Modelling Finite Populations

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

It is common to model the dynamics of an evolutionary algorithm by considering a typical or representative population. We consider modelling a genetic algorithm using two different descriptions of this typical population. The first using cumulants describing the distribution of fitnesses, the second using the distribution itself. Empirically the cumulant description is found to give a much more accurate description of the evolution despite the fact that it gives a cruder description of the typical distribution. The reason for the success of the cumulant approach is because cumulants are self-averaging quantities. The meaning of this statement in this context is explored. The full distribution description is exact in the limit of an infinite population. It is shown that even in simple problems this limit is only approached for very large populations. Furthermore, the infinite population limit can give misleading predictions for the observable long time behaviour of an evolutionary algorithm.

1 Introduction

One approach to understanding evolutionary algorithms (EAs) is to model their dynamics on different test problems. An accurate model can provide considerable insight into how EAs work. For realistic problems, exact models are usually intractable and it is necessary to make some approximations. If these approximate models are sufficiently accurate to reproduced an observed behaviour of an EA we can be fairly confident that they capture

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

the process giving rise to that behaviour. We can then gain intuition about the EA from our simplified model. However, obtaining good approximations is a subtle undertaking. In this paper, we consider two commonly used approximations; the statistical mechanics approach, and the infinite population approach. In particular, we consider an important difference in the way they represent an EA.

The dynamics of an EA can be exactly modelled as a Markov chain in a state space consisting of all possible states of a population (Nix and Vose, 1991; Rudolph, 1998; Schmitt, 2001; Vose, 1999). However, such a state space is usually exponential both in the problem size and the population size. To model realistic problems some approximation is required to reduce the size of the state space. A common approximation is to consider the evolution of a 'typical' population. In this case, we no longer need to consider a state space that is exponential in the size of the population. However, in making this approximation we ignore the stochastic nature of the evolution—each time we run an EA we obtain different results depending on our initial population and the random choices made in the selection, mutation and crossover operations. To model an EA precisely we therefore need to model an ensemble of populations. In the full state space we could do this by considering the probability of the population being in a particular state. When approximating the dynamics using a typical population we clearly lose some accuracy through ignoring these stochastic fluctuations. However, we will see that the size of this loss of accuracy depends on how we represent our typical population.

One approach that uses a typical population to model the ensemble of populations is the statistical mechanics formalism (Prügel-Bennett and Shapiro, 1994, 1997). In fact, corrections due to stochastic fluctuations around the typical population have also been calculated within this formalism (Prügel-Bennett, 1997), however, in most cases these corrections are rather small. Another approximation (used both in population biology and modelling EAs) is to consider an infinite population (Vose, 1999; Vose and Liepins, 1991). Here, a state space consisting of all possible states that a member of the population can be in is considered. To prevent confusion, we will call these string states, although the conclusions we draw will hold for EAs using any representation of the search space. An EA is then described by the fraction of the population in each string state. For an infinite population all operators are deterministic so there is a unique population at each generation. The formalism is therefore mathematically exact in this limit. However, our interest will be in considering how the infinite population provides an approximation to the dynamics of a finite population. We will see that the infinite population approximation is not always accurate.

In the statistical mechanics approach cumulants are used to model the dynamics of the ensemble. This provides an efficient representation of the population and thus gives an accurate model of the dynamics with relatively few variables. However, the cumulants have another property which is even more important to their success in modelling the dynamics of EAs, namely that the ensemble average is a typical representative of the ensemble. This property, known in the physics literature as *self-averaging*, is not shared by either the moments or the occupation probabilities for the string states (as used in infinite population models). A consequence of this is that a model that describes the population in terms of string state probabilities will provide a poor approximation of the dynamics for finite populations. This is not because it fails to describe selection well, but because using ensemble averages of the probabilities of string states gives a non-representative member of the population.

To illustrate this subtle observation we consider the evolution of a generational genetic algorithm (GA) undergoing selection and mutation on some very simple problems. Furthermore we will show that although the infinite population approximation is reached in the limit of large populations, the size of the population needed to be close to the limit can often be huge even for simple problems.

2 Modelling Ones-Counting

System Definition

We consider a GA solving the ones-counting problem. That is, the fitness of a string $\boldsymbol{S} = (S_1, S_2, \ldots, S_L)$, where $S_i \in \{0, 1\}$, is equal to

$$F(\boldsymbol{S}) = \sum_{i=1}^{L} S_i.$$

We will consider a generational GA undergoing selection and mutation. We denote the size of the population by P, and label the members of the population as $\boldsymbol{S}_\alpha$, where $\alpha = 1, \ldots, P$. The initial population is assumed to consist of random strings. The fitness of member α we denote by $F_\alpha = F(\boldsymbol{S}_\alpha)$. We consider two types of selection; proportional and a scaled Boltzmann selection. In both case we will use Roulette wheel sampling. The probability of selecting a member of the population α is

$$p_\alpha = \frac{w(F_\alpha)}{Z}, \qquad Z = \sum_{\alpha'=1}^{P} w(F_{\alpha'})$$

where: $w(F_\alpha) = F_\alpha$ for proportional selection; and $w(F_\alpha) = \exp(\beta F_\alpha/\sigma)$ for scaled Boltzmann selection, where σ is the standard deviation of the distribution of fitnesses and β controls the strength of selection. In the mutation step we independently mutate the allele at site i with a probability $u = U/L$, so that on average we change the alleles at U sites.

We could also model crossover although to do so accurately requires that we keep track of the average correlation in the population (Prügel-Bennett, 2001; Rattray, 1996; Rattray and Shapiro, 1996). This has some advantage in the statistical mechanics formalism as it reduces the higher cumulants. However, it adds nothing to the argument presented in this paper so we will assume that there is no crossover.

Modelling Procedure

Under selection and mutation the only quantities relevant to the dynamics are the fitnesses of the string, F_α. This reduces the number of states we need to consider from 2^L to $L+1$. We can model the GA described above by taking a set of quantities, Ξ, describing a typical population and working out how these quantities change under mutation and selection

$$\Xi(t) \overset{selection}{\longrightarrow} \Xi^s(t) \overset{mutation}{\longrightarrow} \Xi^{sm}(t) = \Xi(t+1).$$

We will consider two types of quantities for describing a typical population

Cumulants where $\Xi = \{K_1, K_2, K_3, K_4\}$ and K_i are the cumulants describing the distributions of fitnesses in the population

String state probabilities where $\Xi = \{\rho(F)|F = 0, \ldots, L\}$ and $\rho(F)$ gives the proportion of the population with fitness F.

We note that the moments can be constructed from the state probabilities since

$$\mu_n = \sum_{F=0}^{L} \rho(F)\, F^n.$$

Naively, one may think that we can go from moments to cumulants so that the cumulant description is an approximation to the string state probability description. However, we must remember that we are trying to model an ensemble by considering the evolution of a typical population of the ensemble. We find that typical population by taking the ensemble average. We cannot compute the ensemble average of the cumulants from the ensemble average of the moments since they are non-linearly related (except for the first moment which equals the first cumulant). In contrast, the state probabilities and moments are linearly related so that the ensemble average of the moments are identical to the moments of the ensemble averaged probability. In the infinite population limit there is a unique probability distribution so the state probabilities, momentum and cumulant description are all equivalent.

Modelling Finite Populations

We can finesse the problem of modelling mutation on a finite population by considering the evolution of the GA to takes place as follows

1. We start from some probability distribution for the fitnesses in the population
2. We draw a finite sample independently from this probability distribution
3. Each member of the finite sample produces a large (infinite) number of seeds in proportion to its fitness
4. We mutate the seeds (an infinite population) to give us a probability distribution for the fitnesses
5. Return to 2.

This algorithm gives identical behaviour on average to the original GA. We have added an additional procedure, step 2, which involves sampling from a probability distribution. However, selecting for fitter individuals, step 3, and mutation, step 4, are now deterministic operations and can be computed exactly. In the infinite population approximation we miss out step 2. We can view step 2 as a correction to the infinite population approximation.

Each time we draw a finite sample from a probability distribution we are likely to obtain a different finite population. Thus we obtain an ensemble of distributions. The key approximation that we are going to make in modelling the GA dynamics is to replace the ensemble by the ensemble average. That is, we keep a single distribution which we hope is a good representative of the ensemble. The effect of selection is usually a non-linear function of the cumulants or probability distribution so that fluctuations from the average will lead to systematic corrections (an exception, however, occurs in the 'diffusion limit' when selection is linear in both the cumulants and moments (Rattray and Shapiro, 2001)). When modelling with cumulants these corrections are small (and can be computed (Prügel-Bennett, 1997)). However, we will see that this is not the case when using string state probabilities.

Sampling from a Distribution

The probability of drawing a sample of P individuals from a probability distribution $\rho(F)$, so that there are $n(F)$ members in state F is given by the multinomial distribution

$$\Pr(\{n(F)\}) = P! \prod_{F=0}^{L} \frac{\rho(F)^{n(F)}}{n(F)!} \left[\sum_{F=1}^{L} n(F) = P \right]$$

where we use the convention $\left[\mathit{predicate} \right]$ is equal to 1 if *predicate* is true or 0 otherwise. We can compute the average distribution after selection as

$$\rho^s(F) = \left\langle \frac{n(F)\, w(F)}{\sum_{F'} n(F')\, w(F')} \right\rangle_{\{n(f)\}} \tag{1}$$

where $\left\langle \cdots \right\rangle_{\{n(f)\}}$ denotes the average over all ways of drawing a finite distribution from $\rho(F)$. We can compute the average in equation (1) exactly by exploiting a couple of identities

$$\begin{aligned}
\rho^s(F) &= \left\langle n(F)\, w(F) \int_0^\infty \mathrm{e}^{-t \sum_{F'} n(F')\, w(f')} \mathrm{d}t \right\rangle_{\{n(f)\}} \\
&= \int_0^\infty \frac{\partial}{\partial z(F)} \left\langle \mathrm{e}^{\sum_{F'} n(F')\, w(F')(z(F') - t)} \right\rangle_{\{n(F)\}} \Bigg|_{z(F)=0} \mathrm{d}t \\
&= \int_0^\infty \frac{\partial}{\partial z(F)} \left(\sum_{F'} \rho(F') \mathrm{e}^{w(F')(z(F') - t)} \right)^P \Bigg|_{z(F)=0} \mathrm{d}t \\
&= P \rho(F)\, w(F) \int_0^\infty \mathrm{e}^{-tw(F)} \left(\sum_{F'} \rho(F') \mathrm{e}^{-tw(F')} \right)^{P-1} \mathrm{d}t.
\end{aligned}$$

The remaining integral can be computed numerically to very high accuracy.

A similar approach is used to compute the average of cumulants after selection. In this case we consider the cumulant generating function

$$G(\beta) = \log \left(\sum_F n(F)\, \mathrm{e}^{\beta F} \right).$$

The cumulants after selection are obtained by taking derivatives of $G(\beta)$, i.e. $K_n^s = G^{(n)}(\beta)$ We can use an integral representation of the logarithm to compute the average cumulants after selection in an analogous way to how we computed the average state probabilities. The details of this calculation have been given many times before (see, for example, (Prügel-Bennett, 1997; Prügel-Bennett and Shapiro, 1994)) and exploits a trick first used in reference (Derrida, 1984). In the weak selection limit we can obtain an expansion for the average cumulants after selection in terms of the cumulants of our typical population before selection. However, for strong selection (the situation we consider in this paper) we need to construct the occupation probabilities $\rho(F)$ from our cumulants. However, as we keep only a few cumulants the distribution is under-determined. In previous papers we have used the Gram-Charlier expansion to obtain an approximation for this distribution (Prügel-Bennett, 1997; Prügel-Bennett and Shapiro, 1997). This seems to work well although it does not guarantee that the distribution is positive everywhere. In this paper we use a maximum entropy approach which gives a more principled method for converting from a cumulant expansion to a distribution (see appendix A).

It might seem that there is little difference between the cumulant model and the string state probability model since in the cumulant model we generate the string state probabilities before doing selection. However, the difference is that we compute the average cumulants after selection in the cumulant model rather than the average string state probabilities. Using these integral representations we can compute the ensemble average after sampling (i.e. the average over the multinomial) exactly for both the cumulants and the state probabilities. Since we can calculate the effect of mutation exactly we can compare the two models for the ensemble, one using the average cumulants and one using the average string state probabilities.

3 Comparison with Simulations

Figures 1 shows the predictions made using the cumulants and the state probabilities for ones-counting with $L = 100$ for a GA with population size $P = 100$ and mutation rate $U = 1$. Figure 1(a) shows the result for proportional selection and figure 1(b) shows the result for Boltzmann selection with a rate $\beta = 1$. Also shown are simulation results for the same system averaged over 10 000 runs (error bars are shown, but appear to have zero width on the scale shown). In addition, we show the infinite population result. It is evident that all predictions have some errors for the intermediate dynamics, although the predictions using the cumulants are considerably better than those using the state probabilities. The errors using cumulants arise because we are only using four cumulants. As there is no crossover the higher cumulants become quite substantial over time and give rise to a significant correction to the predictions using just four cumulants. The finite size corrections for the string state probabilities give only a small improvement over the infinite population result and give a very poor estimate of the finite population dynamics.

Why are the predictions so different?

The poor predictions when using the full probability distribution rather than the cumulants is rather surprising. The probability distribution contains more information that the cumulants and can capture information about all cumulants. The only difference in our modelling procedure is that to compute the typical population after selection we are averaging the probability distribution rather than the cumulant. However, this is crucial—

(a)

(b)

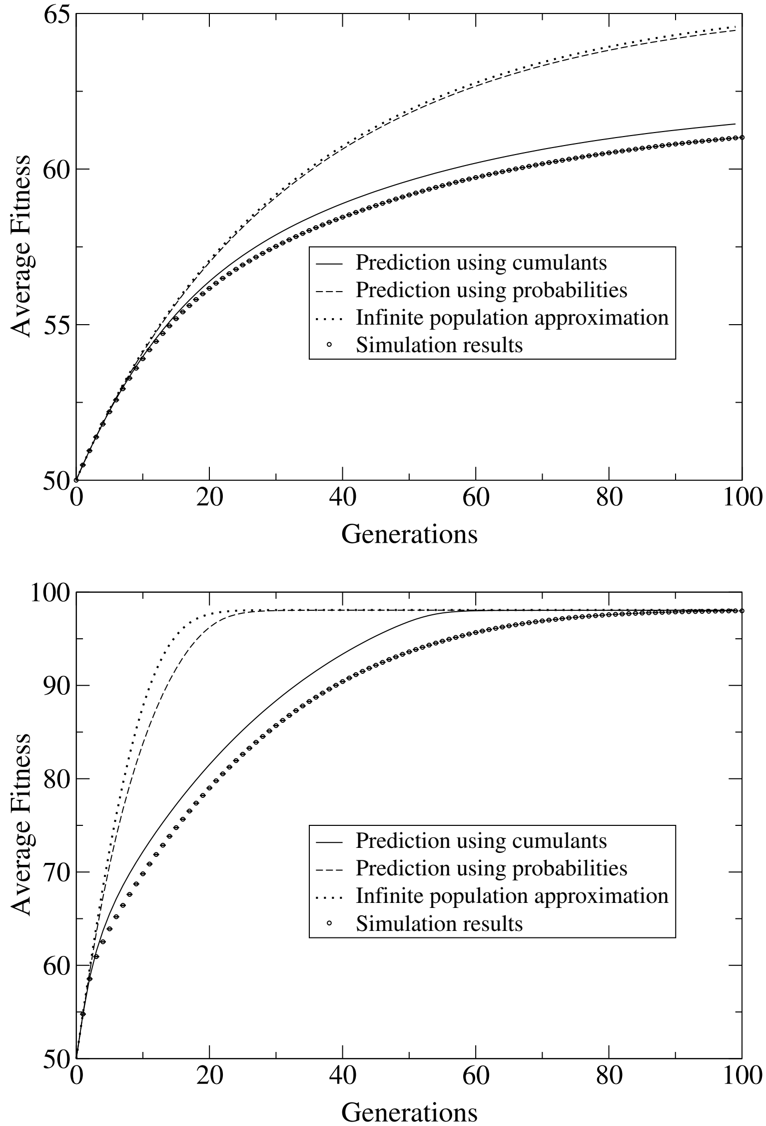

Figure 1 Predictions for the evolution of a GA on the ones-counting problem using cumulants and state probabilities. Graph (a) is for proportional selection and graph (b) is for scaled Boltzmann selection with a selection rate of $\beta = 1.0$. The results shown are for a string length $L = 100$, with a population of size $P = 100$, and a mutation rate of $U = 1$. Also shown are simulation results averaged over 10 000 runs and the infinite population result.

averaging cumulants gives a typical member of an ensemble while averaging probability distributions does not.

It is easy to understand how different measures lead to different averages by considering a simple example. Figure 2 illustrates two ways to find a typical member of an ensemble. We assume we have an ensemble of two distributions which we want to represent by a single distribution. We can generate this 'typical' distribution by finding the average cumulants of the distribution and then create a distribution with these average cumulants or we can average the distribution directly. We get different answers either way. Neither is correct, as we are throwing away information. However, when we average the distributions directly we obtain a much wider distribution than either of the original distributions. We face the same problem when modelling the GA dynamics. Averaging the distributions directly gives raise to a very wide distribution that is not a typical member of the ensemble.

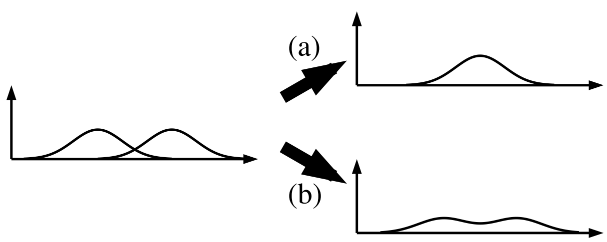

Figure 2 Obtaining a typical distributions by (a) averaging the cumulants and (b) averaging the distributions.

When modelling the dynamics of an ensemble using cumulants we can envisage each population in the ensemble as a point in cumulant space. The ensemble consists of a cloud of points which shifts at each generation. We illustrate this schematically in figure 3. In cumulant space the cloud is compact with the centre of the cloud representing a typical member of the ensemble. We can obtain a more accurate description of the dynamics by averaging over the cloud (this corresponds to averaging over the sample-to-sample fluctuations). Extending the calculation to obtain these corrections is described in reference (Prügel-Bennett, 1997).

In the physics of disordered systems, quantities whose average values are typical members of an ensemble are known as self-averaging. The free energy (which is the cumulant generating function) is usually found to be a self-averaging quantity while the partition function (which is the moment generating function) is often not. This seems to be the case here. The cumulants are self-averaging while the probability distributions and moments are not. In the space of probability distributions the cloud of points representing an ensemble does not form a dense cloud, but is widely distributed around the space. Its average is not a representative distribution of the ensemble. We can easily understand this when we remember that the space of probability distributions is represented by points on a simplex. Finite populations will lie on the edges of the simplex (since many states will not be represented in the population), while the average distribution will lie in the interior of the simplex.

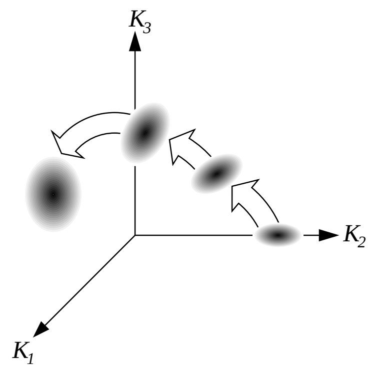

Figure 3 Schematic illustration of the evolution of an ensemble of GA runs in cumulant space

A simple thought experiment helps to illustrate the problem that can arise in averaging probability distributions rather than cumulants. We consider a population that undergoes selection only and we use an infinite selection pressure so that we select only the fittest individual in the population. We know that the dynamics finishes at the first generation and that all members of the population have the fitness of the fittest individual. However, let us consider modelling this using cumulants and the probability of string states. In either case after selection all populations in the ensemble will consist of a spike at the fittest member. The position of the spike will differ for different members of the ensemble as it depends on the fittest member in the particular initial population. If we use a cumulant description then the average first cumulant will be at the average of the fittest individuals in each population in the ensemble, while the second cumulant will be zero for all members of the ensemble so the average will be zero. Thus the cumulant average is indeed a typical member of the ensemble. In contrast, if we average the distributions we are averaging a set of peaks which have a distribution of positions. Consequently, in this case we obtain a distribution with a finite width—a distribution that does not occur in any member of the ensemble. In fact, if we use the distribution to model the ensemble the population would not converge until the global optimum is reached. This is a situation we know does not occur.

4 Reaching the Infinite Population Limit

Infinite populations models are widely used in modelling EAs so it is important to understand how accurate they are in modelling finite populations. As we saw in the previous section infinite population models give poor predictions of the dynamics for small populations. How large does the population need to be for the infinite population to be a good approximation to the actual dynamics? We can answer this empirically. For models such as ones-counting without crossover it is possible to simulate very large populations very quickly (see appendix B). In figure 4, we show the evolution of a GA for the ones-counting problem, with $L = 100$, and a mutation rate $U = 1$, for different sizes of

(a)

(b)

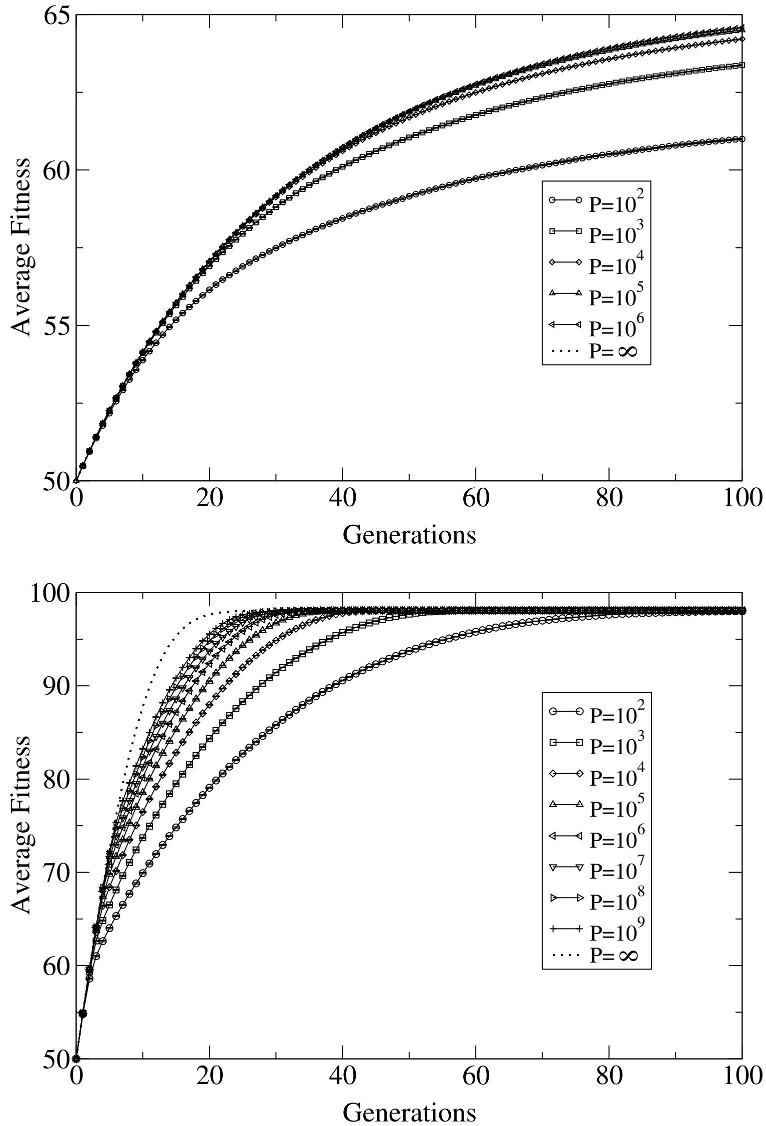

Figure 4 Evolution of the GA for the ones-counting problem with $L = 100$. The average fitness is shown as a function of the number of generations when using (a) proportional selection and (b) scaled Boltzmann selection with $\beta = 1$. In both cases the mutation rate is $U = 1$. The graphs are computed empirically and averaged over 10 000 runs. In graph (a) the data points for $P = 10^5$, $P = 10^6$ and the infinite population result lies on top each other.

population. Figure 4(a) shows the result for proportional selection while figure 4(b) shows the result for Boltzmann selection with $\beta = 1.0$.

In the case of the relative weak proportional selection, the infinite population limit is reached for populations of the order of 10^5, however this is not the case when we use stronger selection schemes, which are often used in real EAs. For Boltzmann selection, we get a significant discrepancy from the infinite population limit even when using population sizes as large as 10^9. Notice that this is for a simple problem with no local optima.

The infinite population model clearly gives an unreliable estimate for the dynamics of a GA. However, in both cases shown here the infinite population approximation appears to give a good estimate for the long time behaviour. Is it possible that this is a generic property? Unfortunately, this is not the case as a simple counter-example shows. We consider a modified ones-counting problem where the fitness is a function of the Hamming distance from the all ones state. We denote this Hamming distance by H. We consider a 'trap function' as shown in figure 5, below. The trap function consists of the usual ones-counting fitness function except for a fit region when there are less than 10 ones in the string.

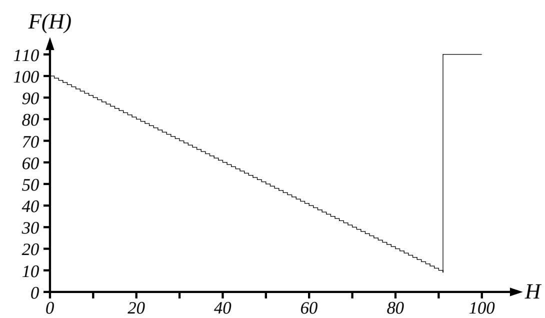

Figure 5 Trap function used to illustrate problems in using the infinite population approximation.

The trap function is a dramatic example of a fitness functions where the search is dominated by the dynamics. Figure 6 shows the evolution of a population consisting on 10^9 individuals and that for an infinite population. In the finite population case the population moves towards the all ones state (i.e. $H = 0$). For the population to reach the global optimum a member of the population must undergo a mutation of 90 or more bits—the probability of this happening is of the order of 10^{-170}. Such a mutation will never be observed. Thus the long time behaviour of the infinite population model is irrelevant to the observable long time behaviour of the finite population.

The trap function has been constructed to illustrate that the infinite population model can give very misleading information about finite models, but this behaviour is also to be expected for most hard optimisation problems. Finding the global optimum for a difficult instance of an NP-hard problem takes a time that grows exponentially in the size of the problem (unless $P = NP$). In an infinite population model with a sufficiently strong selection strength, the global optimum will give a significant contribution to the equilibrium probability solution. However, for a finite population, a GA will not reach the

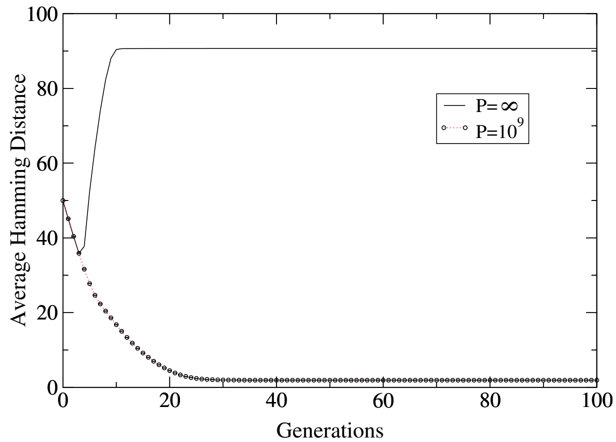

Figure 6 Evolution of a GA on the trap function shown in figure 5. The data points show the average fitness against number of generations for a GA with a population of size 10^9, a mutation rate of $U = 1$ and using scaled Boltzmann selection with $\beta = 1.0$. The results were averaged over 1 000 runs. Also shown is the infinite population result (solid line).

global solution after any plausible number of generations. Thus, once again, the long time behaviour of an infinite population model may tell us little about the observable long time behaviour of a real GAs on difficult problems.

5 Conclusion

The success of a model for an EA can depend critically on the statistics used to describe the system. In this paper, we have used two sets of statistics to model the evolution of a GA on the ones-counting problem. In both cases, the statistics describe the distribution of fitness for a typical or representative population of the ensemble. One set of statistics we have used are the string state probabilies while the other set of statistics are the first few cumulants describing the probability distribution. The cumulant model gives a much more accurate model of the dynamics than using the complete distribution. This may seem surprising as the cumulant model contains much less information than the complete distribution. However, both models are approximations. A complete model should describe the evolution of a ensemble of populations. In cumulant space the ensemble of populations forms a compact set whose average is a typical member of the population. In contrast, in distribution space the ensemble of populations lie on the edges of a simplex and the average of this ensemble does not correspond to a typical member of the population. Using cumulants we have a prescription for obtaining a typical member of the ensemble—we just take the ensemble average. We cannot, however, do this using the probability distribution.

The difficulty arises not because we are using a probability distribution to represent a typical population in our ensemble (after all, we construct a probability distribution from the cumulants in order to compute selection). Rather, the difficulty arises because we need a prescription for obtaining a representative member of the ensemble after selection. In the cumulant model we obtain a representative member of the ensemble by taking the average over the ensemble. This does not work when using the probability distributions themselves.

This paper is deliberately polemic. We have attempted to emphasise the point that, although the infinite population limit might be mathematical exact, it can give substantially worse predictions of the behaviour of real evolutionary algorithms than a clearly approximate approach such as the statistical mechanics formalism. Furthermore, the rather laboured trap function example illustrates that the infinite population limit can sometimes be totally irrelevant to the observable long time behaviour. Thus, although infinite population results are attractive because they are mathematically exact, it can often be more informative to find good approximations that capture the behaviour of real EAs. In this paper, we chose a very simple class of problems to illustrate the pitfalls of using an infinite population model. However, these pitfalls are likely to occur in many situations where we are modelling an ensemble of evolving populations. This is not to say that infinite populations models are always inappropriate. (The author has come across problems where the finite population corrections are irrelevant for understanding the behaviour of an EA, however, in a surprising number of cases finite population effects do seem to be crucial in understanding the behaviour of EAs.) However, the relevance or irrelevance of finite population effects needs to be established either empirically or theoretically. Finally, I wish to make it clear that I am not advocating that the statistical mechanics formalism is the only way to model EAs, but, when we attempt to understand these complicated dynamical systems approximations that capture the underlying behaviour of the system can be more helpful than mathematically exact analysis of limits which might be irrelevant.

A Maximum Entropy

To calculate the effect of selection we need to know the probability distribution for our typical population before selection. In the statistical mechanics approach, however, we only know the first few cumulants (in the example used in this paper we know the first four cumulants). It is therefore necessary to infer the distribution using some inference procedure. One principled way to do this is to use the maximum entropy principle. We can view this as choosing the most likely distribution. To understand this we note that the number of string states with fitness F is $\binom{L}{F}$, so that the probability of a member of the population being in state F given no other information is

$$\rho_0(F) = \frac{1}{2^L}\binom{L}{F}.$$

Now the probability of a population having $n(F)$ individuals with state $n(F)$ is given by

$$\Pr(\{n(F)\}) = P! \prod_{F=0}^{L} \frac{\rho_0(F)^{n(F)}}{n(F)!} = \mathrm{e}^{\,S(\{n(F)\})}$$

where

$$S(\{n(F)\}) = \log(P!) + \sum_{F=0}^{L} n(F)\log\big(\rho_0(F)\big) - \log\big(n(F)!\big).$$

Using Stirling's approximation $\log(n!) \approx n\log(n)$ we obtain the 'entropy'

$$S(\{\rho(F)\}) \approx -P \sum_{F=0}^{L} \rho(F) \log\left(\frac{\rho(F)}{\rho_0(F)}\right).$$

where $\rho(F) = n(F)/P$. The most probable distribution is given by maximising $\Pr(\{n(F)\})$ or equivalently $S(\{\rho(F)\})$ with respect to $\rho(F)$ under the constraints

$$\sum_{F=0}^{L} \rho(F)\, F^n = \mu_n$$

where μ_n are the moments corresponding to the known cumulants of the distribution (and using $\mu_0 = 0$). To find the most likely distribution we therefore maximise the Lagrangian

$$\mathcal{L}(\{\rho(F)\}) = -\sum_{F=0}^{L} \rho(F) \log\left(\frac{\rho(F)}{\rho_0(F)}\right) - \sum_{n=0}^{4} \lambda_n \left(\sum_{F=0}^{L} \rho(F)\, F^n - \mu_n\right)$$

with respect to $\rho(F)$ and the Lagrange multipliers λ_n. Doing this, we find

$$\rho(F) = \rho_0(F)\, \mathrm{e}^{-\sum_{n=0}^{4} \lambda_n F^n - 1}$$

where we have to choose λ_n so that $\rho(F)$ is properly normalised and has the correct first four cumulants.

In practice, it is more convenient use a change of variables. Instead of using the Lagrange multipliers λ_n we find the parameter a_n such that

$$\rho(F) = \rho_0(F)\mathrm{e}^{a_0 + a_1 x + a_2 x^2 + a_3(x^3 - x) + a_4(x^4 - 6x^2 + 3)}$$

where $x = (F - K_1)\sqrt{K_2}$ where K_n is the n^{th} cumulant whose distribution we are trying to find. We can use a gradient search to find the parameter a_n which gives the maximum entropy distribution with the correct first four cumulants. The gradients with respect to the parameters a_n are considerably smaller than those with respect to the original Lagrange multipliers λ_n.

B Simulating Large Populations

For problems whose fitness depend only on the Hamming distance from some reference state it is possible to simulate an asexual GA with large populations very efficiently. This is because we only need to know the occupation number for the Hamming states H. That is, we can completely define the population by a set of $L+1$ numbers $n(H)$ giving the number of individuals in the population with Hamming distance H from a reference string. By assumption the fitness of the state is a function of the Hamming distance $F(H)$ and the probability that an individual is selected is given proportional to $w(F(H))$. The probability of obtaining $n'(H)$ individuals after selection is given by the multinomial

$$\Pr\big(\{n'(H)\}\big) = P! \prod_{H=0}^{L} \frac{p_s(H)^{n'(H)}}{n'(H)!} \left[\sum_{H=0}^{L} n'(H) = P\right]$$

where

$$p_s(H) = \frac{n(H)\, w(F(H))}{\sum_{H'=0}^{L} n(H')\, w(F(H'))}$$

is the probability of selecting an individual with Hamming distance H. To simulate selection we therefore need only generate the random deviates $n'(F)$ from the multinomial distribution given above. Generating n deviates from a multinomial distribution involves generating $n-1$ binomial deviates(Fishman, 1996; Press et al., 1992).

To simulate mutation we note that this can again be done by generating appropriate multinomial deviates. The probability that $n_H(H')$ individuals end up at state H' assuming that there were initially $n(H)$ at state H is given by

$$\Pr\bigl(\{n_H(H')\}\bigr) = P! \prod_{H=0}^{L} \frac{W(H,H')^{n_H(H')}}{n_H(H')!} \left[\sum_{H'=0}^{L} n_H(H') = n(H) \right]$$

where $W(H,H')$ is the probability of a mutation form H to H'. We can precompute these mutations. Furthermore, by ordering states H' so that the more likely mutations appear first, we can reduce the number of binomial deviates we need to generate (Fishman, 1996).

In fact, if we simulate the dynamics described in the *Modelling Finite Populations* part of section 2 we can calculate treat selection and mutation deterministically and need only generate multinomial deviates for performing the sampling step.

References

Derrida, B. (1984). Random-energy model: An exactly solvable model of disordered systems. *Physical Review: B 24*, 2613–2626.

Fishman, G. S. (1996). *Monte Carlo: Concepts, Algorithms and Applications*. Operation Research. New York: Springer.

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

Press, W. H., B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling (1992). *Numerical Recipes in C: The Art of Scientific Computation* (2nd ed.). Cambridge, UK.: Cambridge University Press.

Prügel-Bennett, A. (1997). Modelling evolving populations. *Journal of Theoretical Biology 185*, 81–95.

Prügel-Bennett, A. (2001). Modelling crossover induced linkage in genetic algorithms. *IEEE Transactions on Evolutionary Computation 5*(4), 376–387.

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

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

Rattray, M. (1996). *Modelling the Dynamics of Genetic Algorithms using Statistical Mechanics*. Ph. D. thesis, University of Manchester, UK.

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

Rattray, M. and J. L. Shapiro (2001). Cumulant dynamics of a population under multiplicative selection, mutation and drift. *Journal of Theoretical Population Biology 60*(1), 17–32.

Rudolph, G. (1998). Finite markov chain results in evolutionary computation: A tour d'horizon. *Fundamenta Informaticae 35*(1–4), 67–89.

Schmitt, L. M. (2001). Theory of genetic aglorithms. *Theoretical Computer Science 256*(1–2), 1–61.

Vose, M. D. (1999). *The Simple Genetic Algorithm: Foundations and Theory.* Cambridge, Mass.: MIT Press.

Vose, M. D. and G. E. Liepins (1991). Punctuated equilibria in genetic search. *Complex Systems 5*, 31–44.