

1. • I will talk about my thesis:
 “Simulation of Simple Evolutionary System”
2. • I will give background, and computations involving our model in the simulations we did.
 • Then I will address these four questions, and make concluding remarks.
3. Part I
4. • Population is a collection of length ℓ binary strings
 • Population can be represented by a vector p ; the j th component is the proportion of string j in the population.
 • For example, if P is this, 3rd component of vector p is equal to $1/3$.
5. • \mathcal{R} is the set of length ℓ binary strings and operations under \mathcal{R} are bitwise modulo 2 operations. Z is cardinality of \mathcal{R} .
 • Here are examples of some operations under \mathcal{R} . This is addition, this is multiplication, and this is complement.
6. • Crossover is mask based operator and is defined using the bitwise operations in \mathcal{R} .
 • The use of masks dates back to Geiringer, 1944
 • Crossover exchanges bits in parents u and v using crossover mask to produce children u' and v' using this
 • Results depend on mask m and 1's in mask determines how exchange of bits in parents take place.
 • Here's an example. These parents crossover to produce these offsprings.
 • In this example, since first two bits are 1 in the mask, for this offspring first two bits are from parent u and last two bits are from parent v and vice-versa in other offspring.

- χ_m gives probability of using crossover mask m .
7.
 - Mutation is also mask based operator and is defined using the bitwise operations in \mathcal{R} .
 - Mutation flips bits in x using mutation mask m . Bits of x are flipped in position where mask m has bit 1.
 - In this example, fourth bit of x is flipped because fourth bit of m is 1.
 - μ_m gives probability of using mutation mask m
 8.
 - This flowchart illustrates finite population genetic algorithm for haploids
 - We start from random initial population p .
 - Then randomly select parents u and v ,
 - And crossover to produce u' and v' according to randomly chosen crossover mask
 - We keep one of u' , v' and mutate using randomly chosen mutation mask to produce gamete g
 - We repeat above procedures to form next generation $\tau(p)$
 9.
 - This flowchart illustrates finite population genetic algorithm for diploids
 - Difference from haploids is that each diploid has two haploid components and they crossover and mutate to form a gamete g_0 .
 - Another gamete g_1 is produced from another parent diploid
 - Gametes g_0 and g_1 is combined to form offspring gamete
 10.
 - Random Heuristic Search generalizes simple genetic algorithm
 - Given population p , τ is transition rule that maps p to the next generation p' where p and p' both belong to population space Λ_N . N is population size.

- However, τ is stochastic function of crossover and mutation, and can not be predicted with certainty
 - Finite population evolution sequence forms Markov chain
- 11.
- In infinite population model, population is modeled by vector \mathbf{p} where \mathbf{p} belongs to population space Λ .
 - \mathcal{G} is a function that maps \mathbf{p} to the next generation \mathbf{p}' where j th component is proportion of string j occurs in the next generation.
 - The sequence shows evolution of \mathbf{p} under infinite population model.
 - \mathcal{G} gives the expectation of $\tau(\mathbf{p})$
 - And the variance of finite population in next generation is given by this.
- 12.
- This is our model setup for diploid population. It is infinite population model.
 - We consider diploid genome α with genome length l .
 - Population is modeled by vector \mathbf{q}
 - q_α is prevalence of diploid α
 - $t_\alpha(g)$ is transmission function which is probability of gamete g being produced from parent α
 - \mathbf{q}' is next generation
 - This is the standard evolutionary model from population genetics.
 - This assumes panmixia i.e., parents mate randomly without bias.
- 13.
- Diploids can be determined in terms of haploid distributions.
 - Haploids can be determined in terms of diploid distributions.
 - And evolution equation in terms of haploid distributions can be expressed as this

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

- Square brackets here are Iverson brackets which returns 1 if expression inside is true and returns 0 when expression inside is false.
 - In the matrix form, evolution equation can be expressed as $\mathbf{p}'_g = \mathbf{p}^T M_g \mathbf{p}$
 - \mathbf{p}' is next generation.
 - Where matrix $M(g)$ describes transmission function. If $g = 0$, we call it mixing matrix.
- 14.
- It is mutation distribution we use. μ of i gives probability that i is used as mutation mask. μ is mutation rate.
 - It is crossover distribution we use. χ of i gives probability that i is used as crossover mask. χ is crossover rate.
 - t of g is transmission function. This is expensive to compute because it has three sum terms in it.
- 15.
- Component of Walsh matrix W is given by this, $Z = 2^\ell$
 - Walsh transform of vector w is given by this. and Walsh transform of matrix A is given by this.
 - Here's the algorithm of computing Walsh transform.
 - Using this algorithm, Walsh transform of vector can be computed in $O(Z \log Z)$ and for matrix, it takes $O(Z^2 \log Z)$
- 16.
- Walsh transform of Mixing matrix M is given by the expression

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

- Evolution eqn in Walsh basis takes this form

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

where \widehat{p}' is next generation

17.
 - Now if we compare evolution eqn in Walsh basis to what we had before,
 - we have single sum in computing p' compared to double sum from matrix multiplication before.
 - We need only one mixing matrix compared to 2^l matrices for different g s.
 - Also calculating each mixing matrix in previous eqn required 3 sum terms, which is reduced to 1 sum term in walsh basis.
18.
 - So in summary, Reduction to haploid model and Walsh basis simplify 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
 - Consider $l = 14$, 2^{14} matrices would require 32 TB of memory, while one mixing matrix would require only 2 GB of memory
19.
 - In the 2nd part of this presentation, I talk of distance between finite population and infinite population.
 - If f is finite population and q is infinite population,
 - With naive computation as shown, there would be $2^l \cdot 2^l$ multiplication terms to be computed because α is diploid and have two components
 - We simplify it as shown in eqn (2) where it needs only $2^l + \text{cardinal value of } S_f$ terms
 - S_f is set of diploids that are in finite populations and is byproduct of finite GA.
 - Writing code to compute in the Walsh basis, and running simulations using the Walsh basis is a significant part of my thesis. It represents the "Efficient" part in the thesis title.
20. Part II

21.
 - Our first question concerns about convergence of finite population to infinite population.
 - Chebyshev's inequality, Jensen's inequality and geometric point of view, all, suggest the distance might decrease as $1/\sqrt{N}$
 - But all of 3 implications are from inequalities. The distance might decrease much smaller than $1/\sqrt{N}$.
 - We investigate if the distance in fact can decrease like $1/\sqrt{N}$ in practice.
22.
 - We ran some simulations with $\chi = 0.1$ and $\mu = 0.001$ for different values of l
 - These graphs are in logarithmic scale.
 - This axis is the distance, (point to axis) This axis is population size, (point to axis) this axis is number of generations
 - The thing to notice about these graphs is how flat and planar they are.
23.
 - So we model the data by linear equation
 - Regression shows slope m is nearly equal to negative half $(-1/2)$.
 - This proves distance does decrease as $1/\sqrt{N}$
24.
 - $1/\sqrt{N}$ was theoretical upper bound but our simulation shows it is really very good bound and the distance decreases as $1/\sqrt{N}$.
25.
 - Our 2nd question concerns about oscillation in finite populations.
26.
 - The infinite population sequence $\mathbf{p}, \mathcal{G}(\mathbf{p}), \mathcal{G}^2(\mathbf{p}), \dots$ may converge to a point, and we call that point a fixed point.
 - But under certain conditions, the sequence converges to a periodic orbit between two fixed points, say \mathbf{p}^* and \mathbf{q}^*
 - \mathbf{p}^* is fixed point to population in even time steps and \mathbf{q}^* is fixed point to population in odd time steps

27. • And the conditions are: For some g

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

- Mutation distribution should satisfy this condition and crossover should satisfy this condition
 - We try to answer our 2nd question through simulations; that is can finite populations exhibit oscillations from random initial populations when infinite populations oscillate?
28. • Few people have studied oscillations in the past: Akin 1981, Hastings 1982, Wright and Bidwell 1997, and Wright and Agapie 2001.
29. • However,
- Akin considers - continuous time model, we consider - discrete time model
 - Hastings' study is limited to two bits length, includes only crossover, but no mutation; we consider both crossover and mutation
 - Wright and Bidwell compute specific set of parameter values; there is particular relation between crossover, mutation and population values; in contrast we use random crossover, mutation and population.
 - Wright and Agapie use dynamic mutation that depends upon where population is in the population space while we use static mutation
30. • Simulations were run for both haploid and diploid populations
- To visualize oscillations, distance to fixed points (p^*, q^*) are plotted
31. • These are the results for haploid and diploid population of length 8
- top row is for haploid, bottom row for diploids
 - Population size is in increasing order in columns

- In the figure, Green line is distance of population to p^* , red line is distance to q^*
 - As population size increases, oscillation approaches the behavior exhibited by infinite population
32. • So our answer to the 2nd question is
- Yes, finite populations can exhibit approximate oscillation when infinite populations oscillate
33. • Question 3 concerns the robustness of finite population oscillation under mutation-violation
- By mutation-violation we mean, mutation distribution holding this condition.
 - This means no periodic orbits for infinite population
34. • We introduce violation ϵ as following
- The modification in μ makes the Markov chain regular
 - This means no periodic orbits for finite population
 - We study if finite population can exhibit approximate oscillations in that case
35. • Simulations were run for different values of ϵ .
- Distance of population to limits p and q without violation are plotted
36. • These figures show results for haploid population behavior of string length 8
- As population increases, finite population oscillation converges to infinite population oscillation
 - As ϵ increases, oscillation diminishes
 - Oscillation appears to diminish with time, and in case of infinite population, oscillation dies out after some time.

- Even though finite population also appears to be dying out, since Markov chain is regular, population must visit every population state infinitely. So these population will reoccur and oscillate infinitely often (for $\epsilon = 0.1$).
- 37.
- Our conclusions from this simulation are
 - Finite populations exhibit approximate oscillation even if Markov chain is regular when violation is small
 - If violation becomes larger, finite population oscillation decreases
 - As string length increases, oscillation degrades
- 38.
- Our previous question concerns with the robustness of finite population oscillation under mutation-violation, our next question also concerns with the robustness of finite population but under crossover-violation
 - By crossover-violation, we mean this condition holds.
 - Crossover-violation means no periodic orbit exists for infinite population
- 39.
- We introduce violation ϵ in crossover distribution
 - And investigate if finite population can exhibit approximate oscillations in this case
- 40.
- Simulations were run for different values of ϵ
 - Distances of population to limits p and q without violation are plotted
- 41.
- Graphs show results for haploid population behavior of string length 8
 - Results show similar behavior to that of violation in mutation case
 - However, we noticed that rate of damping of amplitudes of oscillation is slower than in mutation violation

- 42.
 - Conclusions from experiment on violation in crossover are
 - Finite populations can exhibit approximate oscillation if violation is small
 - If violation becomes larger, finite population oscillation decreases
- 43.
 - Overall conclusions from this research are
 - 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 exhibit approximate oscillation for small mutation-violation
 - Finite populations exhibit approximate oscillation for small crossover-violation
- 44.
 - We noticed some peculiar behavior with oscillation in violation case.
 - \mathbf{p}^* and \mathbf{q}^* are limits without violation and \mathbf{z}^* is limit with violation.
 - These dotted lines represent hyperplanes perpendicular to the line joining \mathbf{p}^* and \mathbf{q}^* .
 - Our checks show \mathbf{z}^* lies inside these hyperplanes and also equidistant from \mathbf{p}^* and \mathbf{q}^*
 - In the checks we did for haploid case, infinite population and finite population both lay between these hyperplanes.
 - In the checks we did for diploid case, infinite population lay between these hyperplanes but finite population did not.