Modelling GA Dynamics

Adam Prügel-Bennett and Alex Rogers

October 27, 1999

Abstract

This paper is an introduction to the mathematical modelling of the dynamics of genetic algorithms. The distinguishing feature of this approach is that we consider *macroscopic* properties of the system. After some brief introductory remarks, we look at a generational GA, with tournament selection, tackling the ones-counting problem. Initially we ignore recombination. We start with a two parameter model of the evolution. This is sufficient to explain the qualitative features of the dynamics, although it does not give a good quantitative agreement with simulations. We show how the agreement can be improved by using more parameters to describe the population and by introducing finite population corrections. Finally, we come back to recombination and show how this can be modelled.

1 Why Model?

The aim of mathematical modelling is to gain insights into how things work. For genetic algorithms we would like to know what problems are amenable to GAs, what representation and operators to use, and what are the optimal parameters settings. Modelling will not give precise answers to these questions as they depend on the problem being tackled. Nevertheless, modelling gives us an understanding which we can apply to new problems. To obtain a deeper understanding of how GAs work we must not only model the algorithm but also model different types of problems. This, however, goes beyond the scope of this paper.

This paper is intended as a tutorial in how to accurately model the dynamics of a simple genetic algorithm. The model is, by necessity, only an approximation of reality. By making an accurate model we can convince ourselves that all the important features have been captured. But accuracy often comes at the price of obfuscating the underlying process. A good model should retain the essential features of the process being modelled but with the minimum of unnecessary detail. In what follows we start from a minimalist model, adding details as we delve deeper into what is happening.

So what can we reasonably hope to gain from our efforts? Even modelling simple problems we might expect to compare different strategies such as steady-state versus generational GAs or roulette wheel versus stochastic universal sampling. We might be able to understand how the dynamics is affected by changing a parameter. Although this won't necessarily tell us the optimal parameters for a different problem, it might tell us what we should be look for when tuning our parameters. If we can understand how a GA actually searches a complex search space we might even get insights into what are useful representations and which problems can be successfully tackled using GAs.

2 What to Model

Mathematical modelling can be hard work. It makes sense therefore to start simple. Perhaps the simplest problem of all is ones-counting consisting of a string of ones and zeros, where the fitness is given by the number of ones. In this introduction we will use ones-counting as our example. After all, if we cannot understand ones-counting we have little hope of understanding other models. And we don't understand ones-counting. For example, in a GA with just mutation and selection, how does the equilibrium fitness depend on the string length, the selection rate and the population size? Nobody knows.

Modelling is not, however, confined to ones-counting. The formalism described here has been applied to a modification of ones-counting where each site contributes differently to the fitness [1]; a model where the fitness depends on the states of the neighbouring loci and has exponentially many local optima [2, 1]; the subset-sum problem [3]; learning the weights of a perceptron [4]; and a basin with a barrier [5, 6]. This last problem will be presented elsewhere in this workshop [7].

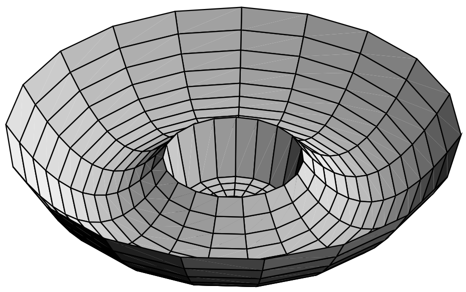

Figure 1: Basin with a barrier in two dimensions.

3 How to Model

3.1 Naming the Parts

We consider the ones-counting problem where the objective is to maximizing the number of ones in a binary string of length L. We denote the string by $\vec{S} = (S_1, S_2, \cdots, S_L)$. The 'fitness' of a string is given by the number of 1's

$$E = \sum_{i=1}^{L} S_i.$$

To solve this we use a simple generational genetic algorithm. The size of the population is fixed. We label the members of the population by $\alpha = 1$, 2, ..., P. The string of member α we denote by $\vec{S}^\alpha$ and the fitness by E_α. The genetic algorithm evolves by selection, mutation and recombination. We use binary tournament selection where two members of the population are drawn at random and the fitter of the two is copied into a mating pool. This is carried out P times. Each member of the mating pool undergoes mutation. In mutation we decide at each site of the string to change its state with a probability u

$$S_i^\alpha \to S_i^{\alpha\prime} = \begin{cases} S_i^\alpha & \text{with probability } 1-u \\ 1 - S_i^\alpha & \text{with probability } u. \end{cases}$$

Recombination is performed by pairing the members of the mating pool and generating two children from each pair using uniform crossover. That is, from each pair $\{\vec{S}^\alpha, \vec{S}^\beta\}$ we produce a pair of strings $\{\vec{S}^\mu, \vec{S}^\nu\}$ where

$$S_i^\mu = \begin{cases} S_i^\alpha & \text{with probability } a \\ S_i^\beta & \text{with probability } 1-a. \end{cases}$$

and $\vec{S}^\nu$ is the complementary child. We usually take $a = 1/2$. If there is no recombination the new population is set equal to the mating pool after selection and mutation. We now have a well defined system to be modelled. It depends on the parameters L, P, u and whether or not we perform crossover. But how are we going to describe the dynamics? We give an outline of our approach next.

3.2 Modelling Philosophy

Figure 2 shows the distribution of fitnesses for a population evolving under selection and mutation. The overall trend is clear: the population is initially widely distributed around $L/2$ — the mean fitness of a random string. As the population evolves the average fitness increases while the width of the population narrows. Clearly many of the details, such as the small scale fluctuations, will vary from run to run. The property we are usually most interested in is the maximum fitness. However, the evolution of the population will depend more on the pack than the leader. Thus for modelling we will focus on coarse statistical properties of the population such as the average fitness and width of the population. If we wish to know the maximum fitness in the population we can estimate this from the statistical properties.

The distinguishing feature of this approach is the use of macroscopic properties of the population to model the dynamics. Because the macroscopic properties don't fully describe the system we may need to infer other properties. We can do this on the basis of what are the most likely values consist with the known macroscopic values. One way of making such inferences is using maximum entropy arguments. For the simple model discussed here we can get away without resorting to maximum entropy. The use of maximum entropy is discussed in detail by Jonathan Shapiro's article in this volume. For this brief overview we consider as macroscopic variables the mean fitness, variance in the fitness and later the skewness as correlation. In doing so we are making implicit assumptions about other properties of populations. For example, we are assuming that all the higher order statistics are zero. The results will therefore be approximate, although we can improve the approximation by including more macroscopic variables.

The most important statistical properties of the population are the mean fitness, μ, and the variance, σ^2,

$$\mu = \frac{1}{P}\sum_{\alpha=1}^{P} E_\alpha \tag{1}$$

$$\sigma^2 = \frac{1}{P}\sum_{\alpha=1}^{P} (E_\alpha - \mu)^2 . \tag{2}$$

The width of the distribution is well characterized by the standard deviation, σ (i.e. the square root of the variance). Figure 3 shows the average fitness and variance in the fitness for five different runs of a GA. The graphs tell us the same story as the histograms of figure 2, although the details are now somewhat clearer. The evolution splits into roughly three regimes. There is an initial transient behaviour which in this case lasts about 10 generations. In this period the variance is rapidly reduced by the relatively strong selection pressure produced by tournament selection. After this transient period the loss in variance due to selection is balanced by the gain in variance due to mutation. Selection however increases the average fitness until the population reaches an equilibrium at around 150 generations where selection and mutation balance exactly. The population then remains at this equilibrium indefinitely.

One obvious feature of figure 3 is that there are substantial fluctuations from one run to another. Which run should we model? We want to know about typically runs. But a typical run has random fluctuations even though these fluctuations differ from run to run. Do we ignore the fluctuations or model a specific run? One way out of this dilemma is to model an ensemble of populations.

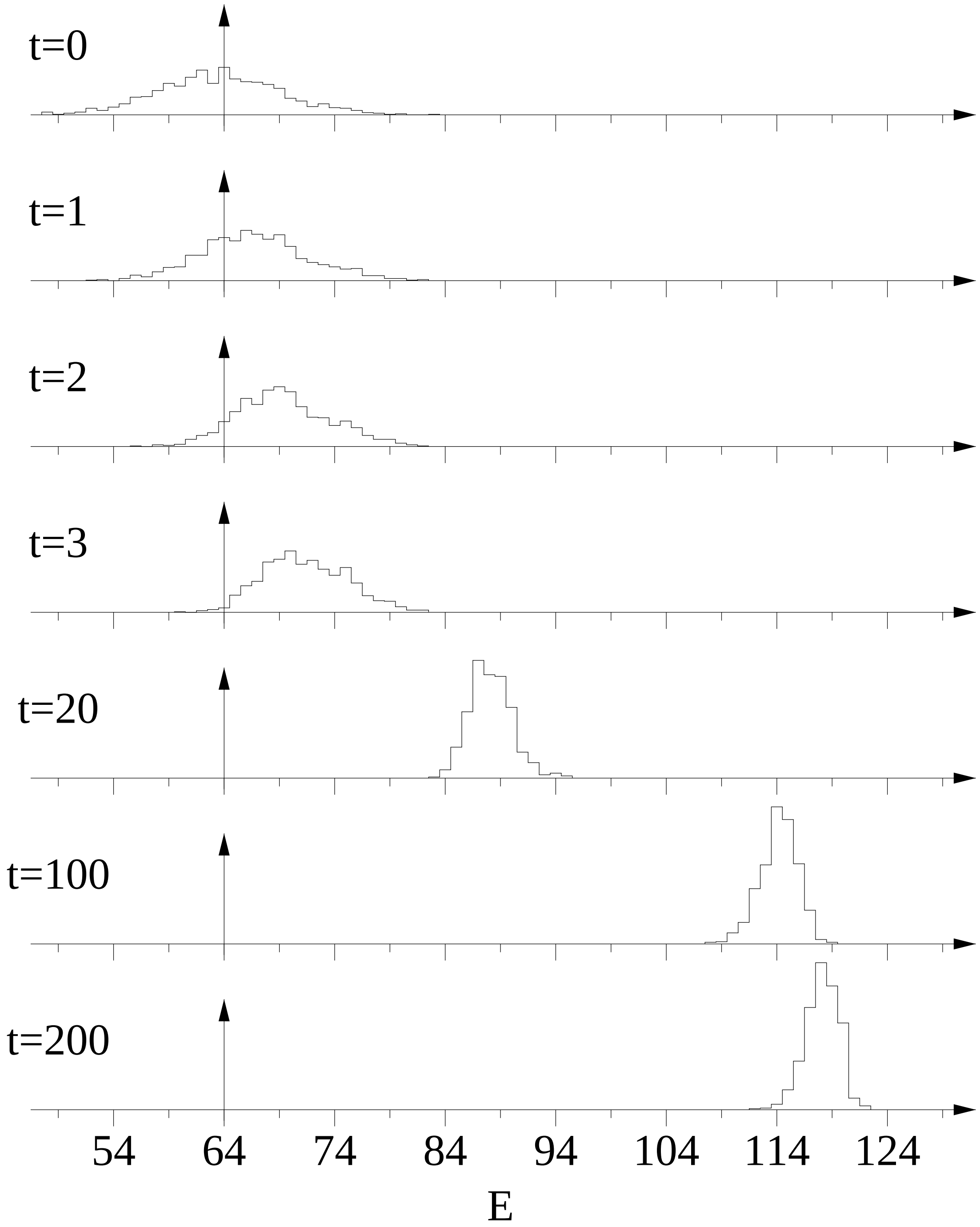

Figure 2: Histograms of the fitnesses of a population after 0, 1, 2, 3, 20, 100 and 200 generations. The figures come from a simulation with $L = 128$, $P = 1000$ and $u = 1/L$. There was no crossover.

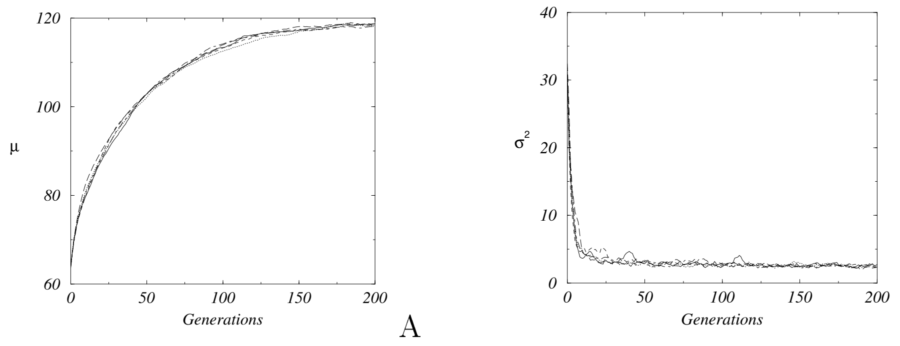

Figure 3: A) average and B) variance versus generations are shown for 5 runs of a GA. The same parameters are used as those in graph 2.

That is, to model a large (infinite) set of populations evolving in parallel. The statistical properties of the ensemble won't fluctuate, even though the members of the populations will. This is the approach we adopt here. We will consider the ensemble average of μ and σ^2 as a function of number of generations. We denote these by $\bar{\mu}(t)$ and $\overline{\sigma^2}(t)$. As we try to improve the accuracy of our model we will need to include more statistical properties of the ensemble in our model.

Our approach throughout will be to consider how the ensemble variables are altered by the genetic operators. Let us denote the set of ensemble variables at generation t by $\Xi(t) = \{\bar{\mu}(t), \overline{\sigma^2}(t), \ldots\}$. We then consider apply selection, mutation and recombination. Each genetic operator changes the ensemble variables

$$\Xi(t) \xrightarrow{sel} \Xi_s(t) \xrightarrow{mut} \Xi_{sm}(t) \xrightarrow{cross} \Xi_{smc}(t) = \Xi(t+1).$$

We will calculate how these genetic operators change an arbitrary set of ensemble variables. Then starting from an initial ensemble $\Xi(0)$ we iteratively apply each operators in turn to predict the whole evolution.

3.3 A Zeroth Order Model

We start with the simplest model we can. Firstly, we assume the population is infinite. In this case there are no fluctuations and we can consider the evolution of a single population rather than an ensemble. Secondly, we assume that the distribution of fitnesses is Gaussian. As a consequence it is sufficient to consider only $\mu(t)$ and $\sigma^2(t)$. We'll also ignore crossover for the time being as that adds some unnecessary complications.

We denote the distribution of fitness as $\rho(E)$, which, by assumption, is Gaussian

$$\rho(E) = \frac{\mathrm{e}^{-(E-\mu(t))^2/2\sigma^2(t)}}{\sqrt{2\pi}\,\sigma(t)}.$$

The initial population consists of random strings. As the members of the population are binary strings the fitnesses of the initial population will be binomially distributed with mean $L/2$ and variance $L/4$. If L is large this distribution is very well approximated by a Gaussian. We now consider how a Gaussian distribution would be changed by selection and mutation.

Selection

In tournament selection two members are randomly drawn from the population and the fitter member is copied into the mating pool. The probability that a member with fitness E is fitter than a randomly chosen member E' is

$$P(E > E') = \int_{-\infty}^{E} \rho(E')\mathrm{d}E'.$$

This is exactly proportional to the rank of E — it is well known that the two selection mechanism give similar results. The probability distribution of the winner is therefore given by

$$\rho_s(E) = 2\rho(E) \int_{-\infty}^{E} \rho(E')\mathrm{d}E'$$

where the factor 2 ensures $\rho_s(E)$ is properly normalized. The distribution $\rho_s(E)$ is not Gaussian, but it is surprisingly close. Figure 4 shows the effect of selection on a Gaussian.

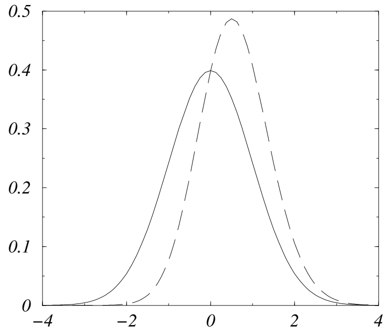

Figure 4: Tournament selection on a zero mean unit Gaussian. The solid curve shows the original Gaussian, while the dashed curve shows the distribution after selection.

To calculate the effect of selection on the average and variance we need only perform two integrals

$$\mu_s = \int_{-\infty}^{\infty} E\,\rho_s(E) = \mu + \frac{\sigma}{\sqrt{\pi}} \tag{3}$$

$$\sigma_s^2 = \int_{-\infty}^{\infty} (E-\mu)^2 \rho_s(E) = \left(1 - \frac{1}{\pi}\right)\sigma^2. \tag{4}$$

These results were first obtain by Blickle and Theile [8]. One feature which is clear from figure 4 and equation (4) is that tournament selection strongly reduces the variance in the population. This will often lead to premature convergence. Tournament selection can be 'softened' by including a probabilistic element. Extension of this type of analysis to soft tournament (or ranking selection) is given in [6].

Mutation

Since we have an infinite population we need only calculate what mutation does on average. So how does mutation change a site variable S_i? (For the moment we drop the member label, α.) Mutation leaves S_i unchanged with probability $(1-u)$ and changes it to $1-S_i$ with probability to u. Denoting the site variable after mutation by S_i^m, and the average over all possible mutation by $\langle \cdots \rangle_m$, then

$$\langle S_i^m \rangle_m = (1-u)S_i + u(1-S_i) = u + (1-2u)S_i.$$

In fact, this is the only average we need since $S_i^m \in \{0,1\}$ so for any power of S_i^m we have $(S_i^m)^n = S_i^m$. The average fitness of a string after mutation is given by

$$\begin{aligned}
\langle E_m \rangle_m &= \Big\langle \sum_i S_i^m \Big\rangle_m = \sum_i \langle S_i^m \rangle_m \\
&= u \sum_i 1 + (1-2u) \sum_i S_i = uL + (1-2u)E.
\end{aligned}$$

Averaging over all members of the population gives

$$\mu_m = uL + (1-2u)\mu. \tag{5}$$

Similarly we can calculate how mutation changes E^2

$$\left\langle E_m^2 \right\rangle_m = \Big\langle \Big(\sum_i S_i^m \Big)^2 \Big\rangle_m = \sum_i \left\langle (S_i^m)^2 \right\rangle_m + \sum_{i \neq j} \left\langle S_i^m S_j^m \right\rangle_m$$

but $(S_i^m)^2 = S_i^m$ and $\left\langle S_i^m S_j^m \right\rangle = \langle S_i^m \rangle_m \left\langle S_j^m \right\rangle_m$, since mutation acts independently on each spin, therefore

$$\begin{aligned}
\left\langle E_m^2 \right\rangle_m &= \sum_i \langle S_i^m \rangle_m + \sum_{i \neq j} \langle S_i^m \rangle_m \left\langle S_j^m \right\rangle_m \\
&= \sum_i \left(\langle S_i^m \rangle_m - \langle S_i^m \rangle_m^2 \right) + \Big(\sum_i \langle S_i^m \rangle_m \Big)^2.
\end{aligned}$$

To find the variance we average $\langle E_m^2 \rangle_m - \langle E_m \rangle_m^2$ over all member of the population. After a little algebra we find

$$\sigma_m^2 = u(1-u)L + (1-2u)^2 \sigma^2 \tag{6}$$

(The algebra can be simplified by redefining the string variables S_i^α to be ± 1 — when extending the analysis it pays to make this redefinition.)

The equations for mutations are exact. We can use them to compute the initial distribution. If we mutate the states of each string with probability 1/2 we immediately reach a random distribution. Putting $u = 1/2$ in equations (5) and (6) gives $\mu = L/2$ and $\sigma^2 = L/4$.

Putting it Together

From equations (3–6) we can compute the whole evolution. The mean and variance from one generation to the next is related by

$$\mu(t+1) = uL + (1-2u)\mu(t) + \frac{(1-2u)}{\sqrt{\pi}}\sigma(t) \tag{7}$$

$$\sigma^2(t+1) = u(1-u)L + (1-2u)^2\left(1-\frac{1}{\pi}\right)\sigma^2(t) \tag{8}$$

The equation for the variance is a linear difference equation of the form $x(t+1) = a + bx(t)$, which has the general solution

$$x(t) = x(\infty) + \big(x(0) - x(\infty)\big)\,\mathrm{e}^{-t/\tau}$$

where $x(\infty) = a/(1-b)$ and $\tau = -\log(b)$. This describes a trajectory starting at $x(0)$ and converging at an exponential rate to $x(\infty)$. For the variance $a = u(1-u)L$, $b = (1-2u)^2(1-1/\pi)$, the initial variance is $\sigma^2(0) = L/4$, while the final variance is given by $\sigma^2(\infty) = \sigma_{eq}^2 = u(1-u)L/(1-b)$ the characteristic time decay time is $\tau_{\sigma^2} = -\log(b)$. For small mutation rates ($u \ll 1$) $\sigma_{eq}^2 \approx \pi u L$ and $\tau_{\sigma^2} \approx \log(1-1/\pi) \approx 0.38$. This describes the loss of variance during the initial transient period.

If we ignore the transient period then $\sigma(t)$ remains constant and the equation for μ is a linear difference equation with $\mu(\infty) = \mu_{eq} = L/2 + (1-2u)\sigma_{eq}/(2u\sqrt{\pi})$ and with a characteristic time $\tau_\mu = -\log(1-2u)$. For small mutation rates $\mu_{eq} \approx (L + \sqrt{L/u})/2$ and $\tau_\mu = 2u$. Note, that if $u < 1/L$ this predicts $\mu_{eq} > L$, which is clearly impossible — our approximations are too crude. Nevertheless, this simple model does explain the overall behaviour. There is an initial transient behaviour as the variance decays rapidly to its equilibrium value. Then there is a long period as the mean approaches its equilibrium, after which the population reaches a fixed distribution. Furthermore, we have obtained estimates for the characteristic times associated with each phase and estimates for the equilibrium values of μ and σ^2.

Figure 5 shows a comparison between the theory and simulations. The qualitative features of the evolution are well captured although the details are clearly wrong. In particularly the equilibrium average fitness is over estimated, although we know from our discussion above that the approximations must break down at some point. Still not bad for such a simple model.

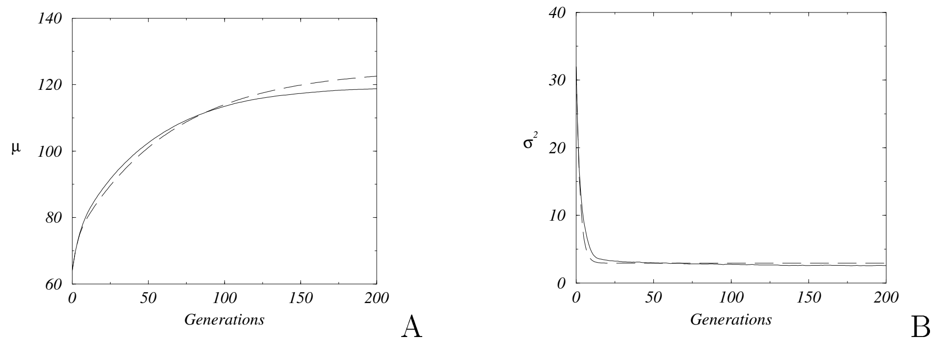

Figure 5: Simulations (solid line) and theory (dashed line) for A) the average fitness and B) the variance. The curves are for the same system as that used in figure 2. The simulations are average over 1000 runs. The theory is computed by iterating equations (7) and (8).

3.4 Increasing the Accuracy

We made two unjustified assumptions: the population was approximated by a Gaussian and the population was infinite. In this section and section 3.5 we indicate how to relax these restrictions.

A distribution can be described by a cumulant expansion (see appendix A) . The first two cumulants are the average and variance. The third cumulant which measures the 'skewness' is defined by

$$\kappa_3 = \frac{1}{P} \sum_{\alpha=1}^{P} \left(E_\alpha - \mu\right)^3 .$$

Actually the skewness is usually defined to be κ_3/σ^3. Higher cumulants a slightly more complicated. For a Gaussian all the high order cumulants (κ_n for $n > 2$) are zero. Figure 6 shows three distribution with a negative, zero and positive third cumulant.

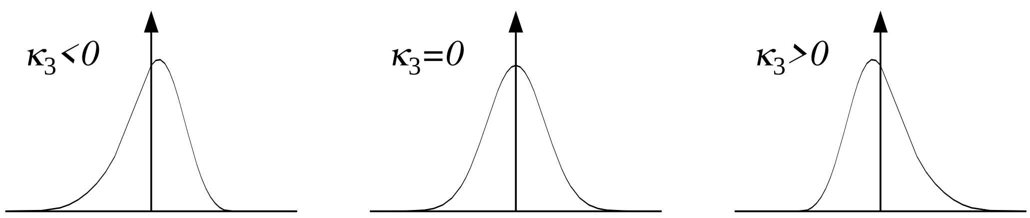

Figure 6: Three distribution with skewness, $\kappa_3 < 0$, $\kappa_3 = 0$ and $\kappa_3 > 0$.

We can include higher cumulants in our model by using the Gram-Charlier expansion (see appendix A)

$$\rho(E) = \frac{\mathrm{e}^{-(E-\mu)^2/2\sigma^2}}{\sqrt{2\pi}\,\sigma}\left(1+\frac{\kappa_3}{\sigma^3}\,h_3\Big(\frac{E-\mu}{\sigma}\Big)\right)$$

where $h_3(x) = (x^3 - 3x)/3!$. This is easily extensible to higher cumulant — see appendix A. The Gram-Charlier expansion has the correct first three cumulants but it does not guarantee that $\rho(E) > 0$. We can compute the effect of selection in an analogous way to what we did above, although the integrals are hairier. We find

$$\mu_s = \mu + \frac{\sigma}{\sqrt{\pi}}\left(1 - \frac{s^2}{96}\right) \tag{9}$$

$$\sigma_s^2 = \sigma^2\left(1 - \frac{1}{\pi} + \frac{s}{2\sqrt{\pi}} + \frac{s^2}{48\pi} - \frac{s^4}{9216\pi}\right) \tag{10}$$

$$\begin{aligned}\kappa_3^s = \sigma^3 \Bigg(&\frac{1}{2\sqrt{\pi}}\left(\frac{4}{\pi}-1\right) + s\left(1-\frac{3}{2\pi}\right) - \frac{s^2}{192\sqrt{\pi}}\left(\frac{12}{\pi}+1\right) \\ &+\frac{s^3}{64\pi} + \frac{s^4}{1536\pi^{3/2}} - \frac{s^6}{442368\pi^{3/2}}\Bigg)\end{aligned} \tag{11}$$

where $s = \kappa_3/\sigma^3$.

The mutation results for the first two cumulants, equations (5) and (6), are exact (i.e. we did not assume a Gaussian). We can compute the result of mutation on the third cumulant in an analogous way (again the algebra becomes tedious)

$$\kappa_3^m = (1-2u)^3\kappa_3 - u(1-u)(1-2u)(2\mu - L) \tag{12}$$

Higher order results can be found in [9] (in fact, the effect of mutation to all orders can be obtained relatively painlessly from a generating function).

The initial population will be symmetrically distributed around a fitness $L/2$. Consequently $\kappa_3(0) = 0$. The evolution equations are less easy to understand by inspection, but to calculate the full evolution we need only iterate selection equations (9–11) and the mutation equations (5), (6) and (12). Figure 7 compares the first two cumulants computed using the three cumulant approximation with simulation results. Introducing the third cumulant improves the accuracy although this good agreement is rather fortunate.

The evolution of the third cumulant is shown in figure 8. The third cumulant initially grows. We might expect this as selection positively skews the distribution of a Gaussian — see equation (11) (this is true for ranking and tournament selection, but not in general). Eventually, however, the population becomes negatively skewed — we can just about detect this in the histograms of figure 2.

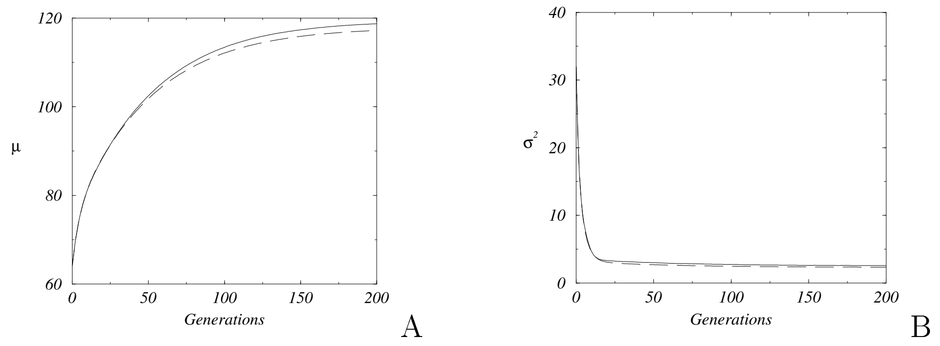

Figure 7: Simulations (solid line) and theory (dashed line) for A) the average fitness and B) the variance. The curves are for the same system as that used in figure 2 and 5. The theory curve is now computed using three cumulants.

Mutation skews the distribution negatively when μ is greater than $L/2$, as can be seen from equation (12). Note that the third cumulant tells us little about the skewness on its own. The quantity κ_3/σ^3 (which is the technical definition of skewness) give a 'dimensionless' measure of skewness. Thus even though initially κ_3 has a large positive value the actual skewness is small because the width of the distribution, σ, is large. Later on when κ_3 is negative the skewness is more marked because the distribution is then relatively narrow.

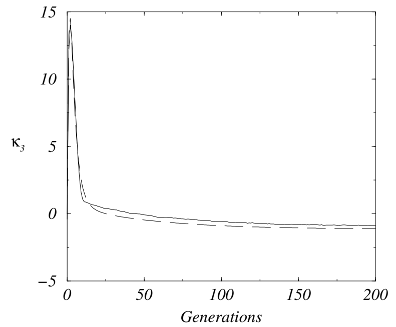

Figure 8: The evolution of the third cumulant is shown for simulations (solid line) and theory (dashed line). The results are shown for the same system as that use in all the previous graphs.

The theoretical prediction for κ_3, although qualitatively similar to the simulations, is not particularly accurate. We could improve accuracy by including the fourth cumulant, but we would then find κ_4 was inaccurate. We can continue *ad infinitum*. For some infinite population models we can do better and solve the model exactly. But for finite population models we have the unsatisfying feature that we must truncate the expansion somewhere and hope for the best. Still this often gives a very respectable approximation.

3.5 Finite Population Effects

Finite population corrections can be significant. Figure 9 shows the evolution of the average fitness for different sizes of population. There is a dramatic deterioration in the quality of the solutions as the population size becomes very small.

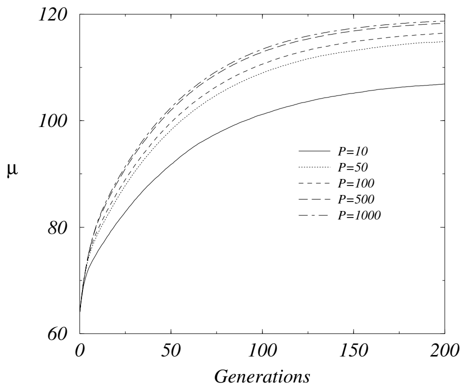

Figure 9: Example of finite population effects. The evolution of the average fitness is shown for a population of size $P = 10$, 50, 100, 500 and 1000. The other parameters are $L = 128$ and $U = 1/N$. The results are from simulations averaged over 1000 runs.

Modelling finite population introduces a number of new complications. Still the size of the population can substantially alter the dynamics, so we have to understand how to model them. Finite populations alter selection in two ways. Firstly selection now involves random sampling — that is, the number of offspring an individual has is proportional to its fitness only on average. Secondly, the evolution becomes stochastic, so we must consider the evolution of an ensemble. (At first this just involves adding a bar to the variables, but for highly accurate modelling the fluctuations give rise to many correction terms.)

Sampling

We can model selection as a two stage process: reproduction and sampling. In reproduction an infinite population, the mating pool, is generated by reproducing each individual exactly in proportion to their fitness. We then choose a random sample of P members from this infinite population to create a new population. We can think of this in biology terms. Each individual produces a large number of seeds proportional to their fitness, but only a random sample of P seeds actually survive. Furthermore, rather than considering mutation on the finite population we can mutate the infinite population before sampling. In terms of biology, the mutations occur in the seeds, but do not effect which ones survive. This means that we don't have to calculate finite population corrections for mutations.

There are two sources of sampling corrections. One comes because we draw a finite sample from an infinite distribution. The second comes from approximating a discrete sum by an integral (we should probably give this correction a different name). We compute first the corrections coming from drawing a finite sample from a probability distribution.

We assume that we have an infinite population with a distribution $\rho(E)$, with mean μ, variance σ^2, etc. We now draw a sample of P individuals. We denote the number of individuals with fitness E by $n(E)$. The average fitness after sampling is given by

$$\mu_{smpl} = \sum_E n(E)E.$$

Taking the average over all ways of drawing the population we find

$$\bar{\mu}_{smpl} = \frac{1}{P}\sum_E \langle n(E)\rangle_{smpl} E$$

where $\langle\cdots\rangle_{smpl}$ signifies an average over all ways of doing sampling. But $\langle n(E)\rangle_{smpl} = P\rho(E)$ so

$$\bar{\mu}_{smpl} = \sum_E \rho(E)E = \mu$$

Sampling does not change the average fitness, but the story is different with other cumulants. The variance is given by

$$\sigma^2_{smpl} = \frac{1}{P}\sum_E n(E)E^2 - \frac{1}{P^2}\sum_{E,E'} n(E)\, n(E')\, E\, E'.$$

We average over all ways of drawing samples. For roulette wheel selection the probability of a particular draw is given by a multinomial distribution. It is straightforward to show for a multinomial distribution

$$\langle n(E)\, n(E')\rangle_{smpl} = (P^2 - P)\rho(E)\rho(E') - P\rho(E)\delta_{E,E'}.$$

(see appendix B). Substituting this into the average of σ^2_{smpl} we find

$$\begin{aligned}
\overline{\sigma^2}_{smpl} &= \left(1 - \frac{1}{P}\right)\left(\sum_E \rho(E)E^2 - \sum_{E,E'} \rho(E)\,\rho(E')\,E\,E'\right) \\
&= \left(1 - \frac{1}{P}\right)\sigma^2.
\end{aligned}$$

This is the familiar formula for the unbiased estimate of the variance. A similar calculation for the third cumulant after sampling gives

$$\bar{\kappa}_3^{smpl} = \left(1 - \frac{1}{P}\right)\left(1 - \frac{2}{P}\right)\kappa_3.$$

Fisher called these sampled cumulants k-statistics [10]. They can be found efficiently using a generating function.

We are still not done. There is a further finite population effect that arises when turning a sum to an integral. One way to obtain an approximation for this correction is to consider the correction for neutral selection. In this case the ranks are randomly assigned to the individuals. We need to calculate

$$\kappa_2' = \sum_\alpha (p_\alpha - p_\alpha^2) E_\alpha^2 - \sum_{\alpha \neq \beta} p_\alpha \, p_\beta \, E_\alpha \, E_\beta$$

where we are assuming neutral selection so we can average over p_α and E_α separately. This gives

$$\kappa_2' = (1 - \left\langle p_\alpha^2 \right\rangle_\alpha) \left(\sum_\alpha E_\alpha^2 - \sum_{\alpha \neq \beta} E_\alpha \, E_\beta \right)$$

where we have use $\sum_\alpha p_\alpha = 1$ and $1 = (\sum_\alpha p_\alpha)^2 = \sum_\alpha p_\alpha^2 + \sum_{\alpha \neq \beta} p_\alpha \, p_\beta$. Now for tournament selection each member of the population is drawn according to its rank. In neutral selection the ranks are arbitrarily assigned. Each rank will occur once, so

$$\left\langle p_\alpha^2 \right\rangle = \frac{\sum_{r=0}^{P-1} r^2}{\left(\sum_{r=0}^{P-1} r\right)^2} = \frac{4P - 2}{3P(P-1)}$$

(see appendix C). Similarly the correction for the third cumulant is given by

$$\left\langle p_\alpha - 3p_\alpha^2 + 2p_\alpha^3 \right\rangle_\alpha = 1 - \frac{4P - 6}{P(P-1)}.$$

Putting together these corrections with the correction due to the randomness in sampling we find for sampling

$$\bar{\mu}_{smpl} = \mu_s \tag{13}$$

$$\overline{\sigma^2}_{smpl} = \left(1 - \frac{1}{P}\right)\left(1 - \frac{4P-2}{3P(P-1)}\right)\sigma_s^2 \tag{14}$$

$$\bar{\kappa}_3^{smpl} = \left(1 - \frac{1}{P}\right)\left(1 - \frac{2}{P}\right)\left(1 - \frac{4P-6}{P(P-1)}\right)\kappa_3^s \tag{15}$$

where μ_s, σ_s^2 and κ_3^s are our calculation from the statistics after sampling.

The main consequence of sampling is that it reduce the variance in the population. This limits the search and leads to a poorer performance. The size of the sampling corrections depend on how we perform the selection. Ranking selection together with stochastic universal sampling [11] would reduce the size of the sampling corrections giving improved performance. The effect of stochastic universal sampling has been studied in [6] and will be reported in this workshop [7]. Other types of GAs, such as steady state GAs, give different sampling corrections. A comparison of these schemes is given in [12].

Fluctuations

So far we have ignored fluctuations. However, to deal with fluctuations we must consider an ensemble of populations. Each population in the ensemble will have its own set of statistics μ, σ^2, κ_3, etc. The ensemble can be thought of as a probability distribution, $P(\mu, \sigma^2, \kappa^3)$, over the population statistics. So far we have calculate the means of this distribution. However, we should average all quantities over the whole ensemble. For formulas that are linear in the statistical variables, such as mutation and sampling this average makes no different. However, selection is non-linear in the statistics. For example, we found that selection depends on the skewness and higher powers of the skewness. But

$$\overline{\left(\frac{\kappa_3}{\sigma^3}\right)} \neq \frac{\bar{\kappa}_3}{\bar{\sigma}^3}$$

where the bar denotes the ensemble average. In fact, even in the two cumulant system $\bar{\sigma}^2 \neq \overline{\sigma^2}$. Thus, there will systematic corrections to the average fitness coming from fluctuations in σ^2. To calculate the corrections we need to know how big the fluctuations are. To do so requires keeping information such as the covariance between the statistical variables

$$v_{n,m} = \overline{\kappa_n \kappa_m} - \bar{\kappa}_n \bar{\kappa}_m$$

where $\kappa_1 = \mu$ and $\kappa_2 = \sigma^2$. These variable may be of interest in themselves. For example, $v_{1,1}$ tells us about the fluctuations in the average fitness between runs.

Fortunately the corrections from these fluctuations are often small and they can usually be neglected. A full treatment of fluctuations was given in [9]. There it was shown that by using a sufficient number of ensemble parameters very good agreement with simulations could be achieved.

Putting in sampling corrections

The sampling corrections can be put into our model by applying the sampling equations (13–15) after selection and mutation. If we consider the two variable system (i.e. we ignore the skewness), then the sampling term reduces the equilibrium variance by approximately $1 - 7/(3P)$. As a consequence the equilibrium fitness is reduced by approximately $7\sqrt{L/u}/(3P)$ — if $u = 1/L$ then μ_{eq} is reduced by approximately $7L/3P$. Decreasing P reduces the average fixed point fitness, but this only becomes significant when P is small. Figure 10 shows a comparison of simulations with theory for populations of size 10, 100 and 1000. The theory capture the dominant behaviour, although the theory tends to underestimate the equilibrium average fitness. Partly this is due to truncating the cumulant expansion (the theory over-estimates the average fitness when using an even number of cumulants and under-estimates it when using an odd number).

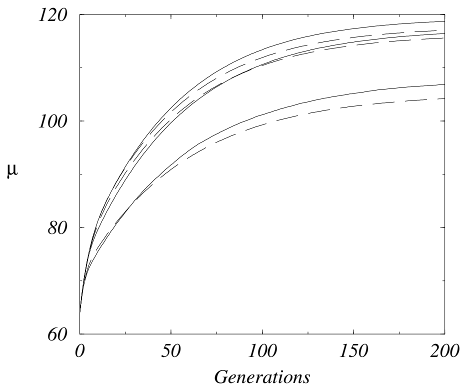

Figure 10: Comparison of simulations (solid curves) and theory (dashed curves) for a GA with population size $P = 10$, 100 and 1000. In both cases the higher curves represent large populations. The other parameters are $L = 128$ and $u = 1/L$. There is no crossover.

3.6 Recombination

So far we have not used recombination. In the remainder of this paper we describe how to treat this important operator.

Recombination is achieved by pairing each string in the mating pool and from every pair of parents $\{\vec{S}^\alpha, \vec{S}^\beta\}$ producing a pair of children $\{\vec{S}^\mu, \vec{S}^\nu\}$ where

$$S_i^\mu = \begin{cases} S_i^\alpha & \text{with probability } a \\ S_i^\beta & \text{with probability } 1 - a. \end{cases}$$

and $\vec{S}^\nu$ is the complement of S_i^μ. In this way we conserve all the variables in the mating pool and prevent further convergence due to sampling in crossover. We could also have generated a mating pool of $2P$ individuals and produced a single child from each pair of parents. This would have reduced the convergence due to sampling in selection but at the price of introducing sampling into recombination. The sampling effects would be similar in the two strategies.

Recombination is a mixed blessing: it complicates the theory, but it substantially reducing the higher cumulants, allowing much better agreement between theory and simulations. Up to now we did not need to know the state of the strings. The only thing that mattered was their fitness. The additional complications arise when using recombination because the effect of crossover depends on the states of the strings. To see this consider two strings having the same fitness. These strings may be identically or they may be very different. With no crossover this made no difference, but when we perform crossover, the results will be very different. Combining two identical strings will give a child with exactly the same configuration, and hence the same fitness as the parents. Crossing over non-identical strings can give rise to a child with a very different fitness to the parents.

Recombination therefore depends on the pairwise correlations

$$q_{\alpha,\beta} = \frac{1}{L}\sum_{i=1}^{L}(2S_i^\alpha - 1)(2S_i^\beta - 1).$$

With this definition, $q_{\alpha,\beta} = 1$ if the strings α and β are identical at each site, and $q_{\alpha,\beta} = -1$ if the two strings differ at each site. Let $\vec{S}^\mu$ be the string produced by combining $\vec{S}^\alpha$ and $\vec{S}^\beta$. We can calculate the effect of recombination in a similar manner that we calculated mutation. We suppose that with a probability a we choose an allele from parent α and with probability $1-a$ we choose the allele from parent β. Thus, on average

$$\langle S_i^\mu \rangle_c = a S_i^\alpha + (1-a) S_i^\beta$$

where $\langle \cdots \rangle_c$ signifies the average over all ways of performing crossover. Just as for mutation, this is the only average we need to know to calculate the effect of recombination, since $(S_i^\mu)^n = S_i^\mu$. We now proceed as we did for mutation. The fitness on average after crossover is given by

$$\begin{aligned}\left\langle F_\mu^c \right\rangle_c &= \sum_{i=1}^{L} \langle S_i^\mu \rangle_c = a\sum_{i=1}^{L} S_i^\alpha + (1-a)\sum_{i=1}^{L} S_i^\beta \\ &= aF_\alpha + (1-a)F_\beta\end{aligned}$$

Averaging over all children and all ways of pairing the mating pool we find

$$\mu_c = \mu. \tag{16}$$

That is, crossover does not change the average fitness. We can compute $\langle (F_\mu^c)^2 \rangle_c$, just as we did for mutation in equation (6), and use it to calculate σ_c^2. The algebra is straightforward if tedious (it would simplify if we had used $S_i^\alpha \in \{-1, 1\}$). We find

$$\sigma_c^2 = \left(a^2 + (1-a)^2\right)\sigma^2 + 2a(1-a)\frac{L(1-q)}{4} \tag{17}$$

where

$$q = \frac{1}{P(P-1)} \sum_{\alpha \neq \beta} q_{\alpha\beta}.$$

Equation (17) tells us that recombination drives the variance towards $L(1-q)/4$ at a rate $\tau_c = -\log(a^2 + (1-a)^2)$. This rate is reaches a maximum of $\log(2)$ when $a = 1/2$, that is when there is maximal mixing of the parents. A similar calculation gives for the third cumulant

$$\kappa_3^c = \left(a^3 + (1-a)^3\right) \kappa_3 + \frac{L}{2P(P-1)} \sum_{\alpha \neq \beta} (E_\alpha - \mu) q_{\alpha\beta}. \tag{18}$$

A reasonable assumption is that the correlation and fitnesses are independent in which case

$$\kappa_3^c = \left(a^3 + (1-a)^3\right) \kappa_3. \tag{19}$$

When $a = 1/2$ then $\kappa_3^c = \kappa_3/4$.

We see explicitly the complication added by recombination. We now need to know about the mean correlation q and how the pair correlation and fitnesses are related. Thus to model the dynamics accurately we must model q. We show how we can model q below. However, if we are just interested in the fixed point fitness then we can make the approximation $\sigma^2 = L(1-q)/4$. This is the value which the variance is being driven to by recombination. From the definition of the variance

$$\sigma^2 = \frac{L(1-q)}{4} + \sum_{i \neq j} \left(\left\langle S_i^\alpha S_j^\alpha \right\rangle_\alpha - \left\langle S_i^\alpha \right\rangle_\alpha \left\langle S_j^\alpha \right\rangle_\alpha \right)$$

where $\langle \cdots \rangle_\alpha$ denotes the average over all members of the population. Our fixed point assumption is that $\langle S_i^\alpha S_j^\alpha \rangle_\alpha - \langle S_i^\alpha \rangle_\alpha \langle \S_j^\alpha \rangle_\alpha$ vanishes on average. This is known in the biological literature as *linkage equilibria*. Selection pushes the population out of linkage equilibrium — as we are using a selection mechanism that has a strong selection pressure the system will quickly move out of linkage equilibrium. However, later in the evolution, when selection is balanced by mutation, recombination has a chance of pushing the population back towards linkage equilibria. If we assume linkage equilibria, then recombination leaves the variance unchanged. Recombination decrease the higher cumulants towards their linkage equilibrium values. In general the cumulants, κ_n, are driven towards their linkage equilibria values at a rate $(n-1)\log(2)$ (assuming $a = 1/2$. As a consequence the fitness distribution will be much closer to a Gaussian for a sexual population than for an asexual population.

To understand the full dynamics we need model the correlation, q. This approach was first used in [3] and is used in this workshop in [7]. To model the correlation we introduce the idea of common ancestry. If two variables, S_i^α and

S_i^β, from the same site but in different individuals, are inherited from the same ancestor we write $S_i^\alpha \sim S_i^\beta$. We define the degree of common ancestry to be

$$C_{\alpha\beta} = \frac{1}{L}\sum_{i=1}^{L} [S_i^\alpha \sim S_i^\beta]$$

where

$$[\,predicate\,] = \begin{cases} 1 & \text{if } predicate \text{ is true} \\ 0 & \text{otherwise.} \end{cases}$$

$C_{\alpha\beta}$ is the proportion of the two string that come from the same ancestors. The correlation between two strings can be broken down into a piece where the states come from a common ancestor and a piece where the states come from different ancestors. If the states come from the same ancestor they are obviously identical. We assume that the part of the strings that come from different ancestors are randomly correlated. The proportion of ones in string α is E_α/L. Thus we would expect if $S_i^\alpha \not\sim S_i^\beta$ that

$$\left\langle S_i^\alpha S_i^\beta \right\rangle = \frac{E_\alpha E_\beta}{L^2}.$$

Putting this together we would expect the average correlation between two strings to be

$$\begin{aligned} q_{\alpha\beta} &= \frac{1}{L}\sum_{i=1}^{L}\Big([S_i^\alpha \sim S_i^\beta] + [S_i^\alpha \not\sim S_i^\beta](2S_i^\alpha - 1)(2S_i^\beta - 1)\Big) \\ &= C_{\alpha\beta} + (1 - C_{\alpha\beta})\left(1 - \frac{2(E_\alpha + E_\beta)}{L} + \frac{4E_\alpha E_\beta}{L^2}\right). \end{aligned}$$

Assuming that the degree of common ancestry is independent of the fitness of string we find the average correlation is given by

$$q = C + (1 - C)\frac{(L - 2\mu)^2}{L^2}$$

where

$$C = \frac{1}{P(P-1)}\sum_{\alpha\neq\beta} C_{\alpha\beta}.$$

We must now compute how the correlation is changed by selection, mutation and crossover. The easiest operator is crossover as this does not change the correlation — it just shuffles the states variables S_i^α between strings. It is easy to calculate how selection causes a change in the degree of common ancestry. If we assume string α is selected n_α times then

$$C_s = \frac{1}{P(P-1)}\left(\sum_{\alpha=1}^{P} n_\alpha(n_\alpha - 1)C_{\alpha\alpha} + \sum_{\alpha\neq\beta} n_\alpha n_\beta C_{\alpha\beta}\right)$$

but $C_{\alpha\alpha} = 1$. We assume that the number of times a string is selected, n_α, is independent of $C_{\alpha\beta}$. We can average over all ways of doing selection. The average decouple from the sum. The expression can then be simplified by noting

$$P^2 = \left(\sum_\alpha n_\alpha\right)^2 = \sum_\alpha n_\alpha^2 + \sum_{\alpha\neq\beta} n_\alpha n_\beta.$$

After a little algebra we find

$$C_s = 1 - \frac{P - \langle n_\alpha^2\rangle_\alpha}{P-1}(1-C).$$

We still have to compute $\langle n_\alpha^2\rangle_\alpha$. The probability of choose n_α copies of member α is given by a multinomial distribution (see appendix B). Thus on average

$$\left\langle n_\alpha^2 \right\rangle_s = P(P-1)p_\alpha^2 + Pp_\alpha.$$

Averaging over all members of the population

$$\left\langle n_\alpha^2 \right\rangle_\alpha = \frac{1}{P}\sum_{\alpha=1}^{P}\left(P(P-1)p_\alpha^2 + Pp_\alpha\right)$$

Now the probability of choosing an individual is proportional to its rank — $p_\alpha = 2r_\alpha/P(P+1)$ where $r_\alpha \in \{0, 1, 2, \cdots, P-1\}$. Since we are averaging over all members of the population, all ranks are included so

$$\left\langle n_\alpha^2 \right\rangle_\alpha = \frac{1}{P}\sum_{r=0}^{P-1} P(P-1)\frac{4r^2}{P^2(P-1)^2} + 1 = \frac{7}{3} - \frac{2}{3P}$$

and

$$C_s = 1 - \frac{(3P-1)(P-2)}{3P(P-1)}(1-C). \tag{20}$$

Finally we must calculate how mutation changes the correlation. Remember the correlation is defined by

$$q_{\alpha,\beta} = \frac{1}{L}\sum_{i=1}^{L}(2S_i^\alpha - 1)(2S_i^\beta - 1)$$

Mutation causes a change of state $(2S_i^\alpha - 1) \to -(2S_i^\alpha - 1)$ with probability u. Thus on average

$$\langle(2S_i^{\alpha\prime} - 1)\rangle_m = (1-2u)(2S_i^\alpha - 1)$$

and consequently

$$q_m = (1-2u)^2 q. \tag{21}$$

It is straightforward to turn this into an expression relating the ancestry C before and after mutation.

Putting it Together

Figure 11 shows the simulations and theory for a GA with recombination. Introducing recombination dramatically increases the speed of the search because a large variance is maintain for much longer. Only for very small populations (e.g. $P = 10$) does the variance fall rapidly towards zero. The agreement with simulations becomes worse for small populations because of the growth in the higher cumulants. (For $P = 10$ we have $\kappa_4 > \sigma^4$, while for $P = 100$, $\kappa_4 < \sigma^4$.)

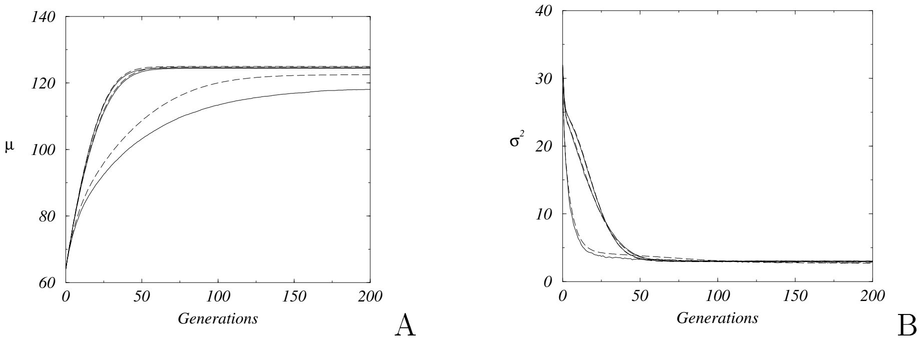

Figure 11: A) the average fitness and B) the variance versus generation measured from simulations (solid lines) and theory results (dashed lines) for populations of size $P = 10$, 100 and 1000. The curves with higher average fitness correspond to larger populations. The results are for $L = 128$ and $u = 1/L$. The simulations are averaged over 1000 runs.

4 Historical Note

Modelling of biological populations begun in the twenty and thirties with the seminal works of Ronald Aylmer Fisher [13] and Sewall Wright [14]. However, the main interest of population geneticists was the frequency of particular traits. These, after all, are what could be measured experimentally. Still, Fisher, Wright and their follows have developed a huge body of theory. Fisher and Wright introduced the diffusion approximation for describing models close to linkage equilibrium. These were further developed by Motoo Kimura [15]. The use of cumulants to describe evolving populations dates back at least to [16] and used more recently by [17].

Heinz Mühlenbein has modelled the dynamics of genetic algorithms for many years [18] (see also his talk in this volume). Investigations of the effect of selection was carried out by [8]. A similar approach to that discussed here has been used on the 'Royal Road function' by Erik van Nimwegen, James P. Crutchfield, and Melanie Mitchell [19]. We have not touched on modelling using Markov chain analysis, which has a long history both in biology [14] and also for modelling GAs [20, 21, 22, 23, 24].

5 Discussion

We have gone through a lot of calculations very rapidly. Let us recap on what we have done. We started looking at a two parameter (mean and variance) model. This gave a reasonable qualitative picture of the evolution, although quantitatively it didn't give very good agreement. We might have made much more use of the model, for instance by examining annealing schedules for the mutation rate. We showed that we could improve the accuracy of the model by introducing higher cumulants, but we do so at the price of obtaining more complicated equations. We then computed the finite population corrections. These corrections can become quite substantial, especially for small populations. Finite population effects are particularly difficult to calculate for tournament selection. Finally we looked at recombination. This has the effect of maintaining a broad population, while reducing the higher cumulants. One result of this is that we don't need to calculate so many cumulants. However, to understand the initial dynamics we need to know the average correlation in the population. This again complicates the analysis.

We have learned a lot about the dynamics for this type of GA and this particular problem. Other types of GA have been studied, such as the steady-state GA [25]. By comparing algorithms even on relatively simple problems we can understand how they differ. For example, in the steady-state GA there are effectively two selections (choosing the fit member and the unfit member) and two samplings going on at each step. Consequently, to a first approximation, the steady-state GA appears to go at twice the rate of a generation GA, although, it also correlates at twice the rate, so it may not explore the solution space so well. This is apparent, even in a simple problem such as ones-counting, from carefully modelling the dynamics. This illustrates that it is not always necessary to consider realistic problems to obtain deep insights into how GAs work.

Ones-counting is, however, a pathologically easy problem to learn. If we were really interested in efficiently solving such a problem, we certainly wouldn't use a genetic algorithm. Ones-counting tells us relatively little about how we should select parameters in more realistic problems. To understand that we have to look at different problems. As we described in section 2 many models have been examined within this formalism, although there is still much more work to do in this direction. The formalism discussed in the paper is relatively new and many of the techniques are still being developed. The results obtained are approximations, but generally very accurate. More importantly by modelling GAs we can begin to understand how they work.

Acknowledgements

Thanks to Mathematica for doing the hairy integrals.

A Cumulant Expansion

The characteristic function of a distribution is defined to be its Fourier transform

$$\phi(\omega) = \sum_E \rho(E)\,\mathrm{e}^{\mathrm{i}\omega E}.$$

The Taylor expansion of the characteristic function is given by

$$\phi(\omega) = \sum_{n=0} \frac{\mu_n (\mathrm{i}\omega)^n}{n!}$$

where μ_n are the moments of the distribution define by

$$\mu_n = \frac{\partial^n \phi(\omega)}{\partial(\mathrm{i}\omega)} = \sum_E \rho(E)\,E^n.$$

Since $\rho(E)$ is normalized $\mu_0 = 1$. The first moment, μ_1, is the average of the distribution.

The cumulant generating function is defined to be the logarithm of the characteristic function

$$G(\omega) = \log(\phi(\omega)) = \sum_{n=1} \frac{\kappa_n(\mathrm{i}\omega)}{n!}$$

The cumulants are related to the moments

$$\begin{aligned}
\kappa_1 &= \sum_E \rho(E)\,E = \mu_1 \\
\kappa_2 &= \sum_E \rho(E)\,(E-\kappa_1)^2 = \mu_2 - \mu_1^2 \\
\kappa_3 &= \sum_E \rho(E)\,(E-\kappa_1)^3 = \mu_3 - 3\mu_2\mu_1 + 2\mu_2^2 \\
\kappa_4 &= \sum_E \rho(E)\,(E-\kappa_1)^4 - 3\kappa_2^2 \\
&= \mu_4 - 4\mu_2\mu_1 - 3\mu_2^2 + 12\mu_2\mu_1^2 - 6\mu_1^4
\end{aligned}$$

Although, the moments and cumulants are trivially related for a single distribution their relation is very non-trivial when we have to average over an ensemble of distributions. Cumulants, although apparently complicated, all scale linearly with L, whereas the n^{th} moment will be of order L^n. This makes it easier to model with the cumulants than the moments.

Gram-Charlier Expansion

The Gram-Charlier expansion is given by

$$\rho(E) = \frac{\mathrm{e}^{-(E-\mu)^2/2\sigma^2}}{\sqrt{2\pi}\,\sigma} \left(1 + \sum_{n=3} \frac{\kappa_n}{\sigma^n} h_n\left(\frac{E-\mu}{\sigma}\right)\right)$$

where $h_3(x) = (x^3 - 3x)/3!$, $h_4(x) = (x^4 - 6x^2 + 3)/4!$, etc. are related to the Hermite polynomials. They can be generated from the recurrence relation

$$h_n(x) = \frac{1}{n}(x h_{n-1}(x) - h_{n-2}(x))$$

with $h_0(x) = 1$ and $h_1(x) = x$.

B Multinomial Distribution

A multinomial distribution arises when we consider making M random draws (with replacement) from N different objects. We label these objects $\alpha = 1, \ldots, N$ and denote the probability of drawing object α as p_α. If we make M draws the probability of drawing n_α objects of type α is given by

$$p(\vec{n}) = M! \prod_{\alpha=1}^{N} \frac{p_\alpha^{n_\alpha}}{n_\alpha!}\, \delta\left(\sum_\alpha n_\alpha - M\right).$$

To calculate averages of multinomial distributions we introduce a generating function

$$F(\vec{l}) = \left\langle \mathrm{e}^{\sum_\alpha n_\alpha l_\alpha} \right\rangle$$

where $\langle \cdots \rangle$ denotes the average over the multinomial distribution. To calculate averages, such as $\langle n_\alpha n_\beta \rangle$, we take the appropriate derivatives of the generating function

$$\langle n_\alpha n_\beta \rangle = \left. \frac{\partial^2 F(\vec{l})}{\partial l_\alpha l_\beta} \right|_{\vec{l}=0}.$$

The multinomial coefficients occur when expanding the product of a sum

$$\left(\sum_{\alpha=1}^{N} a_\alpha\right)^M = \sum_\alpha M! \prod_{\alpha=1}^{N} \frac{a_\alpha^{n_\alpha}}{n_\alpha!}\, \delta\left(\sum_\alpha n_\alpha - M\right)$$

but this is just what we need to compute the generating function. Setting $a_\alpha = p_\alpha \mathrm{e}^{l_\alpha}$ we find

$$F(\vec{l}) = \left(\sum_\alpha p_\alpha \mathrm{e}^{l_\alpha}\right)^M.$$

It is now quite straightforward to compute all the averages we need

$$\begin{aligned} \langle n_\alpha \rangle &= P p_\alpha \\ \left\langle n_\alpha^2 \right\rangle &= P(P-1)p_\alpha^2 + P p_\alpha \\ \langle n_\alpha n_\beta \rangle &= P(P-1)p_\alpha p_\beta. \end{aligned}$$

C Sums

We make use of the following sums

$$\sum_{r=1}^{P-1} r = \frac{P(P-1)}{2}$$

$$\sum_{r=1}^{P-1} r^2 = \frac{P(P-1)(2P-1)}{6}$$

$$\sum_{r=1}^{P-1} r^3 = \frac{P^2(P-1)^2}{4}.$$

References

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

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

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

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

[5] 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 101–139, California, 1997. Morgan Kaufmann.

[6] A. Rogers and A. Prügel-Bennett. The dynamics of a genetic algorithm on a model hard optimization problem. *Complex Systems*, 1999. submitted for publication.

[7] A. Rogers and A. Prügel-Bennett. A solvable modle of a hard optimization problem. In *Theoretical Aspects of Evolutionary Computing*, 1999.

[8] T. Blickle and L. Theile. A mathematical analysis of tournament selection. In L. J. Eshelman, editor, *Proceedings of the Sixth International Conference on Genetic Algorithms*, pages 9–16, Hillsdale, New Jersey, 1995. Lawrence Erlbaum Associates.

[9] A. Prügel-Bennett. Modelling evolving populations. *J. Theor. Biol.*, 185:81–95, 1997.

[10] R. A. Fisher. Moments and product moments of sampling distributions. *Proceedings of the London Mathematical Society,* series 2, 30:199–238, 1929.

[11] J. E. Baker. Reducing bias and inefficiency in the selection algorithm. In *Proceedings of the Second International Conference on Genetic Algorithms.* Lawrence Erlbaum Associates (Hillsdale), 1987.

[12] A. Rogers and A. Prügel-Bennett. Genetic drift in genetic algorithm selection schemes. *IEEE Transactions on Evolutionary Computation*, 1999. accepted for publication.

[13] R. A. Fisher. *The Genetical Theory of Natural Selection.* Oxford University Press, Oxford, 1930.

[14] S. Wright. *Evolution and the Genetics of Populations.* Chicago Univ. Pr, 1968.

[15] M. Kimura and N. Takahata. *Population Genetics, Molecular Evolution, and the Neutral Theory: Selected Papers.* University of Chicago Press, 1994.

[16] M. G. Bulmer. *The Mathematical Theory of Quantitative Genetics.* Clarendon Press, Oxford, 1980.

[17] M. Turelli and N. H. Barton. Genetic and statistical analysis of strong selection on polygenic traits: What, me normal? *Genetics*, 138:913, 1994.

[18] H. Mühlenbein and D. Schlierkamp-Voosen. Predictive models for the breeder genetic algorithm. *Evolutionary Computation*, 1(1):25–49, 1993.

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

[20] D. E. Goldberg and P. Segrest. Finite Markov chain analysis of genetic algorithms. In John J. Grefenstette, editor, *Proceedings of the Second International Conference on Genetic Algorithms*, 1987.

[21] A. Nix and M. D. Vose. Modeling genetic algorithms with markov chains. *Annals of Mathematics and Artificial Intelligence*, 5:79–88, 1991.

[22] M. D. Vose and G. E. Liepins. Punctuated equilibria in genetic search. *Complex Systems*, 5:31–44, 1991.

[23] M. D. Vose. Modelling simple genetic algorithms. In D. Whitley, editor, *Foundations of Genetic Algorithms 2*. Morgan Kaufmann (San Mateo, CA), 1993.

[24] M. D. Vose. *The Simple Genetic Algorithm: Foundations and Theory.* MIT Press, Cambridge, Mass., 1999.

[25] A. Rogers and A. Prügel-Bennett. Modelling the dynamics of steady-state genetic algorithms. In W. Banzhaf and C. Reeves, editors, *Foundations of Genetic Algorithms 5*, pages 57–68, San Francisco, 1999. Morgan Kaufmann.