The Mixing Rate of Different Crossover Operators

Adam Prügel-Bennett*
Image, Speech and Intelligent Systems Research Group
Department of Electronics and Computer Science
University of Southampton
Highfield, Southampton SO17 1BJ, United Kingdom

Abstract

In order to understand the mixing effect of crossover a simple shuffling problem is considered. The time taken for the strings in a population to become mixed is calculated for different crossover procedures. Uniform crossover is found to mix the population fastest, while single-point crossover causes very slow mixing. Two-point crossover extrapolates between these two limiting cases.

1 INTRODUCTION

One of the benefits of using an evolutionary algorithm is that it opens up the possibility of using a crossover operator. In early Genetic Algorithms single-point or two-point crossover were used to recombine pairs of strings [1–3]. Later multi-point crossover [4] and uniform crossover [5–7] were studied. A short controversy raged over which of these operators was best. Of course, this will depend on the problem being treated. Indeed, for many problems it is necessary to design a problem specific crossover operator [8]. Nevertheless, an important practical issue is to identify characteristics of crossover which allow for a more rational choice of crossover operator.

In this paper we study one such aspect of crossover, namely how efficiently it mixes the solutions. This is a problem independent aspect of crossover, which tells us about how quickly the population will explore the space of solutions. To study this we consider a problem which can be viewed as generalized card shuffle. We start with a pack of cards consisting of N suits with L cards in each suit. The pack is divided into N hands sorted

* email: apb@ecs.soton.ac.uk

according to their suit. The hands are then shuffled by pairing them and swapping cards using a uniform, single-point or multi-point crossover strategy. We study how the hands become shuffled over time. In the language of Genetic Algorithms the hands correspond to strings or chromosomes, the cards to genes, while the suits correspond to different alleles.

We define an order parameter to measure the degree of mixing within the population. As the strings become mixed the order parameter decays to the value of a totally mixed population. We define the mixing rate to be the asymptotic rate at which the order parameter decays. The mixing rates for single-point, two-point and uniform crossover are calculated analytically. As one might expect, uniform crossover mixes faster than multi-point crossover which mixes faster than single-point crossover, however, the difference in the rate between uniform and single-point crossover, is more dramatic than one might naively expect. This does not, of course, imply that uniform crossover is always best. Another important factor in choosing a crossover operator is its average cost due to the disruption it causes. A measure of this cost might be the average loss in fitness caused by crossover (this 'interface energy' was computed, for example, in [9, 10] for one particular problem). For problems where the spatial position of the loci have a meaning, uniform crossover may be too disruptive, whereas single-point crossover may produce only a small change in the fitness. However, this cost will depend on the problem being considered. In contrast, mixing of the allele between strings is problem independent.

On completing this paper, a different approach to the same problem was brought to my attention. This approach by Rabani, Rabinovich and Sinclair [11] considered crossover to be a special case of a quadratic dynamical system (QDS)—a generalization of a Markov chain to a process depending on combining two members of a population. In contrast to Markov chains there is no known procedure for solving a QDS. The authors, however, develop tight bounds on the convergence times for the probability distribution of a subclass of QDS to its equilibrium distribution. They use this result to obtain mixing rates for different crossover operators. Rabani *et al.*'s approach is extremely powerful, but has a different character to the approach taken here. In this paper, we consider the evolution of a single statistical property of the population, namely a measure of the degree of shuffling. By concentrating on this simple quantity the problem simplifies considerably. For example, we can quite easily obtain an exact expression for the evolution of this quantity for uniform crossover in a finite population at each generation. This is a more detailed, although less general result than that obtained by Rabani *et al.* The two approaches have different strengths. Rabani *et al.* approach gives a rigorous mathematical framework. Obtaining useful results from such a framework is notoriously hard. The fact that they succeed in finding such a result is very impressive. The tack taken in this paper, which I would not pretend to be of comparable significance to Rabani *et al.*'s general result, is to find a statistical quantity for which we can compute the dynamics. This approach, though often only approximate, allows complex systems to be modelled which are beyond the scope of a rigorous framework—modelling of more complex systems is discussed in the final section of this paper.

2 MODEL

2.1 SHUFFLING

We briefly introduce the shuffling problem together with the notation we shall be using. To denote the individual strings (hands) we use the vector

$$\mathbf{S}^{\mu}(t) = (S_1^{\mu}(t), S_2^{\mu}(t), \ldots, S_L^{\mu}(t))$$

where the superscript $\mu = 1, \ldots, N$, denotes the different strings and $S_i^{\mu}(t)$ denotes the alleles (suits) at position i and generation t. We assume that there are L cards in each hand and that the order is always maintained (i.e. at every time step each hand contains cards 1 to L). We denote the allele types (suits) by a Greek letter. Initially $S_i^{\mu}(0) = \mu$, that is, the genes (cards) of the μ^{th} member of the population are initially in allele state μ—the alleles carry information about the ancestors. To form the next generation the strings are paired at random. Each pair, $(\mathbf{S}^{\mu}(t), \mathbf{S}^{\nu}(t))$, is shuffled to form a new pair $(\mathbf{S}^{\mu}(t+1), \mathbf{S}^{\nu}(t+1))$ according to

$$\begin{aligned} S_i^{\mu}(t+1) &= \chi_i \, S_i^{\mu}(t) + (1 - \chi_i) \, S_i^{\nu}(t) \\ S_i^{\nu}(t+1) &= (1 - \chi_i) \, S_i^{\mu}(t) + \chi_i \, S_i^{\nu}(t) \end{aligned} \tag{1}$$

where $\chi_i \in \{0, 1\}$ depends on the type of crossover operator. The χ_i's are independently chosen for different pairs.

We consider the following types of crossover.

Uniform Crossover. The alleles are chosen randomly from either of the parents,

$$\chi_i = \begin{cases} 1 & \text{with probability } a \\ 0 & \text{with probability } 1 - a. \end{cases} \tag{2}$$

The parameter a defines the degree of bias. In the unbiased case $a = 1/2$.

Single-Point Crossover. A locus A $(1 \leq A < L)$ is chosen on the string and all alleles up to and including A are taken from one string while all the other alleles are taken from the second string,

$$\chi_i = \begin{cases} 1 & \text{if } i \leq A \\ 0 & \text{if } i > A. \end{cases} \tag{3}$$

Two-Point Crossover. The strings are treated as loops, along which we choose two crossing points, A and B. The strings are swapped between these two points,

$$\chi_i = \begin{cases} 1 & \text{if } i \leq A \text{ or } i > B \\ 0 & \text{if } A < i \leq B. \end{cases} \tag{4}$$

Note that we can either choose A and B independently or we can choose A at random and choose B to be a fixed distance from A. In particular we can choose $0 < A \leq N/2$ and $B = A + N/2$ to ensure that a maximum number of alleles are swapped at each 'generation'.

2.2 MIXING RATE

We need to define a measure of shuffling. To do so we first define the 'overlaps', between the strings at time t and the initial strings, to be the number of alleles of type α in $\mathbf{S}^{\mu}(t)$ — that is, the number of alleles that originated in string $\mathbf{S}^{\alpha}(0)$

$$m_{\alpha}^{\mu}(t) = \frac{1}{L}\sum_{i=1}^{L}\left[S_i^{\mu}(t) = \alpha\right] \tag{5}$$

where we have used the notational convention for the indicator function

$$\left[\text{PREDICATE}\right] = \begin{cases} 1 & \text{PREDICATE is true} \\ 0 & \text{PREDICATE is false.} \end{cases} \tag{6}$$

Since every allele ancestor remains in the population and every allele must come from somewhere

$$\sum_{\mu=1}^{N} m_{\alpha}^{\mu}(t) = \sum_{\alpha=1}^{N} m_{\alpha}^{\mu}(t) = 1. \tag{7}$$

We are now ready to define an order parameter, $Q(t)$, measuring the degree of mixing

$$Q(t) = \frac{1}{N}\sum_{\alpha=1}^{N}\sum_{\mu=1}^{N}\left(m_{\alpha}^{\mu}(t)\right)^2. \tag{8}$$

Although this measure is *ad hoc*, it is one of the simplest measures—it depends only on a count of the cards in each hand; a linear measure will not work because of equation (7). Initially $m_{\alpha}^{\mu}(0) = \left[\alpha = \mu\right]$ (the expression in squared brackets is just the Kronecker delta) so $Q(0) = 1$. When the population is totally mixed the order parameter falls to

$$Q(\infty) = \frac{N + L - 1}{N\,L}. \tag{9}$$

(the derivation of this is given in an appendix). We define the mixing rate as the (asymptotic) reduction towards the mixed state produced by a shuffle

$$r = \lim_{t\to\infty}\left(\frac{Q(t+1) - Q(\infty)}{Q(t) - Q(\infty)}\right).$$

3 RESULTS

3.1 UNIFORM CROSSOVER

We can compute the change in the order parameter and consequently the mixing rate exactly for uniform crossover. From equation (1) it follows that

$$\begin{aligned} \left[S_i^{\mu}(t+1) = \alpha\right] &= \chi_i\left[S_i^{\mu}(t) = \alpha\right] + (1-\chi_i)\left[S_i^{\nu}(t) = \alpha\right] \\ \left[S_i^{\nu}(t+1) = \alpha\right] &= (1-\chi_i)\left[S_i^{\mu}(t) = \alpha\right] + \chi_i\left[S_i^{\nu}(t) = \alpha\right] \end{aligned} \tag{10}$$

Since the χ_i's are chosen independently we can average over them. From equation (2) we see that $\langle \chi_i \rangle = a$, where $\langle \cdots \rangle$ denotes the expectation value. To calculate the change in the order parameter we consider

$$\big(m_\alpha^\mu(t+1)\big)^2 = \frac{1}{L^2}\sum_i \big[S_i^\mu(t+1)=\alpha\big] + \frac{1}{L^2}\sum_{i\neq j}\big[S_i^\mu(t+1)=\alpha\big]\,\big[S_j^\mu(t+1)=\alpha\big]$$

where we have separated out the $i=j$ term and used the property of indicator functions $\big[\cdots\big]^2 = \big[\cdots\big]$. We can now substituting in equation (10) to obtain an expression in terms of variables at time t. Averaging out the χ_i's we obtain after a little algebra

$$\Big\langle \big(m_\alpha^\mu(t+1)\big)^2\Big\rangle = \frac{2a(1-a)}{L}\big(m_\alpha^\mu(t)+m_\alpha^\nu(t)\big) + \big(a\, m_\alpha^\mu(t) + (1-a)m_\alpha^\nu(t)\big)^2 .$$

This gives the square of the overlaps at time $t+1$ in terms of the overlaps at time t. Putting this into equation (8) we obtain an equation for $Q(t+1)$ in terms of the overlaps at time t. But, using equations (7) and (8) we can simplify this to

$$Q(t+1) = r\,Q(t) + (1-r)Q(\infty) \tag{11}$$

where

$$r = \frac{1-2a(1-a)}{1-1/N}$$

and $Q(\infty)$ is given in equation (9). Equation (11) is a linear recursion relation which has the well known solution

$$Q(t) = Q(\infty) + r^t\,\big(Q(0)-Q(\infty)\big)\,. \tag{12}$$

Starting from an ordered state we have $Q(0)=1$. The characteristic mixing time is given by

$$\tau = \frac{-1}{\log(r)} \approx \frac{-1}{\log\big(1-2a(1-a)\big)}. \tag{13}$$

The optimal shuffling is achieved when $a=1/2$ in which case $\tau \approx 1/\log(2) \approx 1.44$. Every τ generations the distance $Q(t)-Q(\infty)$ is decreased by $1/e$ — for $a=2$ this distance is halved each generation.

3.2 SINGLE-POINT CROSSOVER

Single-point crossover is more complicated than uniform crossover because the loci are no longer independent of each other—nearby loci are more likely to come from the same ancestor than distant loci. To calculate the effect of single-point crossover we adopt a new approach. We consider the strings to be made up from regions or blocks of genes where the regions are bounded by the points where crossover has occurred. We denote the number of regions in a string $\mathbf{S}^\mu(t)$ by $R^\mu(t)$. We consider, below, how the number of regions grow over time.

We can express the overlaps in terms of block variables. We denote the size of the n^{th} block of $\mathbf{S}^\mu(t)$ by $x_n^\mu(t)$ and the allele type of the n^{th} block by $\beta_n^\mu(t)$. Then the overlap can be written as

$$m_\alpha^\mu(t) = \frac{1}{L}\sum_{n=1}^{R^\mu(t)} x_n^\mu(t)\,\big[\beta_n^\mu(t)=\alpha\big]\,. \tag{14}$$

From equation (8) we find the order parameter is given by

$$Q(t) = \frac{1}{N\,L^2} \sum_{\mu=1}^{N} \sum_{n=1}^{R^\mu(t)} \left(\left(x_n^\mu(t)\right)^2 + \sum_{\substack{m=1 \\ m\neq n}}^{R^\mu(t)} x_n^\mu(t)\, x_m^\mu(t) \left[\beta_n^\mu(t) = \beta_m^\mu(t)\right] \right) \tag{15}$$

where we have used the identities

$$\sum_{\alpha=1}^{N} \left[\beta_n^\mu(t) = \alpha\right] = 1$$

and

$$\sum_{\alpha=1}^{N} \left[\beta_n^\mu(t) = \alpha\right] \left[\beta_m^\mu(t) = \alpha\right] = \left[\beta_n^\mu(t) = \beta_m^\mu(t)\right].$$

The first term in equation (15) gives the contribution to the order parameter coming from the single blocks while the second term gives the contribution coming from different blocks which, by chance, come from the same ancestors. We do not know the block sizes, x_n^μ. However, as the crossing points are randomly chosen the block sizes will be random except for the constraint

$$\sum_{n=1}^{R^\mu(t)} x_n^\mu(t) = L.$$

Given $R^\mu(t)$ blocks then on average

$$\begin{aligned} \left\langle \left(x_n^\mu(t)\right)^2 \right\rangle &= \frac{L(2L - R^\mu(t) + 1)}{R^\mu(t)(R^\mu(t)+1)} \\ \left\langle x_n^\mu(t)\, x_m^\mu(t) \right\rangle &= \frac{L(L+1)}{R^\mu(t)(R^\mu(t)+1)}. \end{aligned} \tag{16}$$

The calculation of these average values are given in a separate appendix. We are left having to calculate the average number of blocks that come from the same ancestor

$$\sum_{n=1}^{R^\mu(t)} \sum_{\substack{m=1 \\ m\neq n}}^{R^\mu(t)} \left[\beta_n^\mu(t) = \beta_m^\mu(t)\right].$$

If each block could come from any ancestor then the probability of two blocks being the same is $1/N$. However, at the first generation the two blocks are guaranteed to come from separate ancestors. To incorporate this fact we assume that the probability of two blocks coming from the same ancestor is $(1 - 1/t)/N$. Using this we find

$$\sum_{n=1}^{R^\mu(t)} \sum_{\substack{m=1 \\ m\neq n}}^{R^\mu(t)} \left[\beta_n^\mu(t) = \beta_m^\mu(t)\right] = \frac{\left(R^\mu(t) - 1\right) R^\mu(t) \left(1 - 1/t\right)}{N}. \tag{17}$$

Putting equations (16) and (17) into equation (15) we obtain

$$\left\langle Q(t)\right\rangle = \frac{N(2L-R(t)+1)+(L+1)(R(t)-1)(1-1/t)}{N\,L\,(R(t)+1)} \tag{18}$$

where $R(t)$ is the average number of regions per string.

We now consider how the number of regions grow over time. Crossover will produce a new region provided that the cutting point occurs within a block rather than at one of the ends. Since there are $L-1$ possible cutting points and $R^{\mu}(t)-1$ region boundaries, there will be a probability of $(L-R^{\mu}(t))/(L-1)$ that the cutting site occurring within a block. Thus, if at time t there are $R(t)$ regions then the expected number of regions at the next shuffle will be

$$\left\langle R(t+1)\right\rangle = R(t) + \frac{L-R(t)}{L-1}. \tag{19}$$

This is again a linear recursion, with solution

$$R(t) = L-(L-1)\left(1-\frac{1}{L-1}\right)^{t}. \tag{20}$$

The derivation of $\left\langle Q(t)\right\rangle$ is not exact. In particular, we have ignored fluctuations in the number of blocks (equation (18) is not linear in R(t), so fluctuations in R(t) will give rise to systematic corrections). Nevertheless, the corrections are so small that equations (18) and (20) are in almost perfect agreement with simulation results.

The asymptotic behaviour of $Q(t)$ is dominated by the rate of production of new regions. From equation (20) we see that the number of regions increases towards its maximum L at a rate $1-1/(L-1)$. As a consequence the characteristic mixing time is $-1/\log(1-1/(L-1)) \approx L-1$.

3.3 TWO-POINT CROSSOVER

We consider the case when we choose a crossing point A uniformly between 1 and $L/2$ and set the second crossing point B to be at $A+L/2$. This ensures that exactly half the genes are swapped at each generation. As a consequence at the first generation $Q(1) = 1/2$. Thereafter each half of the string can be treated as a single string (of length $L/2$) which behaves as though they were experiencing single-point crossover. Thus for $t \geq 2$ starting from $R(t)$ *regions in each half of the string* the expected number after shuffling grows as

$$\left\langle R(t+1)\right\rangle = R(t) + \frac{L-2R(t)}{L}. \tag{21}$$

Using the boundary condition $R(1) = 1$ we find

$$\left\langle R(t)\right\rangle = \frac{L}{2} - \left(\frac{L}{2}-1\right)\left(1-\frac{2}{L}\right)^{t-1}. \tag{22}$$

Following a similar calculation to that for equation (18) we find

$$\left\langle Q(t)\right\rangle = \frac{(N+1)(L-R(t)+1)+(L+2)(R(t)-1)(1-1/(2(t-1)))}{N\,L\,(R(t)+1)} \tag{23}$$

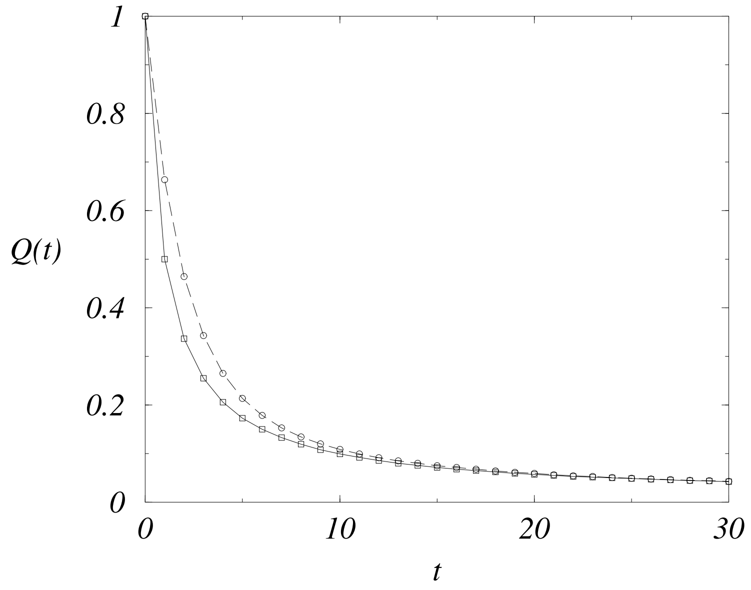

Figure 1 The evolution of the order parameter is shown for two different types of two-point crossover. The solid line shows the case where the separation between the crossing points is set to $L/2$, while the dashed line shows the case when the crossing points are chosen independently. A population size of 100 and string length of 100 was used. The simulations where averaged over 1 000 runs. The errors in the mean are less than the width of the line.

Again the asymptotic decay of $Q(t)$ is dominated by the speed at which new regions are produced. From equation (22) this gives a characteristic shuffling time of $-1/\log(1 - 2/L)) \approx L/2$.

An alternative method for performing two-point crossover would be to choose the two crossing points independently. This complicates the calculations and we have performed simulations only. Figure 1 shows simulation results. Fixing the distance between crossing points to $L/2$ speeds up the initial shuffling. However, the asymptotic mixing rate is the same for both types of crossover.

4 DISCUSSION

We have obtained analytic expressions for the mixing rate for uniform, single-point and two-point crossover. In figure 2 the analytic expressions are plotted for a population size and string length of 100. The curve for uniform crossover is exact. Those for single-point and two-point crossover are approximate, but for these parameters the theory and simulations differ by less than 5×10^{-4}.

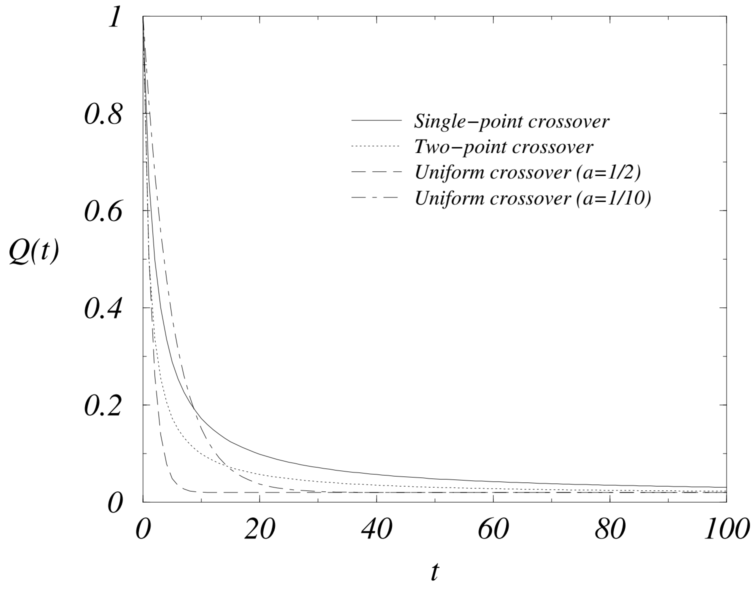

Figure 2 The evolution of the order parameter is shown for uniform, single-point and two-point crossover. In all three cases the population size and string length was set to 100.

Although at the first step, the crossover strategies appear comparable for single-point, two-point and uniform crossover with $a = 1/2$, the shuffling rate continues to reduce rapidly for uniform crossover, but slows down very considerably for single-point and two-point crossover. Two-point crossover shuffles at twice the rate of single-point crossover. In fact, if we rescaled the time so that one generation of two-point crossover equals two generations of single-point crossover the two curves for $Q(t)$ lie almost on top of each other. Multi-point crossover, in which n-points are chosen at random and where every other region is taken from the second parent, will extrapolate between the two-point and uniform crossover curves.

For uniform crossover the mixing rate is $1 - 2a(1-a)$ (ignoring finite population corrections). This is maximized when the bias, a, is set to $1/2$. For $a \ll 1/2$ the mixing rate will be much reduced and the initially reduction in the order parameter will be much slower than that for single-point or two-point crossover. However, provided $a \gg 1/L$ the asymptotic mixing rate will be much faster for the biased uniform crossover than single or two-point crossover. Thus after a long enough period biased uniform crossover will achieve a better mixing than its rivals. This is illustrated in figure 2 where we have also plotted $Q(t)$ for uniform crossover with $a = 1/10$.

One referee of this paper asked the pertinent question, so what? The answer to this is that modelling has both short and long term objectives. The short term objective is to find an accurate description of the system being studied. The much bolder long term objective is to obtain an understanding of the important features that determine the efficiency of the search performed by a Genetic Algorithm, with the hope of creating a principled approach to designing GAs. This is an accumulative process which can be achieved by understanding the detailed behaviour of simple systems. In this paper we identified a measure of the degree of mixing which allowed us to compute the mixing rates. The result that uniform crossover is faster than two-point crossover which in turn is faster than single-point crossover is, of course, not surprising. A result which was more unexpected (at least, for the author) was that uniform crossover even with a very strong bias towards one parent still has a much faster asymptotic mixing rate than single-point crossover. The mixing rates are important as they provide a measure for how fast crossover causes the population to decorrelate from its initial state. The shorter the decorrelation time the faster the exploration of the search space. Asymptotically single-point crossover is no faster than a mutation of one allele per generation (they both require touching each loci, which takes of order $L\log(L)$ trials), while uniform crossover is dramatically faster, taking $\mathcal{O}(\log(L))$ steps. These results agree with Rabani *et al.*, although we have also calculated how the degree of mixing changes for intermediate times.

To understand what crossover does in a GA, however, necessitates the modelling of the interaction of crossover with the other operators such as selection and mutation. This is part of the long term objective of modelling, of which this paper is only a small contribution. A statistical mechanics analysis of GAs has been carried out for various simple problems (e.g. [9, 10, 12–21]), which use statistical properties of the population to model the system dynamics. However, these studied did not treat the problem of spatial correlations that arise when using single or multi-point crossover. Recently the author has completed a paper treating the dynamics of a GA with two-point crossover [22]. The additional spatial correlations considerably complicate the analysis—not because of the difficulty of modelling crossover (which can be done exactly), but because of the difficulty in estimating the effect of selection on the spatial correlations produced by crossover. This paper uncovers a small part of the story of crossover. It shows how the amount of mixing caused by crossover evolves over generations. How important this is to the larger picture only time will tell.

APPENDIX: TOTALLY MIXED STATE

The overlap parameters $M_\alpha^\mu(t) = L\, m_\alpha^\mu(t)$ define a partitioning of the L alleles in string μ amongst the N initial strings. Total mixing corresponds to a random partitioning so that the probability of the overlaps being $(M_1^\mu, M_2^\mu, \ldots, M_N^\mu)$ is given by the multinomial distribution

$$\frac{1}{N^L}\frac{L!}{M_1^\mu!\; M_2^\mu! \cdots M_N^\mu!}\left[\sum_{\alpha=1}^{N} M_\alpha^\mu = L\right].$$

To find averages over a multinomial probability distribution we note that the multinomial coefficients appear in the following expansion

$$\left(\sum_{\alpha=1}^{N} x_\alpha\right)^L = \sum_{\{M_\alpha\}} L! \left(\prod_{\alpha=1}^{N} \frac{x_\alpha^{M_\alpha}}{M_\alpha!}\right)\left[\sum_{\alpha=1}^{N} M_\alpha = L\right] \tag{24}$$

where the sum is over all integer values of the M_α's. We can use equation (24) to compute averages over the multinomial distribution using the observation

$$\begin{aligned}\left\langle M_\alpha \right\rangle &= \frac{1}{N^L}\frac{\partial}{\partial x_\alpha}\sum_{\{M_\alpha\}} L!\left(\prod_{\alpha=1}^{N}\frac{x_\alpha^{M_\alpha}}{M_\alpha!}\right)\left[\sum_{\alpha=1}^{M} M_\alpha = L\right]\Bigg|_{x_\alpha=1} \\ &= \frac{\partial}{\partial x_\alpha}\frac{\left(\sum_{\alpha=1}^{N} x_\alpha\right)^L}{N^L} = \frac{L\left(\sum_{\alpha=1}^{N} 1\right)^{L-1}}{N^L} = \frac{L}{N}\end{aligned}$$

similarly

$$\begin{aligned}\left\langle M_\alpha(M_\alpha - 1)\right\rangle &= \frac{\partial^2}{\partial x_\alpha^2}\frac{\left(\sum_{\alpha=1}^{N} x_\alpha\right)^L}{N^L}\Bigg|_{x_\alpha=1} \\ &= \frac{L(L-1)\left(\sum_{\alpha=1}^{N} 1\right)^{L-2}}{N^L} = \frac{L(L-1)}{N^2}.\end{aligned}$$

From the definition (8) and assuming totally shuffling

$$Q(\infty) = \frac{N}{L^2}\left\langle (M_\alpha^\mu)^2\right\rangle = \frac{N+L-1}{N\,L}.$$

APPENDIX: STATISTICS OF BLOCK SIZES

We need to calculate $\left\langle (x_n^\mu(t))^2\right\rangle$ and $\left\langle x_n^\mu(t) x_m^\mu(t)\right\rangle$. We first abstract this problem. We consider a string with L sites which is split into R regions where the position of each region boundary is independently chosen. We then ask what is the probability distribution for the sizes of the regions? For this calculation we can drop the superscript μ and the time index as these are irrelevant. Thus the length of a region we denote by x_n. Our single constraint is

$$\sum_{n=1}^{R} x_n = L.$$

We can write the probability of a partitioning $\mathbf{x} = (x_1, x_2, \ldots, x_R)$ as

$$p(\mathbf{x}) = \frac{1}{Z}\mathrm{e}^{-\sum_n x_n}\left[\sum_{n=1}^{R} x_n = L\right], \qquad Z = \operatorname*{Tr}_{\{\mathbf{x}\}} \mathrm{e}^{-\sum_n x_n}\left[\sum_{n=1}^{R} x_n = L\right]$$

where we have used the shorthand

$$\operatorname*{Tr}_{\{\mathbf{x}\}} \equiv \sum_{x_1=1}^{\infty}\sum_{x_2=1}^{\infty}\cdots\sum_{x_R=1}^{\infty}$$

($p(\mathbf{x})$ is not a multinomial distribution as there are no combinatorial factors preferring equi-partitioning.) In the definition of $p(\mathbf{x})$ we have included the term

$$\mathrm{e}^{-\sum_n x_n}$$

which is a constant in light of the Kronecker delta, however, it will act as a regularization term in what follows.

Our aim is to find

$$\left\langle x_n^2 \right\rangle = \operatorname*{Tr}_{\{\mathbf{x}\}} x_n^2 \, p(\mathbf{x}).$$

We can replace the Kronecker delta with its integral representation

$$\left[\sum_{n=1}^{R} x_n = L \right] = \int_{-\pi}^{\pi} \mathrm{e}^{\,\mathrm{i}\lambda \left(\sum_{n=1}^{R} x_n - L \right)} \frac{\mathrm{d}\lambda}{2\pi}.$$

Giving

$$\left\langle x_n^2 \right\rangle = \frac{1}{Z} \int_{-\pi}^{\pi} \mathrm{e}^{\,\mathrm{i}\lambda L} \operatorname*{Tr}_{\{\mathbf{x}\}} x_n^2 \mathrm{e}^{\,-(\mathrm{i}\lambda+1) \sum_{n=1}^{R} x_n} \frac{\mathrm{d}\lambda}{2\pi}.$$

We can perform the sums (notice the regularization term ensures that sums converge) to give

$$\begin{aligned}
\left\langle x_n^2 \right\rangle &= \frac{1}{Z} \int_{-\pi}^{\pi} \mathrm{e}^{\,\mathrm{i}\lambda L} \left(\frac{2\mathrm{e}^{\,2\mathrm{i}\lambda+2}}{(\mathrm{e}^{\,\mathrm{i}\lambda+1}-1)^{R+2}} - \frac{\mathrm{e}^{\,\mathrm{i}\lambda+1}}{(\mathrm{e}^{\,\mathrm{i}\lambda+1}-1)^{R+1}} \right) \frac{\mathrm{d}\lambda}{2\pi} \\
&= \frac{1}{Z} \int_{-\pi}^{\pi} \mathrm{e}^{\,\mathrm{i}\lambda L} \left(\frac{2}{R(R+1)} (-D_\lambda^2 + \mathrm{i}\mathrm{D}_\lambda) - \frac{\mathrm{i}}{\mathrm{R}} \mathrm{D}_\lambda \right) \frac{1}{(\mathrm{e}^{\,\mathrm{i}\lambda+1}-1)^{R}} \frac{\mathrm{d}\lambda}{2\pi}
\end{aligned}$$

where $D_\lambda = \partial/\partial\lambda$. We can integrate by parts so that the remaining integral cancels with the normalization factor Z. We thus obtain

$$\left\langle x_n^2 \right\rangle = \frac{L(2L-R+1)}{R(R+1)}.$$

We can calculate $\left\langle x_n \, x_m \right\rangle$ in a similar way. However it is easier to observe that

$$L^2 = \left(\sum_n x_n \right)^2 = \sum_n x_n^2 + \sum_{n \neq m} x_n x_m$$

thus

$$\left\langle x_n \, x_m \right\rangle = \frac{L - \left\langle x_n^2 \right\rangle}{L-1}.$$

References

[1] J. H. Holland. *Adaptation in Natural and Artificial Systems.* University of Michigan Press (Ann Arbor), 1975.

[2] K. A. De Jong. *An Analysis of the Behaviour of a Class of Genetic Adaptive Systems.* PhD thesis, University of Michigan, 1975.

[3] D. E. Goldberg. *Genetic Algorithms in Search, Optimization & Machine Learning.* Addison-Wesley (Reading, Mass), 1989.

[4] W. M. Spears and K. A. De Jong. An analysis of multi-point crossover. In Gregory J. E. Rawlins, editor, *Foundations of Genetic Algorithms*, pages 301–315. Morgan Kaufmann (San Mateo), 1991.

[5] D. H. Achley. *A connectionist machine for genetic hillclimbing.* Kluwer Academic Publishing, 1987.

[6] G. Syswerda. Uniform crossover in genetic algorithms. In *Proceedings of the Third International Conference on Genetic Algorithms*, pages 2–9. Morgan Kaufmann (San Mateo), 1989.

[7] L. J. Eshelman, R. A. Caruana, and J. D. Schaffer. Biases in the crossover landscape. In *Proceedings of the Third International Conference on Genetic Algorithms*, pages 10–19. Morgan Kaufmann (San Mateo), 1989.

[8] P. Galinier and J. K. Hao. Hybrid evolutionary algorithms for graph coloring. *Journal of Combinatorial Optimization*, 3(4):379–397, 1999.

[9] A. Prügel-Bennett and J. L. Shapiro. An analysis of genetic algorithms using statistical mechanics. *Physical Review Letters*, 72(9):1305–1309, 1994.

[10] A. Prügel-Bennett and J. L. Shapiro. The dynamics of a genetic algorithm for simple random Ising systems. *Physica D*, 104:75–114, 1997.

[11] Rabani Y., Rabinovich Y., and Sinclair A. A computational view of population genetics. *Random Structures & Algorithms*, 12(4):313–334, 1998.

[12] M. Rattray. The dynamics of a genetic algorithm under stabilizing selection. *Complex Systems*, 9(3):213–234, 1995.

[13] M. Rattray and J. L. Shapiro. The dynamics of genetic algorithms for a simple learning problem. *Journal of Physics: A.*, 29:7451–7473, 1996.

[14] A. Prügel-Bennett. Modelling evolving populations. *Journal of Theoretical Biology*, 185:81–95, 1997.

[15] J. L. Shapiro and A. Prügel-Bennett. Genetic algorithms dynamics in two-well potentials with basins and barriers. In R. K. Belew and M. D. Vose, editors, *Foundations of Genetic Algorithms 4*, pages 101–116, San Francisco, 1997. Morgan Kaufmann.

[16] M. Rattray and J. L. Shapiro. Noisy fitness evaluations in genetic algorithms and the dynamics of learning. In R. K. Belew and M. D. Vose, editors, *Foundations of Genetic Algorithms 4*, pages 117–139, San Francisco, 1997. Morgan Kaufmann.

[17] S. Bornholdt. Probing genetic algorithm performance of fitness landscapes. In R. K. Belew and M. D. Vose, editors, *Foundation of Genetic Algorithms 4*, pages 141–154, San Francisco, 1997. Morgan Kaufmann.

[18] E. van Nimwegen, J. P. Crutchfield, and M. Mitchell. Finite populations induce metastability in evolutionary search. *Physics Letters A*, 229:144–150, 1997.

[19] A. Rogers and A. Prügel-Bennett. Genetic drift in genetic algorithm selection schemes. *IEEE Transactions on Evolutionary Computation*, 3(4):298–303, 1999.

[20] A. Prügel-Bennett. On the long string limit. In W. Banzhaf and C. Reeves, editors, *Foundations of Genetic Algorithms 5*, pages 45–56, San Francisco, 1999. Morgan Kaufmann.

[21] A. Rogers and A. Prügel-Bennett. The dynamics of a genetic algorithm on a model hard optimization problem. *Complex Systems*, 11(6):437–464, 2000.

[22] A. Prügel-Bennett. Modelling crossover induced linkage in genetic algorithms. Preprint, 2000.