

5

Optimizing Epochal Evolutionary Search: Population-Size Independent Theory

Erik van Nimwegen and James P. Crutchfield
Computer Methods in Applied Mechanics and Engineering,
special issue on Evolutionary and Genetic Algorithms
in Computational Mechanics and Engineering,
D. Goldberg and K. Deb, editors (in press 1999)
Santa Fe Institute Working Paper 98-06-046.

Epochal dynamics, in which long periods of stasis in population fitness are punctuated by sudden innovations, is a common behavior in both natural and artificial evolutionary processes. We use a recent quantitative mathematical analysis of epochal evolution to estimate, as a function of population size and mutation rate, the average number of fitness function evaluations to reach the global optimum. This is then used to derive estimates of and bounds on evolutionary parameters that minimize search effort.

5.1 Engineering Evolutionary Search

Evolutionary search refers to a class of stochastic optimization techniques—loosely based on processes believed to operate in biological evolution—that have been applied successfully to a variety of different problems; see, for example, Refs. [4, 11, 18, 28, 38, 49, 60, 94, 103] and references therein. Unfortunately, the mechanisms constraining and driving the population dynamics of evolutionary search on a given problem are not well understood.

In mathematical terms, evolutionary search algorithms are nonlinear stochastic population-based dynamical systems. The complicated dynamics exhibited by such systems has been appreciated for decades in the field of mathematical population genetics [69]. For example, the effects on evolutionary behavior of the rate of genetic variation, the population size, and the function to be optimized typically cannot be analyzed separately; there are strong, nonlinear interactions between them. These complications make an empirical approach to the question of whether and how to use evolutionary search problematic in general. The work presented here continues an attempt to unify and extend theoretical work that has been done in the areas of evolutionary search theory,

molecular evolution theory, and mathematical population genetics. The goal is to obtain a more general and quantitative understanding of the emergent mechanisms that control the population dynamics of evolutionary search and other population-based dynamical systems.

Here we focus on a class of population dynamics that we call *epochal evolution*. In epochal evolution, long periods of stasis in the average fitness of the population are punctuated by rapid innovations to higher fitness. In our analysis of this behavior [137] and chapter 4, we view large degeneracies in the genotype-fitness mapping to be the main source of the epochal nature of the evolutionary dynamics. For every genotype, we assume that there is a range of genotypic change that mutation can induce, without affecting phenotype or fitness. In this way, the space of genotypes is broken into strongly and weakly connected sets with respect to fitness. Sets of equal fitness genotypes are strongly connected, while sets of different fitness are only weakly connected to each other. Epochs occur because members of the evolving population have to search through most of the space of neutral variants before higher fitness genotypes are discovered. This structural view of the search space motivates several simplifying assumptions that lead to the class of fitness functions we analyze in the following. Stated in the simplest possible terms, all of the resulting population dynamical behavior derives from the interplay of this architecture, the infinite-population dynamics, and the stochasticity and discretization arising from finite-population sampling.

Considering our choice of focus, one might object that important details of real biological evolution, on the one hand, or of alternative evolutionary search algorithms, on the other hand, are not described by the resulting class of epochal evolutionary dynamical systems. Our response is simple: One must start somewhere. The bottom line is that the results and their predictive power justify the approach. Moreover, along the way we come to appreciate a number of fundamental trade-offs and basic mechanisms that drive and inhibit epochal evolutionary search.

Our results show that a detailed dynamical understanding, announced in Ref. [137] and expanded in chapter 4, can be turned to a very practical advantage. Specifically, we determine how to set mutation rate to reach, in the fewest steps, the global optimum in a wide class of fitness functions that give rise to epochal evolution. Moreover, the resulting detailed analytic understanding of the behavior identifies the mechanisms causing the dynamical instabilities that emerge when setting parameters for optimal, efficient search.

As we outline at the end, our analysis provides several insights that are useful knowledge for engineers applying evolutionary search to even more complicated optimization problems—and, for that matter, for the theory of evolutionary dynamics in biology—whenever genetic neutrality and degenerate genotype-to-fitness maps occur. Specifically, the results apply to cases in which epochal evolution is observed in the population dynamics.

5.2 Landscape Architecture

The fitness functions characteristic of problems that evolutionary search or (say) simulated annealing are being used for in practice are very complicated, almost by definition. On the one hand, detailed knowledge of the fitness function implies that one does not

need to run an optimization method to find high fitness solutions. On the other hand, assuming no knowledge at all leads to a completely random set of fitness functions for which it is known that *any* optimization algorithm performs as well on average as random search [146]. Not surprisingly, reality is a middle ground.

Therefore, our strategy to understand the workings of evolutionary search algorithms is to assume some specific structure in the fitness function that is germane to epochal evolution and to assume that, beyond this, there is no other structure. That is, apart from the structure we impose, the fitness function is as simple as can be.

There is a concomitant and compelling biological motivation for our focus on epochal dynamics. This is the common occurrence in natural evolutionary systems of “punctuated equilibria”—a process first introduced to describe sudden morphological changes in the paleontological record [64]. Similar behavior has been recently observed experimentally in bacterial colonies [37] and in simulations of the evolution of tRNA secondary structures [47]. This class of behavior appears general enough to also occur in artificial evolutionary systems, such as evolving cellular automata [22, 104] and populations of competing self-replicating computer programs [1].

How are we to think of the mechanisms that cause this evolutionary behavior? The evolutionary biologist Wright introduced the notion of “adaptive landscapes” to describe the (local) stochastic adaptation of populations to themselves and to environmental fluctuations and constraints [150]. This geographical metaphor has had a powerful influence on thinking about natural and artificial evolutionary processes. The basic picture is that the evolving population stochastically crawls along a surface determined, perhaps dynamically, by the fitness of individuals, moving to peaks and very occasionally hopping across fitness “valleys” to nearby, and hopefully higher fitness, peaks.

More recently, it has been assumed that the typical fitness functions of combinatorial optimization and biological evolution can be modeled as “rugged landscapes” [88, 100]. These are functions with wildly fluctuating fitnesses even at the smallest scales of single-point mutations. Moreover, it is assumed that these “landscapes” possess a large number of local optima. With this picture in mind, the common interpretation of punctuated equilibria in evolving populations is that of a population being “stuck” at a local peak in the landscape, until a rare mutant crosses a valley of relatively low fitness to a higher peak.

At the same time, an increasing appreciation has developed, in contrast to this rugged landscape view, that there are substantial degeneracies in the genotype-to-phenotype and the phenotype-to-fitness mappings. The crucial role played by these degeneracies has found important applications in molecular evolution; e.g. see Ref. [48]. When these degeneracies are operating, different genotypes in the population fall into a relatively small number of distinct fitness classes of genotypes with approximately equal fitness. Moreover, due to the high dimensionality of genotype spaces, sets of genotypes with approximately equal fitness tend to form components in genotype space that are connected by paths made of single-mutation steps. Note that, due to intrinsic or even exogenous effects (e.g. environmental), in addition there simply may not exist a deterministic “fitness” value for each genotype. In this case, fluctuations can induce variation in fitness such that genotypes with neighboring similar average fitness values are not distinct at the level of selection. Thus, genotype-to-fitness degeneracies also occur when there is noise in the fitness evaluation of individuals.

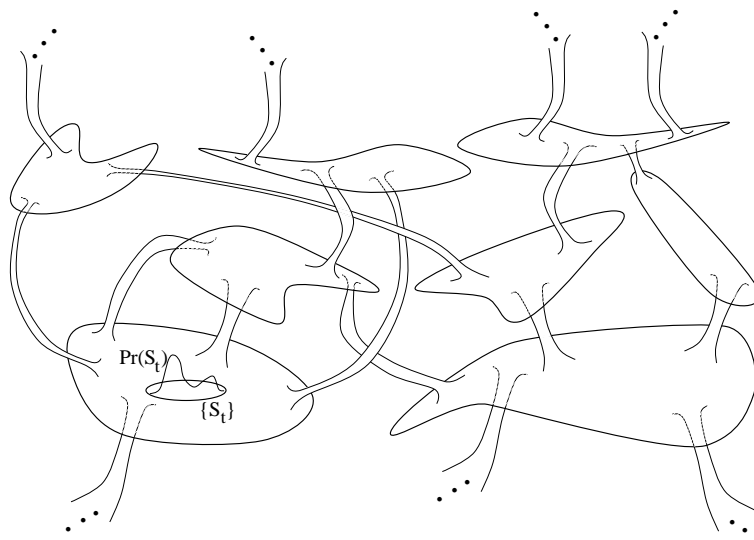


Figure 5.1: Subbasin and portal architecture underlying epochal evolution. Roughly speaking, a population diffuses in the subbasins (large sets) until a portal (a tube) to a higher fitness subbasin is found.

When these biological facts are taken into account we end up with an alternative view to both Wright’s “adaptive landscapes” and the more recent “rugged landscapes”. That is, the genotype space decomposes into a set of *neutral networks* or *subbasins* of approximately isofitness genotypes that are entangled with each other in a complicated and largely unstructured fashion; see Fig. 5.1. Within each neutral network selection is effectively disabled and neutral evolution dominates. This leads to a rather different interpretation of the processes underlying punctuated equilibria. Instead of the population being stuck at a local optimum in genotype space as suggested by the “landscape” models, the population drifts randomly over a neutral network of isofitness genotypes until a connection or *portal* to a neutral net of higher fitness is discovered [79].

Some of the first steps in understanding the biological consequences of neutral evolution, in single neutral networks, were taken by Kimura in the 1960’s using stochastic process methods adapted from statistical physics [91]. Despite the early progress in neutral evolution, a number of fundamental problems remain only partly solved [31]. Although we will analyze neutral evolution in the following, we also emphasize the global architectural structure that connects the neutral networks and drives and constrains epochal evolutionary search. That is, the search space view of entangled neutral networks provides a natural and quite general architecture for studying epochal evolution.

This intuitive view of biologically plausible genotype-space architectures—as a relatively small number of connected neutral nets—is the one that we adopt in the following analysis. We formalize it by making several more specific (simplifying) assumptions

scape” to the view of a diffusion process constrained by the subbasin-portal architecture induced by degeneracies in the genotype-to-phenotype and phenotype-to-fitness mappings. Moreover, as will become more apparent, our approach is not simply a shift in architecture, but it also focuses on the dynamics of populations as they move through the subbasins to find portals to higher fitness. A side benefit is that our approach is not limited to evolutionary processes for which a potential energy function exists, as the “landscape” models are.

5.3 The Royal Staircase Fitness Function

Under the assumptions specified in the previous paragraphs the class of fitness functions, referred to as the *Royal Staircase*, delineated is equivalent to the following specification:

1. Genomes are represented by binary strings $s = s_1 s_2 \cdots s_L, s_i \in \{0, 1\}$, of length $L = NK$.
2. Reading the genome from left to right, the number $I(s)$ of consecutive 1s is counted.
3. The fitness $f(s)$ of string s with I consecutive ones, followed by a zero, is $f(s) = 1 + \lfloor I(s)/K \rfloor$. The fitness is thus an integer between 1 and $N + 1$.

Note that the (single) global optimum is the genome $s = 1^L$; namely, the string of all 1s.

From this it is easy to see that we have chosen N (consecutive) sets of K bits to represent the different fitness classes. These sets we call *blocks*. The first block consists of the first K bits on the left, i.e. $s_1 \cdots s_K$. The second block consists of bits $s_{K+1} \cdots s_{2K}$ and so on. For each of these blocks there is one *aligned* configuration consisting of K 1s and $2^K - 1$ *unaligned* configurations. If the first block is unaligned, the string obtains fitness 1. If the first block is aligned and the second unaligned, it obtains fitness 2. If the first two blocks are aligned and the third unaligned, it obtains fitness 3, and so on up to the globally optimal string with all aligned blocks and fitness $N + 1$.

Without affecting the evolutionary dynamics or the underlying architecture of genotype space, we could have chosen a different aligned-block pattern than the K -1s configuration. In fact, we could have chosen different aligned configurations for the different blocks without affecting the dynamics. Furthermore, since we will not be analyzing position-dependent operators—such as, crossover—we could have chosen the locations of the bits of each block to be anywhere in the genome without affecting the dynamics. The only constraint, other than the block’s ordering, is that we have N disjoint sets of K bits.

Notice that the proportion ρ_n of strings with fitness n is given by:

$$\rho_n = 2^{-K(n-1)} (1 - 2^{-K}) . \tag{5.1}$$

The net result is that the Royal Staircase fitness functions implement the intuitive idea that increasing fitness is obtained by setting more and more bits correctly. One can only set correct bit values in groups of K bits at a time and in blocks from left to right. A

genome's fitness is proportional to the number of blocks it has set correctly. Finally, note that due to the modularity of the subbasin-portal architecture, and of the resulting theory we present below, the restriction to uniform block size and equal fitness contributions of each block could also be lifted.

The Royal Staircase fitness functions thus realize our view of the underlying architecture as sets of isofitness genomes that occur in nested neutral networks of smaller and smaller volume; as shown in Fig. 5.2. Despite their simplicity, which is quite helpful for our population dynamics analyses, the preceding series of observations show that the Royal Staircase fitness functions describe a large class of genotype space architectures for epochal evolution.

5.4 The Genetic Algorithm

For our analysis of evolutionary search we have chosen a simplified form of a genetic algorithm (GA) that does not include crossover and that uses fitness-proportionate selection. The GA is defined by the following steps.

1. Generate a population of M bit strings of length $L = NK$ with uniform probability over the space of L -bit strings.
2. Evaluate the fitness of all strings in the population.
3. Stop the algorithm, noting the generation number t_{opt} , if a string with optimal fitness $N + 1$ occurs in the population. Else, proceed.
4. Create a new population of M strings by selecting, with replacement and in proportion to fitness, strings from the current population.
5. Mutate each bit in each string of the new population with probability q .
6. Go to step 2.

The total number E of fitness function evaluations is $E = Mt_{\text{opt}}$. We are interested in the average number E of evaluations per GA run required over a large number R of runs.

The main motivation for leaving out crossover is that this greatly simplifies our analysis. The benefit is that we can make detailed quantitative predictions of the GA's behavior. Moreover, we believe that from the point of view of optimization, the addition of crossover to the genetic operators only marginally improves the efficiency of the search for fitness functions that typically show epochal evolution. We comment on this later in section 5.8. Additional discussion and supporting evidence for this claim can be found in section 4.6.5 and in Ref. [104].

The selection method we use is simple fitness-proportionate selection. There is a wide variety of selection mechanisms used and studied in the evolutionary search literature [61]. In fact, many practitioners prefer other selection schemes over simple fitness-proportionate selection. Our use of fitness-proportionate selection is motivated by two observations. First, it is the most appropriate selection scheme for the analysis of biological systems, in which we are also interested. It is, therefore, the first selection

scheme that we study. Second, our analysis is most straightforward and the results easiest to understand for fitness-proportionate selection. Fortunately, our analyses can be readily extended to other selection schemes, such as truncation (elite) selection, ranking schemes, and tournament selection. We will report on these extensions of our methods elsewhere.

Notice that our GA effectively has two parameters: the mutation rate q and the population size M . A given search problem, then, is specified by the fitness function in terms of N and K . The central goal of the following analysis is to find those settings of M and q that minimize the number E of fitness function evaluations for given N and K .

5.5 Observed Population Dynamics

The typical dynamics of a population evolving on a landscape of connected neutral networks, such as defined above, alternates between long periods of stasis in the population's average fitness (*epochs*) and sudden increases in the average fitness (*innovations*). (See, for example, Fig. 1 of chapter 4.) As was first pointed out in the context of molecular evolution in Ref. [79], through neutral mutations, the best individuals in the population diffuse over the neutral network (*subbasin*) of isofitness genotypes until one of them discovers a connection (*portal*) to a neutral network of even higher fitness. The fraction of individuals on this new network then grows rapidly, reaching an equilibrium after which the new subset of most-fit individuals diffuses again over the new neutral network. In addition to the increasing attention paid to this type of epochal evolution in the theoretical biology community [47, 51, 77, 108, 50, 142], recently there has also been an increased interest by evolutionary search theorists [6, 70].

The GA just defined is the same as that studied in our earlier analyses [137] and chapter 4. Also, the Royal Staircase fitness function defined above is very similar to the *Royal Road* fitness function that we used there. It should not come as a surprise, therefore, that qualitatively the GA's experimentally observed behavior is very similar to that reported in Ref. [137] and chapter 4. Moreover, most of the theory developed there for epochal evolutionary dynamics carries over to the Royal Staircase class of fitness functions.

We now briefly recount the experimentally observed behavior of typical Royal Staircase GA runs. The reader is referred to chapter 4 for a detailed discussion of the dynamical regimes this type of GA exhibits under a range of different parameter settings.

Figure 5.3 shows the GA's behavior with four different parameter settings. The vertical axes show the best fitness in the population (upper lines) and the average fitness in the population (lower lines) as a function of the number of generations. Each figure is produced from a single GA run. In all of these runs the average fitness $\langle f \rangle$ in the population goes through stepwise changes early in the run, alternating epochs of stasis with sudden innovations in fitness. Later in the run, the average fitness has higher fluctuations that tend to obscure the epochal nature of the dynamics. Notice also that $\langle f \rangle$ roughly tracks the epochal behavior of the best fitness in the population. Notice, too, that the best fitness often shows a series of "spikes", corresponding to the discovery of a higher fitness string that is soon lost. Eventually, one of these discoveries *fixates* in the population. Finally, for each of the four settings of N and K we have chosen the

5.5 Observed Population Dynamics

values of M and q such that the total number E of fitness function evaluations to reach the global optimum for the first time is roughly minimal. Thus, the four plots illustrate the GA's typical dynamics close to optimal (q, M) -parameter settings—the analysis for which begins in the next section.

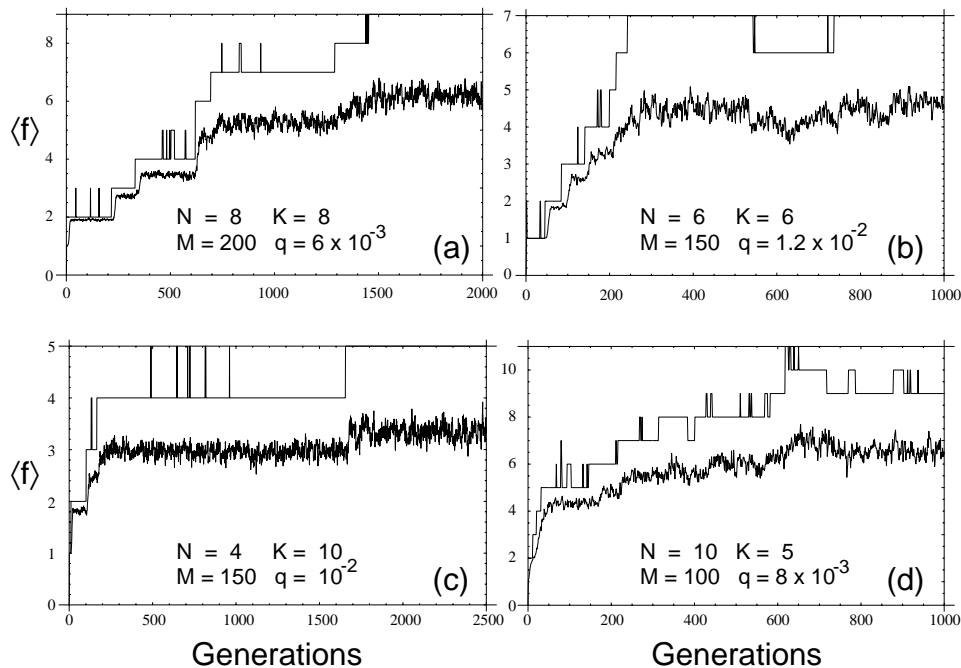


Figure 5.3: Examples of the Royal Staircase GA population dynamics with different parameter settings. The four plots show best fitness in the population (upper lines) and average fitness in the population (lower lines) as a function of time, measured in generations. The fitness function and GA parameters are given in each plot. In each case we have chosen q and M in the neighborhood of their optimal settings (see later) for each of the four values of N and K .

There is a large range, almost a factor of 10, in times to reach the global optimum across the runs. Thus, there can be a strong parameter dependence in search times. Moreover, the *standard deviation* of the total number E of fitness function evaluations is roughly as large as the average E . That is, there are large run-to-run variations in the time to reach the global optimum, even with the parameters held constant. This is true for all parameter settings with which we experimented, of which only a few are reported here.

Figure 5.3(a) plots the results of a GA run with $N = 8$ blocks of $K = 8$ bits each, a mutation rate of $q = 0.006$, and a population size of $M = 200$. During the epochs, the best fitness in the population jumps up and down several times before it finally jumps up and the new more-fit genotype stabilizes in the population. This transition is reflected in the average fitness also starting to move upward. In this particular run, it took the GA

approximately 2.9×10^5 fitness function evaluations to reach the global optimum for the first time. Over 250 runs the GA takes on average 5×10^5 fitness function evaluations to reach the global optimum for these parameters. The inherent large per-run variation means in this case that some runs take less than 10^5 function evaluations and that others take many more than 10^6 .

Figure 5.3(b) plots a run with $N = 6$ blocks of length $K = 6$ bits, a mutation rate of $q = 0.012$, and a population size of $M = 150$. The GA reached the global optimum after approximately 3.6×10^4 fitness function evaluations. On average, the GA uses approximately 5×10^4 fitness function evaluations to reach the global fitness optimum. Notice that somewhere after generation 500 the global optimum is lost again from the population. It turns out that this is a typical feature of the GA's behavior for parameter settings close to those that give minimal E . The global fitness optimum often only occurs in relatively short bursts after which it is lost again from the population. Notice also that there is only a small difference in $\langle f \rangle$ depending whether the best fitness is either 6 or 7.

Figure 5.3(c) shows a run for a small number ($N = 4$) of large ($K = 10$) blocks. The mutation rate is $q = 0.01$ and the population size is again $M = 150$. As in all three other runs we see that the average fitness goes through epochs punctuated by rapid increases of average fitness. We also see that the best fitness in the population jumps up several times before the population fixates on strings with higher fitness. The GA takes about 2×10^5 fitness function evaluations on average to reach the global optimum for these parameter settings. In this run, the GA just happened to have taken about 2.5×10^5 fitness function evaluations.

Finally, Fig. 5.3(d) shows a run with a large number ($N = 10$) of smaller ($K = 5$) blocks. The mutation rate is $q = 0.008$ and the population size is $M = 100$. Notice that in this run, the best fitness in the population alternates several times between fitnesses 7, 8, and 9 before it reaches the global fitness optimum of 11. After it has reached the global optimum for several time steps, the global optimum disappears again and the best fitness in the population alternates between 9 and 10 from then on. It is notable that this intermittent behavior of the best fitness is barely discernible in the behavior of the average fitness. It appears to be lost in the "noise" of the average fitness fluctuations. The GA takes about 10^5 fitness function evaluations on average at these parameter settings to reach the global optimum; while in this particular run the GA took only 6.3×10^4 fitness function evaluations.

Again, we stress that there are large fluctuations in the total number of fitness evaluations to reach the global optimum between runs. One tentative conclusion is that, if one is only going to use a GA for a few runs on a specific problem, there is a relatively large operating regime in parameter space for which the GA's performance is statistically equivalent. On the one hand, the large fluctuations in the GA's search dynamics make it hard to predict for a single run how long it is going to take to reach the global optimum. On the other hand, as we will see below, this prediction is not very sensitive to the precise parameter settings.

5.6 Statistical Dynamics of Evolutionary Search

In Refs [137] and chapter 4 we developed the statistical dynamics of genetic algorithms to analyze the behavioral regimes of a GA searching the Royal Road fitness function. The analysis here builds on those results. Due to the strong similarities we will only briefly introduce this analytical approach to evolutionary dynamics. The reader is referred to chapter 4 for an extensive and detailed exposition. There the reader also will find a review of the connections and similarities of our work with the alternative methodologies for GA theory developed by Vose and collaborators [109, 139, 138], by Prügell-Bennett, Rattray, and Shapiro [117, 118, 119, 116], in the theory of molecular evolution [32, 33], and in mathematical population genetics.

From a microscopic point of view, the state of an evolving population is only fully described when a list \mathcal{S} of all genotypes with their frequencies of occurrence in the population is given. On the microscopic level, the evolutionary dynamics is implemented via the conditional transition probabilities $\Pr(\mathcal{S}'|\mathcal{S})$ that the population at the next generation will be the microscopic state \mathcal{S}' ; see Refs. [69] and [109]. For any reasonable genetic representation, however, there will be an enormous number of these microscopic states \mathcal{S} and their transition probabilities. This makes it almost impossible to quantitatively study the dynamics at this microscopic level.

More practically, a full description of the dynamics on the level of microscopic states \mathcal{S} is neither useful nor typically of interest. One is much more likely to be concerned with relatively coarse statistics of the dynamics, such as the evolution of the best and average fitness in the population or the waiting times for evolution to produce a genotype of a certain quality. The result is that quantitative mathematical analysis faces the task of finding a description of the evolutionary dynamics that is simple enough to be tractable numerically or analytically and that, moreover, facilitates the predicting the quantities of interest to a practitioner.

With these issues in mind, we specify the state of the population at any time by some relatively small set of *macroscopic* variables. Since this set of variables intentionally ignores vast amounts of microscopic detail, it is generally impossible to exactly describe the GA's dynamics in terms of these macroscopic variables. To achieve the benefits of a coarser description, however, we assume that, given the state of the macroscopic variables, the population has equal probabilities to be in any of the microscopic states consistent with the specified macroscopic state. This *maximum entropy* assumption lies at the heart of statistical mechanics in physics.

We assume in addition that in the limit of infinite population size, the resulting equations of motion for the macroscopic variables become closed. That is, for infinite populations, we assume that we can exactly predict the state of the macroscopic variables at the next generation, given the present state of the macroscopic variables. This limit is analogous to the thermodynamic limit in statistical mechanics. The corresponding assumption is analogous to *self-averaging* of the dynamics in this limit.

The key, and as yet unspecified step, in developing such a thermodynamic model of evolutionary dynamics is to find an appropriate set of macroscopic variables that satisfies the above assumptions. In practice this is difficult. Ultimately, the suitability of a set of macroscopic variables has to be verified by comparing theoretical predictions with experimental measurements. In choosing such a set of macroscopic variables one is

guided by our knowledge of the fitness function and the search’s genetic operators.

To see how this choice is made imagine that strings can have only two possible values for fitness, f_A and f_B . Assume also that under mutation all strings of type A are equally likely to turn into type- B strings and that all strings of type B have equal probability to turn into strings of type A . In this situation, it is easy to see that we can take the macroscopic variables to be the relative proportions of A strings and B strings in the population. Any additional microscopic detail, such as the number of 1s in the strings, is not required or relevant to the evolutionary dynamics. Neither selection nor mutation distinguish different strings within the sets A and B on the level of the proportions of A ’s and B ’s they produce in the next generation.

Similarly, our approach describes the state of the population at any time only by the distribution of *fitness* in the population. That is, we group strings into equivalence classes of equal fitness and assume that, on the level of the fitness distribution, the dynamics treats all strings within a fitness class as equal. At the macroscopic (fitness) level of the dynamics, we know that a string of fitness n has the first $n - 1$ blocks aligned and the n th block in one of the $2^K - 1$ other unaligned configurations. The maximum entropy assumption specifies that for all strings of fitness n , the n th block is *equally likely* to be in any of the $2^K - 1$ unaligned configurations and that each of the blocks $n + 1$ through N are equally likely to be in any of their possible 2^K block configurations.

Various reasons suggest that the maximum entropy approximation will not be valid in practice. For example, the convergence due to finite-population sampling makes it hard to believe that all unaligned block configurations in all strings are random and *independent*. For large populations, fortunately, the assumption is compelling. In fact, in the limit of very large population sizes, typically $M > 2^L$, the GA’s dynamics on the level of fitness distributions accurately captures the fitness distribution dynamics found experimentally (see chapter 4). In any case, we will solve explicitly for the fitness distribution dynamics in the limit of infinite populations using our maximum entropy assumption and then show how this solution can be used to solve for the finite-population dynamics.

The essence of our statistical dynamics approach then is to describe the population state at any time during a GA run by a relatively small number of macroscopic variables—variables that in the limit of infinite populations self-consistently describe the dynamics at their own level. After obtaining this infinite population dynamics explicitly, we then use it to solve for the GA’s dynamical behaviors with finite populations.

Employing the maximum entropy principle and focusing on fitness distributions is also found in an alternative statistical mechanics approach to GA dynamics developed by Prügel-Bennett, Rattray, and Shapiro [117, 118, 119]. In their approach, however, maximum entropy is assumed with respect to the ensemble of entire GA *runs*. Specifically, in their analysis the *average* dynamics, averaged over many runs, of the first few cumulants of the fitness distribution are predicted theoretically. In more recent work, the model has been used to calculate ensemble cumulants, such as variance of the average dynamics over an ensemble of runs [116]. These macroscopic variables are generally unsuited, however, to capture the epochal behavior illustrated in Fig. 5.3 of the previous section. In contrast, our statistical dynamics approach applies maximum entropy only to the population’s *current* state, given its current fitness distribution. The result is that for finite populations we do not assume, as we do for infinite populations, that the GA

dynamics is self-averaging. That is, two runs, with equal fitness distributions \vec{P} at time t , are not assumed to have the same future macroscopic behavior. They are assumed only to have the same probability distribution of possible futures.

5.6.1 Generation Operator

The macroscopic state of the population is determined by its fitness distribution, denoted by a vector $\vec{P} = \{P_1, P_2, \dots, P_{N+1}\}$, where the components $0 \leq P_n \leq 1$ are the proportion of individuals in the population with fitness $n = 1, 2, \dots, N + 1$. We refer to \vec{P} as the *phenotypic quasispecies*, following its analog in molecular evolution theory [32, 33, 34]. Since \vec{P} is a distribution, we have the normalization condition:

$$\sum_{n=1}^{N+1} P_n = 1. \quad (5.2)$$

The average fitness $\langle f \rangle$ of the population is given by:

$$\langle f \rangle = \sum_{n=1}^{N+1} n P_n. \quad (5.3)$$

In the limit of infinite populations and under our maximum entropy assumption, we can construct a generation operator \mathbf{G} that maps the current fitness distribution $\vec{P}(t)$ deterministically into the fitness distribution $\vec{P}(t + 1)$ at the next time step; that is,

$$\vec{P}(t + 1) = \mathbf{G}[\vec{P}(t)]. \quad (5.4)$$

The operator \mathbf{G} consists of a selection operator \mathbf{S} and a mutation operator \mathbf{M} :

$$\mathbf{G} = \mathbf{M} \cdot \mathbf{S}. \quad (5.5)$$

The selection operator encodes the fitness-level effect of selection on the population; and the mutation operator, the fitness-level effect of mutation. Below we construct these operators for our GA and the Royal Staircase fitness function explicitly. For now we note that the infinite population dynamics can be obtained by iteratively applying the operator \mathbf{G} to the initial fitness distribution $\vec{P}(0)$. Thus, the macroscopic equations of motion are formally given by

$$\vec{P}(t) = \mathbf{G}^{(t)}[\vec{P}(0)]. \quad (5.6)$$

Recalling Eq. (5.1) it is easy to see that the initial fitness distribution $\vec{P}(0)$ is given by:

$$P_n(0) = 2^{-K(n-1)} (1 - 2^{-K}), \quad 1 \leq n \leq N, \quad (5.7)$$

and

$$P_{N+1}(0) = 2^{-KN}. \quad (5.8)$$

As shown in Ref. [137] and chapter 4, despite \mathbf{G} 's nonlinearity, it can be easily linearized such that the t th iterate $\mathbf{G}^{(t)}$ can be directly obtained by solving for the eigenvalues and eigenvectors of \mathbf{G} .

For very large populations ($M > 2^L$) the dynamics of the fitness distribution obtained from GA simulation experiments is accurately predicted by iteration of the operator \mathbf{G} . (See Ref. [137] and chapter 4.) It is noteworthy, though, that this “infinite” population dynamics is qualitatively very different from the behavior shown in Fig. 5.3. For large populations strings of *all* fitnesses are present in the initial population and the average fitness increases smoothly and monotonically to an asymptote over a small number of generations. (This limit is tantamount to an exhaustive search, requiring as it does $\mathcal{O}(2^L)$ fitness function evaluations.) Despite this seeming lack of utility, in the next section we show how to use the infinite-population dynamics given by \mathbf{G} to describe the finite-population behavior.

5.6.2 Finite-Population Dynamics

There are two important differences between the infinite-population dynamics and that with finite populations. The first is that with finite populations the components P_n cannot take on arbitrary values between 0 and 1. Since the number of individuals with fitness n in the population is necessarily integer, the values of P_n are quantized to multiples of $1/M$. The space of allowed finite-population fitness distributions thus turns into a regular lattice in $N + 1$ dimensions with a lattice spacing of $1/M$ within the simplex specified by normalization (Eq. (5.2)). Second, the dynamics of the fitness distribution is no longer deterministic. In general, we can only determine the conditional probabilities $\Pr[\vec{Q}|\vec{P}]$ that a certain fitness distribution \vec{P} leads to another \vec{Q} in the next generation.

Fortunately, the probabilities $\Pr[\vec{Q}|\vec{P}]$ are simply given by a multinomial distribution with mean $\mathbf{G}[\vec{P}]$, which in turn is the result of the action of the infinite-population dynamics. This can be understood in the following way. In creating the population for the next generation individuals are selected, copied, and mutated, M times from the *same* population \vec{P} . This means that for each of the M selections there are equal probabilities q_n to produce an individual of fitness n . The probabilities q_n give the *expected* proportions of fitness n strings in the next generation. The actual proportions Q_n of individuals with fitness n in the next generation are thus given by a multinomial sample of size M with from the distribution of expected proportions q_n . Moreover, since for an infinite population the expected proportions q_n are equal to the actual proportions Q_n , the probabilities q_n are equal to the components of the infinite population distribution $\mathbf{G}[\vec{P}]$.

Putting these observations together, if we write $Q_n = m_n/M$, with $0 \leq m_n \leq M$ being integers that sum to M , we have:

$$\Pr[\vec{Q}|\vec{P}] = M! \prod_{n=1}^{N+1} \frac{\left(\mathbf{G}_n[\vec{P}]\right)^{m_n}}{m_n!}. \quad (5.9)$$

In mathematical population genetics, such multinomial sampling Markov models are known as Wright-Fisher models [69, pp. 66-70]. We see that for any finite-population fitness distribution \vec{P} the operator \mathbf{G} still gives the GA’s *average* dynamics over one time step, since the *expected* fitness distribution at the next time step is $\mathbf{G}[\vec{P}]$. Note that the components $\mathbf{G}_n[\vec{P}]$ need not be multiples of $1/M$. Therefore the *actual* fitness

distribution \vec{Q} at the next time step is not $\mathbf{G}[\vec{P}]$, but is instead one of the lattice points of the finite-population state space. Since the variance around the expected distribution $\mathbf{G}[\vec{P}]$ is proportional to $1/M$, \vec{Q} is likely to be close to $\mathbf{G}[\vec{P}]$.

5.6.3 Epochal Dynamics

For finite populations, the expected change $\langle d\vec{P} \rangle$ in the fitness distribution over one generation is given by:

$$\langle d\vec{P} \rangle = \mathbf{G}[\vec{P}] - \vec{P}. \quad (5.10)$$

Assuming that some component $\langle dP_i \rangle$ is much smaller than $1/M$, the actual change in component P_i is likely to be $dP_i = 0$ for a long succession of generations. That is, if the size of the *flow* $\langle dP_i \rangle$ in some direction i is much smaller than the lattice spacing ($1/M$) for the finite population, we expect the fitness distribution to not change in direction (fitness) i . In Ref. [137] and chapter 4 we showed this is the mechanism that causes epochal dynamics for finite populations. More formally, each epoch n of the dynamics corresponds to the population being restricted to a region in the n -dimensional lower-fitness subspace of fitnesses 1 to n of the macroscopic state space. Stasis occurs because the flow out of this subspace is much smaller than the finite-population induced lattice spacing.

As Fig. 5.3 illustrates, each epoch in the average fitness is associated with a constant value of the best fitness in the population. More detailed experiments reveal that not only is $\langle f \rangle$ constant on average during the epochs, in fact, up to fluctuations, the entire fitness distribution \vec{P} is constant during the epochs. We denote by \vec{P}^n the average fitness distribution during the generations when n is the highest fitness in the population. As was shown in chapter 4, each epoch fitness distribution \vec{P}^n is the unique fixed point of the operator \mathbf{G} restricted to the n -dimensional subspace of strings with $1 \leq f \leq n$. That is, if \mathbf{G}^n is the projection of the operator \mathbf{G} onto the n -dimensional subspace of fitnesses from 1 up to n , we have:

$$\mathbf{G}^n[\vec{P}^n] = \vec{P}^n. \quad (5.11)$$

The average fitness f_n in epoch n is then given by:

$$f_n = \sum_{j=1}^n jP_j^n. \quad (5.12)$$

Thus, the fitness distribution \vec{P}^n during epoch n is obtained by finding the fixed points of \mathbf{G} restricted to the first n dimensions of the fitness distribution space. We will construct the operators \mathbf{G}^n explicitly below for our GA and solve analytically for the epoch fitness distributions \vec{P}^n as a function of n , K , and q .

The global dynamics can be viewed as an incremental discovery of successively more dimensions of the fitness distribution space. In most realistic settings, it is typically the case that population sizes M are much smaller than 2^L . Initially, then, only strings of low fitness are present in the initial population, i.e. see Eq. (5.7). The population then stabilizes on the epoch fitness distribution \vec{P}^n corresponding to the best fitness n in the

initial population. The fitness distribution fluctuates around \vec{P}^n until a string of fitness $n + 1$ is discovered and spreads through the population. The population then settles into fitness distribution \vec{P}^{n+1} until a string of fitness $n + 2$ is discovered, and so on, until the global optimum at fitness $N + 1$ is found. In this way, the global dynamics can be seen as stochastically hopping between the different epoch distributions \vec{P}^n .

Whenever mutation creates a string of fitness $n + 1$, this string may either disappear before it spreads (seen as the isolated jumps in best fitness in Fig. 5.3) or it may spread, leading the population to fitness distribution \vec{P}^{n+1} . We have been calling the latter process an *innovation*. Fig. 5.3 also showed that it is possible for the population to fall from epoch n (say) down to epoch $n - 1$. This happens when, due to fluctuations, all individuals of fitness n are lost from the population. We refer to this as a *destabilization* of epoch n . For some parameter settings, such as shown in Figs. 5.3(a) and 5.3(c), this is very rare. The time for the GA to reach the global optimum is mainly determined by the time it takes to discover strings of fitness $n + 1$ in each epoch n . For other parameter settings, however, such as in Figs. 5.3(b) and 5.3(d), the destabilizations play an important role in how the GA reaches the global optimum. In these regimes, destabilization must be taken into account in calculating search times. This task is accomplished in the next chapter.

5.6.4 Selection Operator

We now explicitly construct the generation operator \mathbf{G} for the limit of infinite population size by constructing the selection operator \mathbf{S} and mutation operator \mathbf{M} . First, we consider the effect of selection on the fitness distribution. Since we are using fitness-proportionate selection, the proportion P_i^s of strings with fitness i after selection is proportional to i and to the fraction P_i of strings with fitness i before selection; that is,

$$P_i^s = c i P_i . \quad (5.13)$$

The constant c can be determined by demanding that the distribution remains normalized:

$$1 = \sum_{i=1}^n P_i^s = c \sum_{i=1}^{N+1} i P_i . \quad (5.14)$$

Since the average fitness $\langle f \rangle$ of the population is given by:

$$\langle f \rangle = \sum_{i=1}^{N+1} i P_i , \quad (5.15)$$

we have

$$P_i^s = \frac{i P_i}{\langle f \rangle} . \quad (5.16)$$

We can thus define a (diagonal) operator \mathbf{S} that works on a fitness distribution \vec{P} and produces the fitness distribution \vec{P}^s after selection by:

$$\left(\mathbf{S} \cdot \vec{P} \right)_i = \sum_{j=1}^{N+1} \frac{\delta_{ij} j}{\langle f \rangle} P_j . \quad (5.17)$$

Notice that this operator is nonlinear since, by Eq. (5.15), the average fitness $\langle f \rangle$ is a function of the distribution \vec{P} on which the operator acts.

5.6.5 Mutation Operator

The components of the mutation operator \mathbf{M} give the probabilities \mathbf{M}_{ij} that a string of fitness j is turned into a string with fitness i under mutation. These components naturally fall into three categories. First, consider the components \mathbf{M}_{ij} with $i < j$. These strings are obtained if mutation leaves the first $i - 1$ blocks of the string unaltered and disrupts the i th block in the string. The effect of mutation on the blocks $i + 1$ through N is immaterial for this transition. Multiplying the probabilities that the preceding $i - 1$ blocks remain aligned and that the i th block becomes unaligned we have:

$$\mathbf{M}_{ij} = (1 - q)^{(i-1)K} (1 - (1 - q)^K), \quad i < j. \quad (5.18)$$

Second, the diagonal components \mathbf{M}_{jj} are obtained when mutation leaves the first $j - 1$ blocks unaltered and does *not* mutate the j th block to be aligned. The maximum entropy assumption says that the j th block is random and so the probability P_a that mutation will change the unaligned j th block to an aligned block is given by:

$$P_a = \frac{1 - (1 - q)^K}{2^K - 1}. \quad (5.19)$$

This is the probability that at least one mutation will occur in the block times the probability that the mutated block will be in the correct configuration. Thus the diagonal components are given by:

$$\mathbf{M}_{jj} = (1 - q)^{(j-1)K} (1 - P_a). \quad (5.20)$$

Finally, we calculate the probabilities for increasing-fitness transitions \mathbf{M}_{ij} with $i > j$. Depending on the approximation used, these transition probabilities might depend on the value n of the current best fitness in the population. As we show below in section 5.8, the reason for this is that all individuals in the population during epoch n are relatively recent descendants of strings in the highest fitness class n . Therefore, the blocks j through $n - 1$ in strings with fitness $j < n$ during epoch n tend to reflect memory of the fact that they are relatively recent descendants of a string with fitness n . This will of course influence the probabilities \mathbf{M}_{ij} with $i > j$ differently in each epoch n . It turns out, however, that for the theoretical predictions of our chosen observables—such as, epoch durations and total number of fitness function evaluations—the maximum entropy assumption gives results that are very similar to results obtained with epoch-dependent approximations.

Using the maximum entropy assumption, we obtain epoch-independent transition probabilities \mathbf{M}_{ij} by assuming that all the blocks j through N are random. The j th block is equally likely to be in any of $2^K - 1$ unaligned configurations. All succeeding blocks are equally likely to be in any one of the 2^K configurations, including the aligned one. In order for a transition to occur from state j to i , first of all the $j - 1$ aligned blocks have to remain aligned, then the j th block has to become aligned through the mutation.

The latter has probability P_a . Furthermore, the following $i - j - 1$ blocks all have to be aligned. Finally, the i th block has to be unaligned. Putting these together, we find that:

$$\mathbf{M}_{ij} = (1 - q)^{(j-1)K} P_a \left(\frac{1}{2^K} \right)^{i-j-1} \left(1 - \frac{1}{2^K} \right), \quad i > j. \quad (5.21)$$

As a technical aside, note that for the case $i = N + 1$ the last factor does not appear.

The generation operator \mathbf{G} is obtained by taking the product of the selection operator \mathbf{S} with the mutation operator \mathbf{M} :

$$\mathbf{G} = \mathbf{M} \cdot \mathbf{S}. \quad (5.22)$$

The restricted generation operators \mathbf{G}^n are obtained by setting all components with $i > n$ or $j > n$ to zero, where the component indices of the mutation and selection operators run from 1 to n .

5.7 Quasispecies Distributions and Epoch Fitness Levels

In this section, we solve for the metastable fitness distribution \vec{P}^n during each epoch n . In section 5.9 we will use the \vec{P}^n to calculate the waiting times for an innovation from epoch n to $n + 1$ to occur.

The quasispecies fitness distribution \vec{P}^n is given by a fixed point of the operator \mathbf{G}^n . To obtain this fixed point we linearize the generation operator by taking out the factor $\langle f \rangle$, thereby defining a new operator $\tilde{\mathbf{G}}^n$ via:

$$\mathbf{G}^n = \frac{1}{\langle f \rangle} \tilde{\mathbf{G}}^n. \quad (5.23)$$

The operator $\tilde{\mathbf{G}}^n$ is just an ordinary (linear) matrix operator and the quasispecies fitness distribution is nothing other than the principal eigenvector of this matrix. The principal eigenvalue f_n of $\tilde{\mathbf{G}}^n$ is the average fitness of the quasispecies distribution. In this way, obtaining the quasispecies distribution reduces to calculating the principal eigenvector of the matrix $\tilde{\mathbf{G}}^n$. Again the reader is referred to chapter 4.

As in Ref. [137] and chapter 4, the local stability of the epochs can be analyzed by calculating the eigenvalues and eigenvectors of the Jacobian matrix $\mathbf{D}\mathbf{G}^n$ around each epoch fitness distribution \vec{P}^n . The components $\mathbf{D}\mathbf{G}_{ij}^n$ of the Jacobian around epoch n are given by:

$$\mathbf{D}\mathbf{G}_{ij}^n = \left[\frac{\partial \mathbf{G}_i(\vec{P})}{\partial P_j} \right]_{\vec{P}=\vec{P}^n} = \frac{\tilde{\mathbf{G}}_{ij}^n - j\vec{P}_j^n}{f_n}. \quad (5.24)$$

Just as in chapter 4, the eigenvectors \vec{U}^i of the Jacobian are given by $\vec{U}^i = \vec{P}^i - \vec{P}^n$, with corresponding eigenvalues $e_i^n = f_i/f_n$. Thus, the spectra of the Jacobian matrices are simply determined by the spectrum of the generation operator $\tilde{\mathbf{G}}^n$. The eigenvalues e_i^n determine the bulk of the GA's behavior. Since the matrix $\tilde{\mathbf{G}}^n$ is generally of modest size, i.e. its dimension is determined by the number of blocks N , we can easily obtain

5.7 Quasispecies Distributions and Epoch Fitness Levels

numerical solutions for the epoch fitnesses f_n and the epoch quasispecies distributions \vec{P}^n . At the same time one also obtains the eigenvalues e_i^n and eigenvectors \vec{U}^i of the Jacobian.

For a clearer understanding of the functional dependence of the epoch fitness distributions on the GA's parameters, however, we will now develop analytical approximations to the epoch fitness levels f_n and quasispecies distributions \vec{P}^n .

In order to explicitly determine the form of the quasispecies distribution \vec{P}^n during epoch n we will approximate the matrix $\tilde{\mathbf{G}}^n$. As we saw in section 5.6.5, the components M_{ij} (and so of \mathbf{G}) naturally fall into three categories. Those with $i < j$, those with $i > j$, and those on the diagonal $i = j$. Components with $i > j$ involve at least one block becoming aligned through mutation. For block lengths K that are not too small, these components are generally much smaller than those only involving the destruction of aligned blocks or for which there is no change in the blocks. We therefore approximate $\tilde{\mathbf{G}}^n$ by neglecting terms proportional to P_a . The distribution \vec{P}^n is then simply the result of a balance between selection expanding the higher fitness strings and mutation disrupting blocks, whereby high fitness strings turn into lower fitness strings.

Under this approximation we have for the components of $\tilde{\mathbf{G}}^n$:

$$\tilde{\mathbf{G}}_{ij}^n = j(1-q)^{(i-1)K}(1-(1-q)^K), \quad i < j, \quad (5.25)$$

and

$$\tilde{\mathbf{G}}_{jj}^n = j(1-q)^{(j-1)K}. \quad (5.26)$$

The components with $i > j$ are now all zero. The result is that $\tilde{\mathbf{G}}^n$ is upper triangular. As is well known in general matrix theory, the eigenvalues of an upper triangular matrix are given by its diagonal components. Therefore, the average fitness f_n in epoch n , which is given by the largest eigenvalue, is equal to the largest diagonal component of $\tilde{\mathbf{G}}^n$. That is,

$$f_n = n(1-q)^{(n-1)K}. \quad (5.27)$$

Notice that the matrix elements only depend on q and K through the effective parameter λ defined by:

$$\lambda = (1-q)^K. \quad (5.28)$$

λ is just the probability that a block will not be mutated.

The principal eigenvector \vec{P}^n is the positive and normalized solution of the equation:

$$\sum_{j=1}^n \left(\tilde{\mathbf{G}}_{ij}^n - f_n \delta_{ij} \right) P_j^n = 0. \quad (5.29)$$

Since the components of $\tilde{\mathbf{G}}^n$ depend on λ in such a direct way, we can analytically solve for this eigenvector. We find that the quasispecies components are given by:

$$P_i^n = \frac{(1-\lambda)n\lambda^{n-1-i}}{n\lambda^{n-1-i}-i} \prod_{j=1}^{i-1} \frac{n\lambda^{n-j}-j}{n\lambda^{n-1-j}-j}. \quad (5.30)$$

For the component P_n^n this reduces to

$$P_n^n = \prod_{j=1}^{n-1} \frac{n\lambda^{n-j} - j}{n\lambda^{n-1-j} - j}. \quad (5.31)$$

The above equation can be re-expressed in terms of the epoch fitness levels f_j :

$$P_n^n = \lambda^{n-1} \prod_{j=1}^{n-1} \frac{f_n - f_j}{f_n - \lambda f_j}. \quad (5.32)$$

Note that it is straightforward to increase the accuracy of our analytical approximations by including terms proportional to P_a in the matrix $\tilde{\mathbf{G}}^n$ and then treating these terms as a perturbation to the upper triangular matrix. Using standard perturbation theory, one can then obtain corrections due to block alignments. We will not pursue this here, however, since the current approximation is accurate enough for the optimization analysis.

5.8 Quasispecies Genealogies and Crossover's Role

Before continuing on to solve this problem, we digress slightly at this point for two reasons. First, we claimed in a previous section that *all* individuals in the population during epoch n are descendants of strings with fitness n . We will demonstrate this now by considering the genealogies of strings occurring in a quasispecies population \vec{P}^n . Second, since the argument is quite general and only depends on the effects of selection, it has important implications for population structure in metastable states (such as fitness epochs) and, more specifically, for the role of crossover in epochal evolutionary search.

For the n th epoch, let the set of “suboptimal” strings be all those with fitness lower than n ; their proportion is simply $1 - P_n^n$. This proportion is constant on average during epoch n . Over one generation, the suboptimal strings in the next generation will be either descendants of suboptimal strings in the current generation or mutant descendants of “optimal” strings with fitness n . Let r denote the average number of offspring per suboptimal individual. The fact that the total proportion of suboptimal strings remains constant gives us the equation:

$$1 - P_n^n = r(1 - P_n^n) + \frac{(1 - \mathbf{M}_{nn})n}{f_n} P_n^n. \quad (5.33)$$

The last term is the proportion of individuals of fitness n that are selected and mutate to suboptimal strings. From this we can solve for r and find:

$$r = 1 - \frac{(1 - \mathbf{M}_{nn})n P_n^n}{f_n(1 - P_n^n)} = \frac{1 - \lambda^{1-n} P_n^n}{1 - P_n^n}, \quad (5.34)$$

where we have used the previous analytical approximations to $n\mathbf{M}_{nn}$ and f_n , Eqs. (5.26) and (5.27), in the last equality.

We see that in every generation only a fraction $r < 1$ of the suboptimal individuals are descendants of suboptimal individuals in the previous generation. This means that after t generations, only a fraction r^t of the suboptimal individuals are the terminations of lineages that solely consist of suboptimal strings. There is a fraction of $1 - r^t$ strings that have one or more ancestors with fitness n in the t preceding generations. After a certain number of generations t_c the fraction r^{t_c} becomes so small that less than one individual (on average) has only suboptimal ancestors. That is, after approximately t_c generations in epoch n , the whole quasispecies will consist of strings that are descendants of a string with fitness n somewhere in the past. Explicitly, we find that

$$t_c = \frac{\log [M(1 - P_n^n)]}{\log \left[\frac{1 - \lambda^{1-n} P_n^n}{1 - P_n^n} \right]}. \quad (5.35)$$

As expected, t_c is proportional to the logarithm of the total number $M(1 - P_n^n)$ of suboptimal strings in the quasispecies.

The above result can be generalized to the case in which the suboptimal strings are defined to be all those with fitnesses 1 to $n - i$. One can then calculate the time until all strings in these classes are descendants of strings with fitness $n - i + 1$ to n to find that these lower classes are taken over even faster by descendants of strings with fitness n .

The preceding result is significant for a conceptual understanding of the structure of epoch populations. All suboptimal individuals in the population have an ancestor of optimal fitness that is a relatively small number of generations in the recent past. The result is that in genotype space all suboptimal individuals are always relatively close to some individual of optimal fitness. The suboptimal individuals never wander far from the individuals of optimal fitness. More precisely, the average duration of suboptimal lineages is $1/(1 - r)$ generations. That is, all suboptimal individuals are typically found within a Hamming distance of $Lq/(1 - r)$ from optimal-fitness individuals.

The individuals with optimal fitness, however, *can* wander through genotype space as long as they do not leave the neutral network of optimal fitness strings—those with fitness n in epoch n . If the population is to traverse large regions of genotype space in order to discover a string of fitness larger than n , it must do so along this neutral network. In short, this is the reason we believe the existence of neutral networks, consisting of approximately equal fitness strings that percolate through large parts of the genotype space, is so important for evolutionary search; cf. Ref. [79]. If strings of fitness n were to form a small island in a sea of lower fitness strings that are at relatively large Hamming distances from islands with fitness higher than n , then there would be little chance that a suboptimal fitness mutant will drift far enough from the island of fitness n strings to discover another island with higher fitness.

This result—that all strings in the metastable population are relatively recent descendants of strings in the highest fitness class—generalizes to other selection schemes such as elite selection, rank selection, and tournament selection. Furthermore, this result also provides some insight into the effects of adding crossover to evolutionary search algorithms on problems that show epochal evolution. Let's assume that we add crossover to our current GA; see also the discussion of similar crossover experiments in section 4.6.5. The initial population still has a distribution of fitnesses given by Eq. (5.7). The best fitness in the initial population might be (say) 3, corresponding to the first two

blocks being aligned and the third block unaligned. It is unlikely that crossovers will lead quickly to strings of fitness 4. Although the initial population will have strings with the 3rd block aligned, these strings are very unlikely to also have the first two blocks aligned. This means that these strings have low fitness and so are unlikely to be selected as the parent of a crossover event. Moreover, relatively quickly, the entire population will become descendants of strings with fitness 3 that, by definition, have the third block unaligned. Crossover events will thus almost never lead to the creation of strings of higher fitness; at least not by combining building blocks [74].

The positive contribution of crossover is that an aligned block may be formed from two parents each with an unaligned 3rd block if the crossover point falls within the 3rd block and if the resulting complementary subblocks are themselves aligned. The negative effect is that crossover may also combine lower fitness strings with higher fitness strings so as to produce two lower fitness offspring. Thus, with nothing else said or added, one concludes that the effect of crossover in our GA is marginal. Experiments with single-point crossover confirm this. The global optimum is found somewhat more quickly, but the improvement in search time is very small.

These arguments are specific to the Royal Staircase (and also Royal Road) classes of fitness function. However, for search dynamics exhibiting epochal evolution it is generally the case that the population structure is a cloud of strings *localized* around strings of the current best fitness during the epochs. Therefore, the effect of crossover will mostly be to increase the amount of mixing *within* the quasispecies cloud. That is, during the epochs crossover acts as a local mixing operator very much as mutation does. Note that this is not a result of our fitness function demanding that blocks be found consecutively.

In fact, by the same arguments as sketched above, crossover only marginally improves the search on Royal Road functions (chapter 4), which were specifically *designed* as fitness functions for which crossover should be most beneficial [105]. As was shown in Ref. [134], for simple genetic algorithms, the population size must be set unreasonably high in order for crossover to produce globally optimal strings before the population has converged. In order for recombination operators to be useful on these types of building block functions, more elaborate genetic algorithms—such as, messy GAs [63, 62]—must be constructed. However, as shown below, building-block-based fitness functions, such as the Royal Staircase and the Royal Road, can be efficiently searched with mutation-only genetic algorithms, provided that the mutation rate is set correctly.

5.9 Mutation Rate Optimization

In the previous sections we argued that the GA's behavior can be viewed as stochastically hopping from epoch to epoch until the search discovers a string with fitness $N + 1$. Assuming for the moment that the total time to reach this global optimum is dominated by the time the GA spends in the epochs, we will derive a way to tune the mutation rate q such that the time the GA spends in an epoch is minimized.

During epoch n no string in the population has the n th block aligned. Thus, the main contribution to the waiting time in epoch n is given by the time it takes for a mutant with the n th block set correctly to appear. As we have seen, the probability P_a to mutate a

5.9 Mutation Rate Optimization

single block such that it becomes aligned is given by:

$$P_a = \frac{1 - (1 - q)^K}{2^K - 1} = \frac{1 - \lambda}{2^K - 1}, \quad (5.36)$$

where again λ is the probability that a block will not be mutated at all. Obviously, the higher P_a , the more likely mutation is to produce a new aligned block. Every generation each individual in the population has a probability P_a of aligning the n th block. Aligning the n th block only creates a string of fitness $n + 1$ when all the $n - 1$ blocks to its left are aligned as well. That is, only the fraction P_n^n of the population with fitness n will produce a fitness $n + 1$ string by aligning the n th block. Therefore, the probability C_{n+1} that a string of fitness $n + 1$ will be created over one generation is given by:

$$C_{n+1} \approx M P_n^n P_a. \quad (5.37)$$

Our claim is that, as a first approximation, maximizing C_{n+1} minimizes the number of generations the population spends in epoch n .

In section 5.7 we derived an analytic approximation, Eq. (5.31), to the proportion P_n^n of individuals in the highest fitness class during epoch n as a function of $\lambda = (1 - q)^K$. Since P_a also only depends on q through λ and λ is monotonic in q , we can maximize C_{n+1} as a function of λ instead of q . Ignoring proportionality constants, the function to maximize is:

$$C_{n+1} \propto (1 - \lambda) P_n^n(\lambda). \quad (5.38)$$

Using Eq. (5.31) for the dependence of P_n^n on λ and differentiating the above function C_{n+1} with respect to λ , we find that the optimum λ_o satisfies:

$$\frac{n(1 - \lambda_o)}{\lambda_o} \left[\sum_{i=1}^{n-1} \frac{i \lambda_o^i}{n \lambda_o^i - n + i} - \frac{(i-1) \lambda_o^{i-1}}{n \lambda_o^{i-1} - n + i} \right] = 1. \quad (5.39)$$

The solution of this equation gives the optimum $\lambda_o(n)$ which is only a function of the epoch number n . This is an important observation, since it means that the optimal value of the mutation rate q_o takes the following general form as a function of n and K :

$$q_o = 1 - \sqrt[K]{\lambda_o(n)}. \quad (5.40)$$

Once we solve for the function $\lambda_o(n)$, we can immediately obtain the dependence of q_o on K using Eq. (5.40).

In this calculation we assumed that the waiting time for *discovering* a higher fitness string dominated the time spent in an epoch. This amounts to assuming that as soon as a string of fitness $n + 1$ is created, copies of this string spread through the population and the population stabilizes onto epoch $n + 1$. In fact, it is quite likely that the string with fitness $n + 1$ will be lost through a deleterious mutation in the aligned blocks or via sampling, before it gets a chance to spread in the population. Recall the transitory jumps in the best fitness seen in Fig. 5.3. In chapter 4 we used a diffusion equation approximation to calculate the probability π_n that a string with fitness $n + 1$ will spread. We found that to a good approximation it is given by:

$$\pi_n = \frac{1 - \left(1 - \frac{1}{M}\right)^{2M\gamma_n + 1}}{1 - \left(1 - P_{n+1}^{n+1}\right)^{2M\gamma_n + 1}} \approx 1 - e^{-2\gamma_n}, \quad (5.41)$$

where

$$\gamma_n = \frac{f_{n+1} - f_n}{f_n}, \quad (5.42)$$

and the last approximation in Eq. (5.41) holds for relatively large population sizes ($P_n^n M \gg 1$). Equation (5.41) is an adaptation of diffusion equation results first obtained by Kimura [89]. Notice that the spreading probability π_n only depends on the relative difference of the average fitness in epoch $n + 1$ and epoch n . Using Eq. (5.27) for f_n we find:

$$\gamma_n = \left(1 + \frac{1}{n}\right) \lambda - 1. \quad (5.43)$$

Thus, we find that $\pi_n(\lambda)$ is approximately given by:

$$\pi_n(\lambda) = 1 - e^{-2\left(1 + \frac{1}{n}\right)\lambda + 2}. \quad (5.44)$$

The probability C'_{n+1} to create a string of fitness $n + 1$ that spreads through the population is given by:

$$C'_{n+1} = C_{n+1} \pi_n(\lambda). \quad (5.45)$$

Thus, taking the spreading probability π_n into account, we want to maximize C'_{n+1} in order to minimize the time spent in epoch n . Note that, also in this case, the dependence on q and K enters only through λ . For each n there is an optimal value of λ from which the optimal mutation rate can be determined as a function of K .

The optimal value λ_o in this case is the solution of:

$$\begin{aligned} \frac{n(1 - \lambda_o)}{\lambda_o} &+ \left[\sum_{i=1}^{n-1} \frac{i\lambda_o^i}{n\lambda_o^i - n + i} - \frac{(i-1)\lambda_o^{i-1}}{n\lambda_o^{i-1} - n + i} \right] \\ &+ (1 - \lambda_o) \frac{2\left(1 + \frac{1}{n}\right) e^{-2\left(1 + \frac{1}{n}\right)\lambda_o + 2}}{1 - e^{-2\left(1 + \frac{1}{n}\right)\lambda_o + 2}} = 1. \end{aligned} \quad (5.46)$$

Numerically, the solution $\lambda_o(n)$ is well approximated by:

$$\lambda_o(n) = 1 - \frac{1}{3n^{1.175}}, \quad (5.47)$$

as shown in Fig. 5.4, which plots $1 - \lambda_o$ as a function of n . The solid line is the numerical solution obtained from Eq. (5.46); the dashed line is the approximation Eq. (5.47).

For large n , using Eqs. (5.40) and (5.47) we can approximate the optimal mutation rate by:

$$q_o = \frac{1}{3(nK)n^{0.175}}. \quad (5.48)$$

Thus, the optimal mutation rate drops as a power-law in both n and K . This implies that, generally for the Royal Staircase fitness functions, the mutation rate should decrease as a GA run progresses so that the search will find the global optimum as quickly as possible. We pursue this idea more precisely elsewhere by considering an adaptive mutation rate scheme for the GA. In the following, the mutation rate is assumed to be constant during the runs.

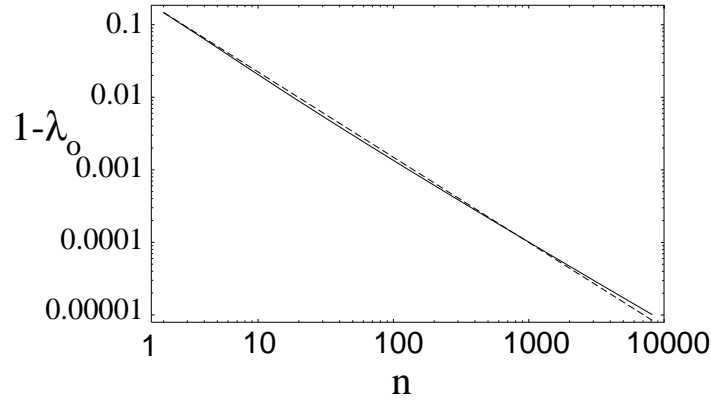


Figure 5.4: Optimal λ as a function of n . The vertical axis shows $1 - \lambda_0$ on a logarithmic scale. The horizontal axis plots n on a logarithmic scale. The solid line is the numerical solution of Eq. (5.46) and the dashed line is the approximation given by Eq. (5.47).

5.10 Fitness Function Evaluations

In the preceding sections we derived an expression for the probability C'_{n+1} to create, over one generation in epoch n , a string of fitness $n + 1$ that will stabilize by spreading through the population. From this we now estimate the total number E of fitness function evaluations the GA uses on average before an optimal string of fitness $N + 1$ is found. As a first approximation, we assume that the GA visits all epochs, that the time spent in innovations between them is negligible, and that epochs are always *stable*. By epoch stability we mean that it is highly unlikely that strings with the current highest fitness will disappear from the population through a fluctuation, once such strings have spread. These assumptions appear to hold for the parameters of Figs. 5.3(a) and 5.3(c). They may hold even for the parameters of Fig. 5.3(b), but they probably do not for Fig. 5.3(d). For the parameters of Fig. 5.3(d), we see that the later epochs ($n = 8, 9,$ and 10) easily destabilize a number of times before the global optimum is found. We develop an extension of the analysis that addresses this more complicated behavior in the next chapter.

The average number T_n of generations that the population spends in epoch n is simply $1/C'_{n+1}$ —the inverse of the probability that a string of fitness $n + 1$ will be discovered and spread through the population. For a population of size M , the number of fitness function evaluations per generation is M , so that the total number E_n of fitness function evaluations in epoch n is given by $T_n M$. More explicitly, we have from this and Eqs. (5.37) and (5.45) that:

$$E_n = \frac{2^K - 1}{(1 - \lambda) P_n^n \pi_n}. \quad (5.49)$$

This says that, given our approximations, the total number of fitness function evaluations in each epoch is independent of the population size M . The epoch lengths, measured

in generations, are inversely proportional to M , while the number of fitness function evaluations per generation is M . Substituting our analytical expression for P_n^n , Eq. (5.31), we have:

$$E_n(\lambda) = \frac{2^K - 1}{(1 - \lambda)\pi_n(\lambda)} \prod_{i=1}^{n-1} \frac{n\lambda^{n-i-1} - i}{n\lambda^{n-i} - i}. \quad (5.50)$$

The total number of fitness function evaluations $E(\lambda)$ to reach the global optimum is simply given by the sum of $E_n(\lambda)$ over all epochs n from 1 to N :

$$E(\lambda) = \sum_{n=1}^N \frac{2^K - 1}{(1 - \lambda)\pi_n(\lambda)} \prod_{i=1}^{n-1} \frac{n\lambda^{n-i-1} - i}{n\lambda^{n-i} - i}. \quad (5.51)$$

The optimal mutation rate for an entire run is obtained by minimizing the above expression for E with respect to λ . In general, this minimum E_o can only be determined numerically. However, since the last epoch N is in general by far the longest epoch, a rough approximation to the minimum of E can be obtained by taking $\lambda = \lambda_o(N)$, i.e. setting the mutation rate optimally for the last epoch. Substituting Eq. (5.47) into Eq. (5.51) we find, numerically, that the minimum number of fitness function evaluations E_o as a function of N and K is approximately given by:

$$E_o \approx 2.2 N^{3.1} 2^K. \quad (5.52)$$

Somewhat surprisingly perhaps, the block length only enters in the factor 2^K . This factor 2^K is simply the number of block states and is also exactly the ratio between the volume of the neutral nets with fitness n and $n+1$. That is, the factor 2^K accounts for the amount of neutral net volume that has to be searched before a higher fitness string is discovered. Note that in E_o the factor that accounts for the genotypic structure scales roughly as the cube of the number of blocks, *independent* of the block length K . Although this search-time scaling for our mutation-only GA seems quite efficient, it must be compared to simple random-mutation hill climbing algorithms [105] for which it can be shown that the optimum is found on average in $\approx N^2 2^K$ fitness function evaluations.

Note that we set $\pi_N = 1$ in Eq. (5.51), since by definition the GA stops at the *first* occurrence of the globally optimum string. For the last epoch N it is only necessary to create a string of fitness $N + 1$, it does not need to spread through the population.

5.11 Theory versus Experiment

We now turn to examining the dependency of E on the mutation rate q in the neighborhood of the optimum. Figure 5.5 compares Eq. (5.51) to the number (solid lines) of fitness function evaluations estimated from 250 GA runs for the four different settings of N and K of Fig. 5.3. Each plot shows E as a function of mutation rate q for a set of different population sizes M . In addition, each of the four plots in Fig. 5.5 shows, as a dashed line, the population-size independent theoretical approximation to the total number E of fitness function evaluations as a function of q .

Figure 5.5(a) has $N = 8$ blocks of $K = 8$ bits. The dependence of E on mutation rate is shown over the range from $q = 0.001$ to $q = 0.012$. The optimal mutation

5.11 Theory versus Experiment

rate occurs somewhere around $q_o = 0.006$. Each solid line shows the experimental dependence of E on q for a fixed population size. At high mutation rates, the smaller population sizes have higher E . At high mutation rates, the set of solid lines in Fig. 5.5 show, from left to right, $E(q)$ for population sizes $M = 150$, $M = 200$, $M = 250$, $M = 300$, and $M = 350$, respectively.

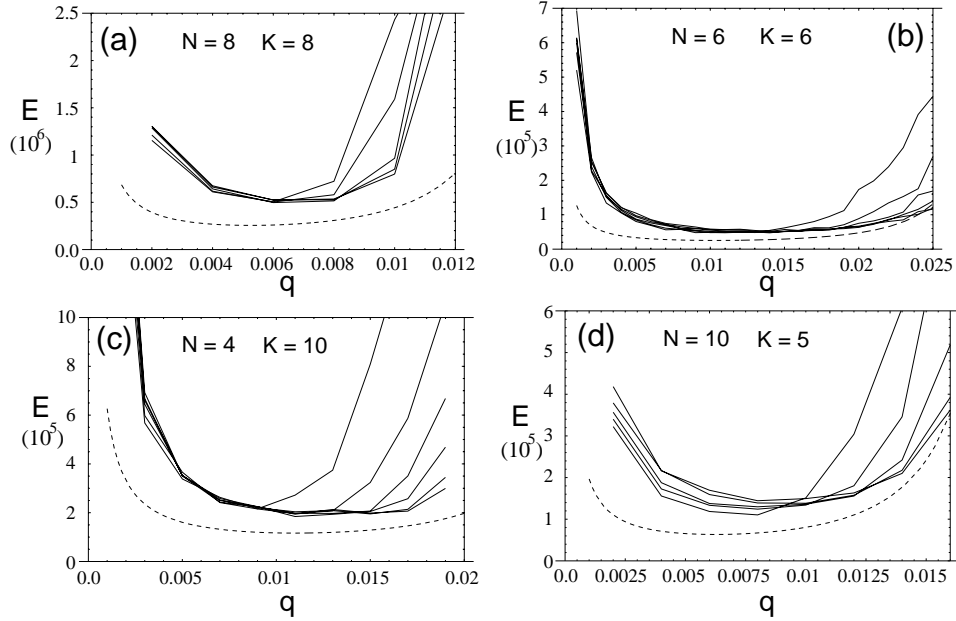


Figure 5.5: Comparison of experimental results (solid lines) and theoretical predictions (dashed lines) of the total number E of function evaluations to reach the global optimal as a function of the mutation rate q , for four different fitness functions as determined by N and K . The fitness function parameters are the same as those used in Fig. 5.3. The dashed lines give the theoretical predictions according to Eq. (5.51). The solid lines are the results of experiments with different population sizes M and show the experimentally estimated E as a function of q for a constant population size M . Each data point on the solid lines gives E averaged over 250 GA runs. For low mutation rates, the solid lines approximately overlap, indicating that E is approximately independent of M in this regime. For large mutation rates, small population sizes have larger E than large populations. Plot (a) shows, from left to right, population sizes $M = 150$, $M = 200$, $M = 250$, $M = 300$, and $M = 350$; plot (b) $M = 60$, $M = 90$, $M = 120$, $M = 150$, $M = 180$, and $M = 210$; plot (c) $M = 50$, $M = 80$, $M = 110$, $M = 140$, $M = 170$, and $M = 200$; and finally, plot (d) $M = 100$, $M = 150$, $M = 200$, $M = 250$, and $M = 300$.

Figure 5.5(b) has parameters $N = 6$ and $K = 6$. $E(q)$ is shown over the range $q = 0.001$ to $q = 0.025$. Again, at large mutation rates the smaller population sizes have higher $E(q)$. The solid lines from top to bottom in the high mutation rate regime

show $E(q)$ for population sizes $M = 60$, $M = 90$, $M = 120$, $M = 150$, $M = 180$, and $M = 210$, respectively. The optimal mutation rate occurs near $q_o = 0.012$.

Figure 5.5(c) has $N = 4$ blocks of length $K = 10$ as parameter settings. $E(q)$ is shown over the range $q = 0.001$ to $q = 0.02$, with the optimum occurring around $q_o = 0.011$. The solid lines show the experimental $E(q)$ for population sizes $M = 50$, $M = 80$, $M = 110$, $M = 140$, and $M = 200$.

Finally, Fig. 5.5(d) has $N = 10$ blocks of length $K = 5$ bits. The horizontal axis ranges from $q = 0.001$ to $q = 0.016$ with the optimal mutation rate occurring around $q_o = 0.008$. The population sizes are, from top to bottom at high q , $M = 100$, $M = 150$, $M = 200$, $M = 250$, and $M = 300$, respectively.

Note that for figures 5.5(b), 5.5(c), and 5.5(d) the tick marks on the vertical axis have a scale of 10^5 fitness function evaluations, while Fig. 5.5(a) uses a scale of 10^6 fitness function evaluations. Figure 5.5 shows that for all these different parameter settings, the theory, which is independent of population size M , reasonably predicts the location of the optimal mutation rate q_o . It also predicts moderately well the shape of the curve around the optimum.

It is notable that the theory consistently underestimates $E(q)$. Furthermore, it underestimates $E(q)$ more in the low mutation rate regime than in the high. We believe that both of these offsets can be explained by the effects of finite-population sampling. We assumed that all unaligned blocks in members of the current highest fitness class are random and *independent* of each other. This last assumption does not hold in general [31]. Due to finite-population sampling and the resulting tendency of the population to converge, strings in the highest fitness class are correlated with each other. This means that if one individual in the highest fitness class has its n th unaligned block at a large Hamming distance from the desired configuration (with that block aligned), then many individuals in the highest fitness class also tend to have their n th block at large Hamming distances from the desired configuration. This genetic correlation among the individuals increases the epoch durations. Moreover, this effect is more severe for small mutation rates which, along with small population sizes, increase population convergence.

It turns out that this effect is difficult to analyze quantitatively. In spite of this it appears, as Fig. 5.5 shows, that for low mutation rates on up to the optimal mutation rate, the total number E of fitness function evaluations is indeed approximately independent of population size M . Moreover, the theory still accurately predicts the location of the optimal mutation rate q_o . Also, although the exact value of E is underestimated, the largest deviation in E from the experimental minimum is 47% for the parameters of Fig. 5.5(a), whereas the minimal deviation is 37% for the parameters of Fig. 5.5(c). Thus, the theory predicts the correct order of magnitude for the minimal number of fitness function evaluations.

As was noted above, the experimental curves for different population sizes appear to collapse onto each other in the low mutation rate regime. As the mutation rate increases, the solid curves break off this common curve one by one and do so delayed in proportion to population size. As the mutation rate increases it appears that progressively larger population sizes are necessary to maintain the search's efficiency.

Of course, this effect is not explained by our population-size independent theory. Intuitively, however, it can be understood as follows. As the mutation rate increases beyond the optimum, the mean proportion P_N^N of individuals with fitness N (during the

last epoch) decreases quite rapidly. When P_N^N is on the order of $1/M$, on average there are very few members with fitness N in the population. These fitness N strings can be easily lost due to a sampling fluctuation. When this happens, the epoch destabilizes, as can be seen occurring in Fig. 5.3(d). E starts rising rapidly for these small population sizes, since epoch N is likely to destabilize several times before an optimal string is found. For larger population sizes, however, the epochs are still stable.

We analyze this destabilization dynamics in detail in the next chapter. Roughly speaking, the population size can be optimized by determining P_N^N for the optimal mutation rate and then by lowering the population size as far as possible without destabilizing epoch N . Note also that there is a critical mutation rate q_c beyond which epoch N is absolutely unstable, even for *infinite* population size. This critical mutation rate is analogous to the error threshold in the theory of molecular evolution [33, 133]. For mutation rates beyond this error threshold, the optimum genome essentially will never be found. The error threshold q_c is also calculated explicitly in the following chapter. The resulting analytical theory ends up predicting the entire *search efficiency surface* $E(q, M)$.

5.12 Conclusion and Future Analyses

In summary, our findings here are the following.

1. At fixed population size M , there is a smooth cost function $E(q)$ as a function of mutation rate q . It has a *single* and *shallow* minimum q_o .
2. The optimal parameter settings roughly occur in the regime where the highest epochs $N - 1$, N , and $N + 1$ are marginally stable; see Fig. 5.3.
3. At the optimal parameter settings the average number of fitness function evaluations is $\mathcal{O}(N^{3.1}2^K)$. This is less efficient than a simple mutational hill climber, $\mathcal{O}(N^22^K)$.
4. For lower mutation rates than q_o the total number of fitness function evaluations $E(q)$ grows steadily and becomes almost independent of the population size M .
5. Crossover's role in reducing search time is marginal due to population convergence during the epochs.
6. There is a mutational error threshold in q that bounds the upper limit of the GA's efficient search regime. Above the threshold, which is population-size independent, suboptimal strings of fitness N cannot stabilize in the population. For small populations, this error threshold occurs at lower values of q .
7. The function $E(q)$ appears to be concave. That is, for any two parameters q_1 and q_2 , the straight line connecting these two points is everywhere above the curve $E(q)$. We conjecture that this is always the case for mutation-only genetic algorithms with a unique global optimum. This feature is useful in that a steepest-descent algorithm applied to parameter q will lead to the unique optimum q_o .

Synopsizing the results this way, we anticipate some of the results developed for the population-size dependent theory of chapter 6.

Having come to this point one must ask, What relevance do these results have for a practitioner applying evolutionary search to complicated optimization problems? First of all, our analysis applies to problems showing epochal evolution. When epochal evolution occurs for a realistic search problem it may either be a result of the population getting stuck on local fitness optima or it may be a result of large neutral subspaces through which the population diffuses. From the practitioner's standpoint, it is important to distinguish these two cases, since their search dynamics are very different. A population stuck on a local optimum can be identified by the fact that the population forms a cloud around a *single genotype* in genotype space. Therefore, the genotypic content of the population is constant on average. In contrast, for the second case with diffusion over neutral networks, only the *average fitness* is constant over time while the population cloud of genotypes *drifts* in an erratic way through genotype space. It should therefore be relatively straightforward for the practitioner to determine what *kind* of epochal evolution is occurring by monitoring these features.

On the one hand, if the population is stuck at a local optimum, it is most often beneficial to decrease the selection pressure or to increase the mutation rate until the population begins to "move" again through genotype space. Basically, in this situation the population has already discovered the highest fitness genotype in the region of genotype space around the local optimum. No higher fitness strings are likely to be discovered unless the population starts to move through genotype space.

On the other hand, if the population is diffusing over a neutral net of roughly equal fitness strings, it is straightforward to determine an optimal mutation rate experimentally. Our theory indicates that the optimal mutation rate occurs when the product $qP_n^n(q)$ of the proportion of individuals P_n^n on the highest fitness neutral net and the mutation rate q is maximal. In short, the optimal mutation rate occurs when the total number of mutations in individuals of the current highest fitness class is maximal. After the optimal mutation rate is determined (holding population size constant), one can lower the population size until the product $P_n^n M$ is so small that further decreases in M compromise the stability of current highest fitness individuals. Since the proportion P_n^n can be measured experimentally, the procedure just outlined allows for the maximization of the product qP_n^n for complicated search problems.

If the product qP_n^n happens to be too hard to obtain experimentally, a rough estimate of the optimal parameter settings can still be obtained by noting that the optimal parameter settings typically occur when the current highest fitness members in the population are only marginally stable. That is, the mutation rate is as high as can be during each epoch without destroying all individuals of the highest fitness class. (This behavioral regime is analyzed in more detail in the next chapter.)

Finally, we note that the analysis presented here can be adapted in a straightforward manner to other selection schemes—such as, elite and tournament selection. Results for these will be reported elsewhere.

Acknowledgements

The authors thank Melanie Mitchell for helpful discussions. This work was supported at the Santa Fe Institute by NSF Grant IRI-9705830, ONR grant N00014-95-1-0524 and by Sandia National Laboratory contract AU-4978.

