# A Computational View of Population Genetics (preliminary version)

Yuval Rabani[†] Yuri Rabinovich[†] Alistair Sinclair[‡]

## Abstract

This paper contributes to the study of nonlinear dynamical systems from a computational perspective. These systems are inherently more powerful than their linear counterparts (such as Markov chains), which have had a wide impact in Computer Science, and they seem likely to play an increasing role in future. However, there are as yet no general techniques available for handling the computational aspects of discrete nonlinear systems, and even the simplest examples seem very hard to analyze. We focus in this paper on a class of quadratic systems that are widely used as a model in population genetics and also in genetic algorithms. These systems describe a process where random matings occur between parental chromosomes via a mechanism known as "crossover": i.e., children inherit pieces of genetic material from different parents according to some random rule. Our results concern two fundamental quantitative properties of crossover systems:

1. We develop a general technique for computing the rate of convergence to equilibrium. We apply this technique to obtain tight bounds on the rate of convergence in several cases of biological and computational interest. In general, we prove that these systems are "rapidly mixing", in the sense that the convergence time is very small in comparison with the size of the state space.
2. We show that, for crossover systems, the classical quadratic system is a good model for the behavior of finite populations of small size. This stands in sharp contrast to recent results of Arora *et al* and Pudlak, who show that such a correspondence is unlikely to hold for general quadratic systems.

[†]Department of Computer Science, University of Toronto, Toronto, Ontario M5S 1A4, Canada. Email: {rabani,yuri}@cs.toronto.edu.

[‡]Computer Science Division, University of California, Berkeley CA 94720-1776, U.S.A. Email: sinclair@cs.berkeley.edu.

## 1 Introduction

It is well known that linear dynamical systems (in the form of branching processes and Markov chains) have had a major impact in algorithmic and analytical applications in Computer Science (see, e.g., [2] and the surveys [12,14]). In contrast, and despite their inherently greater power, the study of *nonlinear* dynamical systems in Computer Science is still in its infancy (though we mention some recent exceptions below). Motivated by the belief that nonlinear systems will play an increasing role in future, we aim in this paper to contribute to research on their basic properties, and to develop analytical tools for them analogous to those that have been essential in applications of linear systems.

Many nonlinear systems of computational interest can be viewed as *quadratic dynamical systems* (QDS). In a QDS, the probability distribution (also called the "population") $p_t$ at time $t$ evolves under random pairwise interactions, so that

$$p_{t+1}(x) = \sum_{uvw} p_t(u)p_t(v) \left(\tfrac{1}{2}\beta_{uvwx} + \tfrac{1}{2}\beta_{uvxw}\right), \qquad (1)$$

where $u, v, w, x$ range over the set of states (or "types"), and $\beta_{uvwx}$ denotes the probability that, in an interaction between types $u$ and $v$, the outcome is the pair of types $w$ and $x$. Thus interactions occur between pairs of types $u, v$ ("parents") selected independently at random from the current population $p_t$; the effect of an interaction is to replace the parents by two "offspring" types created according to the distribution $\beta_{uv**}$. The coefficients $\beta_{uvwx}$ satisfy $\sum_{wx}\beta_{uvwx} = 1$. Note that (1) is a natural generalization of a Markov chain, whose dynamical equation has the form $p_{t+1}(x) = \sum_u p_t(u)\alpha_{ux}$. A QDS is *symmetric* if the probabilities $\beta_{uvwx}$ satisfy the symmetry conditions $\beta_{uvwx} = \beta_{vuwx} = \beta_{uvxw}$, and the reversibility condition $\beta_{uvwx} = \beta_{wxuv}$. Symmetry of QDSs is analogous to reversibility of Markov chains.

The above framework is quite general, and captures several important systems from the natural sciences: e.g., Boltzmann's model of an ideal gas (where the types are velocities of gas molecules, and interactions are collisions between them) [17]; the Hardy-Weinberg model of population genetics (where the types are the genotypes of some species, and interactions are matings between parents) [15]; and Volterra's model of a predator-prey ecology (where the types are the predator and the prey, and interactions correspond to procreation and kills) [10]. The first two of these examples are symmetric.

In Computer Science, quadratic systems arise in genetic algorithms [11], a class of heuristics for combinatorial optimization based loosely on the paradigm of natural selection. This approach is attractive, but the resulting algorithms are usually impossible to analyze (though see [20,4]). The key ingredient of such algorithms is a "mating" operator, which in many cases is modeled by equation (1).† Nonlinear systems have also been used as an analytical tool in various contexts: examples included Valiant's construction of small-sized monotone circuits for majority [21], Clarkson *et al*'s approximation algorithm for the center point [6], and the analysis of on-line fair scheduling by Ajtai *et al* [1]. The last two examples hint at the power of nonlinear systems in providing a more compact and theoretically more elegant representation of linear systems whose states are vectors in a high dimensional space.

In this paper, we focus on two questions of central importance to computational applications of QDSs. The first question concerns the *rate of convergence* of the system to its equilibrium state. (Recently, Rabinovich *et al* [19] have shown that, under a mild technical condition, any symmetric QDS eventually converges to a stationary population.) The rate of convergence is fundamental in computational contexts, and tools for analyzing it have been essential in applications of linear systems. In the nonlinear case, however, there is as yet very little general theory (such as eigenvalues) to guide us, and even the simplest systems appear extremely hard to analyze. For example, the rate of convergence of the Boltzmann equation is a long-standing open problem in theoretical physics [5].

The second question we address concerns the feasibility of representing the evolution of a finite population by a QDS. The dynamical system (1) operates on probability distributions, and hence implicitly assumes that the population consists of an infinite number of individuals. Thus (1) is actually a convenient theoretical abstraction of an analogous physical, biological or computational process in which the population size is finite, and of manageable size. The question is: when does the QDS faithfully represent the behavior of the finite population, so that this abstraction is justified? This is an instance of the following more general question: when is it possible to efficiently *simulate* a QDS? (If the QDS faithfully represents a small finite population, then this finite system offers an efficient simulation.) This is not an interesting issue in the case of linear systems such as Markov chains: the probability distribution at time $t$ is represented precisely by a single point (a population of size 1) which performs a random walk as prescribed by the transition probabilities of the chain. In the quadratic case, however, it seems that in order to obtain a single random sample from the population $p_t$, we require two independent random samples from $p_{t-1}$, four from $p_{t-2}$, and so on, so that the population size required for a faithful simulation grows exponentially with time. This intuition was formalized recently by Arora *et al* [3] and by Pudlak [18], who proved that a QDS is capable of solving any problem in PSPACE in a polynomial number of steps. This means that we cannot expect efficient simulation to be possible in general (unless RP = PSPACE). In particular, a QDS is in general not a good representation of the behavior of small finite populations. Thus it is an interesting practical and theoretical question to determine under which additional assumptions such a correspondence does hold.

In this paper, we give a complete analysis of both the above questions — rate of convergence and faithful representation — for a subclass of QDSs that are extensively studied in population genetics, and are also frequently employed in the mating operator of genetic algorithms. Here the types are chromosomes, represented as strings of length $n$ over a finite alphabet, and the interactions are "crossovers": i.e., given a pair of parent strings $u, v$, the first child, $w$, is obtained by selecting a subset $S \subseteq \{1, \ldots, n\}$ of positions from some probability distribution, and taking symbols at positions $i \in S$ from $u$, and the remainder from $v$. The second child, $x$, is constructed in the same way using the complement of $S$. By choosing the probability distribution over $S$ (or "crossover distribution") appropriately, one can cast any desired reproductive rule in this framework. (For precise definitions and concrete examples, see Section 2.)

It is a classical result dating back to 1944 [8] that any such system (under an obvious non-degeneracy assumption) converges to a stationary population in which the symbols at different positions are mutually independent; moreover, the probability distribution at each position can be determined from the initial population. However, despite the wide use of this model, there were apparently no precise quantitative results about the rate of convergence or the relationship with finite populations.

Our analysis settles these questions as follows. First, we prove that any crossover system converges extremely fast to stationarity, namely after only $A \log n$ steps, where the factor $A$ has a natural interpretation in terms of the rate at which the crossover distribution separates positions in the string. We also show that this bound is tight (up to a small constant factor) for several specific crossover distributions that are used in practice. This is apparently the first time that precise convergence rates have been given for these systems: the best previous analysis, using much more complex methods, gave only much weaker asymptotic information about the rate [15].

Our second contribution demonstrates that, for crossover systems, the QDS model is an appropriate mathematical abstraction for the behavior of relatively small populations of individuals. Specifically, we prove that $t$ steps of the quadratic system accurately represent $t$ generations of a finite population of size only $O(n^2 t)$. This linear dependence on time is a dramatic improvement over the obvious exponential bound mentioned above, and provides a positive counterpart to the negative results of Arora *et al* and Pudlak about simulating a general QDS. Again, this seems to be the first result of its kind. The question of finite populations has recently been addressed in the population genetics literature [13], but the notion of simulation adopted there is much weaker than ours, and the resulting finite system cannot be said to faithfully reflect the original nonlinear one.

Our techniques are elementary, but rely crucially on the novel observation that a crossover system has an

†In genetic algorithms, the situation is further complicated by the interleaving of mating with a "selection" operator, in which fitter types are favored; see, e.g., [11].

equivalent formulation as an inverse process in which individuals randomly select genetic material from their ancestors. This approach parallels the use of inverse processes in the analysis of certain card-shuffling Markov chains [7]. The inverse process allows a surprisingly clean treatment of finite populations. For the bounds on convergence rate, we further interpret the inverse process as a random partitioning process on sets, and analyze a suitable stopping time. This process seems to be of independent combinatorial interest.

The remainder of this paper is organized as follows. In Section 2 we give some definitions and notation concerning crossover systems, quote some basic facts about them, and present the inverse process that underlies all our results. In Section 3 we consider the rate of convergence, and in Section 4 finite populations.

## 2 Crossover operators

### 2.1 Definitions and basic facts

As indicated above, we consider a classical abstract genetic model in which chromosomes in some species are represented as strings of length $n$ over a finite alphabet. Each position in the string is the locus of a gene, and the symbol in that position is the corresponding gene value, or *allele*. Mathematically there is no essential loss of generality in assuming that there are only two possible alleles at each locus. Therefore, we shall work throughout with the set $\{0,1\}^n$ of all 0-1 strings of length $n$, which we shall also refer to as *types*. Our results can be generalized in an obvious fashion to handle more than two possible alleles at each locus.

The *population at time t* is a probability distribution $p_t$ over $\{0,1\}^n$, such that $p_t(v)$ is the proportion of individuals of type $v$. The population evolves in discrete time as a quadratic dynamical system of the form given in equation (1). In this special case, mating is controlled by a probability distribution, $\Pi$, on the set of all subsets of the set of bit positions $\{1,\ldots,n\}$. In a mating between parents $u,v$, a child $w$ is determined by the following random process:

(i) select a subset $S$ randomly according to $\Pi$;

(ii) set $w_i = \begin{cases} u_i & \text{if } i \in S; \\ v_i & \text{otherwise.} \end{cases}$

Thus the child $w$ receives bits from parent $u$ in positions $i \in S$, and from $v$ in the remaining positions. The dynamical equation for the system may now be written formally as

$$p_{t+1}(w) = \sum_S \sum_{\substack{u:u_S=w_S \\ v:v_{\bar S}=w_{\bar S}}} p_t(u)p_t(v)\Pi(S). \qquad (2)$$

Here, for a subset $S \subseteq \{1,\ldots,n\}$ and a vector $v \in \{0,1\}^n$, $\bar S$ denotes the complement $\{1,\ldots,n\} - S$, and $v_S$ is the projection of $v$ onto $S$ (i.e., the vector obtained from $v$ by deleting the bits in positions $i \in \bar S$). It is not hard to see that this process can be formulated as a symmetric QDS as defined in equation (1): simply introduce a second child $x$ such that $w \oplus x = u \oplus v$ and symmetrize.

An equivalent view of this process is that $S$ specifies a sequence of positions along the chromosome at which the parent from which the child inherits alleles changes: these events are known as *crossovers*. We shall refer to the distribution $\Pi$ as a *crossover distribution.* The crossover distribution controls recombination, or the relationship between pieces of genetic material passed from parents to children. Standard examples of crossover distributions include the following.

**Example 1** *Uniform crossover.* Here crossovers occur independently at random at all positions with probability $\frac{1}{2}$. Equivalently, $\Pi$ is the uniform distribution on all subsets of $\{1,\ldots,n\}$.

**Example 2** *One-point crossover.* Here there is a single crossover at a position selected uniformly at random, i.e., $\Pi$ is the uniform distribution on all sets of the form $\{1,\ldots,i\}$, for $1 \le i \le n$. A common generalization is $k$-point crossover, in which exactly $k$ crossovers occur at randomly chosen distinct positions. This model is particularly popular in genetic algorithms.

**Example 3** *The Poisson model.* This is the most widely used model in population genetics, and dates back to 1919 [9]. A crossover occurs between positions $i$ and $i+1$ with probability $p_i \le \frac{1}{2}$, independent of other crossovers. (The name of this model derives from the fact that adjacent loci in the idealized string model are actually separated by large distances in the physical chromosome, and crossovers between a pair of these loci are presumed to occur in the physical model according to a Poisson process in the separating interval.)

There is currently much interest in finding crossover distributions that model real biological mating more accurately: see, e.g., [16] for an extensive discussion. The chief drawback of the Poisson model is the assumption of independence of adjacent intervals. Although we will illustrate our techniques for the above standard examples, we stress that they can be applied to *any* crossover distribution.

The following terminology will be useful in describing the behavior of crossover operators. A subset $S$ is said to *separate* positions $i,j \in \{1,\ldots,n\}$ if $i \in S$ and $j \in \bar S$ or vice versa. We shall call a crossover distribution $\Pi$ *non-degenerate* if, for all pairs of distinct positions $i,j$, there exists some subset $S$ with $\Pi(S) > 0$ that separates $i$ and $j$. All the above examples are plainly non-degenerate.

For any given initial population $p_0$, the dynamical system (2) defines a *trajectory* $(p_0, p_1, \ldots, p_t, \ldots)$ in population space. The first question that naturally arises is: what is the asymptotic behavior of trajectories? We say that a trajectory $(p_t)_{t=0}^{\infty}$ *converges to* a population $p$ if $p_t \to p$ pointwise as $t \to \infty$. The following classical result, first proved by Geiringer in 1944 [8], states that every trajectory converges to a stationary population in which the loci behave independently. (A *stationary* population is one which remains invariant under the operator of equation (2).)

**Theorem 1** [8] *For any non-degenerate crossover distribution, every trajectory $(p_t)$ converges to a stationary population $p_\infty$ defined by $p_\infty(v) = \prod_{i=1}^n c_i(v_i)$, where $c_i(0)$ and $c_i(1)$ are the probabilities of values 0 and 1 respectively at position $i$ in the initial population.* □

**Remarks:** (i) Plainly, the stationary population itself is not interesting and can be constructed by direct methods. Our concern here is with the transient behavior of the system, i.e., with its approach to stationarity.

(ii) The non-degeneracy assumption is inessential. Suppose some pair of positions $i, j$ is not separated by $\Pi$; this phenomenon is known as *complete linkage.* For a degenerate crossover distribution, the bit positions can be partitioned into equivalence classes under the relation of complete linkage; asymptotically, these classes are mutually independent, i.e., Theorem 1 still holds but with the index $i$ generalized in the obvious way to run over classes. $\square$

### 2.2 An inverse process

Our aim in this subsection is to reformulate the crossover process in a way that will ease our subsequent analysis. We will view a random individual from the population $p_t$ as a random variable over a sample space of labeled trees that describe the sequence of matings by which the individual was created. As will become clear shortly, this can be seen as an inverse process in which individuals select bits from their ancestors.

A *t-step derivation* is a complete binary tree of height $t$, in which some of the leaves are labeled with $n$-bit strings, and the interior nodes are labeled with strings from $\{\ell, r, *\}^n$ in a way to be described below. We picture the tree "upside-down", with its root at level $t$; each interior node, at level $i$ say, has two *parents* at level $i-1$. The idea is that the nodes at level $i$ represent the time-$i$ ancestors of the individual at time $t$, while the $\ell/r$ labels indicate the transitions that brought about its creation: an $\ell$ (respectively, $r$) label in position $i$ indicates that the $i$th bit was inherited from the left (respectively, right) parent. These symbols are called *active* symbols. Active symbols occur only where the corresponding bit is eventually inherited by the root. More formally, the labeling is defined as follows:

(i) The root is always labeled with a string from $\{\ell, r\}^n$.

(ii) Every other interior node is labeled with a string from $\{\ell, r, *\}^n$; if the node is a right (left) parent of a node with label $L$, then it has a symbol other than $*$ in each position $i$ for which $L_i = r$ ($L_i = \ell$), and a $*$ in all the remaining positions.

(iii) If a leaf is a right (left) parent of a node with a label which contains a symbol $r$ ($\ell$), then this leaf is labeled with an $n$-bit string.

We call nodes whose labels contain an active symbol *active nodes*, and leaves with a label *active leaves.* Given such a derivation $T$, we define an $n$-bit string $\mathrm{root}(T)$ by propagating bits from leaf labels to the root as indicated by the $\ell/r$ labels on the active nodes. This string is our individual from the population at time $t$.

It remains to specify a probability distribution over the labels so that the distribution of $\mathrm{root}(T)$ is indeed $p_t$. This is done as follows:

1. Choose the label $L$ on the root by selecting a subset $S$ at random from $\Pi$ and setting $L_i = r$ if $i \in S$, and $L_i = \ell$ otherwise.
2. For level $j = t-1$ down to 1, for each active node at level $j$ choose the label $L$ by selecting a subset $S$ at random from $\Pi$ and setting

$$L_i = \begin{cases} r & \text{if } i \in S \text{ and symbol } i \text{ is active;} \\ \ell & \text{if } i \in \bar{S} \text{ and symbol } i \text{ is active;} \\ * & \text{otherwise.} \end{cases}$$

3. Choose the labels on active leaves independently from $p_0$.

We denote by $\mathcal{D}_t$ the resulting sample space over $t$-step derivations. The following claim should be intuitively clear, and may readily be verified by induction.

**Lemma 2** *If the derivation $T$ is drawn at random from $\mathcal{D}_t$, the distribution of* $\mathrm{root}(T)$ *is precisely $p_t$.* $\square$

The sample space $\mathcal{D}_t$ will play a central role in our analysis in the remainder of the paper. Note that the construction of the sample space given above can be viewed as an inverse process in which individuals select bits from their parents.

## 3 The rate of convergence

Let $\Pi$ be an arbitrary non-degenerate crossover distribution. We have seen in Theorem 1 that all trajectories under $\Pi$ converge asymptotically to a well-defined stationary population. In this section we shall investigate the quantitative question of how many generations elapse before this asymptotic behavior sets in. To make this question precise, let us define the *mixing time* as

$$\tau(\epsilon) = \max_{p_0} \min\{t : \|p_{t'} - p_\infty\| \leq \epsilon \quad \forall t' \geq t\},$$

where $\|\cdot\|$ denotes variation distance and $\epsilon \in (0, 1]$. Our aim is to derive tight bounds on $\tau$ as a function of $n$, $\epsilon$ and the crossover distribution $\Pi$. In Section 3.1 we will obtain upper bounds on the mixing time; in Section 3.2, we will demonstrate that these bounds are essentially tight.

Our main tool throughout will be a random set partitioning process that is closely related to the inverse process introduced in the previous section. The process is a natural one and we believe it to be of independent combinatorial interest. It is the following: choose a subset $S$ at random according to $\Pi$, and partition the set $\{1, \ldots, n\}$ by intersecting it with $S$. Let the resulting sets be $K_1$ and $K_2$. Then partition $K_1$ and $K_2$ in similar fashion, by selecting further random subsets $S_1, S_2$ independently. Continue this process until only singleton sets remain. More formally, the process $(*)$ is defined as follows:

$k = 0$; $\sigma_0 = \{\{1, \ldots, n\}\}$
**repeat**
  $\sigma_{k+1} = \emptyset$
  **for** each $K \in \sigma_k$ with $|K| > 1$ **do**
    select $S$ at random according to $\Pi$
    $K_1 = S \cap K$; $K_2 = \bar{S} \cap K$
    $\sigma_{k+1} = \sigma_{k+1} \cup \{K_1, K_2\}$
  $k = k + 1$
**until** all sets in $\sigma_k$ are singletons

The relationship between this process and the derivation model introduced in the previous section should be clear. The partition $\sigma_k$ corresponds to the set of active labels at level $t-k$ of the tree; each active label is represented by a subset $K$ containing the positions of its active symbols, and the partition of $K$ into $K_1, K_2$ corresponds to the splitting of these symbols into $\ell, r$. The process stops when all sets in $\sigma_k$ are singletons; at this point, there are precisely $n$ active leaves, each of which contributes one bit to the root.

### 3.1 Upper bounds on the mixing time

We begin with a simple observation which relates the variation distance at time $t$ to a structural property of elements of the sample space $\mathcal{D}_t$.

**Lemma 3** *Let $T$ be a derivation drawn at random from $\mathcal{D}_t$. If $t$ is large enough that* $\Pr[T$ has $n$ active leaves$] \geq 1-\epsilon$*, then* $\|p_t - p_\infty\| \leq \epsilon$.

**Proof:** If $T$ has $n$ active leaves, all bits in root$(T)$ are descended from distinct ancestors at time 0. Hence, conditional on this event, the distribution of root$(T)$ is precisely $p_\infty$. □

The condition that $T$ has $n$ active leaves corresponds precisely to the stopping rule for the process $(*)$. Thus we can obtain upper bounds on the mixing time by bounding the tail distribution of this stopping time. Let $Z$ be the random variable (stopping time) that counts the number of iterations of the process $(*)$ (i.e., the value of $k$ at the conclusion), and for $\epsilon \in (0,1]$ define

$$\tau'(\epsilon) = \min\{t : \Pr[Z > t] \leq \epsilon\}.$$

The foregoing discussion, together with Lemma 3, immediately yields the following result.

**Theorem 4** $\tau(\epsilon) \leq \tau'(\epsilon)$. □

The detailed analysis of the stopping time $\tau'$ of course depends on $\Pi$; however, it is not hard to derive an estimate which, though apparently a little crude, actually gives good bounds in many cases. Define

$$r_{ij}(\Pi) \;=\; \Pr[S \text{ does not separate } i \text{ and } j],$$

where $S$ is drawn at random from $\Pi$, and set

$$r(\Pi) \;=\; \max_{i,j}\; r_{ij}(\Pi).$$

Note that our assumption of non-degeneracy implies that $r(\Pi) < 1$.

**Theorem 5** *The mixing time $\tau$ satisifies*

$$\tau(\epsilon) \;\leq\; \log_{1/r(\Pi)}(n^2\epsilon^{-1}) \;=\; \frac{1}{\ln r(\Pi)^{-1}}\left(2\ln n + \ln\epsilon^{-1}\right).$$

**Proof:** Consider some pair of positions $i, j$, and assume that, at the start of iteration $k$ of the above random process, $i$ and $j$ belong to the same set $K \in \sigma_k$. The probability that $i$ and $j$ belong to different sets in $\sigma_{k+1}$ is clearly at least $1 - r(\Pi)$. Since all iterations in the process are independent, if $k$ is at least $\log_{r(\Pi)^{-1}}(n^2\epsilon^{-1})$, then we can be sure that $i$ and $j$ belong to different sets in $\sigma_k$ with probability at least $1 - \epsilon/n^2$. But since there are only $\binom{n}{2}$ distinct pairs $i, j$, we know that *every* pair is separated in $\sigma_k$ with probability at least $1-\epsilon$. Clearly, once all pairs have been separated, $\sigma_k$ must consist entirely of singletons. Thus $\tau'(\epsilon) \leq \log_{r(\Pi)^{-1}}(n^2\epsilon^{-1})$, and the result follows from Theorem 4. □

**Remarks:** (i) Note that the bound on the mixing time in Theorem 5 is extremely small compared to the number of types, which is $2^n$. There is an initial delay (the "relaxation time") that depends only logarithmically on the string length $n$, followed by an exponential decay at rate $\ln r(\Pi)^{-1}$.

(ii) A weak (asymptotic in $\ln\epsilon^{-1}$) bound on the mixing time is given in [15, Theorem 6.6.1], by a much more complex method that involves solving the evolutionary equation explicitly. This bound, however, provides no information about the relaxation time, which is of central importance in computational applications. □

Theorem 5 immediately yields upper bounds on the mixing time for the classical crossover distributions defined in Section 2.1.

**Corollary 6** *The mixing time $\tau_{\rm unif}$ for uniform crossover satisfies*

$$\tau_{\rm unif}(\epsilon) \leq 2\log_2 n + \log_2 \epsilon^{-1}.$$

**Proof:** For uniform crossover $r(\Pi) = \frac{1}{2}$. □

**Corollary 7** *The mixing time $\tau_{\rm 1pt}$ for one-point crossover satisfies*

$$\tau_{\rm 1pt}(\epsilon) \leq n\ln n + n\ln\epsilon^{-1}.$$

**Proof:** For one-point crossover, $r(\Pi) = 1 - \frac{1}{n}$ (achieved for any pair of *adjacent* positions $i, j$), and, using the fact that $\ln(1+\alpha) \geq \alpha$, we immediately obtain the upper bound $2n\ln n + n\ln\epsilon^{-1}$. A closer examination shows, however, that it is enough to separate only the $n-1$ pairs of adjacent positions, rather than all $\binom{n}{2}$ pairs as in the proof of Theorem 5. This observation gets rid of the factor of 2 in the first term. □

**Corollary 8** *The mixing time $\tau_{\rm poisson}$ for the Poisson model satisfies*

$$\tau_{\rm poisson}(\epsilon) \leq A(2\ln n + \ln\epsilon^{-1}),$$

*where* $A = -1/\ln(\max_i\{1-p_i\})$.

**Proof:** We show that for the Poisson model,

$$r(\Pi) = \max_{1\leq i\leq n-1}(1-p_i).$$

This will imply the claim.

Consider any two loci $i < j$. Note that $1 - r_{ij}(\Pi)$ is exactly the probability that the number of crossovers in the interval between $i$ and $j$ is odd. We shall use the following notation. Let $P = \prod_{k=i}^{j-1}(1-p_k)$. $E_k$ denotes the event that exactly $k$ crossovers occur between $i$ and $j$; EVEN denotes the event that an even number of crossovers occur between $i$ and $j$; and ODD denotes the complement of EVEN. We have

$$\Pr[E_k] \;=\; P \times \sum_{\{i_1,i_2,\ldots,i_k\}} \prod_{l=1}^{k} \frac{p_{i_l}}{1-p_{i_l}},$$

where the sum is over all $k$-subsets of $\{i, i+1, \ldots, j-1\}$. Now it is not hard to see that

$$\Pr[\mathrm{EVEN}] - \Pr[\mathrm{ODD}] = \prod_{k=i}^{j-1}(1-2p_k).$$

Since $\Pr[\mathrm{EVEN}] + \Pr[\mathrm{ODD}] = 1$, we get

$$1 - r_{ij}(\Pi) = \Pr[\mathrm{ODD}] = \frac{1-\prod_{k=i}^{j-1}(1-2p_k)}{2}.$$

Since all the $p_k$ are at most $\frac{1}{2}$, this quantity is monotonically increasing in $j$. Therefore, the minimum is achieved for $j = i+1$, and thus $r(\Pi)$ is achieved at a pair of adjacent positions. □

Of course, one would not expect a single closed expression such as Theorem 5 to provide a tight bound on the relaxation time in every case. To obtain sharper bounds in specific cases, one can analyze the stopping time $\tau'(\epsilon)$ directly and appeal to Theorem 4. For example, we used this in the proof of Corollary 7 to remove the factor of 2 from our upper bound on $\tau_{1\mathrm{pt}}(\epsilon)$. As another illustration, we can also get the following improved bound for the Poisson model that is considerably sharper than that of Corollary 8 when the crossover probabilities $p_i$ vary widely.

**Theorem 9** *The mixing time $\tau_{\mathrm{poisson}}$ for the Poisson model satisfies*

$$\tau_{\mathrm{poisson}}(\epsilon) \le 2\min\{t : \textstyle\sum_i (1-p_i)^t \le \epsilon^{1/2}\}.$$

**Proof:** Let $r_{ij} = r_{ij}(\Pi)$. By the argument in the proof of Corollary 8, for any two loci $i < j$, we have $r_{ij} \le \min\{1-p_i, 1-p_{j-1}\}$.

Consider $t$ for which $\sum_i (1-p_i)^t \le \epsilon^{1/2}$. Since $(1-p_i)^t$ is the probability that positions $i, i+1$ remain unseparated at time $t$, we may conclude that at time $t$ the probability that any pair of adjacent positions remains unseparated is at most $\epsilon^{1/2}$. The probability that *any* pair of positions (not necessarily adjacent) remains unseparated at time $2t$ is bounded above by $\sum_{i<j}(r_{ij})^{2t}$. We get

$$\begin{aligned}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}(r_{ij})^{2t} &\le \sum_{i=1}^{n-1}\sum_{j=i+1}^{n}(1-p_i)^t(1-p_{j-1})^t \\ &\le \left(\sum_{i=1}^{n-1}(1-p_i)^t\right)^2 \\ &\le \epsilon. \qquad \square\end{aligned}$$

Note that the bound in Theorem 9 can be effectively computed numerically, for any values of $p_i$ and $\epsilon$. We shall see in the next subsection that this bound is tight to within a factor of $\mathrm{O}(\log\log\log n)$.

### 3.2 Lower bounds on the mixing time

We now turn to lower bounds on the mixing time. Our aim here is to show that the upper bounds we obtained in the previous section are tight, to within small constant factors. (In the case of the Poisson model, we achieve a factor that is not quite constant.) The analysis in this subsection is somewhat technical, and may safely be skipped by the casual reader who is prepared to take these statements on trust.

Our upper bounds did not depend on the initial population. Of course, we cannot expect this to be the case for lower bounds. In what follows, we will always use the initial population $p_0(0^n) = p_0(1^n) = \frac{1}{2}$. Thus the stationary population will be the uniform distribution over all strings.

As is to be expected, our estimates of the mixing time consist of two terms: one, containing $\epsilon$, is the *asymptotic part*, and reflects the behavior of the system for small $\epsilon$, while the other more interesting term (the *relaxation time*) measures the time needed for the system to reach equilibrium. For definiteness, we may formally define the relaxation time as $\tau(\frac{1}{4})$. (The choice of the constant $\frac{1}{4}$ is arbitrary.)

It is a straightforward matter to verify that the asymptotic part in our bounds is tight.

**Theorem 10** *For any crossover distribution,*

$$\tau(\epsilon) \ge \log_{1/r(\Pi)}(2\epsilon)^{-1}.$$

**Proof:** Let $i, j$ be any pair of positions on which $r(\Pi)$ is achieved. For any distribution $p$ on $n$-bit strings, let $p^{\{i,j\}}$ denote the projection of $p$ onto the positions $i, j$ (i.e., $p^{\{i,j\}}$ is a distribution on 2-bit strings). Then, by the triangle inequality for variation distance,

$$d_t = \|p_t - p_\infty\| \;\ge\; \|p_t^{\{i,j\}} - p_\infty^{\{i,j\}}\|$$

If $i, j$ are separated in the first $t$ rounds, the latter expression is 0. If, however, $i, j$ are not separated (this happens with probability at least $\epsilon$) the variation distance is $\frac{1}{2}$, which proves the theorem. □

Obtaining good lower bounds on the relaxation time is much harder. Essentially one has to identify a *distinguishing property* that separates $p_t$ from the uniform distribution, i.e., a large deviation whose probability is small under the uniform distribution, but not under $p_t$. This property will be different for each crossover distribution. To analyze large deviations in $p_t$, we will again use the process $(*)$: more specifically, we will view strings in $p_t$ as being constructed by taking the partition $\sigma_t$ created by the process $(*)$ at time $t$, and assigning to each block the value 0 or 1 independently with probability $\frac{1}{2}$. We believe that the analysis of large deviations in this process is of independent interest.

We now show that the upper bounds on the relaxation times of uniform and one-point crossover in Corollaries 6 and 7 are tight to within small constant factors.

**Theorem 11** *For uniform crossover, the relaxation time is at least $\left(\frac{1}{2} - \mathrm{o}(1)\right)\log_2 n$.*

**Proof:** Let $t = \frac{1}{2}\log_2 n - \log_2 c$, where $c > 1$ is a constant to be specified later. Then $\sigma_t$ surely contains a block $K$ of size $cn^{1/2}$. Consider the set $W$ of strings $w$ with $|\#1(w) - \#0(w)| \geq cn^{1/2}$, where $\#1(w)$ and $\#0(w)$ denote the number of 1s and 0s respectively in $w$. For the uniform distribution, Chernoff bounds give $\Pr[W] \leq 2\mathrm{e}^{-c^2/2}$. On the other hand, under $p_t$ we have $\Pr[W] \geq \frac{1}{2}$, since with probability $\frac{1}{2}$ the bit assigned to $K$ matches the majority in the rest of the string. Taking $c = 3$ makes the variation distance at least $\frac{1}{4}$. □

**Remark:** Using a considerably more involved analysis, we can improve the constant in the above theorem to $1 - \mathrm{o}(1)$. More specifically, we can show that $\tau_{\mathrm{unif}}(\gamma) \geq \log_2 n - C$ for constants $\gamma$ and $C$. The constant $\gamma$ here is rather less than $\frac{1}{4}$, but as we have noted the choice of $\frac{1}{4}$ in the definition of relaxation time is arbitrary. We defer the proof to the full version of the paper. □

**Theorem 12** *For one-point crossover, the relaxation time is at least* $\left(\frac{1}{2} - \mathrm{o}(1)\right)n \ln n$.

**Proof:** Let $t = cn \ln n$, where $0 < c < 1$ is a constant to be specified later. The proof proceeds in two stages: First, we show that with probability close to 1, the number of pairs of adjacent positions that are in the same block at time $t$ is at least $n^{\frac{1}{2}+\delta}$, for some small $\delta$. Then, we show how this implies a large variation distance.

**Stage 1:** Since for this system all partitions $\sigma_i$ consist of blocks of consecutive positions, the process $(*)$ can be viewed as inserting "wedges" in the "slots" between adjacent positions. We say that two adjacent positions are *separated* if there is a wedge between them, otherwise they are *unseparated* (i.e., separated adjacent positions are in different blocks; unseparated adjacent positions are in the same block).

We consider the following related process. Balls are dropped into $n$ bins, empty at first. At each time step $i \leq t$, some balls are dropped according to a probability distribution $P_i$. All distributions $P_i$ share the following property:

- For any subset of bins $K$, the probability that all $|K|$ bins receive a ball simultaneously is at most $n^{-|K|}$.

Call the distributions having this property "legal." Consider the following game: start with all bins empty; at each step an adversary is permitted to pick an *arbitrary* legal distribution $P$, then balls are dropped into the buckets according to $P$.

The relation between this game and our question is the following. The bins are analogous to the slots between adjacent positions, and the balls are analogous to the wedges. In the process $(*)$, the distribution under which new wedges are inserted is determined by the current partition, and it has the above-mentioned property. (Notice that no matter what the partition is, any block is further partitioned with probability proportional to the number of slots between its adjacent positions.) So, the choice of adversary strategy in the bins and balls game can only make things worse for our analysis.

Thus, instead of bounding the number of unseparated adjacent positions, we show the following, stronger, claim:

**Claim** *Let $0 < \delta \ll 1$. There is an absolute constant $c$ such that for $t = cn \ln n$, regardless of the strategy of the adversary, with probability $1 - \mathrm{o}(1)$ there remain at least $n^{\frac{1}{2}+\delta}$ empty bins at time $t$.*

To prove the claim, let $3 \leq k \leq \sqrt{\ln n}$ be an integer, and let $t_0$ be a time at which the number of empty bins is at least $n^{1-\frac{1}{k}}$. We show that, with probability at least $1 - n^{-\mathrm{O}\left(\frac{1}{k}\right)}$, the number of empty bins at time $t_0 + \left(\frac{1}{k-1} - \frac{1}{k} - 2\delta\right) n \ln n$ is at least $n^{1-\frac{1}{k-1}+\delta}$. Taking $k = 3$ and assuming $\delta$ is small gives $c \approx \frac{1}{6}$ (since the random variable $t_0$ is non-negative). However, a slightly more careful argument, summing over $k = 3, 4, \ldots, \lfloor\sqrt{\ln n}\rfloor$ and taking $\delta = \frac{\ln \ln n}{\ln n}$, yields $c = \frac{1}{2} - \mathrm{o}(1)$.

So now assume that $n^{1-\frac{1}{k}}$ bins are empty, and set $t = \left(\frac{1}{k-1} - \frac{1}{k} - 2\delta\right) n \ln n$. Our argument uses large deviation bounds for martingales. In order to apply these bounds, we need to limit the number of bins that can get a ball in a single step. Let $A$ denote the event that at all times between $t_0$ and $t_0 + t$ at most $k$ of the above empty bins receive a ball. Let $B$ denote the event that at time $t_0 + t$ at least $n^{1-\frac{1}{k-1}+\delta}$ bins are empty. We wish to estimate $\Pr[B]$. We do this indirectly, by estimating $\Pr[A \rightarrow B] = \Pr[\neg A \vee B]$ and $\Pr[\neg A]$. We then use $\Pr[B] \geq \Pr[A \rightarrow B] - \Pr[\neg A]$. The reason for doing this is that the estimation of $\Pr[A \rightarrow B]$ can be simplified by considering the following modified game. As long as $A$ holds (i.e., at most $k$ bins receive a ball in each step), we follow the game determined by the adversary strategy. As soon as $A$ is about to be violated (i.e., more than $k$ bins are to receive a ball), we switch to a more accommodating sequence of distributions where $A$ is never violated. We bound $\Pr[B]$ for this modified game. Clearly, the same bound holds for $\Pr[A \rightarrow B]$ in the original game.

First, we bound $\Pr[\neg A]$. By our assumptions,

$$\Pr[\neg A] \leq \frac{1}{n^{k+1}} \binom{n^{1-\frac{1}{k}}}{k+1} n \ln n \leq \frac{\ln n}{n^{\frac{1}{k}}}. \qquad (3)$$

We now bound $\Pr[A \rightarrow B]$, or rather $\Pr[B]$ in the modified game. For $i = 0, 1, \ldots, t$, define the random variable $Y_i$ to be the expectation of the number of bins that will remain empty at the end of the modified game, as computed at time $t_0 + i$. This expectation is, of course, over the strategy of adversary. Clearly, $\mathrm{E}[Y_{i+1} \mid Y_i] = Y_i$, and the sequence of random variables $Y_0, Y_1, .., Y_t$ is a martingale. Moreover, since we are assuming that $A$ holds, $|Y_i - Y_{i+1}| \leq k$. Thus, using Azuma's inequality, we have

$$\Pr[Y_t - Y_0 < -k\lambda\sqrt{t}] \leq \mathrm{e}^{-\lambda^2/2}. \qquad (4)$$

What is $Y_0$? Let $Z_1, Z_2, \ldots, Z_{n^{1-\frac{1}{k}}}$ be indicator random variables with $Z_j = 1$ iff the $j$th empty bin (at time $t_0$) remained empty at time $t_0+t$. By our assumptions, the probability that any bin receives a ball in one time step is at most $\frac{1}{n}$; therefore $p_j = \Pr[Z_j = 1] \geq \left(1-\frac{1}{n}\right)^t$. By linearity of expectation,

$$Y_0 = \sum\nolimits_j p_j \;\geq\; n^{1-\frac{1}{k}}\left(1-\tfrac{1}{n}\right)^t.$$

After substituting in the value of $t$, a routine calculation reveals that $Y_0 \geq n^{1-\frac{1}{k-1}+2\delta}$.

On the other hand, taking $\lambda = \sqrt{2\ln n}$ and recalling that $k \geq 3$, we get

$$Y_0 - k\lambda\sqrt{t} \geq n^{1-\frac{1}{k-1}+2\delta} - 2\sqrt{n}\ln n \geq n^{1-\frac{1}{k-1}+\delta},$$

for all sufficiently large $n$. Recalling inequality (4) and plugging in our bound for $Y_0$ and our choice of $\lambda$, we get

$$\Pr\left[Y_t < n^{1-\frac{1}{k-1}+\delta}\right] \leq n^{-1}.$$

Thus, in the original game, $\Pr[A \to B] \geq 1 - n^{-1}$. Putting this together with inequality (3),

$$\Pr[B] \geq \Pr[A \to B] - \Pr[\neg A] \geq 1 - \frac{1}{n} - \frac{\ln n}{n^{\frac{1}{k}}} \geq 1 - n^{-\mathrm{O}(\frac{1}{k})}.$$

This concludes the proof of the Claim, and hence also Stage 1 of the proof of the Theorem.

**Stage 2:** Given a distribution $D$ on the $n$-bit strings, let $X_i$, $i = 1, 2, \ldots, n-1$, be $\pm 1$ random variables with $X_i = 1$ iff positions $i$ and $i+1$ in the string are equal, and $-1$ otherwise. Let $X = \sum_i X_i$.

When $D$ is the uniform distribution, the $X_i$ are independent, and using Chernoff bounds we get $\Pr[X > n^{\frac{1}{2}+\delta}] \leq \mathrm{e}^{-n^{2\delta}/2}$. On the other hand, when $D$ is $p_t$, from Stage 1 the number of unseparated adjacent positions in $\sigma_t$ is at least $n^{\frac{1}{2}+\delta}$ with probability $1 - \mathrm{o}(1)$, and therefore $\Pr[X > n^{\frac{1}{2}+\delta}] \geq \frac{1}{2} - \mathrm{o}(1)$ (because if $i, i+1$ are separated then $X_i = 1$ with probability $\frac{1}{2}$, so the sum of $X_i$ over such positions is non-negative with probability $\frac{1}{2}$). Taking $\delta = \frac{\ln\ln n}{\ln n}$, as required in Stage 1, guarantees a variation distance of at least $\frac{1}{4}$ for sufficiently large $n$. □

We conclude this section with a result that shows that the upper bound on the mixing time of the Poisson model given in Theorem 9 is tight to within a factor that is close to constant. The proof is somewhat technical, and for reasons of space we give only a very brief sketch of it; the details may be found in the full version of the paper.

**Theorem 13** *The upper bound on the relaxation time of the Poisson model obtained by setting $\epsilon = \frac{1}{4}$ in Theorem 9 is tight to within a factor of* $\mathrm{O}(\log\log\log n)$.

**Proof (sketch):** Given the sequence of probabilities $p_i$ defining the Poisson model, we introduce a new parameter which we call the *rate* of the sequence. For $1 \leq i \leq n-1$, define $\tilde{p}_i$ as the rounding of $p_i$ up to the closest power of 2; i.e., $\tilde{p}_i = 2^{-m} \geq p_i \geq 2^{-(m+1)}$. Let $A_j$ denote the set of positions with $\tilde{p}_i = 2^{-j}$, and let $a_j = |A_j|$. Now, the rate $R$ of the sequence $(p_i)$ is

$$R = \max_{j: a_j > 0} 2^{j+1}\big(\log(a_j+1) + \log\log\log n + 2\big).$$

Now on the one hand it can be shown that $T \leq R$, where $T$ is the upper bound in Theorem 9 obtained by setting $\epsilon = \frac{1}{4}$. And on the other hand, it can be shown using a suitable distinguishing property that $\tau_{\mathrm{poisson}}(\frac{1}{4}) \geq R/\mathrm{O}(\log\log\log n)$. □

## 4 Finite populations

In this section, we address the issue of the relationship between the idealized quadratic crossover system and an analogous system based on a finite population. As explained in the Introduction, a naïve analysis leads to a combinatorial explosion in the population size required for a faithful correspondence, as a function of time.

The standard finite population analog of the quadratic system (2) is the following. We maintain a finite population $F_t$ of individuals of size $m$. The initial finite population, $F_0$, is just a random $m$-sample from $p_0$. Given the finite population $F_t$ at time $t$, we construct $F_{t+1}$ by repeating the following $m$ times: pick two parents $u, v$ uniformly with replacement from $F_t$, generate a child $w$ of $u, v$ exactly as in the quadratic system, and add $w$ to $F_{t+1}$. (Other finite population models are also sometimes used. For example, one can construct a random pairing of the individuals in $F_t$ and mate each pair to produce a pair of children. $F_{t+1}$ consists of these pairs. Our results carry over to such alternative schemes with obvious modifications.)

Let us denote by $f_t$ the distribution over types that is induced by picking an individual u.a.r. from the random finite population $F_t$. Clearly we cannot expect $f_t$ to be identical to $p_t$ because the finiteness introduces correlations between parents. The extent of this discrepancy should decrease with population size $m$ and increase with time $t$. For a general QDS, as explained in the Introduction, the PSPACE-completeness results of [3,18] imply that $f_t$ deviates substantially from $p_t$ unless $m$ is very large as a function of $t$. Here we will show in contrast that, for any crossover system, $f_t$ in fact closely resembles $p_t$ even when $m$ is of size only $\mathrm{O}(n^2 t)$.

Recall the inverse process based on derivations presented in Section 2.2. The key observation is that we can readily augment this process to take account of a finite population. Since the population is finite, each individual has an *index* between 1 and $m$. A *derivation* of an individual at time $t$ is again a labeled complete binary tree, except that now each active node also has an index that specifies which individual in the finite population is represented by the node. We also enforce the following consistency condition: if two nodes at some level have the same index, then all labels and indices in the subtrees rooted at these nodes must be identical. (This is because the nodes represent the same individual.) For a derivation $T$, the $n$-bit string $\mathrm{root}(T)$ is defined by propagating bits from the leaves to the root exactly as before.

The sample space $\Delta_t$ over these derivations is defined in similar fashion to $\mathcal{D}_t$ in Section 2.2, except that we must handle the indices correctly. This is done as follows:

1. Choose the index on the root u.a.r. from $\{1, \ldots, m\}$, and the label $L$ on the root as before.
2. Repeat for levels $j := t-1$ down to 1:
   (i) For each index $i$ at level $j+1$, choose two indices $i_r$ and $i_\ell$ independently and u.a.r. from $\{1, \ldots, m\}$. For each level $j+1$ active node indexed $i$, label its right parent $i_r$ and its left parent $i_\ell$.
   (ii) For each index $i$ at level $j$, choose a random label $L(i)$ as before. Label the level $j$ active nodes indexed $i$ with $L(i)$.
3. Repeat the above indexing process for the active leaves. Draw an $m$-sample from $p_0$ and label the active leaves indexed $i$ with the $i$th element of the $m$-sample.

The following counterpart to Lemma 2 states that this sample space accurately reflects what is going on in the finite population model. We omit the straightforward proof.

**Lemma 14** *If the derivation $T$ is drawn at random from $\Delta_t$, the distribution of* $\mathrm{root}(T)$ *is precisely $f_t$.* □

We now have two labeled tree models, $\mathcal{D}_t$ and $\Delta_t$, that represent $p_t$ and $f_t$ respectively. The discrepancy between these two models is captured precisely by the concept of a *collision*, which is defined as a pair of active nodes in the same level of a derivation in $\Delta_t$ that share the same index. We call a derivation that contains one or more collisions a *colliding derivation*. Fortunately, provided the population size is not too small, collisions are rare: the reason for this is that there are at most $n$ active nodes at each level. The next lemma makes this precise.

**Lemma 15** *If $T$ is drawn at random from $\Delta_t$, then*

$$\Pr[T \text{ is a colliding derivation}] \leq \frac{2n^2 t}{m}.$$

**Proof:** Fix a level $i$, and consider all trees in $\hat{\Delta}_t$ that do not have a collision at levels $i+1$ through $t$. The active nodes at level $i$ are taken from among the parents of active nodes at level $i+1$: there are at most $2n$ such candidates for each tree. Since there are no collisions up to level $i+1$, the labels for these candidates are drawn independently and uniformly at random from $\{1, 2, \ldots, m\}$. The chance of a collision at level $i$ is thus at most $\binom{2n}{2}/m \leq \frac{2n^2}{m}$. Therefore, for all $i$,

$$\Pr[\text{collision at level } i \mid \text{no collision above level } i] \leq \frac{2n^2}{m}.$$

The result follows by summing over levels $i$. □

Armed with this fact, we can now prove the main result of this section, which quantifies the discrepancy between $p_t$ and $f_t$ in terms of the population size.

**Theorem 16** *For an arbitrary crossover distribution, the quadratic system $p_t$ and the system $f_t$ based on a finite population of size $m$ satisfy $\|p_t - f_t\| \leq \frac{8n^2 t}{m}$.*

**Proof:** We introduce a slightly modified version, $\Delta'_t$ of the sample space $\Delta_t$, defined as follows: a derivation in $\Delta'_t$ is constructed in exactly the same way as for $\Delta_t$, except that if in step 2(i) some index is drawn more than once for some level $j$ we discard the tree and start again. Call the distribution on $\mathrm{root}(T)$ induced by this modified process $r_t$. We claim that

$$\|r_t - f_t\| \leq \tfrac{4n^2 t}{m} \qquad \text{and} \qquad \|p_t - r_t\| \leq \tfrac{4n^2 t}{m}.$$

The result then follows from the triangle inequality.

To see the first of the above inequalities, note that for any derivation $T \in \Delta'_t$ we have

$$\Pr_{\Delta'_t}[T] = \Pr_{\Delta_t}[T]/(1-\epsilon),$$

where $\epsilon$ is the collision probability, which is bounded by Lemma 15. The remaining $\epsilon$-fraction of derivations has probability 0 in $\Delta'_t$. This implies a variation distance of at most $2\epsilon$.

To see the second inequality, follow in parallel the process in Section 2.2 for generating $\mathcal{D}_t$ and the process in this section for generating $\Delta'_t$, as they progress level by level from the root to the leaves. As long as no collision occurs in the second process, the two processes produce identical distributions (if we forget the indices in $\Delta'_t$ on completion). If the second process does have a collision, it restarts and we make a worst-case assumption that it produces an arbitrary tree. This again yields a variation distance of at most $2\epsilon$. □

We conclude that, for any $\delta > 0$, the finite population system and the quadratic system remain within variation distance $\delta$ for at least $t$ steps provided only that the population size is at least $m = \lceil \frac{8n^2 t}{\delta} \rceil$. It is an intriguing open question whether a similarly close correspondence holds for other natural quadratic systems.

**Remark:** One consequence of Theorem 16 is that, in the case of crossover systems, the QDS can be simulated efficiently using a small finite population (of size only $\mathrm{O}(n^2 t)$, where $t$ is the number of steps to be simulated). It is worth noting that this simulation can in fact be streamlined; specifically, it is possible to sample *exactly* from the time-$t$ distribution $p_t$ of the quadratic system in time only $\mathrm{O}(nt)$. This fact again relies crucially on the inverse process introduced in Section 2. To see this, note that it is enough to generate the active nodes of a tree $T$ from the distribution $\mathcal{D}_t$. Then, $\mathrm{root}(T)$ can be determined and taken as the sample. Since there are at most $n$ active labels (and therefore $n$ active nodes) at any level, the active portion of $T$ can be generated using the process described before Lemma 2 in time $\mathrm{O}(nt)$. □

## Acknowledgements

We are grateful to David Aldous, Charlie Rackoff and Joel Spencer for enlightening discussions on the material presented here, and to Terry Speed for help with the biological literature.

## References


[1] M. Ajtai, J. Aspnes, M. Naor, Y. Rabani, L. Schulman and O. Waarts. Fairness in scheduling. *Proceedings of the 6th Annual ACM Symposium on Discrete Algorithms,* 1995, pp. 477–485.

[2] M. Ajtai, J. Komlós and E. Szemerédi. Deterministic simulation in LOGSPACE. *Proceedings of the 19th Annual ACM Symposium on Theory of Computing,* 1987, pp. 132–140.

[3] S. Arora, Y. Rabani and U. Vazirani. Simulating quadratic dynamical systems is PSPACE-complete. *Proceedings of the 26th Annual ACM Symposium on Theory of Computing,* 1994, pp. 459–467.

[4] E.B. Baum, D. Boneh and C. Garrett. *On genetic algorithms.* Preprint, NEC Research Institute, Princeton, 1994.

[5] E. Carlen and M. Carvalho. Strict entropy production bounds and stability of the rate of convergence to equilibrium for the Boltzmann equation. *Journal of Statistical Physics* **67** (1992), pp. 575–608.

[6] K.L. Clarkson, D. Eppstein, G.L. Miller, C. Sturtivant and S.H. Teng. Approximating center points with iterated Radon points. *Proceedings of the 9th Annual ACM Symposium on Computational Geometry,* 1993, pp. 91–98.

[7] P. Diaconis. *Group representations in probability and statistics.* Lecture Notes Monograph Series Vol. 11, Institute of Mathematical Statistics, Hayward, California, 1988.

[8] H. Geiringer. On the probability theory of linkage in Mendelian heredity. *Annals of Mathematical Statistics* **15** (1944), pp. 25–37.

[9] J.B.S. Haldane. The combination of linkage values and the calculation of distances between the loci of linked factors. *Journal of Genetics* **8** (1919), pp. 299–309.

[10] M. Hirsch and S. Smale. *Differential equations, dynamical systems, and linear algebra.* Academic Press, New York, 1974.

[11] J. Holland. *Adaptation in natural and artificial systems.* MIT Press, 1992.

[12] R. Kannan. Markov chains and polynomial time algorithms. *Proceedings of the 35th Annual IEEE Symposium on Foundations of Computer Science,* 1994, pp. 656–671.

[13] K. Lange. A stochastic model for genetic linkage equilibrium. *Theoretical population biology* **44** (1993), pp. 129–148.

[14] L. Lovász. *Random walks on graphs: a survey.* Report /dcs/tr–1029, Department of Computer Science, Yale University, 1994. To appear in: *Combinatorics,* a volume dedicated to P. Erdös (D. Miklos, V.T. Sos, T. Szonyi eds.), published by the Janos Bolyai Mathematical Society, 1995.

[15] Y. Lyubich. *Mathematical structures in population genetics.* Springer-Verlag Lecture Notes in Biomathematics, Volume **22**, 1992.

[16] M.S. McPeek and T.P. Speed. Modeling interference in genetic recombination. *Genetics* **139** (1995), pp. 1031–1044.

[17] O. Penrose. *Foundations of statistical mechanics.* Pergamon Press, Oxford, 1970.

[18] P. Pudlak. Complexity Theory and Genetics. *Proceedings of the 9th Annual IEEE Symposium on Structure in Complexity Theory,* 1994, pp. 383–395.

[19] Y. Rabinovich, A. Sinclair and A. Wigderson. Quadratic dynamical systems. Full version, in preparation. (Preliminary version appeared in *Proceedings of the 33rd Annual IEEE Symposium on Foundations of Computer Science*, 1992, pp. 304–313.).

[20] Y. Rabinovich and A. Wigderson. Analysis of a simple genetic algorithm. *Proceedings of the 4th International Conference on Genetic Algorithms,* 1991, pp. 215–221.

[21] L.G. Valiant. Short monotone formulae for the majority function. *Journal of Algorithms* **5** (1984), pp. 363–366.