Statistical Dynamics of the Royal Road Genetic Algorithm

Erik van Nimwegen,[†]
James P. Crutchfield,[†‡] **and**
Melanie Mitchell[†]
[†]**Santa Fe Institute, 1399 Hyde Park Road, Santa Fe, NM 87501**
[‡]**Physics Department, University of California, Berkeley, CA 94720-7300**

To appear in **Theoretical Computer Science** (1998).

Abstract.

Metastability is a common phenomenon. Many evolutionary processes, both natural and artificial, alternate between periods of stasis and brief periods of rapid change in their behavior. In this paper an analytical model for the dynamics of a mutation-only genetic algorithm (GA) is introduced that identifies a new and general mechanism causing metastability in evolutionary dynamics. The GA's population dynamics is described in terms of flows in the space of fitness distributions. The trajectories through fitness distribution space are derived in closed form in the limit of infinite populations. We then show how finite populations induce metastability, even in regions where fitness does not exhibit a local optimum. In particular, the model predicts the occurrence of "fitness epochs"—periods of stasis in population fitness distributions—at finite population size and identifies the locations of these fitness epochs with the flow's hyperbolic fixed points. This enables exact predictions of the metastable fitness distributions during the fitness epochs, as well as giving insight into the nature of the periods of stasis and the innovations between them. All these results are obtained as closed-form expressions in terms of the GA's parameters.

An analysis of the Jacobian matrices in the neighborhood of an epoch's metastable fitness distribution allows for the calculation of its stable and unstable manifold dimensions and so reveals the state space's topological structure. More general quantitative features of the dynamics—fitness fluctuation amplitudes, epoch stability, and speed of the innovations—are also determined from the Jacobian eigenvalues. The analysis shows how quantitative predictions for a range of dynamical behaviors, that are specific to the finite population dynamics, can be derived from the solution of the infinite population dynamics. The theoretical predictions are shown to agree very well with statistics from GA simulations. We also discuss the connections of our results with those from population genetics and molecular evolution theory.

Contents

1. Epochal Evolution

Metastability is a commonly observed phenomenon in many population-based dynamical systems. In such systems—including evolutionary search algorithms, models of biological evolution, and ecological and sociological systems—the state of a population is often described as the distribution of certain features of interest over the population. A commonly observed qualitative behavior is that the distribution of these features alternates between periods of stasis and sudden change. For extended periods of time the system seems to stabilize on some feature distribution, which is then disrupted by a brief burst of change. We call this type of behavior "epochal evolution", where the term "epoch" denotes an extended period of apparent stability. We use the term "innovation" to refer to the sudden change between epochs. Such behavior, often referred to as "punctuated equilibria", has been reported in many applications of evolutionary search as well as in models of biological and molecular evolution (e.g.,[3, 15, 23, 25, 31, 33, 41]). Epochal behavior has also been observed in natural phenomena such as the outbreak of epidemics [4], the progression of diseases such as cancer [1] and AIDS [5] in individuals, rapid large-scale ecological changes, and the sudden rise and fall of cultures. In natural systems and in many models, epochal behavior is undoubtedly the result of a complicated and poorly understood web of mechanisms.

In this paper we identify the mechanism that underlies the occurrence of epochal behavior in a simplified mutation-only genetic algorithm. In the mechanism's most general form, metastability is induced in an area of state space where the local "flow" of the dynamics is small compared to a scale set by the population's finite size. The dynamics becomes too weak to drive changes in the finite population. More specifically, we will see that the metastability can be associated with an "entropic barrier" that the finite population must pass in moving through the "slow" part of state space. Metastability due to such entropic barriers is contrasted here with the more traditional explanation of metastability as being induced by "fitness barriers". In the latter the population stabilizes around a local fitness optimum in sequence space and must pass through a "valley" of lower fitness to find a higher-fitness optimum. We believe that the generality and simplicity of the mechanism for metastability introduced here makes it likely to play a role in the occurrence of epochal dynamics in the more general and complicated cases alluded to above.

1.1 Search and Evolution

Genetic algorithms (GAs) are a class of stochastic search techniques, loosely based on ideas from biological evolution, that have been used successfully for a great variety of different problems (e.g., [2, 10, 16, 20]). However, the mechanisms that control the dynamics of a GA on a given problem are not well understood. GAs are nonlinear population-based dynamical systems. The complicated dynamics exhibited by such systems has been appreciated in the field of mathematical population genetics for decades. On the one hand, these complications make an empirical approach to the question of when and how to use evolutionary search problematic. On the other hand, the lack of a unified theory capable of quantitative predictions in specific situations has rendered the literature largely anecdotal and of limited generality. The work presented in this paper tries to unify and extend theoretical work that has been done in the areas of GA theory, the theory of molecular evolution, and mathematical population genetics. The goal is to obtain a more general and unified understanding of the mechanisms that control the dynamics of GAs and other population-based dynamical systems.

Vose and his colleagues have previously studied GA dynamics by describing the state of a genetic algorithm at a certain time as a vector in a high-dimensional Euclidean space. Each dimension of this space either represents a certain string [45] or the state of the population as a whole [35]. The dynamics is then described by a nonlinear matrix operator acting on this vector to produce the state at the next time step. Although this "microscopic" approach is formally very clear and precise, in practice the huge sizes of these matrices make it impossible to obtain specific quantitative results. In this paper, a genetic matrix operator is constructed that is similar in spirit to the genetic operators discussed in [35] and [45] but that acts on vectors representing fitness distributions only, *averaging out* all other structure of the microscopic state of the population. The operator, therefore, has a much lower dimensionality. This will make quantitative analyses of this operator possible, allowing specific quantitative predictions to be made about the GA's observed behavior.

A more macroscopic theoretical approach was developed by Prügel-Bennett, Shapiro, and Rattray. Their approach uses ideas from statistical mechanics to analyze the dynamics of GAs [38, 39, 40]. Their formalism also focuses on the evolution of fitness distributions, but generally only the *average* evolution of the first few cumulants of the fitness distribution is calculated. This averaging of the dynamics over a large number of runs makes it impossible to describe the epochal structure of the dynamics in which we are interested. The statistical mechanics approach does, however, provide some insights into the roles that the different genetic operators play in the dynamics and shares with our approach the idea of averaging out most microscopic degrees of freedom to obtain a macroscopic description of the dynamics.

Another theoretical framework of relevance is Eigen's theory of molecular evolution [11, 14]. In the simplest form of this theory, one considers a large population of self-replicating molecules in a reaction vessel. Since the total concentration of molecules is kept constant, there is an effective selection for molecules that replicate fast and efficiently. It is assumed that the molecules make errors during replication, thus introducing mutations. The differential equations that describe the change in concentrations of the different molecular types are analogous to the genetic operator equations that we will develop in this paper. Although, in contrast to our mesoscopic approach, they are defined only on the microscopic states of concentrations of individual genotypes. We will explain how some theoretical concepts from molecular evolution theory, such as the quasispecies and the error threshold, generalize to analogous concepts in our description of the GA dynamics.

Finally, the theory of mathematical population genetics has a long history of analyzing the behavior of evolving populations. Many important results were obtained in the 1930s by the trio of Fisher, Wright, and Haldane. In the 1960s Kimura developed a new way of analyzing evolving populations using diffusion equations [27] that were originally developed in the context of statistical physics [19, 30, 37]. We will make use of this type of analysis several times and will show how methods developed in the context of mathematical population genetics bear on the dynamics of GAs as well.

1.2 Organization of the Analysis

Our analysis of epochal evolution in a mutation-only genetic algorithm first appeared in [44]. The present work goes into considerably more depth. Section 2 introduces the simplified GA used. In section 3 we present an overview of the wide range of different dynamical behaviors that our simple GA exhibits. We discuss the qualitative features of these different dynamical behaviors and pose ourselves a set of questions that we would like to answer using our theoretical model.

The bulk of the remainder is devoted to the development and analysis of this theoretical model. We have termed our type of analysis "statistical dynamics", since it combines a dynamical systems approach, on the one hand, with a statistical physics and stochastic process approach, on the other. The infinite population behavior is treated as the dynamics of a deterministic nonlinear dynamical system. In constructing this dynamical system we have to choose suitable "mesoscopic" state variables (in this case, fitness distributions) that capture the complicated microscopic state of the system in terms of a much lower dimension state space. Moreover, we require that the description of the system and its behavior in terms of these variables should be closed in the limit of infinite populations. That is, for infinite populations we assume that the dynamics of fitness distributions is fully specified by the fitness distributions themselves and does not depend on the exact underlying ("microscopic") distribution of genotypes. This condensation of the microscopic states using a few "order parameters", that describe the dynamics in the limit of infinite system size, is a well-known procedure from statistical physics. With this setting established, we augment the solution of the nonlinear dynamical system with a statistical treatment of the finite population behavior. In doing so, we make use of simple stochastic differential equations, such as the Fokker-Planck equation. These three features—describing the system in terms of a small set of statistical order parameters, deriving and solving the deterministic nonlinear dynamical systems equations in the infinite population limit, and then augmenting this solution with simple stochastic differential equations to capture the finite-population dynamics—is the essence of our statistical dynamics approach.

In section 4 we introduce the state space of fitness distributions in terms of which the GA's dynamics is defined, as well as motivate the use of this particular state space. Section 5 develops an exact solution of the dynamics in this state space in the limit of infinite populations. Specifically, we solve in closed form for the trajectory through fitness distribution space that is followed by an infinite population and analytically characterize the asymptotic behavior of the dynamics. Section 6 is concerned with the finite-population dynamics and presents the main results. This section builds on the results from section 5 to quantitatively analyze a wide range of dynamical features that derive from the finite-population dynamics. In particular, we identify the mechanism that leads to the fitness epochs, solve for their locations in fitness distribution space, and show that the fitness levels of the epochs are left unaltered under the introduction of genetic crossover. We then calculate the stable and unstable manifold dimensions of the metastable epochs, the fitness fluctuation amplitudes during the epochs, the speed of innovations between epochs, and the stability of the epochs. All these results are obtained analytically as functions of the model parameters and are shown to agree with statistics estimated from our GA simulations. Major players in the derivation of the results of section 6 are the eigenvalues and eigenvectors of the Jacobian of the generation operator that describes the dynamics in the limit of infinite populations. Section 7 discusses the average durations of the epochs and describes how the model breaks down in predicting these average durations. Section 8 discusses the results of our paper and looks ahead to future work.

2. A Simple Genetic Algorithm on the Royal Road Fitness Function

2.1 The Fitness Function

The Royal Road fitness functions assign a fitness $f(s)$ to a string s as the sum of fitness contributions f_i from N different nonoverlapping bit sets ("blocks") s_i of s. We will consider bit strings s of length $L = NK$, each of which can be thought to consist of N blocks of length K.

$$\underbrace{\overbrace{101\cdots011}^{K}\,101100010\cdots10100111001}_{NK}$$

For each block of length K there is a particular desired bit configuration (schema). In the above illustration we took the blocks to be sets of K contiguous bits, but it's easy to see that the dynamics of a mutation-only GA is invariant under random permutations of the bits in the string representation. Formally, we have

$$f(s) = \sum_{i=1}^{N} f_i \delta_{s_i, x_i}, \tag{2.1}$$

where the x_i are desired configurations for each block s_i of s and $\delta_{s_i,x_i} = 1$ if and only if $x_i = s_i$, otherwise $\delta_{s_i,x_i} = 0$. A block s_i that exhibits the desired configuration x_i will be called an "aligned" block and blocks that do not exhibit the desired configuration will be called "unaligned". Without loss of generality, this desired configuration can be taken to be the configuration of K 1s: $x_i = 1^K$. The f_i are the fitness contributions from each aligned block i. For simplicity we shall take all f_i to be equal: $f_i = 1$. The fitness of s can then be simply defined as the number of aligned blocks in s. Thus $0 \leq f(s) \leq N$. The number of blocks N and the number of bits per block K are parameters of the fitness function.

Royal Road fitness functions were initially designed to address questions about the processing and recombination of schemata in genetic algorithms. They were thought to lay out a "royal road" for genetic algorithm search [34] and so could test a GA's ability to preferentially propagate useful genetic "building blocks". The Royal Road functions defined in [34] are more general than the ones we are considering in this paper. For instance, the fitness $f(s)$ of a string s does not in general need to be a simple *sum* of the fitnesses f_i of the aligned blocks. Here we will not be concerned with the issues of schemata processing and the building block hypothesis. We use the simple Royal Road fitness functions defined above, because they are simple enough to be analyzed and because the GA's behavior on these fitness functions exhibits a range of qualitatively distinct epochal dynamics.

2.2 The Genetic Algorithm

In our study, we use the following mutation-only genetic algorithm:

1. Generate a population of M strings of length $L = NK$, chosen with uniform probability from the space of all L-bit strings.

2. Evaluate the fitness $f(s)$ of each string s in the population.

3. Create a new population of M strings by choosing strings from the current population with replacement and with probability proportional to fitness. This is sometimes called fitness-proportionate selection.

4. Mutate (i.e., change) each site value in all strings with a fixed probability q.

5. Go to step 2.

As noted before, this algorithm does not include crossover, a genetic operator purported to aid in the evolutionary propagation of important genetic building blocks. Crossover is left out of this first work for two reasons. First, it considerably simplifies the analysis of the GA. Second, the main qualitative features of the dynamics—the occurrence of fitness epochs and innovations between them—are not changed by leaving out crossover. We will address in more detail the effects of crossover on this GA in section 6.5 and show that the fitness levels of the epochs are not changed by including crossover.

This defines our Royal Road GA. It has four control parameters: the number N of blocks, the number K of bits in a block, the mutation rate q, and the population size M.

3. Observed Behavior of the Royal Road GA

In general, the behavior of an evolving population is governed by a complicated interplay of a few major genetic and populational forces—selection, mutation, crossover (if present), and genetic drift.

Selection tends to converge the population onto the current best strings that are found in the population. In this sense selection is installing information in the population: strings in the population become much more similar to one another than a collection of random strings and the entropy of the distribution of strings is therefore decreased. To some extent the bit values that are shared by all members of the population are a reflection of their functionality. For example, under the Royal Road fitness function, once a string with a desired block is discovered, that string is preferentially reproduced and the population converges quickly to strings containing that block. Such convergence reflects the functionality conferred on strings by this block.

Mutation and crossover, for that matter, are largely forces that drive genetic mixing. Information that has been installed in a string is destroyed by mutation randomly flipping bits. At the same time mutation is a force that can provide something *new*: it can create bit strings that were never before present in the population and that might have improved fitness.

A third important force is genetic drift, which is due to the sampling fluctuations induced by the finite size of the population. Genetic drift, which is recognized as a major player in the theory of population genetics, seems to be somewhat neglected in the theory of GAs. For small populations, it turns out that information can be stored in the strings *by accident*. Suppose that there is no mutation or selection. The initial population is chosen at random, so it is likely to contain M different genotypes. At each generation a new population is created by randomly sampling M strings from the old population. At each time step it is likely that certain genotypes will be lost, since they are not sampled, and that other genotypes will multiply. After a number of generations on the order of the population size M, it is likely that there will be only one genotype left in the population, the particular genotype being purely accidental. This process is known as "random genetic drift" and it is one way in which arbitrary information is stored in the string population. Genetic drift plays a major role in the dynamics of GAs with selection and mutation: small populations tend to spontaneously converge, regardless of selection. Note that populations also converge in the presence of crossover. Thus, the only genetic operator capable of prohibiting complete convergence of the population is mutation.

We now present a set of examples of the empirical behavior of our Royal Road GA in order to demonstrate how varied it can be for different parameter settings. This should make it clear that, although we can qualitatively identify the main evolutionary forces, the actual interplay of these forces can be complicated and subtle.

Caption for figure 1: Average fitness (solid lines) and best fitness (diamonds, often appearing together as thick solid lines) in the population over time for nine runs of the Royal Road GA with different parameter settings. Our canonical example of epochal behavior can be found in run 1(d). The parameter settings for this run 1(d) are: $N = 10$ blocks of length $K = 6$, a mutation rate of $q = 0.001$, and a population size of $M = 500$. All other runs were done with parameter settings that can be obtained by consecutive single-parameter changes from the parameters of run 1(d). Note that all runs were done with a fixed string length $NK = 60$. Moving out from run 1(d) by following the large arrows from run to run, a single parameter is changed. The changed parameter and its new setting (increased or decreased according to the thin up or down arrow, resp.) are indicated next to the large arrows. For example, following the large arrows up and down from run 1(d) we arrive at runs 1(a) and 1(g), respectively. As indicated, only the population sizes of run 1(a) and 1(g) differ from the parameters in run 1(d). For run 1(a) a large population size of $M = 5000$ was used and for run 1(g), a small population size of $M = 50$ was used. From the three runs 1(a), 1(d), and 1(g) three arrows point to the three runs 1(b), 1(e), and 1(h) in the middle column. All vary in only one parameter from the settings in the corresponding runs in the left-hand column. Run 1(b) differs from run 1(a) by a decrease of the block size to $K = 3$ and an increase of the number of blocks to $N = 20$. Run 1(e) differs from run 1(d) by an increase of the mutation rate to $q = 0.0075$. And, run 1(h) differs from run 1(g) by an increase of the mutation rate to $q = 0.005$. The three runs in the middle column show how a change in a single parameter with respect to the runs in the left-hand column can make epochal dynamics disappear. The runs in the right-hand column show that a further change in a single parameter can make epochal dynamics reappear. Run 1(c) differs from run 1(b) by a decrease in population size to $M = 300$. Run 1(f) differs from run 1(e) by an increased population size of $M = 7500$. And finally, run 1(i) differs from run 1(h) by an increased block size of $K = 15$. These nine runs demonstrate the variety of dynamical behaviors exhibited by our simple Royal Road GA.

Figures 1(a) through 1(i) show the results of nine runs of the Royal Road GA with different parameter settings. In each, the average fitness in the population over time is shown (solid lines) together with the best fitness in the population at each time step (diamonds). In all runs the total length of the string $L = NK$ was kept constant at $L = 60$ to reduce the number of free parameters. Figure 1(d) is the central parameter setting of our analysis. Following the large arrows, each successive run differs from the predecessor by a single parameter change.

Run 1(d), selected as the canonical example of epochal evolution, was performed with

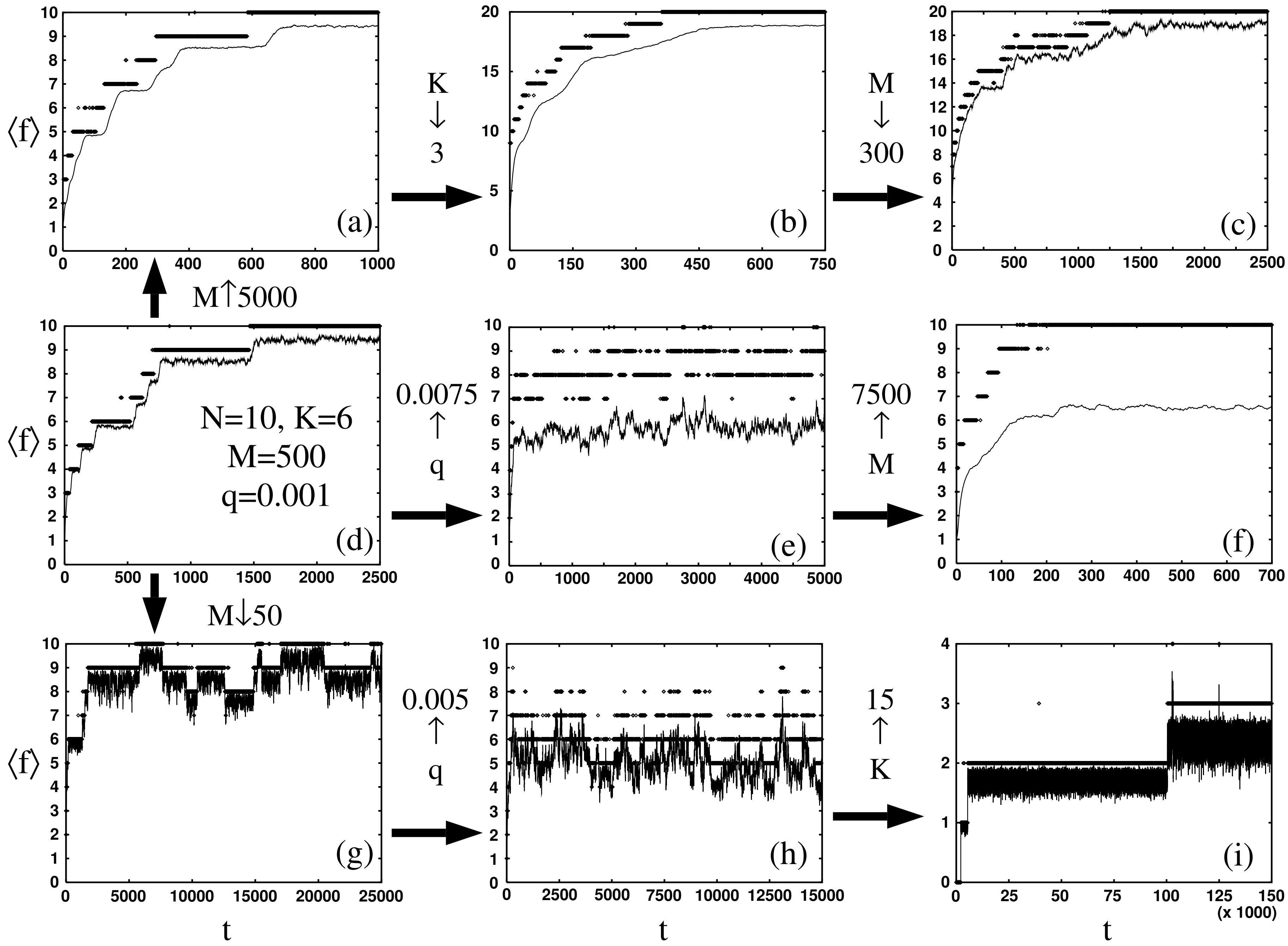

Figure 1: For caption see text.

$N = 10$ blocks of length $K = 6$ bits, a mutation rate of $q = 0.001$, and a population size of $M = 500$. This run clearly shows epochal dynamics in the evolutionary search. This behavior is encountered over a wide range of parameter settings around those of run 1(d). Note that experimentally, the length of the epochs varies greatly from run to run, but the fitness levels at which they occur are the same in each run. The epoch levels therefore constitute a *reproducible* macroscopic feature of the dynamics. Runs 1(a) and 1(g) in the left-hand column show the effect on the search behavior of changing the population size M. Run 1(a) was performed with a relatively large population size of $M = 5000$. The fitness epochs are also clear in this plot. The behavior is less noisy, the search for higher fitness quicker on average, and the innovations between the fitness epochs are less rapid. For the small population size of $M = 50$ in figure 1(g), however, the behavior becomes much noisier and the average fitness in the population shows an interesting intermittent behavior at high fitness levels. The average fitness jumps up and down irregularly between the different fitness epochs. We would like to understand what causes the epochal dynamics in Figures 1(a), 1(d), and 1(g). In particular, we would like to understand how the different population sizes lead to these different classes of search behavior.

The runs shown in the middle column—Figures 1(b), 1(e), and 1(h)—illustrate how change in a single parameter can cause fitness epochs to disappear. In run 1(b), for example, the block length was lowered from $K = 6$ to $K = 3$, otherwise the parameter settings are those of 1(a). Note that since we are keeping string length constant ($L = 60$) throughout the figure mosaic, in run 1(b) there are $N = 20$ blocks. In this plot we see the average fitness increasing strictly monotonically over time. Although well-defined fitness epochs cannot be distinguished anymore, the rate of fitness increase does vary in different regions of the run. Note that the fluctuations in this rate of fitness increase are *not* the same between runs. That is, although the rate of fitness increase is decreasing over time on average, the precise locations of the inflection points in the curve of average fitness against time are not the same between runs.

Run 1(e) has a higher mutation rate ($q = 0.0075$) than that of the canonical run 1(d). Here too the fitness epochs have disappeared from the behavior, but in a markedly different way from that in going from run 1(a) to run 1(b). Overall, fitness jumps up quickly at first after which it moves up and down irregularly in a band of fitnesses between 4.5 and 7. Note also that the best fitness in the population alternates between different values in a fitness band between 7 and 10.

Run 1(h) has a higher mutation rate ($q = 0.005$) than that in run 1(g) ($q = 0.001$). The behavior of run 1(h) is similar to that in run 1(e) although the fitness fluctuations occur over a wider range of amplitudes and seem to have larger variation in the associated time scales. In contrast, in run 1(e) the fitness fluctuates up and down on a time scale that is roughly constant. In run 1(h) fluctuations occur both on short time scales as well as on long time scales. Moreover, a hierarchy of fluctuation amplitude bands can be distinguished. As in the transition from 1(d) to 1(e), we would like to understand why the fitness epochs disappear in going from 1(g) to 1(h) by increasing mutation and what mechanisms cause the different types of fluctuation behavior in 1(e) and 1(h).

Finally, the runs in the rightmost column—Figures 1(c), 1(f), and 1(i)—illustrate how change in another (different) single parameter can make epochal evolution dynamics reappear. In run 1(c) M was lowered to 300, compared with run 1(b) in which $M = 5000$. Although the behavior is quite noisy and fitness increases more gradually on average over a wide range of time steps in the run, at least two fitness epochs can be distinguished in this run. One occurs around an average fitness of 13.5 and one around an average fitness of 16. In general, different fitness epochs occur for different runs with these parameter settings, but if a certain epoch occurs, it always occurs at the same fitness level.

In run 1(f) M was raised from 500 to 7500. Here the noise seen in run 1(e) has largely disappeared. Although the fitness increases smoothly almost everywhere in the run, an epoch can be seen around a fitness value of 6. Notice that in contrast to run 1(c), the epochs reappeared here by *increasing* the population size instead of decreasing it. This illustrates the strong interdependence of the parameter settings. That is, the effect of increasing or decreasing a certain parameter is highly dependent on the values of all other parameters. Notice alsothe large gap between the best fitness and average fitness in this run.

Lastly, in run 1(i) the block size was increased to $K = 15$. Here the fitness epochs that disappeared in run 1(h) can be clearly seen again. The behavior is still very noisy within the epochs, but average fitness increases sharply between them. From the run it is not clear if the population will eventually find the highest possible fitness of 4. Note that the best fitness did reach this highest level around generation 100,000, but did not successfully establish itself in the population, almost immediately dropping back to the lower fitness epoch. Again, our task is to explain why epochal behavior reappeared by changing each parameter in these runs.

The goal of this paper is to understand the dynamical features observed in these nine runs. Why does average fitness follow a pattern of stepwise increase for a large range of parameters? At what fitness levels are epochs going to occur and how many can occur for a given parameter setting? What is the average length of the fitness epochs? What determines the size of the fitness fluctuations within these epochs and what determines the speed of the innovations between the fitness epochs? Why do epochs disappear from the dynamics if some parameters are changed? What causes these different kinds of dynamical behaviors, such as the intermittency of run 1(g), the bounded fluctuations seen in runs 1(e) and 1(h), and the relatively smooth dynamics of run 1(b) and run 1(f)? In this paper we will present quantitative answers to almost all of these questions.

Explaining the GA's behaviors at different parameter settings is a first step towards understanding *how* one should optimally design an evolutionary search algorithm. Unfortunately, making general statements about how to set parameters is problematic at best. Since a GA's algorithmic formulation is naturally specified by independent operations on the population—such as selection, mutation, and crossover—it is tempting to base an explanation of GA dynamics on an understanding of the behaviors under selection, under mutation, and under crossover individually. As illustrated in our discussion of figure 1, changing a single parameter such as population size or mutation rate can have very different quantitative and even qualitative effects on GA behavior. Moreover, the resulting behavior depends on the settings of *all* other GA parameters. Thus, although one intuitively thinks of the GA as being composed of selection, mutation, and crossover, its actual behavior is generally not decomposable into these operators' separate effects. The direct consequence is that an integrative approach to the dynamics—one that treats the component operators simultaneously and on a equal footing and that focuses on the emergent effects of their interaction—is necessary to obtain a basic understanding of GA behavior. The following analysis demonstrates how a subtle interplay between evolutionary operators is responsible for a wide range of GA epochal search behaviors. Furthermore, our statistical dynamics approach enables us to describe the behaviors in terms of the emergence of "effective" dynamical forces, which generally involve selection and mutation as well as finite population effects. In our view the identification and analysis of these forces is prerequisite to developing, for example, prescriptions for optimizing evolutionary search parameters.

4. The Royal Road GA's State Space

Just as the state of a gas or a fluid in a box is fully described only when the locations and velocities of all the molecules are given, so the state of a population in an evolutionary search algorithm is fully specified only when all of its bit strings are listed. Any knowledge we have of the algorithm's behavior must be rooted in an understanding of the behavior of the algorithm at the lowest level of distributions of actual bit strings. The same holds for the behavior of gases and fluids in nature: they can be understood only in terms of the kinetics of their constituent molecules. It is, however, practically impossible to describe the locations of all the molecules and their velocities, just as it is impractical to describe the state of all individuals in the GA population.

This impracticality is much more of a problem from the experimental point of view than from the theoretical point of view. In fact, it is relatively easy to formally write down the equations of motion on the level of whole populations of bit strings for a simple GA, just as it is easy to formally write down the Newtonian equations of motion for all molecules in a gas. The problem is that no experimentalist will produce lists of all strings in the time series of populations to analyze, or even just describe, the GA's behavior. Instead, the experimentalist produces much more coarse-grained data from simulations—recording the statistics of most interest, such as the average fitness and best fitness in the populations or the number of generations necessary to find a string of a certain fitness.

Thus, the problem facing the theoretician is to develop a theory capable of making quantitative predictions about the GA's behavior on the level of observables that an experimentalist measures. In statistical physics this problem is solved by finding suitable "macroscopic" variables, such as temperature, pressure, and volume, that fully and self-consistently describe the thermodynamic state of the system. Specifically, in the limit of infinite system size—the thermodynamic limit—the description of the system in terms of these macroscopic variables becomes closed. That is, in this limit, one does not have to know the detailed microscopic state of the system to obtain the dynamics of the macroscopic variables.[1]

The strategy of our GA analysis is precisely this. Rather than focus on distributions of individuals in the population, we choose distributions of fitnesses in the population as our macroscopic variables. We claim that in the limit of infinite population size—the analogue of the thermodynamic limit—the evolutionary system can be deterministically and self-consistently modeled in terms of these variables. As is generally true in statistical physics, this claim is not based on rigorous mathematical proof. Rather we assume that for very large populations, the dynamics of the population's fitness distribution can be described solely in terms of the fitness distribution. This assumption is supported by the results of our simulations with large populations and by mathematical arguments that describe how the large population dynamics converges to the behavior of infinite populations. Again, the situation is analogous to that in statistical physics. Although it may not be widely appreciated, at present there is no general mathematical proof that the behavior of observables such as pressure, temperature, and volume of a gas can be described in terms of these macroscopic variables only. This fact derives from *experimental* observation.

Mathematically, the dynamics of the macroscopic variables is obtained by making the maximum entropy assumption. This says that all microscopic states that are consistent with a certain macroscopic state are equally likely. In our case, we assume that if a population has a

[1]This is strictly only true for systems in thermodynamic equilibrium.

certain fitness distribution $\Pr(f)$, then the microscopic state of the population—the distribution of genotypes—is equally likely to be any of the microscopic population states consistent with $\Pr(f)$. Starting from the "microscopic" specification of how the GA acts on string populations and using the maximum entropy assumption, a generation operator $\mathbf{G}$ can be constructed that, in the limit of infinite populations, deterministically describes the action of the GA on fitness distributions. In section 5.1 we will give theoretical arguments in support of the maximum entropy assumption, but ultimately its validity must be tested by comparing our theory's predictions against statistics gathered from the simulation experiments.

To implement this approach we describe the state of the population at a given generation by its distribution $\vec{P}(t)$ of string fitnesses, rather than by the distribution of string probabilities $\Pr(s)$ directly. This projects from the 2^L-dimensional space of strings to an N-dimensional space. $\vec{P}(t)$'s components, $P_f(t), 0 \leq f \leq N$, represent the fraction of individuals in the population with fitness f at time t. Thus, the state of the population is described by a $N+1$-dimensional vector $\vec{P}$ whose components sum up to 1; that is,

$$\sum_{f=0}^{N} P_f = 1. \tag{4.1}$$

Therefore, the state space has N independent components (dimensions). The average fitness in the population is given by

$$\langle f \rangle = \sum_{f=0}^{N} f P_f. \tag{4.2}$$

We denote the set of all possible states $\vec{P}$ for a finite population of size M as Λ_M. This state space is given by

$$\Lambda_M = \{\vec{P} \, : \, P_f = \frac{n_f}{M}, \; n_f \in \mathbb{N}, \; \sum_{f=0}^{N} n_f = M\}, \tag{4.3}$$

where n_f is the number of individuals with fitness f in the population. The number of points in the state space Λ_M for finite populations is

$$|\Lambda_M| = \binom{M+N}{N}, \tag{4.4}$$

which is on the order of M^N states for $N \ll M$. The above set Λ_M forms a rectangular lattice within the simplex in $N+1$ dimensions with lattice spacing $1/M$. In the limit of infinite populations the lattice points become dense in the simplex and we can take the state space to be the simplex itself:

$$\Lambda_\infty = \{\vec{P} \in \mathbb{R}^{N+1} \, : \, \sum_{f=0}^{N} P_f = 1, \; P_f \geq 0\}. \tag{4.5}$$

Another important assumption in considering a state space of fitness distributions is that the dynamics of the fitness distribution is not sensitive to the underlying microscopic state of the population. Note that this is an extra restriction in addition to the maximum entropy assumption. Even if it were true that for a certain fitness distribution $\vec{P}$ the GA state is equally likely to be in any microscopic population $\Pr(s)$ consistent with $\vec{P}$, then there could still be an inconsistency if the fitness distribution dynamics was sensitive to some more detailed feature of $\Pr(s)$. In the statistical physics literature this problem is called: lack of "self-averaging". Thus, our approach assumes that the dynamics on the level of fitness distributions is "self-averaging". In other words, for large populations we assume that two microscopic population states with the same fitness distribution tend to give rise to the same fitness distribution dynamics. As before, this assumption is supported by statistics gathered from our simulation experiments. We expect though that self-averaging is not generally valid for arbitrary GA dynamics. That it works so well is, therefore, a specific feature of our mutation-only GA and the Royal Road fitness function. For example, if crossover were employed by our GA, the dynamics of strings with fitness n would be sensitive to *where* the n aligned blocks were located in the strings. Fitness distributions then would be an overly coarse-grained set of macroscopic variables. We would have to augment them to include other variables that appropriately accounted for statistical properties of the aligned-block locations.

5. Infinite Population Dynamics on Fitness Distribution Space

We will first solve for the dynamics of the fitness distribution in the infinite-population limit ($M \to \infty$) by constructing a generation operator $\mathbf{G}$ that incorporates the effects of selection and mutation on the fitness distribution. The state $\vec{P}(t+1)$ of the population at time $t+1$ is obtained by acting with the operator $\mathbf{G}$ on the state $\vec{P}(t)$ of the population at time t:

$$\vec{P}(t+1) = \mathbf{G}[\vec{P}(t)], \tag{5.1}$$

where the operator $\mathbf{G}$ formally can be written as the product,

$$\mathbf{G} = \mathbf{M} \cdot \mathbf{S}, \tag{5.2}$$

of the selection operator $\mathbf{S}$ and the mutation operator $\mathbf{M}$.

This construction is similar in spirit to the generation operators in [45], although since we focus on fitness distributions rather than entire populations, we will be able to analyze the effect of this operator explicitly and quantitatively. The focus on the dynamics of fitness distributions is similar to the analysis in [38], where the dynamics of the first few cumulants of the fitness distribution is analyzed. Here we track the dynamics of the entire fitness distribution.

Equation 5.1 is analogous to a discrete-time version of the differential equations of the Eigen model of molecular evolution [11, 13]. In that model one considers the evolutionary dynamics of the distribution of different self-replicating molecular "genotypes". Therefore, the Eigen model equations are again defined on the microscopic level, not on a coarse-grained level such as that of fitness distributions.

We will now construct the operator $\mathbf{G}$ explicitly for the case of our simple GA. We first consider the alignment dynamics of unaligned blocks and then build the mutation and selection components of $\mathbf{G}$. Once we have this composite expression for $\mathbf{G}$, we turn to explicitly solving for the dynamics.

5.1 Block Alignment Dynamics

We will now present some heuristic theoretical arguments for our maximum entropy assumption: given a fitness distribution, all microscopic populations with that fitness distribution are equally likely to occur. Since the dynamics of the mutation-only GA is obviously independent of where the aligned blocks in strings occur, our maximum entropy assumption boils down to assuming that the bit values in unaligned blocks are essentially random and statistically independent of each other.

Assuming that the unaligned blocks are random we can calculate the probability A that mutation will transform an unaligned block into an aligned block. A random unaligned block is equally likely to be in any of the $2^K - 1$ unaligned block states. If the block has d zeros, the probability that it will turn into an aligned block under mutation is just $q^d(1-q)^{K-d}$. There are $\binom{K}{d}$ different block states which have d zeros. We thus have:

$$A = \frac{1}{2^K - 1} \sum_{d=1}^{K} \binom{K}{d} q^d (1-q)^{K-d} = \frac{1-(1-q)^K}{2^K - 1}. \tag{5.3}$$

The above expression for A is a direct consequence of the maximum entropy assumption that all microscopic populations with the same fitness distribution are equally likely. This is, of course, no guarantee that the GA will actually behave according to this assumption. The reason we expect unaligned blocks to be random on average is because (i) selection in our GA only counts the number of aligned blocks and therefore acts on all unaligned blocks equally and (ii) mutation randomly mixes bit values in blocks. But let's be more precise about our assumption.

Consider a particular unaligned block b in a given string s. We consider two cases. First, string s either has an ancestor in which block b was aligned and then was destroyed or, second, none of its ancestors had b aligned. For low mutation rates, block b is likely to have more bits set in the first case than in the second, since in the first case it is the remnant of a previously aligned block and in the second it is the descendant of well-mixed random unaligned blocks.

Now consider unaligned blocks in strings that belong to the highest fitness class currently in the population at time t during a run. Let $\mathcal{P}(t)$ denote the current population and let $\mathcal{P}_h(t)$ denote the set of strings having the highest fitness in $\mathcal{P}(t)$:

$$\mathcal{P}_h(t) = \{s' \in \mathcal{P}(t) : f(s') \geq f(s) \ \forall s \in \mathcal{P}(t)\}$$

Unaligned blocks in such strings are very likely to have descended from unaligned blocks in ancestral strings. This is because it is unlikely that strings with fitness higher than those of $\mathcal{P}_h(t)$ were ever present in the population in appreciable numbers. Therefore, at any time t in the run we can assume that these unaligned blocks have been subject to mutation for t time steps without ever having been aligned. Assuming that we can take different unaligned blocks in different strings in $\mathcal{P}_h(t)$ to be independent, we can take the state of these blocks to be essentially random. Different unaligned blocks within the *same* string in $\mathcal{P}_h(t)$ can in general be taken to be independent to a high approximation. The assumption that the unaligned blocks in different strings in $\mathcal{P}_h(t)$ are independent is only appropriate for very large populations, as we will see later on. For infinite populations this assumption of independence is exact for blocks in strings that are in $\mathcal{P}_h(t)$. We have therefore argued that unaligned blocks in $\mathcal{P}_h(t)$ can be taken as random, at least for very large populations.

We now turn to the case of unaligned blocks in strings not in $\mathcal{P}_h(t)$. Let s be in $\mathcal{P}(t)$ but not in $\mathcal{P}_h(t)$. The unaligned blocks in s can be divided into two types: blocks that were never aligned in ancestors of s and blocks that were aligned in some higher-fitness ancestor of s but that were destroyed by mutation. The first type of block can also be taken as random and has the same probability A given by equation 5.3 of becoming aligned by mutation. The second type of block is likely to have more bits set to 1 than a random block, so it has a higher probability of being aligned by mutation. In general we will not try to solve for the relative proportions of the two types of blocks in strings s. We can give, however, lower and upper bounds on the probability A_s that an unaligned block in s will become aligned through mutation. The lower bound is obtained by assuming that *all* unaligned blocks are of the first type and thus have never been aligned before, which yields:

$$A_s \geq A. \tag{5.4}$$

An upper bound is obtained by assuming that all blocks have $K-1$ of the K bits set, giving

$$A_s \leq q(1-q)^{K-1}. \tag{5.5}$$

We will see later on that many of our results are largely insensitive to the value of A_s within these bounds. For convenience we will use $A_s = A$ and employ the same probability for all unaligned blocks in all strings.

Again we stress that the above arguments do not prove that unaligned blocks behave as random blocks for our GA. In fact, we will later see some examples were this approximation breaks down for the dynamics of finite populations. However, for very large populations, our experiments show that this maximum entropy assumption gives excellent predictions of the dynamics of the actual GA. In the following sections we will solve for the infinite population dynamics of the GA under this assumption.

5.2 The Mutation Operator M

In the last section we derived an expression for the probability A to align a block through mutation in one time step under the random block approximation. The probability D that mutation will destroy an aligned block is simply given by

$$D = 1 - (1-q)^K. \tag{5.6}$$

Using A and D, we now consider the probability $\mathbf{M}_{ij}$ that mutation turns a string with fitness j (i.e. j aligned blocks) into a string with fitness i. In other words, $\mathbf{M}_{ij}$ is the probability that by mutating every bit with probability q in a string of fitness j this string will turn into a string with fitness i. We can write $\mathbf{M}_{ij}$ as the sum over all the probabilities that k unaligned blocks in the string will be aligned and l aligned blocks will be destroyed such that $j+k-l=i$. Thus, we have

$$\mathbf{M}_{ij} = \sum_{k=0}^{N-j} \sum_{l=0}^{j} \delta_{j+k-l,i} \binom{N-j}{k} \binom{j}{l} A^k (1-A)^{N-j-k} D^l (1-D)^{j-l}. \tag{5.7}$$

In the limit of infinite populations, the operator $\mathbf{M}_{ij}$ acting on a fitness distribution $\vec{P}$ will give the fitness distribution $\vec{P}^m$ after mutation:

$$P_i^m = \sum_{j=0}^{N} \mathbf{M}_{ij} P_j, \tag{5.8}$$

where P_i and P_i^m are the proportions of strings with fitness i before and after mutation, respectively. The mutation operator $\mathbf{M}$ is an ordinary linear matrix operator with the property that

$$\sum_{i=0}^{N} \mathbf{M}_{ij} = 1. \tag{5.9}$$

This is, of course, just another way of saying that summing the probabilities of *all* possible outcomes of mutation gives unity. That is, $\mathbf{M}$ is a stochastic matrix.

5.3 The Selection Operator S

Our simplified GA uses fitness proportionate selection. This means that the proportion P_i^s of strings with fitness i after selection is proportional to both i and the fraction P_i of strings with fitness i before selection:

$$P_i^s = c\, i\, P_i, \tag{5.10}$$

where c is a constant.[1] The constant can easily be obtained by demanding that the vector $\vec{P}^s = (P_0^s, \ldots, P_N^s)$ is normalized:

$$\sum_{i=0}^{N} P_i^s = 1. \tag{5.11}$$

Therefore, we have

$$c = \left[\sum_{i=0}^{N} i P_i\right]^{-1} = \frac{1}{\langle f \rangle}, \tag{5.12}$$

where $\langle f \rangle$ is the average fitness of the population. We can write the entries $\mathbf{S}_{ij}$ of the selection operator as the diagonal matrix

$$\mathbf{S}_{ij} = \frac{i\delta_{ij}}{\langle f \rangle}. \tag{5.13}$$

Notice that, in contrast to the mutation operator, the selection operator is nonlinear because it depends on the average fitness of the distribution it acts on.

5.4 The Generation Operator G

We can now construct the generation operator $\mathbf{G}$ that maps the fitness distribution of a population into the fitness distribution of the population at the next generation. $\mathbf{G}$ is the product of the selection and mutation operators:

$$\mathbf{G}_{ij} = \sum_{k=0}^{N} \mathbf{M}_{ik} \mathbf{S}_{kj}. \tag{5.14}$$

To analyze the dynamics of this operator we first construct a linearized version $\tilde{\mathbf{G}}$. We note that all entries in the generation operator $\mathbf{G}$ are independent of the vector $\vec{P}$ it acts on, apart from the normalization factor $1/\langle f \rangle$ in $\mathbf{S}$. We can take this factor outside the matrix and write

$$\mathbf{S} = \frac{1}{\langle f \rangle} \tilde{\mathbf{S}}, \tag{5.15}$$

[1] In the case where all strings have zero fitness, the GA is not well defined. In all practical situations considered here the fitness distribution will have nonzero proportions of nonzero fitness strings at all times.

giving

$$\tilde{\mathbf{G}} = \mathbf{M} \cdot \tilde{\mathbf{S}}. \tag{5.16}$$

The operator $\tilde{\mathbf{G}}$ is an ordinary $(N+1)$ by $(N+1)$ matrix with nonnegative entries.

The fitness distribution in the population at time t is given by the t^{th} iterate of $\mathbf{G}$ acting on the initial fitness distribution $\vec{P}(0)$. That is,

$$\vec{P}(t) = \mathbf{G}^t[\vec{P}(0)]. \tag{5.17}$$

Since at each iteration the operator $\mathbf{G}$ is just the matrix $\tilde{\mathbf{G}}$ times a constant that depends on the fitness distribution it acts on, $\mathbf{G}^t$ is proportional to the linear operator $\tilde{\mathbf{G}}^t$ times a constant that depends only on $\vec{P}(0)$ and t. Therefore, we can express the fitness distribution $\vec{P}(t)$ at time t as a constant times $\tilde{\mathbf{G}}^t$ acting on $\vec{P}(0)$. Thus, we have

$$\vec{P}_i(t) = \sum_j C[t, \vec{P}(0)] \tilde{\mathbf{G}}^t_{ij} P_j(0). \tag{5.18}$$

We can determine the constant $C[t, \vec{P}(0)]$ easily by requiring that $\vec{P}(t)$ is normalized. The result is that

$$C[t, \vec{P}(0)] = \left[\sum_{i,j} \tilde{\mathbf{G}}^t_{ij} P_j(0) \right]^{-1}. \tag{5.19}$$

Since the initial population consists of random strings of length $L = NK$, the initial fitness distribution $\vec{P}(0)$ can be obtained by considering that at $t = 0$ each block in each string has a probability 2^{-K} to be aligned. We then find that

$$P_i(0) = \binom{N}{i} 2^{-Ki} \left(1 - 2^{-K}\right)^{N-i}. \tag{5.20}$$

We can solve explicitly for $\vec{P}(t)$ by diagonalizing the matrix $\tilde{\mathbf{G}}$. In general, $\tilde{\mathbf{G}}$ will have $N+1$ distinct eigenvectors and eigenvalues.[2] We will denote these eigenvectors $\vec{V}^i$ and their corresponding eigenvalues g_i. These obey

$$\tilde{\mathbf{G}} \cdot \vec{V}^i = g_i \vec{V}^i, \tag{5.21}$$

for each value of i from 0 to N. We further normalize these eigenvectors in probability, so that

$$\sum_{i=0}^{N} V_i^j = 1. \tag{5.22}$$

Defining the matrix $\mathbf{R}$ to contain $\tilde{\mathbf{G}}$'s $N+1$ normalized eigenvectors $\vec{V}^i$ as its columns, the matrix $\mathbf{R}$ and its inverse diagonalize the generation operator, in the sense that

$$\mathbf{G}'_{ij} = (\mathbf{R}^{-1} \cdot \tilde{\mathbf{G}} \cdot \mathbf{R})_{ij} = g_i \delta_{ij}. \tag{5.23}$$

[2] In the very rare cases were the characteristic polynomial of $\tilde{\mathbf{G}}$ has multiple roots, these roots can be separated by an infinitesimal change in the mutation rate q.

Since the eigenvectors are normalized, the matrix $\mathbf{R}$ has the additional property that its columns add up to 1:

$$\sum_{i=0}^{N} \mathbf{R}_{ij} = \sum_{i=0}^{N} V_i^j = 1. \tag{5.24}$$

The generation operator $\tilde{\mathbf{G}}^t$ can now be written in terms of its eigenvalues g_i, the matrix $\mathbf{R}$, and its inverse, as follows

$$\tilde{\mathbf{G}}_{ij}^t = \sum_{k=0}^{N} \mathbf{R}_{ik} g_k^t \mathbf{R}_{kj}^{-1}. \tag{5.25}$$

This allows us to solve for $\vec{P}(t)$ in equation 5.18, obtaining

$$\vec{P}_i(t) = C[t, \vec{P}(0)] \sum_{k,j} \mathbf{R}_{ik} g_k^t \mathbf{R}_{kj}^{-1} P_j(0). \tag{5.26}$$

Equation 5.26 gives an exact expression for the fitness distribution in the population as a function of time in the limit of infinite population size. We can make equation 5.26 more transparent by moving to the basis of $\tilde{\mathbf{G}}$'s eigenvectors. First, we write the fitness distributions in this basis as

$$\alpha_i(t) = \sum_{j=0}^{N} \mathbf{R}_{ij}^{-1} P_j(t). \tag{5.27}$$

Note that the $\alpha_i(t)$ are normalized. We further simplify the expression for $C[t, \vec{P}(0)]$ using equations 5.25 and 5.27 in equation 5.19:

$$C[t, \vec{P}(0)] = \left[\sum_{i,j,k=0}^{N} \mathbf{R}_{ik} g_k^t \mathbf{R}_{kj}^{-1} P_j(0) \right]^{-1} = \left[\sum_{k=0}^{N} g_k^t \alpha_k(0) \right]^{-1}. \tag{5.28}$$

Transforming the equations of motion (equation 5.26) to the basis of the eigenvectors $\vec{V}^i$ we find that the fitness distribution is given by

$$\alpha_i(t) = \frac{g_i^t \alpha_i(0)}{\sum_{j=0}^{N} g_j^t \alpha_j(0)}. \tag{5.29}$$

From this we get a very simple expression for the average fitness $\langle f(t) \rangle$ as a function of time. Again using the fact that the rows of $\mathbf{R}$ sum up to 1, a little algebra gives:[3]

$$\langle f \rangle = \sum_i i P_i = \sum_i g_i \alpha_i. \tag{5.30}$$

From the two preceding expressions, we are left with a simple, direct expression for the behavior of the average fitness as a function of time for infinite populations:

$$\langle f(t) \rangle = \frac{\sum_i g_i^{t+1} \alpha_i(0)}{\sum_j g_j^t \alpha_j(0)}. \tag{5.31}$$

[3]See the next section for a detailed derivation of this property.

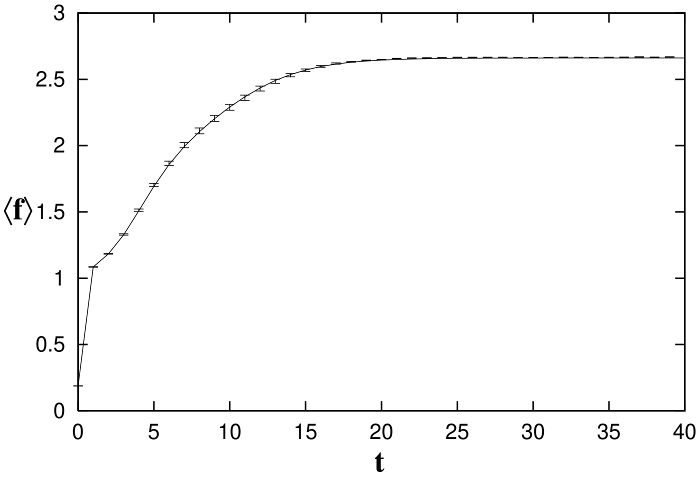

Figure 2: Average fitness $\langle f(t)\rangle$ over time t averaged over 20 GA runs along with the theoretical prediction (solid line) for infinite population obtained from equation 5.31. The error-bars show ± 2 standard deviations from the average fitness over the 20 runs. The parameters here are $N = 3$ and $K = 4$ with $q = 0.01$ and a population size of 10^4.

Figure 2 shows the predicted infinite population dynamics for a particular parameter setting together with the empirical results averaged over 20 runs with a large population. The figure shows that for large populations, typically for $M > 2^{NK}$, the actual average fitness behavior follows the infinite population dynamics, obtained using our maximum entropy assumption, quite closely. This supports our assumption that for large populations, the evolution of the fitness distribution can be described in terms of the fitness distribution only. (Appendix B proves that under the maximum entropy assumption the dynamics of large populations converges towards the deterministic dynamics as given by $\mathbf{G}$.) The small error bars in the figure also show that for these large populations, there are very small fluctuations in the evolution of the average fitness between runs. This fact supports our assumption that the dynamics on the level of fitness distributions is self-averaging in the limit of large populations. Notice, though, that for populations of this size the average fitness increases smoothly as a function of time and there are no fitness epochs. Apparently, fitness epochs occur only for population sizes that are not too large.

Recapitulating, by obtaining the probabilities A and D to align and destroy blocks we constructed a matrix operator that describes the dynamics of the genetic algorithm in the limit of infinite populations. By linearizing $\mathbf{G}$ to form $\tilde{\mathbf{G}}$ and then computing $\tilde{\mathbf{G}}$'s eigenvalues and eigenvectors, we were able to solve exactly for (i) the dynamics of the fitness distribution (equation 5.29) and (ii) the average fitness $\langle f(t)\rangle$ in the population (equation 5.31).

5.5 Properties of the Generation Operator G

The fixed points of the operator $\mathbf{G}$ are those vectors $\vec{V}$ that are mapped to themselves under its action. Whenever the fitness distribution $\vec{P}$ describing the infinite population equals one of those vectors $\vec{V}$, it will remain at $\vec{V}$ from then on. Consider the fixed point equation for the dynamics, $\vec{P} = \mathbf{G}(\vec{P})$. We have

$$\mathbf{G}(\vec{P}) = \frac{\tilde{\mathbf{G}}}{\langle f\rangle} \cdot \vec{P} \; ; \tag{5.32}$$

so for fixed points $\vec{P}$ of $\mathbf{G}$,

$$\tilde{\mathbf{G}} \cdot \vec{P} = \langle f \rangle \vec{P}. \tag{5.33}$$

In words, the fixed points of $\mathbf{G}$ are given by normalized eigenvectors of $\tilde{\mathbf{G}}$, with the extra restriction that $\tilde{\mathbf{G}}$'s eigenvalues are equal to the fitness average of the eigenvector.

Using the stochasticity property of the mutation operator $\mathbf{M}$ (equation 5.9), we find that *all* eigenvectors $\vec{V}^i$ of $\tilde{\mathbf{G}}$ fulfill the above restriction (equation 5.33). First, using equation 5.21, note that

$$\sum_{i,j} \tilde{\mathbf{G}}_{ij} V_j^k = \sum_i g_k V_i^k = g_k. \tag{5.34}$$

This simply states the fact that $\vec{V}^k$ is a normalized eigenvector with eigenvalue g_k. Furthermore, substituting the definition of $\tilde{\mathbf{G}}$ in terms of $\mathbf{M}$ and $\mathbf{S}$, we have

$$\sum_{i,j} \tilde{\mathbf{G}}_{ij} V_j^k = \sum_{i,j} \mathbf{M}_{ij} j V_j^k = \sum_j j V_j^k = \langle f \rangle. \tag{5.35}$$

Thus, we see that

$$g_k = \langle f \rangle \tag{5.36}$$

for all eigenvectors $\vec{V}^k$, which gives a simple interpretation of their eigenvalues as average fitnesses.

This implies, in turn, that all eigenvectors of $\tilde{\mathbf{G}}$ are fixed points of the generation operator $\mathbf{G}$. This might lead one to believe that the infinite population GA dynamics could end up in N different stable states in fitness distribution space.[4] This is not true. The normalized eigenvectors of $\tilde{\mathbf{G}}$ have to be positive definite to be interpreted as fitness distributions and, in general, they are not. In Appendix A we prove that the matrix $\tilde{\mathbf{G}}$ can have only a single positive definite eigenvector. Therefore, there is a unique fixed-point fitness distribution towards which the Royal Road GA will always converge asymptotically. All other fixed points lie outside the simplex Λ_∞ and are therefore unreachable. The asymptotic fixed point corresponds to $\tilde{\mathbf{G}}$'s principal eigenvector—the eigenvector of $\tilde{\mathbf{G}}$ with the largest eigenvalue. This can also be understood by considering equation 5.31, which shows that the largest eigenvalue g_N—having chosen the ordering $g_0 < g_1 < \cdots < g_N$—will dominate the average fitness in the limit of $t \to \infty$. That is,

$$\lim_{t\to\infty} \langle f(t) \rangle = g_N. \tag{5.37}$$

In addition, we see that the fitness distribution $P(t)$ asymptotically approaches the principal eigenvector:

$$\lim_{t\to\infty} \vec{P}(t) = \vec{V}^N. \tag{5.38}$$

Globally, the map $\mathbf{G}$ has one fixed point lying inside the state space Λ_∞ and $N-1$ fixed points lying outside.

Notably, the asymptotic distribution $\vec{V}^N$ over the different fitness classes is analogous to what is called a quasispecies in molecular evolution theory [11]. A species—a particular virus for instance—cannot in general be identified with a certain unique genotype. In fact, the genotypes of members of a certain species can often be thought of as a cloud of points in sequence space that is centered around a genotype of highest fitness. The size and shape of this cloud depend on

[4] Remember that we excluded the possibility of all strings having 0 fitness.

the structure of the fitness variations around this peak and on the interplay between mutation and selection. Mutation causes points to drift away from the peak, while selection tends to replicate only individual genotypes whose fitness remains close to that of the peak.[5] This cloud is called a "quasispecies" and is equivalent to our asymptotically stable eigenvector $\vec{V}^N$. In our case, $\vec{V}^N$ represents a cloud of strings with different fitnesses centered around the string with all blocks aligned. Here, as in the molecular evolution setting, the shape and size of the cloud are obtained by solving for the principal eigenvector $\vec{V}^N$ of the generation operator. Note though that since $\vec{V}^N$ is a fitness distribution, the lower components of $\vec{V}^N$ do not correspond to particular genotypes as in the molecular quasispecies case, but rather to sets of genotypes with equal fitness. Our quasispecies distribution is therefore a "phenotypic" or fitness quasispecies cloud.

5.6 GA Dynamics as a Flow in Fitness Distribution Space

With equation 5.26 we have an exact expression for the evolution of the fitness distribution $\vec{P}(t)$. As mentioned above, this dynamics shows a smooth and strictly monotonic increase of average fitness starting from a random initial population; figure 2. It does not exhibit the fitness epochs illustrated in figure 1. To begin to explain the occurrence of fitness epochs for small populations we must first adopt a different and more geometric view of the infinite population dynamics. We will visualize the infinite population dynamics as a flow in the space of fitness distributions Λ_∞. As a simple example, figure 3 illustrates the dynamics for the parameter settings $N = 3$, $K = 4$, and $q = 0.01$. The state space Λ_∞ can be projected into 3-dimensional Euclidean space for this case. Figure 3 has P_0 on the x-axis, P_1 on the y-axis, and P_2 on the z-axis. Of course, we have $P_3 = 1 - P_0 - P_1 - P_2$.

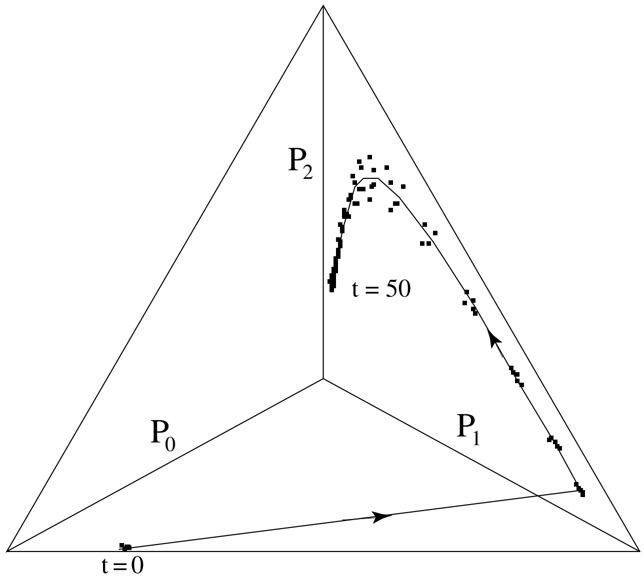

Figure 3: Infinite population dynamics as a trajectory (solid line) in Λ_∞ for $N = 3$, $K = 4$, and $q = 0.01$, together with the results (squares) of five runs with these parameters and $M = 10^4$. Cf. figure 2, which plots the average fitness dynamics.

[5]It is often tacitly assumed that genotypes with fitness close to that of the peak remain genotypically close to the peak as well. See [25] for a discussion of the implications of this *not* being the case.

Figure 3 shows the theoretical infinite population trajectory of the dynamics in Λ_∞ for these parameter settings, together with the empirical dynamics for five runs with a large population ($M = 10^4$). We see that the empirical dynamics follows the infinite population trajectory quite closely. (Compare the average fitness dynamics plotted in figure 2.) This again shows that the evolution of the fitness distribution is well approximated by our maximum entropy assumption for large populations.

We can get an idea of the force driving the fitness distribution along its trajectory through the simplex by considering the difference $d\vec{P} = \mathbf{G}(\vec{P}) - \vec{P}$ of the fitness distribution $\vec{P}$ and the fitness distribution $\mathbf{G}(\vec{P})$ at the next time step. The vector $d\vec{P}$ gives the local direction and magnitude of the change of the fitness distribution at $\vec{P}$ over one time step. We will refer to this change $d\vec{P}$ as the "flow" induced by the generation operator at the point $\vec{P}$ in the state space.[6] Figure 4 shows $d\vec{P}$ in the simplex for the same parameter settings as in figure 3. We now see how the trajectory as shown in figure 3 can be understood as the fitness distribution following the flow $d\vec{P}$ through fitness distribution space. The geometric view of the GA dynamics is that the flow $d\vec{P}$ in the simplex is followed by the fitness distribution in the limit of infinite populations. This will help us understand the occurrence of fitness epochs for finite populations. We will see that in the finite population case the fitness distribution *attempts* to follow this flow, but for reasons to be discussed in the next section, cannot always (and not everywhere) exactly follow it. This, it turns out, is precisely the origin of the fitness epochs, seen only for finite populations.

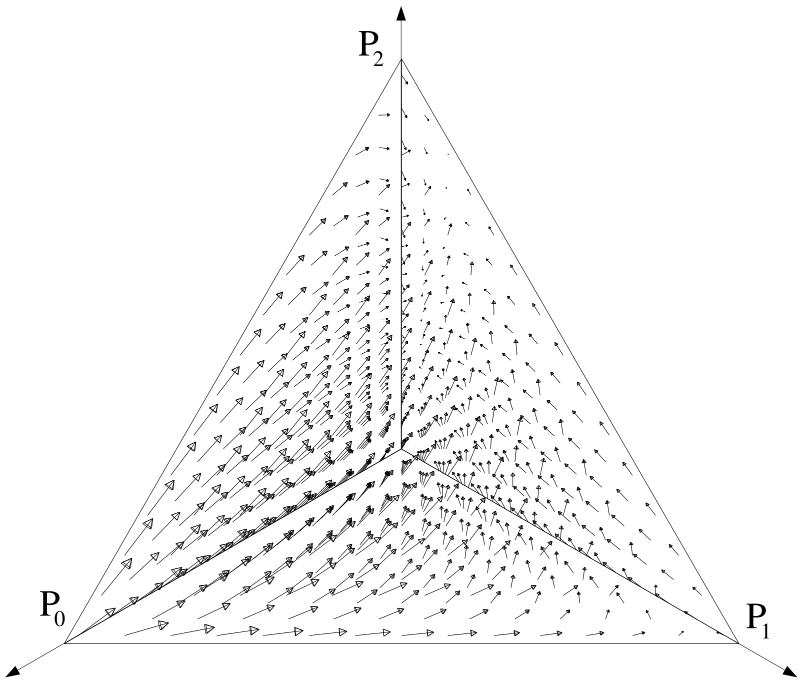

Figure 4: Fitness distribution flow $d\vec{P}$ in the simplex for $N = 3$, $K = 4$, and $q = 0.01$.

[6] We are using "flow" in a somewhat loose, but convenient, sense. In dynamical systems theory "flow" is normally reserved for the collection of all trajectories through state space over some time interval. Our usage of the term "flow" corresponds to the vector field that points along the trajectory for each point in state space. It can therefore be seen as the vector field *generating* the flow at each point of state space.

6. Finite Population Dynamics

As we saw in section 4, the fitness distribution state space for a finite population is given by a rectangular lattice Λ_M over the $N+1$ dimensional simplex Λ_∞ with lattice spacing $1/M$. As in the infinite population case, applying the generation operator $\mathbf{G}$ to a fitness distribution $\vec{P} \in \Lambda_M$ gives the *expected* fitness distribution $\langle \vec{P} \rangle$ at the next time step. Viewed in a slightly different way, $\mathbf{G}_i(\vec{P}) = \langle P_i \rangle$ is the probability that if one string is selected from $\vec{P}$ and mutated, it will turn out to have fitness i. Since the new population is created from M of these selections and mutations, it is clear that the new population is a sample of size M of the distribution $\langle \vec{P} \rangle$. In the infinite population case the *expected* distribution $\langle \vec{P} \rangle$ is always attained; there are no fluctuations in the $\langle P_i \rangle$ in this limit. For a finite population of size M, the finite sampling from $\langle \vec{P} \rangle$ leads to stochasticity in the dynamics. At each time step the population that is actually attained will fluctuate around the expected distribution $\langle \vec{P} \rangle$. We therefore see that the flow in state space for a finite population is given by the same flow operator $\mathbf{G}$ but now over a discrete space and with noise added due to the population's finiteness. It is important to note that the finite population sampling noise added to the operation of $\mathbf{G}$ takes into account the combined effect of the fluctuations in both mutation and selection. Normally, one would consider the sampling fluctuations in mutation and selection to be two distinct sources of fluctuations. In combining the selection and mutation effects into the operator $\mathbf{G}$, all fluctuations are due to the finite population sampling applied to the operation of $\mathbf{G}$.

In general, the probability $\Pr[\vec{P}^n \to \vec{P}^m]$ that a fitness distribution $\vec{P}^n = (n_0, n_1, \ldots, n_N)/M$ will go to a fitness distribution $\vec{P}^m = (m_0, m_1, \ldots, m_N)/M$ under mutation and selection is given by a multinomial sampling distribution with mean $\mathbf{G}(\vec{P}^n)$:

$$\Pr[\vec{P}^n \to \vec{P}^m] = M! \prod_{i=0}^{N} \frac{[\mathbf{G}_i(\vec{P}^n)]^{m_i}}{m_i!}, \tag{6.1}$$

where $\mathbf{G}_i(\vec{P}^n)$ is the expected proportion of individuals with fitness i at the next generation. The multinomial distribution is nothing more than the distribution of a random sample of size M of the expected distribution $\mathbf{G}(\vec{P})$. In Appendix B this distribution is used to prove that for large populations the finite population dynamics approaches the infinite population dynamics arbitrarily closely.

6.1 Fitness Epochs

We will now turn to the fitness epochs that occur for finite population size. As we have seen, $\mathbf{G}(\vec{P})$ gives the expected distribution at the next time step if the current distribution is $\vec{P}$. The expected flow $\langle d\vec{P} \rangle$ in fitness distribution space is given by $\langle d\vec{P} \rangle = \mathbf{G}(\vec{P}) - \vec{P}$. Now let us consider what happens if the absolute value $|\langle dP_i \rangle|$ of a certain component $\langle dP_i \rangle$ of the expected flow is much smaller than the lattice spacing; that is, if $|\langle dP_i \rangle| \ll 1/M$. The fitness distribution $\vec{P}$ for a finite population can only be on the lattice points of Λ_M, since the actual difference dP_i can only be $\ldots -1/M$ 0, $1/M$, $2/M \ldots$. Thus, we expect the actual difference dP_i to be 0 most of the time; this means that the actual component P_i will likely not change for some time. For example, in the previous case with $N = 3$, $K = 4$, and $q = 0.01$, but with finite $M = 100$, we find that if $P_3 = 0$, the dynamics will push P_0, P_1, and P_2 into a region where

$$|\langle dP_3 \rangle| \approx 6.5 * 10^{-4} \ll \frac{1}{M} = 0.01. \tag{6.2}$$