v to produce u' and v'

Keep one of u' , v' , and mutate to produce gamete g

Repeat above to form next generation

Random Heuristic Search

τ is a stochastic transition rule that maps \mathbf{p} to

$$\mathbf{p}' \in \Lambda_N = \{ \langle \frac{X_1}{N}, \dots, \frac{X_Z}{N} \rangle \mid X_i \in \mathbb{Z}^{\geq 0}, \sum X_i = N \}$$

$\tau(\mathbf{p})$ cannot be predicted with certainty

$\mathbf{p}, \tau(\mathbf{p}), \tau^2(\mathbf{p}), \dots$ forms Markov chain

Infinite Population Model

Population modeled as vector $\mathbf{p} \in \Lambda = \{\langle \mathbf{p}_1, \dots, \mathbf{p}_Z \rangle \mid \mathbf{p}_i \geq 0, \sum \mathbf{p}_i = 1\}$

\mathcal{G} maps \mathbf{p} to the next generation \mathbf{p}'

$\mathcal{G}(\mathbf{p})_j$ = proportion of string j in the next generation

The infinite population model

$$\mathbf{p} \rightarrow \mathcal{G}(\mathbf{p}) \rightarrow \mathcal{G}(\mathcal{G}(\mathbf{p})) \rightarrow \dots$$

$$\mathcal{G}(\mathbf{p}) = \mathcal{E}(\|\tau(\mathbf{p})\|)$$

The variance is

$$\mathcal{E}(\|\tau(\mathbf{p}) - \mathcal{G}(\mathbf{p})\|^2) = \frac{1 - \|\mathcal{G}(\mathbf{p})\|^2}{N}$$

Diploid Population Model

Diploid genome: $\alpha = \langle \alpha_0, \alpha_1 \rangle$

Population vector \mathbf{q}_α : prevalence of diploid α

$t_\alpha(g)$: probability that gamete g is produced from parent α

$$\mathbf{q}'_\gamma = \sum_{\alpha} \mathbf{q}_\alpha t_\alpha(\gamma_0) \sum_{\beta} \mathbf{q}_\beta t_\beta(\gamma_1)$$

Diploid Model Reduction to Haploid Model

Diploids in terms of haploids:

$$\mathbf{q}_{\langle\gamma_0, \gamma_1\rangle} = \mathbf{p}_{\gamma_0} \mathbf{p}_{\gamma_1}$$

Haploids in terms of diploids:

$$\mathbf{p}_g = \frac{1}{2} \sum_{\alpha_0, \alpha_1} \mathbf{q}_{\langle\alpha_0, \alpha_1\rangle} ([g = \alpha_0] + [g = \alpha_1])$$

Evolution equation in terms of haploid distribution \mathbf{p} :

$$\mathbf{p}'_{\gamma_0} = \sum_{\alpha_0, \alpha_1} \mathbf{p}_{\alpha_0} \mathbf{p}_{\alpha_1} t_{\langle\alpha_0, \alpha_1\rangle}(\gamma_0)$$

Matrix form:

$$\mathbf{p}'_g = \mathbf{p}^T M_g \mathbf{p} \quad \text{where} \quad (M_g)_{u,v} = t_{\langle u,v\rangle}(g)$$

Specialization to Vose's Haploid Model

Mutation distribution:

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

Crossover distribution:

$$\chi_i = \begin{cases} \chi c_i & \text{if } i > 0 \\ 1 - \chi + \chi c_0 & \text{if } i = 0 \end{cases} \quad c_i = 2^{-\ell}$$

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

Walsh Basis

$$W_{n,t} = Z^{-1/2}(-1)^{n^T t} \text{ where}$$

$$Z = 2^\ell$$

$$\hat{A} = WAW$$

$$\hat{w} = Ww$$

procedure FWT

$n = 2^d \leftarrow$ size of array X where d is positive integer

for $i = 0$ to $d - 1$ do

$m = n/2^i$

$z = m/2$

for $j = 0$ to $2^i - 1$ do

for $k = 0$ to $z - 1$ do

$t1 = m \times j + k$

$t2 = m \times j + z + k$

$a = X[t1]$

$b = X[t2]$

$X[t1] = a + b$

$X[t2] = a - b$

end for

end for

end for

return X

end procedure

Computations in Walsh basis

Mixing matrix M_0 in Walsh basis

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

Evolution eqn in Walsh basis

$$\hat{\mathbf{p}}'_g = 2^{\ell/2} \sum_{i \in g\mathcal{R}} \hat{\mathbf{p}}_i \hat{\mathbf{p}}_{i+g} \hat{M}_{i,i+g} \quad \text{where } g\mathcal{R} = \{gi \mid i \in \mathcal{R}\}$$

Computational Comparison

In Walsh basis

$$\hat{\mathbf{p}}'_g = 2^{\ell/2} \sum_{i \in g\mathcal{R}} \hat{\mathbf{p}}_i \hat{\mathbf{p}}_{i+g} \hat{M}_{i,i+g}$$

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

Before Walsh basis

$$\mathbf{p}'_g = \mathbf{p}^T M_g \mathbf{p}$$

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

Computational Significance

Reduction to haploid model and Walsh basis simplifiy computation, which otherwise for diploid case would have been impractical

Only one mixing matrix as opposed to 2^ℓ is needed to compute next generation

For $\ell = 14$, using 2^{14} mixing matrices with each having $2^{14} \cdot 2^{14}$ entries would require 32 TB of memory, whereas one mixing matrix requires only 2 GB

Distance

Naive implementation:

$$\|\mathbf{f} - \mathbf{q}\|^2 = \sum_{\alpha} (\mathbf{f}_{\alpha} - \mathbf{q}_{\alpha})^2 \longrightarrow 2^{\ell} \cdot 2^{\ell} \text{ terms}$$

Our implementation:

$$S_{\mathbf{f}} = \{\alpha \mid \mathbf{f}_{\alpha} > 0\}$$

$$\|\mathbf{f} - \mathbf{q}\|^2 = \sum_g (\mathbf{p}_g)^2 + \sum_{\alpha \in S_{\mathbf{f}}} \mathbf{f}_{\alpha} (\mathbf{f}_{\alpha} - 2\mathbf{q}_{\alpha}) \rightarrow 2^{\ell} + |S_{\mathbf{f}}| \text{ terms}$$

Part-II: Applications

Question 1: Distance Between Finite and Infinite Population

Chebyshev's inequality:

$$\|\tau(\mathbf{p}) - \mathcal{G}(\mathbf{p})\| \leq \frac{k}{\sqrt{N}}$$

Jensen's inequality:

$$\mathcal{E}(\|\tau(\mathbf{p}) - \mathcal{G}(\mathbf{p})\|) \leq \frac{\sqrt{1 - \|\mathcal{G}(\mathbf{p})\|^2}}{\sqrt{N}}$$

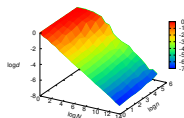
Geometric point of view:

$$\sup_{\xi \in \Lambda} \inf_{\mathbf{p} \in \Lambda_N} \|\xi - \mathbf{p}\| = O(1/\sqrt{N})$$

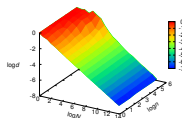
Does the distance decrease in practice like $1/\sqrt{N}$?

Convergence: Results

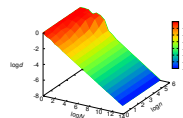
$$\chi = 0.1, \mu = 0.001$$



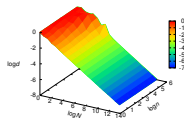
(a) $\ell = 4$



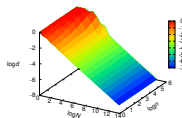
(b) $\ell = 6$



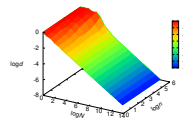
(c) $\ell = 8$



(d) $\ell = 10$



(e) $\ell = 12$

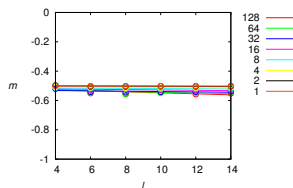


(f) $\ell = 14$

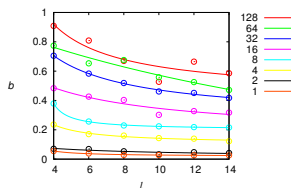
Figure : Convergence of finite population behaviour

Regression

$$\log d = m \log N + b$$



(a) Slope m



(b) Intercept b

Figure : Regression parameter for generation $n \in \{1, 2, 4, 8, 16, 32, 64, 128\}$

$$d \approx N^m e^b$$

From figure (a) above, $m \approx -(\frac{1}{2})$

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

Convergence: Conclusion

Can the distance between finite and infinite population decrease like $1/\sqrt{N}$?

Yes

Question 2

Finite Population Oscillation

Limits

Infinite population evolution

$$\mathbf{p}, \mathcal{G}(\mathbf{p}), \mathcal{G}^2(\mathbf{p}), \dots$$

may converge to a fixed point

$$\mathcal{G}(\omega) = \lim_{n \rightarrow \infty} \mathcal{G}^n(\mathbf{p}) = \omega$$

But under some circumstances, evolution converges to a periodic orbit that oscillates between two fixed points, \mathbf{p}^* and \mathbf{q}^*

Periodic Orbit: Necessary and Sufficient Conditions

For some $g \in \mathcal{R}, g \neq 0$

$$\begin{aligned}-1 &= \sum_j (-1)^{g^T j} \mu_j \\ 1 &= \sum_{k \in \bar{g}\mathcal{R}} \chi_{k+g} + \chi_k\end{aligned}$$

Infinite populations converge to a periodic orbit

Can finite populations also exhibit oscillation from random initial populations?

Previous Works on Oscillation

Akin (1982) proved existence of cycling for continuous-time 2-bit diploid model

Hasting (1981) studied cycling in populations with infinite 2-bit diploid population model

Wright and Bidwell (1997) provided examples of cycling in an infinite haploid model with crossover and mutation for 3 bit and 4 bit populations

Wright and Agapie (2001) described cycling in infinite population for up to 4 bits, and also presented data for cycling in finite population

Difference From Previous Works

Akin considers continuous time, we consider discrete time

Hastings' study is limited to two bits, and only crossover, no mutation

Wright and Bidwell consider specific parameter values

Wright and Agapie use dynamic mutation

Simulation

Simulations were run for both haploid and diploid populations

Random initial population

$$\ell \in \{8, 10, 12, 14\}$$

$$N \in \{4096, 40960, 81920\}$$

To visualize, distance of population to fixed points \mathbf{p}^* and \mathbf{q}^* is plotted

Oscillation: Results

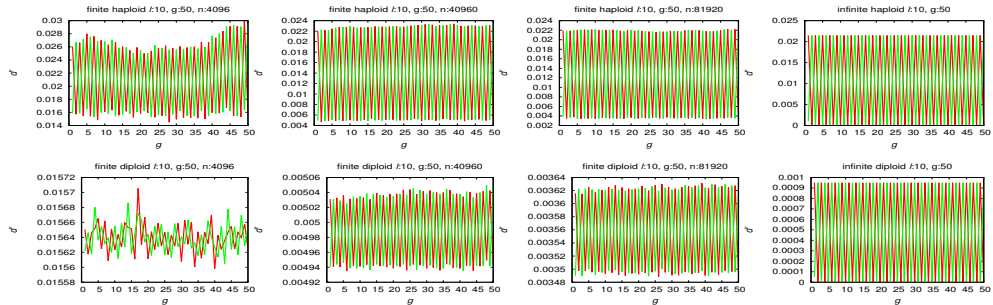


Figure : Infinite and finite population behavior for genome length $\ell = 10$

Oscillation: Conclusion

Finite populations can exhibit approximate oscillations

Question 3

Oscillation Under Mutation-Violation

$$-1 \neq \sum_j (-1)^{g^T j} \mu_j$$

Mutation-Violation

$$\mu_0 := \epsilon$$

$$\mu_i := (1 - \epsilon)\mu_i$$

No periodic orbits for infinite population

This modification makes the Markov chain regular

No periodic orbits for finite population

Can finite population exhibit approximate oscillations?

Simulation

$$\epsilon \in \{0.01, 0.1, 0.5\}$$

$$\ell \in \{8, 10, 12, 14\}$$

$$N \in \{4096, 40960, 81920\}$$

Distance to limits p^* and q^* without violation ($\epsilon = 0$) are plotted

Mutation-Violation: Results

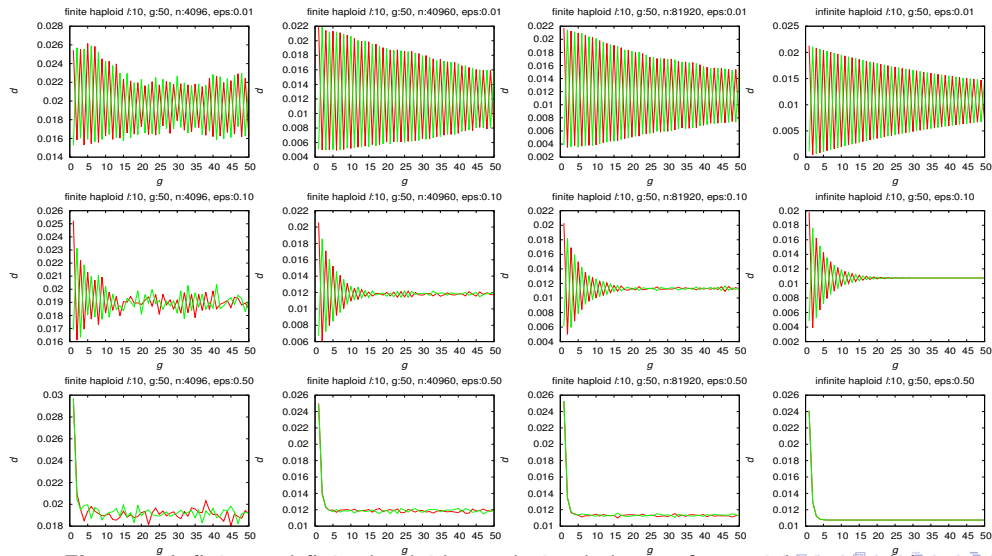


Figure : Infinite and finite haploid population behavior for μ violation and $\ell = 10$

Mutation-Violation: Results

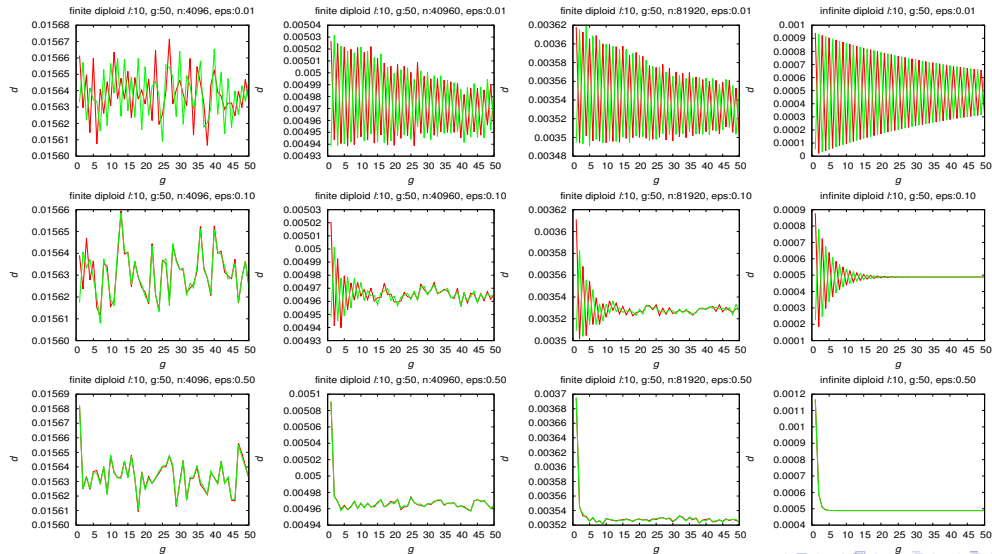


Figure : Infinite and finite diploid population behavior for μ violation and $\ell = 10$

Mutation-Violation: Conclusion

Finite populations can exhibit approximate oscillation when mutation-violation is small

If violation is large, then oscillation decreases

Question 4

Oscillation under Crossover-Violation

$$1 \neq \sum_{k \in \bar{g}\mathcal{R}} \chi_{k+g} + \chi_k$$

Crossover-Violation

$$\chi_i := (1 - \epsilon)\chi_i$$

$$\chi_j := \epsilon \quad j \text{ is chosen where } \chi_j = 0$$

No periodic orbit exists for infinite population

Can finite populations exhibit approximate oscillation?

Simulation

$$\epsilon = \{0.01, 0.1, 0.5\}$$

$$\ell = \{8, 10, 12, 14\}$$

$$N = \{4096, 40960, 81920\}$$

Distance to limits \mathbf{p}^* and \mathbf{q}^* without violation ($\epsilon = 0$) are plotted

Crossover-Violation: Results

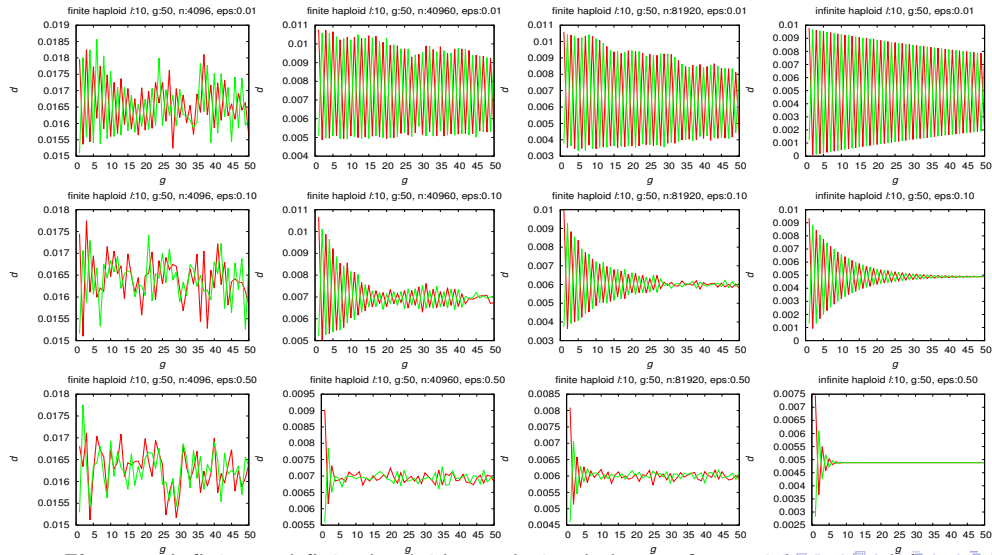


Figure : Infinite and finite haploid population behavior for χ violation and $\ell = 10$

Crossover-Violation: Results

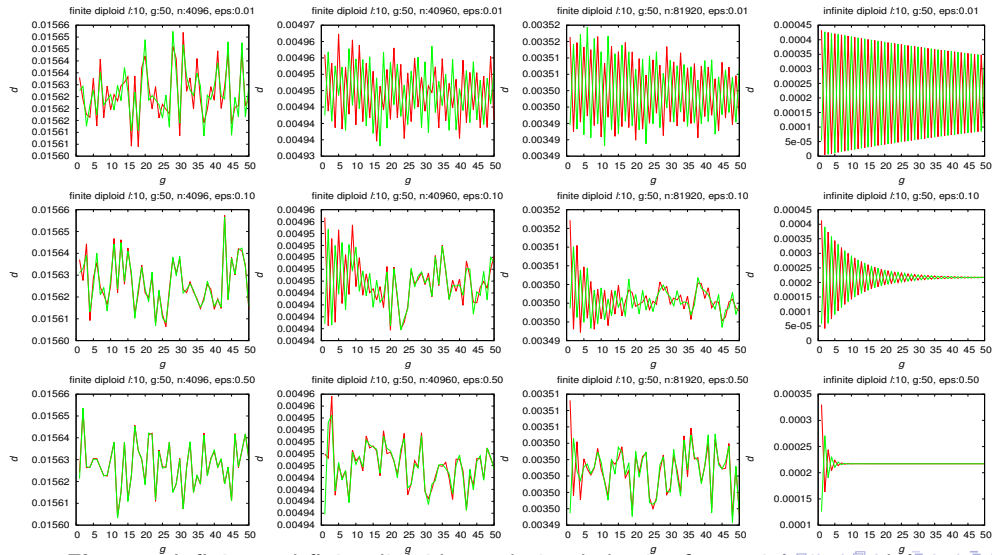


Figure : Infinite and finite diploid population behavior for χ violation and $\ell = 10$

Crossover-Violation: Conclusion

Finite populations can exhibit approximate oscillation when crossover-violation is small

If violation is large, then oscillation decreases

Conclusion

Vose's haploid model makes computation efficient in diploid case by reducing to haploid case

Distance between finite population and infinite population can decrease like $1/\sqrt{N}$

When infinite populations oscillate, finite populations exhibit approximate oscillation

Finite populations can exhibit approximate oscillation for small violation

Finite populations can exhibit approximate oscillation for small crossover-violation

Thank You!!

